UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA – UFPE
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

PAULO FERNANDO LEITE FILHO

ESCALA PSICOMÉTRICA NA AVALIAÇÃO COGNITIVA DO APRENDIZADO DE MÁQUINA

Recife
2024

Paulo Fernando Leite Filho

ESCALA PSICOMÉTRICA NA AVALIAÇÃO COGNITIVA DO APRENDIZADO DE MÁQUINA

Tese de Doutorado apresentada ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de Doutor em Ciência da Computação.

Área de Concentração: Inteligência computacional

Orientador: Prof. Dr. Silvio de Barros Melo

Coorientadora: Prof. Dra. Rita de Cássia Moura do Nascimento

Recife
2024

.Catalogação de Publicação na Fonte. UFPE - Biblioteca Central

Leite Filho, Paulo Fernando.
Escala psicométrica na avaliação cognitiva do aprendizado de máquina / Paulo Fernando Leite Filho. - Recife, 2024.
203f.: il.

Tese (Doutorado) - Universidade Federal de Pernambuco, Centro de Informática, Programa de Pós-Graduação em Ciência da Computação, 2024.
Orientação: Silvio de Barros Melo.
Coorientação: Rita de Cássia Moura do Nascimento.
Inclui referências e anexos.

1. Aprendizado de máquina; 2. Agrupamento; 3. Métricas; 4. Teoria da Resposta ao Item. I. Melo, Silvio de Barros. II. Nascimento, Rita de Cássia Moura do. III. Título.

UFPE-Biblioteca Central

Paulo Fernando Leite Filho

ESCALA PSICOMÉTRICA NA AVALIAÇÃO COGNITIVA DO APRENDIZADO DE MÁQUINA

Tese de Doutorado apresentada ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de Doutor em Ciência da Computação.

Aprovado em: 30/01/2024.

Orientador: Prof. Dr. Silvio de Barros Melo
Coorientadora: Profa. Dra. Rita de Cássia Moura do Nascimento

BANCA EXAMINADORA

_____
Prof. Dr. Ricardo Bastos Cavalcante Prudêncio
Centro de Informática / UFPE

_____
Prof. Dr. Nivan Roberto Ferreira Júnior
Centro de Informática / UFPE

_____
Prof. Dr. Carmelo Jose Albanez Bastos Filho
Escola Politécnica de Pernambuco / UPE

_____
Prof. Dr. Sérgio de Carvalho Bezerra
Centro de Informática / UFPB

_____
Prof. Dr. Christian Robson de Souza Reis
Instituto Aggeu Magalhães / FIOCRUZ

Dedico esta Tese aos meus pais Paulo e Eva, minha esposa Laury e minhas filhas Paloma, Mariana e Agnes.

AGRADECIMENTOS

Gostaria de agradecer primeiramente a Deus pela oportunidade de realizar o sonho do Doutorado em Ciência da Computação na Universidade Federal de Pernambuco (UFPE).

Aos meus pais, Paulo e Eva, que me educaram e sempre me incentivaram a buscar novos horizontes e desafios, na certeza da minha capacidade em conquistá-los. As minhas filhas, Paloma, Mariana e Agnes, a minha esposa Laury, pelo incentivo e compreensão. Foram momentos longe de casa no intuito da realização deste sonho que deixou de ser somente meu e passou a ser nosso. A vocês dedico esta conquista!

Ao meu orientador Professor Silvio de Barros Melo e a coorientadora Professora Cássia Moura, pela competência acadêmica e palavras de inspiração profissional.

Ao Centro de Informática (CIn) da UFPE por me proporcionar a oportunidade de realizar um grande passo na minha formação acadêmica.

Agradeço também àqueles que fizeram parte destes longos anos de estudos no curso de doutorado no CIn, e de minha atuação profissional em pesquisas, nas salas de aula de uma faculdade particular, e no setor de informática de um hospital público, durante o enfrentamento da pandemia do Covid-19, sem a previsão de data e nem hora para acabar. Foram momentos de muito estudo e de trabalho árduo, mas também de muito aprendizado com todos vocês! Agradeço, portanto, a todos que fizeram parte desta jornada acadêmica.

“A aprendizagem não ocorre pela simples transmissão de algo que está fora. A aprendizagem é um fenômeno interpretativo da realidade e que requer o ato de construir e reconstruir a todo instante.”

Paulo Freire

RESUMO

Os métodos de avaliação cognitiva em humanos são fundamentais na análise de modelos complexos obtidos da Inteligência Artificial (IA). O objetivo desta Tese foi avaliar as dificuldades dos algoritmos de *clustering* pela Teoria da Resposta ao Item (TRI), como escala psicométrica na formação de agrupamentos. Foram avaliados pela escala cognitiva baseada no Parâmetro de Dificuldade b dos Modelos de Parâmetros Logísticos (PL) da TRI os agrupamentos formados por 38 algoritmos de *clustering* em 11 *datasets*, e avaliados os resultados das métricas de Precisão, Sensibilidade e Especificidade, como referências para os índices externos de *Rand, Foulkes-Mallows* e *Jaccard.* Estas avaliações foram realizadas nos acertos e erros na formação dos agrupamentos correspondentes às classes pré-existentes nos *datasets*, pelos grupos de algoritmos de *clustering* baseados em métodos clássicos de agrupamento, em *k-means* e no método *Fuzzy.* Estes agrupamentos ocorreram proporcionalmente em condições adversas de desbalanceamento de classes, variações da dimensionalidade, interseção dos elementos nas partições pelo método de votação, decisão para situações críticas e significância das métricas. Os resultados para as atribuições corretas nos níveis de dificuldade e correspondentes às métricas calculadas foram: os algoritmos baseados em *Fuzzy* obtiveram 99,15% no nível fácil e 84,98% no nível difícil da Precisão, 5,99% no nível muito difícil da Especificidade e 4,19% no nível extremamente difícil; os algoritmos baseados em *k-means* com 5,07% na Sensibilidade, e para os algoritmos baseados em métodos clássicos de agrupamento houve bons resultados em nível extremamente difícil na Sensibilidade. Estes resultados proporcionaram diversas vantagens em relação à qualidade dos modelos baseados em algoritmos de *clustering*, informando referências e indicadores de qualidade na formação de agrupamentos em ambientes desfavoráveis, e atribuições dos elementos em agrupamentos que impactam nos resultados das métricas. Conclui-se que o emprego da escala psicométrica durante as atribuições dos agrupamentos em situações adversas, amplia a confiabilidade em sistemas de apoio à decisão, com potencial uso por profissionais em áreas críticas.

Palavras-chave: Aprendizado de Máquina; Agrupamento; Métricas; Teoria da Resposta ao Item.

ABSTRACT

Cognitive assessment methods in humans are fundamental in the analysis of complex models obtained from Artificial Intelligence (AI). The objective of this Thesis was to evaluate the difficulties of clustering algorithms using Item Response Theory (IRT), as a psychometric scale in the formation of clusters. Clusters formed by 38 clustering algorithms in 11 datasets were evaluated using the cognitive scale based on the Difficulty Parameter *b* of the Logistic Parameter Models (LP) of the IRT, and the results of the metrics of Precision, Sensitivity and Specificity were evaluated, as references for the external indices of Rand, Foulkes-Mallows and Jaccard. These evaluations were carried out on the successes and errors in the formation of clusters corresponding to pre-existing classes in the datasets, by groups of clustering algorithms based on classical clustering methods, k-means and the Fuzzy method. These groupings occurred proportionally in adverse conditions of class imbalance, variations in dimensionality, intersection of elements in the partitions by the voting method, decision for critical situations and significance of metrics. The results for correct assignments at difficulty levels and corresponding to the calculated metrics were: the Fuzzy-based algorithms obtained 99.15% at the easy level and 84.98% at the difficult level of Accuracy, 5.99% at the very difficult level of Specificity and 4.19% in the extremely difficult level; the algorithms based on k-means with 5.07% in Sensitivity, and for the algorithms based on classical clustering methods, there were good results in extremely difficult level in Sensitivity. These results provided several advantages in relation to the quality of models based on clustering algorithms, informing references and quality indicators in the formation of clusters in unfavorable environments, and assignments of elements in clusters that impact the results of the metrics. It is concluded that the use of the psychometric scale during the assignment of clusters in adverse situations increases the reliability in decision support systems, with potential use by professionals in critical areas.

Keywords: Machine Learning; Cluster; Metrics; Item Response Theory.

.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

Abreviatura	Significado
+P	Preditos como os Positivos
1PL	Modelo de 1 Parâmetro Logístico
2PL	Modelo de 2 Parâmetros Logísticos
3PL	Modelo de 3 Parâmetros Logísticos
Acc	Precisão, do inglês *Accuracy*
ACE	Exame Cognitivo de *Addenbrooke*, do inglês *Addenbrooke's Cognitive Examination*
AD	Distância média, do inglês *Average Distance*
ADM	Distância média entre médias, do inglês *Average Distance Between Means*
AGI	Inteligência Geral Artificial, do inglês *Artificial General Intelligence*
AIC	Critério de Informação Akaike, do inglês *Akaike Information Criterion*
ANI	Inteligência Artificial Estreita, do inglês *Artificial Narrow Intelligence*
APN	Proporção Média de Não Sobreposição, do inglês *Average Proportion of Non-overlap*
ASI	Superinteligência Artificial, do inglês *Artificial Super Intelligence*
AUC	Área sob a curva, do inglês *Area Under the curve*
BIC	Critério de Informação Bayesiano, do inglês *Bayesian information criterion*
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CCI	Curva Característica do Item
CLARA	*Clustering Large Application*
CNES	Cadastro Nacional de Estabelecimentos de Saúde
DC	Difícil e classificado corretamente
DE	Difícil e classificado incorretamente
DKVMN	Rede dinâmica de memória de valor-chave, do inglês *Dynamic Key-value Memory Network*
E	Fácil, do inglês *Easy*
Easy	*Ensemble and Asymmetric Bagging*
eHEALS	Escala de alfabetização em eSaúde*, do inglês eHealth Literacy Scale*
EM	Máxima Expectativa, do inglês *Expectation Maximization*
ENEM	Exame Nacional do Ensino Médio
FANNY	*Fuzzy Analysis Clustering*
FC	Fácil e classificado corretamente
FDA	Função de Distribuição Acumulada

FE	Fácil e classificado incorretamente
FN	Falso-Negativo
FOM	Figura de mérito, do inglês *Figure Of Merit*
FP	Falso-Positivo
GCS	Escala de Coma de Glasgow, do inglês *Glasgow Coma Scale*
GMAT	Teste de Admissão em Gestão de Pós-Graduação, do inglês *Graduate Management Admission Test*
GRE	Exame de registro de pós-graduação, do inglês *Graduate Registration Examination*
H	Difícil, do inglês *Hard*
IA	Inteligência Artificial
MCMC	Monte Carlo de Cadeia de Markov, do inglês *Markov Chain Monte Carlo*
MDC	Muito difícil e classificado corretamente
MDE	Muito difícil e classificado erroneamente
MFC	Muito fácil e classificado corretamente
MFE	Muito fácil e classificado erroneamente
mGES	Escala de Eficácia da Marcha Modificada, do inglês *Modified Gait Efficacy Scale*
MS	Ministério da Saúde
ML	Modelo Logístico
MLE	Estimativa de Máxima Verossimilhança, do inglês *Maximum-Likelihood Estimation*
MMLE	Estimativa de Máxima Verossimilhança Marginal, do inglês *Marginal Maximum-Likelihood Estimation*
ND	Níveis de dificuldade
NMF	*Nonnegative Matrix Factorization*
Not2ESE	*Notification of Easier - exits and stratified – enters*
NPC	Problemas não-polinomial-completo
OE	Objetivos Específicos
O-grande	Ordem de Grandeza
P	Problemas Polinomiais
PAM	*Partitioning Around Medoids*
PCA	Análise de Componentes Principais, do inglês *Principal Component Analysis*
PL	Parâmetro Logístico
PRISMA	Itens de relatório para revisões sistemáticas e meta-análises, do inglês *Preferred Reporting Items for Systematic reviews and Meta-Analyses*

PSO	Otimização de Enxame de Partículas, do inglês *Particle Swarm Optimization*
QP	Questão de Pesquisa
RASS	Escala de Agitação e Sedação de Richmond, do inglês *Richmond Agitation Sedation Scale*
ROC	Curva característica de operação do receptor, do inglês *Receiver Operating Characteristic Curve*
ROS	*Random Over Sampling*
RUS	*Random Under Sampling*
SAT	Teste de Avaliação Escolar, do inglês *Scholastic Assessment Test*
Sen	Sensibilidade, do inglês *Sensitivity*
SIA	Sistema de Informação Ambulatorial
SIH	Sistema de Informações Hospitalares
SINAN	Sistema de Informação de Agravos de Notificação
SINASC	Sistema de Informações sobre Nascidos Vivos
SIS	Sistemas Informatizados em Saúde
SMOTE	Técnica de sobreamostragem minoritária sintética, do inglês *Synthetic Minority Oversampling Technique*
SOTA	*Self-Organization Tree Algorithm*
Spe	Especificidade, do inglês *Specificity*
STC	Suffix Tree Clustering
SVM	*Support Vector Machine*
TCT	Teoria Clássica dos Testes
tnr	Taxa de verdadeiros negativos, do inglês *True negative rate*
TOEFL	Teste de inglês como língua estrangeira, do inglês *Test of English as a Foreign Language*
UTI	Unidade de Terapia Intensiva
TRI	Teoria de Resposta ao Item
VE	Muito fácil, do inglês *Very Easy*
VH	Muito difícil, do inglês *Very Hard*
V-Measure	Índice de Medida de Variação, do inglês *Variation Measure*
VN	Verdadeiro-Negativo
VP	Verdadeiro-Positivo
XE	Extremamente fácil, do inglês *Extremely Easy*
XH	Extremamente difícil, do inglês *Extremely Hard*

Lista de símbolos

π	Função dos parâmetros para os efeitos da habilidade do algoritmo do teste e as propriedades do item
θ	Habilidade do algoritmo
a	Parâmetro de Discriminação da TRI
b	Parâmetro de Dificuldade da TRI
c	Parâmetro de Probabilidade de acerto ao acaso da TRI

Sumário

1. INTRODUÇÃO

O aprendizado dos algoritmos ocorre de formas distintas para a resolução de problemas com objetivos específicos, possibilitando maior eficiência do algoritmo em diferentes tipos de bases de dados (Hathaliya; Tanwar, 2020). Para que ocorra o aprendizado dos algoritmos de *Aprendizagem de Máquina* é necessária uma base de conhecimento prévio (*i.e.* supervisionado) ou não supervisionados. Os algoritmos de *clustering* utilizam a metodologia de aprendizagem não supervisionada, e têm como objetivo agrupar elementos em *clusters* com base em sua similaridade, visando atribuir a cada amostra um rótulo do *cluster* (Wang *et al.,* 2024a).

Os algoritmos de *clustering* podem ser penalizados na formação dos agrupamentos por fatores como determinação prévia de número de classes em *clusters*, problemas de amostragem de dados, variação de dimensionalidade, desbalanceamento de classes. Há outros fatores similares, que podem ser interpretados como as características das amostras, proporcionando aos algoritmos limitações, comprometendo a formação dos agrupamentos, e causando indeterminações nas pertinências dos elementos nos respectivos grupos. Esta questão demanda o uso de métricas para avaliar os acertos e erros dos algoritmos de *clustering*. Entretanto, estas métricas não avaliam as habilidades destes algoritmos nos acertos e erros de pertinência dos elementos nos respectivos grupos, assim como as situações desfavoráveis citadas no início deste parágrafo (Leite-Filho; Melo; Cassia-Moura, 2021b).

Os algoritmos de *clustering* precisam da avaliação para validar os resultados obtidos, pois os algoritmos aprendem com os dados e em sequência constroem grupos de dados para a resolução de problemas específicos em diversos sistemas de apoio a decisões complexas, e comumente são aplicados em sistemas financeiros, energia, saúde, dentre outros. Avaliar a dificuldade na representação da importância dos elementos contidos nos agrupamentos em relação a suas respectivas pertinências nos grupos gerados por *clustering* é o objetivo desta Tese.

1.1. Contexto e motivação

As aplicações de algoritmos de *clustering* abrangem o diagnóstico do conjunto de dados de saúde dos prontos-socorros (Wartelle *et al*., 2021); análise de dados de expressão gênica para identificar padrões de expressão relacionadas à genética (Ding *et al*., 2024); para diagnóstico de defeitos hereditários, segmentação de imagens médicas, detecção de comunidades sociais, detecção de anomalias, extração de classes, análise de comportamento de consumo (Wang *et al*., 2024b) e outras aplicações. Estas soluções estão diretamente associadas com o objetivo dos algoritmos de *clustering*.

O objetivo do algoritmo de *clustering* é determinar o agrupamento (i.e. *cluster*) intrínseco em um conjunto de dados previamente não rotulados, onde os elementos em cada grupo são indistinguíveis sob algum critério de similaridade, agrupando elementos com características comuns (Rendón *et al*., 2011). A definição de *clusters* utilizada da literatura conceitualiza matematicamente que são conjuntos formados pelos elementos C_1, C_2, ..., C_i, correspondendo a subespaços E_1, E_2, ..., E_k, onde *i* representa a ordem de formação dos *clusters, k* é o *i*-ésimo valor conhecido, de forma que os elementos contidos em C_i sejam semelhantes. Estes elementos são agrupados em um subespaço representado por E_i, e E_k representando o subconjunto de atributos originais dos dados. Portanto, o *cluster* pode ser interpretado como uma formação de conjuntos em um espaço de um sistema ortogonal (Charu, 2014).

Seja $X = \{X_1, X_2, \dots, X_n\}$ conjunto com *n* objetos, onde $X_i \epsilon \mathbb{R}^p$ é um vetor de *p* medidas reais. A ação de geração de clusters é o agrupamento dos elementos de *X* em *k* clusters disjuntos $\{C_1, C_2, \dots, C_k\}$ tais que:

1. $C_1 \cup C_2 \cup \dots \cup C_k = X$
2. $C_i \neq \emptyset, \forall i \; 1 \leq i \leq k$
3. $C_i \cup C_j, \forall i \neq j, 1 \leq i, j \leq k$

Os algoritmos de *clustering* são organizados taxonomicamente em classes, representadas pelas seguintes categorias: hierárquico aglomerativo ou divisivo (Paganin *et al.*, 2022); formação por densidade (Sukassini; T.Velmurugan, 2022); baseado na formação do centróide (Wijaya *et al*., 2021); ajuste do centróide pela densidade dos pontos (Gultom *et al.*, 2018); por *Self-Organizing Map* (Das; Bhuyan, 2017); pelo método de *Fuzzy* (Boongoen; Iam-on, 2018), dentre outros

*(*Maechler *et al*., 2015). Estas categorias podem ser combinadas em *clusters* mistos, para a facilitar a tomada de decisões.

Entretanto, a tarefa de clustering por algoritmos embute os seguintes desafios: i) representação da importância dos elementos contidos nos agrupamentos em relação a suas respectivas pertinências nos grupos; ii) interpretação da qualidade do agrupamento, a partir do uso das distâncias dos elementos do *cluster*, a qual é difícil de interpretar devido à generalização do agrupamento formado em relação ao rótulo diagnóstico atribuído pelos especialistas; iii) a perda de informação devido à geolocalização espacial do elemento no agrupamento; e iv) inadequações pela perda de informação na análise de correspondência múltipla (Wartelle *et al*., 2021). Estes fatores devem ser analisados a partir dos grupos formados, necessitando de métricas que abarquem as dificuldades apresentadas neste parágrafo.

A Teoria da Resposta ao Item (TRI) é um método psicométrico da Psicologia que permite avaliações de confiabilidade de instrumentos de medida, como as avaliações acadêmicas, questionários e similares (Gomes *et al.*, 2018). A TRI também pode ser vista como um método estatístico utilizado em avaliações e questionários para identificar habilidades dos respondentes, e que também possibilita classificar individualmente cada item em distintos níveis de dificuldades nas respostas obtidas (Martínez-Plumed et al., 2019). A TRI é composta por parâmetros importantes que informam diferentes atributos na avaliação de questões. Tem sido considerada como uma solução para selecionar os registros considerados difíceis ou fáceis de classificar em uma escala.

A TRI foi aplicada em áreas mistas, como saúde e educação (Dickson; Krumwiede, 2020), finanças (Vieira; Potrich; Bressan, 2020), avaliação de nível de desconforto de projetos de assentos de aeronaves (Menegon et al., 2019) e outras áreas correlatas. Nestes trabalhos citados, a TRI contribuiu em estudos individuais dos elementos que compõem os respectivos objetivos dos conjuntos de dados. A identificação do nível de significância de cada item em relação ao conjunto possibilita uma tomada da decisão mais otimizada por cada aplicação em suas respectivas áreas. A computação também aplicou as técnicas de TRI para resolver problemas como: construir um sistema de aprendizado que incorpora a TRI em uma estrutura unificada para resolver o problema de *cold-*

start, para novos elementos de um ambiente de aprendizado adaptativo em Aprendizado de Máquina (Pliakos *et al*., 2019). Ao realizar uma associação da TRI com agrupamento e classificações, entende-se que as instâncias correspondem aos itens e os algoritmos de *clustering* correspondem aos respondentes. Este tipo de associação foi referenciada na literatura em experimentos com Aprendizado de Máquinas supervisionadas, como no trabalho de Martínez-Plumed *et al*. (2019).

1.2. Organização da Tese

Esta Tese foi conduzida sob uma perspectiva multidisciplinar, combinada com uma análise comparativa da cognição computacional bioinspirada no aprendizado humano, envolvendo abordagens do tema em Computação.

A Tese foi editada com base nas Normas NBR 6023, 14724 e 10520 (Associação Brasileira de Normas Técnicas, 2011, 2018, 2023), seguindo o modelo de referência e teses apresentadas (Biblioteca Digital de Teses e Dissertações, 2024; Universidade Federal de Pernambuco, 2024), e está organizada da conforme abaixo.

No Capítulo 1 foram apresentados os tópicos introdutórios, contendo na seção 1.1 os conceitos e motivações importantes para a contextualização desta Tese, na seção 1.2 foi descrita a organização da Tese, na seção 1.3 a definição do problema e a hipótese, na seção 1.4 os objetivos e na seção 1.5 os trabalhos acadêmicos publicados (Anexo A – E).

No Capítulo 2 foi descrita a revisão sistemática, na seção 2.2 foram descritos tópicos sobre Aprendizado de Máquina, e nas demais seções deste capítulo foram citados os fatores relevantes sobre algoritmos de *clustering* e suas características.

No Capítulo 3 foi apresentado o método proposto nesta Tese. No Capítulo 4 foram apresentados os resultados obtidos e a discussão com a literatura. No Capítulo 5, a conclusão. No Capítulo 6 foram apresentadas as contribuições deste trabalho para a comunidade acadêmica. No Capítulo 7 foram discutidas as limitações dos resultados encontrados nesta Tese e no capítulo 8 foram propostos trabalhos futuros para a continuidade da pesquisa.

Nos Apêndice A foi apresentado o código fonte utilizado nos experimentos desta Tese. Nos Anexos (A – E) foram apresentados os comprovantes da publicação dos trabalhos acadêmicos.

1.3. Problemas de pesquisa e hipótese

Esta pesquisa teve por objetivo estabelecer uma associação quantitativa e qualitativa dos fatores que impactam na tarefa de agrupamento por algoritmos, bem como estimar a dificuldade dessa tarefa através do parâmetro de dificuldade da TRI, validando com métricas de *performance*. Desta forma, foram selecionadas as seguintes questões de pesquisa (QP):

QP1: Quais são os fatores relacionados às dificuldades na formação dos agrupamentos que serão avaliados?

QP2: Como a literatura aborda a aplicação da TRI como método para determinar as dificuldades na pertinência de elementos nos agrupamentos dos algoritmos de *clustering*?

QP3: Como aplicar a TRI como método comparativo de dificuldades na formação dos agrupamentos?

QP4: Como é possível utilizar a TRI para determinar algoritmos de *clustering* ótimos na determinação da pertinência dos elementos nos respectivos agrupamentos?

QP5: Como será realizada a análise comparativa entre os níveis de dificuldades obtidas pela TRI nos elementos agrupados, as correspondências dos agrupamentos criados pelos algoritmos de *clustering* com as rotulações já existentes nos *datasets?*

QP6: Como será realizada a análise comparativa entre as relações entre o cálculo da dificuldade da TRI com os resultados das métricas de rendimento e formação de *clusters*?

Para a orientação de como foram respondidos os seis problemas de pesquisa acima referidos, serão descritos os conceitos considerados importantes para a apresentação da hipótese desta Tese, que foi relacionar em uma escala psicométrica a comparação entre os grupos de resultados das classificações dos algoritmos não supervisionados e os grupos de Parâmetro de Dificuldade b da TRI. Será identificado o impacto destas dificuldades nas

métricas de rendimento e formação de *clusters*, possibilitando uma melhor avaliação para algoritmos ótimos nos diferentes níveis de formação de agrupamentos. E serão também descritos os conceitos importantes para o entendimento da hipótese proposta, os resultados obtidos nas análises realizadas nesta Tese, a comparação de resultados da literatura com este trabalho, e os avanços na forma de avaliar a cognição de algoritmos de *clustering.*

1.4. Objetivos

1.4.1. *Objetivo geral*

Estabelecer uma associação quantitativa e qualitativa dos fatores que impactam na tarefa de agrupamento por algoritmos, bem como estimar a dificuldade dessa tarefa através do Parâmetro de Dificuldade b da TRI, validando com métricas de *performance*.

1.4.2. *Objetivos específicos (OE)*

- Realizar experimentos de formação de agrupamentos por algoritmos de *clustering* em cada *dataset.*
- Avaliar os experimentos pelo Parâmetro de Dificuldade b da TRI em cada cenário de impacto na formação dos agrupamentos.
- Quantificar os fatores que impactam nos agrupamentos por algoritmos de *clustering*.
- Qualificar os fatores que impactam nos agrupamentos por algoritmos de *clustering*.
- Relacionar o Parâmetro de Dificuldade b com os métodos de métricas nos agrupamentos por algoritmos de *clustering*.

1.5. Trabalhos publicados

Durante a realização desta Tese, os seguintes trabalhos acadêmicos foram publicados:

- LEITE-FILHO, P. F.; MELO, S. B.; CASSIA-MOURA, R. Teoria de Resposta ao Item e o Paradoxo de Moravec na obtenção de algoritmos ótimos. In: Mostra de Extensão, Inovação e Pesquisa POLI/UPE 2021, 2021, Recife. Anais da Mostra de Extensão, Inovação e Pesquisa POLI/UPE 2021. v. 1. p. 165-166, 2021 (ANEXO A).

- LEITE-FILHO, P. F.; MELO, S. B.; CASSIA-MOURA, R. Métricas de rendimentos estatísticos em uma visão ampla dos dados. In: Semana Universitária UPE 2021 - Democratizando a ciência do litoral ao sertão, 2021, Recife. Anais da Semana Universitária UPE 2021 - Democratizando a ciência do litoral ao sertão. p. 1002, 2021 (ANEXO B).

- LEITE FILHO, P. F.; MELO, S. B.. Tomada de decisões baseada no parâmetro *b* da Teoria da Resposta ao Item nas classificações de *ensemble*. **Revista dos Mestrados Profissionais**, v. 11, n. 2, p. 235–249, 2022. DOI: 10.51359/2317-0115.2022.256869. Disponível em: https://periodicos.ufpe.br/revistas/RMP/article/view/256869 (ANEXO C, Qualis Capes B4).

- LEITE-FILHO, P. F.; MELO, S. B.. Significance of Item Response Theory in Cluster Evaluation Metrics. **2023 XLIX Latin American Computer Conference (CLEI),** La Paz, Bolivia,pp. 1-9, 2023, DOI: 10.1109/CLEI60451.2023.10346171 (ANEXO D, Qualis Capes B2).

Vale salientar que o trabalho intitulado “Teoria de Resposta ao Item e o Paradoxo de Moravec na obtenção de algoritmos ótimos” (Leite-Filho; Melo; Cassia-Moura, 2021a), apresentado na Mostra de Extensão, Inovação e Pesquisa POLI/UPE 2021, foi premiado como melhor trabalho na categoria pós-graduação, Engenharia da Computação - Sistemas, durante a Semana Universitária da Universidade de Pernambuco (ANEXO E)**.**

2. REVISÃO SISTEMÁTICA

A revisão sistemática da literatura é usual na área de saúde e tem como objetivo realizar uma busca de trabalhos correlacionados com tópicos de pesquisa, definidos primariamente. Foi reformulada e introduzida na Informática por meio da Engenharia de *Software* baseada em evidências (Kitchenham; Charters, 2007), identificando os estudos já realizados e disponíveis na literatura, proporcionando a continuidade do método formal do rigor científico e identificando as lacunas na pesquisa.

Nesta Tese, uma revisão sistemática da literatura foi realizada de acordo com o método *Preferred Reporting Items for Systematic reviews and Meta-Analyses* (PRISMA) (PAGE *et al*., 2021), utilizando o *Software Rayyan* (Khabsa *et al.*, 2016; Ouzzani *et al*., 2016; Tao *et al.,* 2019), aplicando filtros nas palavras-chaves: *Item Response Theory, Machine Learning, Metrics, Classification, Psychometrics, AND Cognition*.

A busca foi realizada nas bases da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), *SCOPUS, IEEE Xplore* e *ArXiv*, com período de busca de 1991 até 2022. Foram identificados 1.030 artigos. Para filtrar estes resultados, foram adotados os seguintes critérios de seleção: (i) artigos vicinais dos conceitos e evolução da TRI, (ii) artigos correlacionados da TRI com IA, (iii) TRI correlacionada com a dimensionalidade ou amostragem de dados, (iv) contexto humano-computador, e (v) artigos não relacionados para a exclusão. Após filtrar os artigos a partir da leitura dos títulos e resumos, foram identificados os conteúdos dos critérios de seleção citados neste parágrafo, e foram avaliados os métodos de aplicações da TRI com a IA. Ao aplicar estes filtros foram obtidos os seguintes resultados:

- artigos vicinais para os conceitos e evolução da TRI: 334 artigos, sendo mais antigos que 2019;
- artigos correlacionados da TRI com IA: 80 artigos, nos últimos 5 anos;
- TRI correlacionada com dimensionalidade ou amostragem de dados: 26 artigos, nos últimos 5 anos;
- contexto humano-computador: 33 artigos, nos últimos 5 anos;
- os demais artigos não relacionados foram excluídos.

Foi realizado um cruzamento entre os grupos de artigos correlacionados da “TRI com IA” e “TRI correlacionada com dimensionalidade ou amostragem de dados”, nos quais foi aplicado mais um filtro considerando as aplicações em *Machine Learning*, restringindo a 68 artigos de interesse.

Uma lista ordenada pela frequência das palavras mais utilizadas nos artigos foi elaborada, onde foram selecionadas as palavras de maior frequência para cruzamentos entre os artigos, nas quais foram obtidos os seguintes resultados: dos 68 artigos foram identificados 39 artigos relacionados com classificação; seis artigos relacionados apenas *cluster*; 18 artigos relacionados com TRI e alguma técnica de Aprendizado de Máquina; dois artigos relacionando Aprendizagem de Máquina com dimensionalidade dos dados; 12 artigos relacionados com desbalanceamento de classes; oito artigos relacionados com *Ensemble*; 40 artigos relacionando com métricas; e três artigos relacionando Aprendizagem de Máquina, métricas e TRI. Ao realizar o cruzamento de *Machine Learning*, *cluster* e TRI; IA, *cluster* e TRI; *cluster*, métricas e TRI, foram encontradas uma quantidade muito pequena de artigos relacionando estes cruzamentos, evidenciando lacunas para a pesquisa desta Tese.

Esta etapa da revisão sistemática possibilitou observar a presença de lacunas a serem avaliadas nesta Tese. Observou-se que as análises das aplicações da TRI foram pouco exploradas em *Machine Learning* com as métricas, sendo identificada esta lacuna do atual conhecimento para a realização desta Tese. Além disso, não houve um aprofundamento da aplicação da TRI em algoritmos de *cluster* de dados. E devido a complementações de embasamentos teóricos para esta Tese, na bibliografia listada nesta Tese há artigos adicionais, além dos 68 artigos de interesse e dos artigos vicinais os quais estão detalhados na seção do Estado da Arte.

2.1. Estado da Arte

Com a revisão sistemática da literatura foram obtidos distintos grupos de artigos relacionados aos cruzamentos dos tópicos de classificação, *cluster*, TRI e Aprendizado de Máquina, dimensionalidade dos dados, desbalanceamento de classes, *Ensemble* e métricas. Como a proposta desta Tese foi a apresentação de uma solução baseada em uma escala do Parâmetro de Dificuldade *b* da TRI

como otimização de qualidade dos elementos agrupados, foram aplicados filtros para canalizar os resultados nos grupos citados anteriormente na revisão sistemática da literatura, organizados nos parágrafos a seguir. Nesta seção estão incluídos os trabalhos mais relevantes para esta Tese, assim como trabalhos complementares realizados fora do escopo temporal da revisão sistemática, destacando os pontos de importância para esta Tese em cada grupo formado.

No grupo do cruzamento de Aprendizado de Máquina e dimensionalidade da revisão da literatura aplicando *TRI* como filtro no grupo destacam-se trabalhos relevantes como a aplicação do teste adaptativo baseado na TRI como parâmetro avaliativo nas classificações de algoritmos de Aprendizagem de Maquina em dados desbalanceados e com variações de dimensionalidade (Zheng; Cheon; Katz, 2020). Em casos de predição de mortalidade aplicando a TRI como método avaliativo de dificuldades nas estimações das classificações, foram realizadas críticas às métricas de precisão, sensibilidade, especificidade e recall nas avaliações qualitativas dos diagnósticos de cada paciente tratado (Zheng; Cheon; Katz, 2020) e utilização da TRI como proposta de abordagem de otimização da precisão das classificações, categorizando os itens por níveis de dificuldades e aplicando o método de *Ensemble* em Aprendizado de Máquina atribuindo desta forma pesos aos itens classificados (Chen; Ahn, 2020). Ao realizar no grupo do cruzamento de IA e TRI da revisão da literatura aplicando *cluster* como filtro no grupo não foram obtidos trabalhos significativos, destacando novamente esta lacuna como área a ser explorada em pesquisas acadêmicas para o cruzamento de IA e TRI filtrando a pesquisa por *cluster.*

Os trabalhos envolvendo Aprendizado de Máquina e TRI, evidenciam soluções que abrangem: o estabelecimento de uma significância mais específica para os itens tratados por algoritmos de Aprendizado de Máquina (Martínez-Plumed *et a*l., 2016), a classificação de dados utilizando algoritmos supervisionados e o Parâmetro de Dificuldade b da TRI (Martínez-Plumed *et al*., 2019), as aplicações dos parâmetros da TRI em problemas de regressão (Moraes *et al.,* 2022), a otimização de reconhecimento de fala (Oliveira *et al.*, 2022) utilizando modelos desenvolvidos a partir da TRI como o $\beta^3 - IRT$ (Chen *et al.*, 2019a), a avaliação de estratégias de aprendizado utilizados em *meta-learning* (Sousa *et al*., 2016), o aprofundamento de aprendizado utilizando a TRI

em *Ensemble* e outras aplicações que combinam o Aprendizado de Máquina com TRI em estratégias supervisionadas.

Não foi identificado que houve um aprofundamento na literatura da aplicação de TRI e algoritmos de *clustering* ao ser realizada uma busca no grupo do cruzamento de Aprendizado de Máquina e dimensionalidade aplicando *cluster* como filtro no grupo; na verdade, destacam-se poucos trabalhos relevantes, como a obtenção de um modelo de agrupamento probabilístico para classificação de discurso de ódio no *Twitter* (Ayo *et al*., 2021), e aplicação de modelos de TRI unidimensionais ou bidimensionais em dados reais e simulados para examinar casos de algoritmos de *clustering* na identificação de grupos de itens com diferentes dimensões de dados (Bergner *et al*., 2012).

No grupo do cruzamento de Aprendizado de Máquina e dimensionalidade aplicando *saúde* como filtro destacam-se trabalhos relevantes abordando os tópicos deste parágrafo que foram citados no início desta Tese, destacando-se a importância das soluções utilizando Aprendizado de Máquina em diversos contextos. Todavia, há problemas onde a dimensionalidade dos dados proporcionava distintos fatores causadores de dificuldades para as soluções utilizando Aprendizado de Máquina. Para a TRI há críticas relacionadas à estrita dimensionalidade, onde modelos fortes baseados na TRI são obtidos de instrumentos de medição, os quais devem ser coerentes com os dados avaliados. Estes modelos TRI podem ser unidimensionais ou multidimensionais, dependendo da natureza dos dados e o objetivo da análise (Cheng; Cheon; Katz, 2020) (Tezza *et al*., 2018)(Chang; Vattikuti; Chow, 2019). Outro grupo de dados obtido do cruzamento da revisão sistemática destacam-se relevantemente os trabalhos aplicando *saúde* como filtro no grupo de IA e TRI; foram realizadas avaliações transversais utilizando a TRI como ferramenta de análise em dados médicos (Schapira; Walker; Sedivy, 2009), pesquisas sobre saúde pública (Gomes *et a*l., 2018), uso da TRI conjuntamente com Aprendizado de Máquina na predição de mortalidade em unidades intensivas de saúde (Kline *et al.*, 2020), aplicação de soluções aplicando *clusters* em diagnóstico e tratamento da diabetes (Hassan; Mollick; Yasmin, 2022) e outros trabalhos onde a TRI contribuiu como solução para otimização de modelos e ajustes nos dados. A complexidade computacional foi destaque em trabalho envolvendo a aplicação da computação em soluções que exigiam segurança e processamento (Yuan;

Yang, 2022) (Ding *et al*., 2024)(Diogo; Francisco de, 2023)(Wang *et al*., 2024b)(Anand; Kumar, 2023). A análise de complexidade é tradicionalmente calculada em função do tempo de processamento e, para esta avaliação, a grandeza utilizada é medida pelo método utilizado no O-Grande (Benabdellah; Benghabrit; Bouhaddou, 2019), fator este que foi avaliado nos experimentos desta Tese. No grupo do cruzamento de Aprendizado de Máquina e dimensionalidade aplicando *Ensemble* como filtro no grupo, e no grupo do cruzamento de IA e TRI aplicando *Ensemble* como filtro no grupo destacam-se trabalhos relevantes como a aplicação do método de *Ensemble* na classificação de dados desbalanceados (Puri; Gupta, 2022), predição em amostragens de dados utilizando Aprendizado de máquina (Kumar; D., 2016) e outros trabalhos onde o método de *Ensemble* é largamente utilizado para combinar resultados e otimizar a confiança da decisão baseada no conjunto de classificadores utilizados. Destaca-se nos trabalhos obtidos pela revisão sistemática o uso da TRI em conjunto com *Ensemble* na detecção de anomalias não supervisionadas (Kandanaarachchi, 2022a). O uso do algoritmo de votação por maioria ponderada é proposto para resolver a determinação do vetor de correspondência para os cálculos comparativos dos acertos, observando que os algoritmos não-supervisionados não pré-dispõem das classes previamente determinadas (Chen; Ahn, 2020), portanto indicando o método a ser utilizado nos experimentos desta Tese.

No grupo do cruzamento aplicando *tomada de decisão* como filtro no grupo de Aprendizado de Máquina e dimensionalidade, também se destacam trabalhos relevantes onde a tomada de decisão é fundamental para os profissionais de saúde para a obtenção dos *insights* necessários a partir dos distintos tipos de dados, para uma conduta confiante e otimizada de suas decisões. Neste grupo, utilizam-se diferentes tipos de análises, como descritivas, diagnósticas, prescritivas e outras. Estas decisões são baseadas em um processo contínuo de autoaprendizagem, combinando diferentes tecnologias, como *Machine Learning, Big Data Analytics*, IA, Processamento de Linguagem Natural, Visualização de Dados, *Deep Learning*, dentre outros, permitindo assim melhorias significativas nos resultados (Behera; Bala; Dhir, 2019). Ainda utilizando a *tomada de decisão* como filtro no grupo de Aprendizado de Máquina e dimensionalidade destacam-se trabalhos como utilização de algoritmos de *clustering* para avaliar

recorrências de doenças em dados de saúde, e planejamento baseados em similaridades de agrupamentos obtidos (Wartelle *et al*., 2021), uso de combinações de árvores (*i.e. random forests*) como método para decisão ajustadas a subamostras dos dados de saúde (El-shafeiy, Abohany, 2020), e outros trabalhos relacionados ao filtro aplicado nos resultados no grupo de Aprendizado de Máquina e dimensionalidade. Foram relacionados trabalhos onde foram avaliados modelos em dados desbalanceados utilizando *Machine Learning,* com o objetivo de avaliar e modelar soluções baseadas em computação para o diagnóstico de doenças mais atuais, e outras doenças que ainda persistem como grandes causas de mortalidades como COVID-19 (Dorn *et al*., 2021), câncer cervical (Aora; Dhawan; Singh, 2020) e outros trabalhos com relevante significância para a revisão sistemática realizada.

No grupo do cruzamento de IA e TRI aplicando *tomada de decisão* como filtro no grupo, destacam-se trabalhos relevantes como os algoritmos de Aprendizado de Máquina como suporte para a tomada de decisão em diagnostico em saúde, mediante respostas em questionários aplicados para o diagnóstico, evidenciando a aplicação da TRI como proposta de solução nas decisões dos classificadores de Aprendizado de Máquina (Gonzalez, 2021) e o uso da TRI em situações em que a estatística é requerida para um fator decisório, aplicando desta forma a TRI conjuntamente com um método preditivo em saúde (Chang *et a*l., 2022).

No grupo do cruzamento de Aprendizado de Máquina e dimensionalidade aplicando o *SMOTE* como filtro, surgem trabalhos relevantes como a avaliação de superamostragem de dados em datasets desbalanceados utilizando o SMOTE como método para geração de dados artificiais para balancear os conjuntos a serem particionados pelos algoritmos de *clustering* (Tao *et al*., 2019). Para o grupo do cruzamento de IA e TRI aplicando *SMOTE* como filtro não houve trabalhos relevantes para a pesquisa desenvolvida nesta Tese.

Como complemento da revisão sistemática foram incluídos trabalhos englobando outras técnicas aplicadas em algoritmos de *clustering*. O algoritmo k-means é aplicado em diversos trabalhos citados anteriormente, porém há outros algoritmos clássicos de *clustering* que também são utilizados para a formação de agrupamentos. Alguns destes algoritmos foram utilizados nos

experimentos desta Tese para avaliar a aplicação da TRI como método para determinar os níveis de dificuldades na formação dos grupos

Há outros algoritmos mais atuais que foram utilizados nas formações de agrupamentos como o OPTICS e o DBSCAN na avaliação de representação dos dados, análise de densidade ou agrupamento correlacionado com a classificação dos dados (Anand; Kumar, 2023; Hahsler; Piekenbrock; Doran, 2016; Pourbahrami *et al.*, 2020). Há outras técnicas que também devem ser citadas. O algoritmo *MOO-clus* que utiliza como técnica uma estrutura de recozimento multiobjetivo simulada para otimizar simultaneamente dois objetivos: Compacidade e separação de agrupamentos (Scharya; Saha, 2014). O algoritmo de *GK-means,* que utiliza como técnica uma otimização do algoritmo *K-means* para uma medida de similaridade de terceira ordem, determinando critérios de parada para encontrar automaticamente o número de *clusters* (Acharya; Saha, 2014). O algoritmo *Suffix Tree Clustering* (STC), que utiliza como técnica um algoritmo de tempo linear e incremental que determina a formação de agrupamentos com base em frases compartilhadas entre arquivos de texto (Saini *et al*., 2021). O algoritmo LINGO, que utiliza como técnica a estimação das frequências de frases de textos que são extraídas para matrizes de sufixos e posteriormente comparadas com a descrição do grupo de tópicos obtidos com análise semântica latente (Saha; Mitra; Kramer, 2018). O algoritmo MMOEA que utiliza como técnica a visualização múltipla para gerar diferentes soluções de *clustering*, selecionando posteriormente uma combinação de agrupamentos para formar uma solução final de *cluster* (Mitra *et al.*, 2018; Saha; Mitra; Kramer, 2018; Saini *et al.*, 2021). Nos experimentos realizados nesta Tese foram utilizados algoritmos clássicos e alguns dos algoritmos citados neste parágrafo, com o objetivo de compará-los nas dificuldades de atribuição de elementos para a formação dos agrupamentos nos diferentes níveis de complexidade.

Para avaliar a qualidade dos agrupamentos obtidos, são necessários métodos que validam a formação dos *clusters*. Estes métodos estão representados em sua maioria nas validações de *clustering* internas (Liu *et al.*, 2013) e externas (Wu *et al.*, 2009). Estas validações são importantes para referenciar o grau de pureza dos dados agrupados e a qualidade da estabilidade do *cluster,* e medir a qualidade dos grupos selecionados, referenciando a

eficiência dos algoritmos de *clustering* na alocação e representação dos elementos (Anand; Kumar, 2023). Estes métodos de avaliação podem ser comparados ao método de correção de avaliações em educação, e desta forma pode ser traçado um paralelo para a comparação entre as atribuições dos elementos nos referentes agrupamentos e a pertinência dos elementos previamente rotulados, proporcionando uma extensão ao método semi-supervisionado utilizados na avaliação de algoritmos de *clustering* (Aggarwal; Reddy, 2014)(Diogo; Francisco de, 2023)(Qiang *et al.*, 2023).

Nos próximos tópicos deste capítulo estão descritos os conceitos importantes para a compreensão destas limitações e fatores correlacionados, contextualizando a proposta desta Tese. Estes tópicos foram organizados nos seguintes itens: aprendizado de máquina, fatores que influenciam o desempenho dos algoritmos de *clustering*, sistemas de avaliação computacional, Teoria da Resposta ao Item, e avaliação de probabilidades complexas.

2.2. Aprendizado de Máquina

Machine Learning (i.e. aprendizado de máquina) é um campo da inteligência artificial que se concentra no desenvolvimento de algoritmos e modelos que permitem que computadores e sistemas realizem tarefas cujos parâmetros internos se ajustam automaticamente a partir dos dados, em vez de serem explicitamente programados para cada tarefa. Essencialmente, a aprendizagem de máquina permite que máquinas ajustem seu comportamento ou façam previsões com base em padrões detectados nos dados. Neste processo os parâmetros utilizados nos algoritmos de *Machine Learning* são otimizados para a obtenção dos melhores ajustes dos dados, e atribuição do máximo possível de elementos aos respectivos grupos de forma correta.

Para avaliar a dificuldade das amostras dos *datasets* e a capacidade de atribuição em agrupamentos simultaneamente, deve ser observado se a atividade a ser desempenhada é supervisionada, não-supervisionada, por reforço ou outros objetivos. Para os algoritmos de *Machine Learning* supervisionadas o aprendizado ocorre pelo ajuste dos modelos de forma gradativa à amostragem dos dados no grupo de treinamento. O método não-supervisionado realiza ajustes independentes da fase de treinamento,

dependendo apenas de suas habilidades matemáticas e estatísticas implícitas, exigindo uma maior atenção, pois a decisão está totalmente condicionado a autonomia do algoritmo (Weiss; Khoshgoftaar; Wang, 2016) e este método exige que o resultado (*i.e.* agrupamento) seja o mais preciso possível em atingir o objetivo especificado (Huang; Yu; Gu, 2017). Os algoritmos semi-supervisionados são utilizados nas demandas tecnológicas atuais, pois partem de uma base de conhecimento prévio e atualiza o modelo a cada iteração de novos dados não identificados (i.e. não rotulados) (Sanches, 2003). A Aprendizagem por Reforço consiste em penalizar os erros dos algoritmos e bonificar os acertos (Osinenko; Dobriborsci; Aumer, 2022); e o método de Transferência de Aprendizagem consiste em atualizar ou transferir o conteúdo/conhecimento adquirido para o modelo ou tarefa realizada no processo de modelagem (Weiss; Khoshgoftaar; Wang, 2016). Será utilizado nos experimentos desta Tese os algoritmos de clustering, os quais utilizam o método não-supervisionado. Na seção seguinte será explicado o conceito de algoritmos de *clustering* e os métodos utilizados na formação e fatores que influenciam na performance da formação dos agrupamentos realizados pelos algoritmos de c*lustering.*

2.3. Fatores que influenciam o desempenho dos algoritmos de *clustering*

Para um algoritmo de *clustering* obter sucesso no objetivo especificado, faz-se necessária que uma fonte de dados seja apresentada para a aplicação do um método não-supervisionado ou semi-supervisionado.

Ao avaliar a capacidade de um algoritmo em resolver determinados problemas, é importante observar se foi obtida uma solução válida ou apenas uma resposta não validada pelos especialistas humanos. Para alcançar este objetivo se faz necessária a análise da capacidade do algoritmo em resolver problemas, como a dimensionalidade dos dados a serem agrupados, a distribuição dos dados, o pré-processamento de dados, a escolha da métrica utilizada nos dados agrupados, a inicialização dos centroides, a definição do Número de clusters e o tipo de algoritmo (Wang *et al*., 2024a). Além destes fatores, há questões consideradas difíceis para os algoritmos de *clustering*,

como a atribuição dos elementos de forma ideal nos respectivos agrupamentos, ou da contextualização dos grupos formados como resposta para os seres humanos do problema abordado.

As próximas seções abordam conceitos como os objetivos dos algoritmos, os problemas do tamanho das amostras, balanceamento das classes e problemas de fronteira. Estes tópicos são importantes para a estruturação dos *datasets* que servirão de base para a formação dos grupos pelo algoritmo e de sua generalização de novos dados. Serão detalhados nesta seção os fatores que podem reduzir o bom desempenho dos algoritmos de *clustering* em seus objetivos.

2.3.1. Objetivo do algoritmo de clustering

Os objetivos dos algoritmos em pauta são as resoluções de ações previamente designadas, que podem ser: previsão, classificação, *clusterização* e regressão. Nesta Tese a atribuição dos elementos nos respectivos grupos e associá-los de forma semisupervisionada às respectivas rotulações é o objetivo a ser realizado pelos algoritmos de *clustering* nos experimentos realizados. Portanto, aos algoritmos devem ser atribuídos previamente aos objetivos determinados, e a partir destes objetivos deve-se estimar se o algoritmo obtém o maior índice de acertos em relação ao conjunto de dados formado (Lyndon et al., 2018).

2.3.2. Tamanho do grupo para a formação dos agrupamentos

Os algoritmos de *clustering* não necessitam obrigatoriamente de um grupo de treinamento, pois aprendem diretamente com os dados e suas distribuições em um espaço baseado nas características dos *datasets*. Esta distribuição dos elementos neste espaço está geralmente relacionada com os métodos de dissimilaridade utilizados nos *cluster*s. A dissimilaridade é calculada pelas distâncias entre elementos *intra-cluster, inter-cluster*, centróides e densidade dos elementos (Ghesmoune; Lebbah; Azzag, 2016). Quanto maior for a quantidade e mais variados forem os dados a serem agrupados, mais complexa é a ação de organizar os elementos nos respectivos agrupamentos. Os elementos agrupados em classes desiguais (*i.e*. desbalanceadas) proporcionam um

comprometimento no desempenho dos objetivos dos algoritmos, pois os dados devem estar bem representados e equivalentemente distribuídos em seus respectivos grupos, esta diferenciação nas quantidades dos elementos, proporcionam classes majoritárias e minoritárias.

2.3.3. Classes majoritárias e minoritárias

O desbalanceamento nos grupos ou classes pode ocorrer pela diferença de registros nos *datasets* entre os grupos de maior quantidade de exemplos, majoritárias, e as de menor quantidade, minoritárias. Devido a estes fatores, ocorrem erros na atribuição de elementos aos agrupamentos, e o algoritmo torna-se ineficiente em seu objetivo (Jayadeva *et al.*, 2019).

Como a atribuição dos centroides está vinculada à representação dos elementos de características semelhantes, o grupo majoritário geralmente proporciona uma representação vetorial dos grupos de forma equivocada. Uma solução aplicada no balanceamento de classes em algoritmos que utilizam o método supervisionado é balancear os elementos do grupo de treinamento suprimindo dados ou gerando dados artificiais, reduzindo as desproporções das quantidades de amostras analisadas pelos algoritmos. Este balanceamento é realizado pelos métodos de Subamostragem ou de Sobreamostragem de dados (Ghesmoune; Lebbah; Azzag, 2016).

A combinação de Subamostragem com Sobreamostragem para melhorar o desempenho na identificação das classes consiste em realizar uma organização na vizinhança das amostras minoritárias, removendo objetos da classe majoritária. O objetivo é simplificar a tarefa de ajustes nas classes de amostras da classe minoritária, realizando uma super amostragem seletivamente com as características da classe minoritária, criando amostras sintéticas mais próximas das características minoritárias na zona menos segura, ou seja, onde a classe majoritária é dominante (Nnamoko; Korkontzelos, 2020).

2.3.4. Subamostragem

A Subamostragem (i.e. *Undersample*) é uma estratégia que preferencialmente realiza a redução de amostras do conjunto de dados da classe majoritária como pode ser observada na Figura 1. Esta técnica utiliza os dados

disponíveis no *dataset* e reduz os dados da classe majoritária ao mesmo número de amostras da classe minoritária, igualando as quantidades de elementos e balanceando as classes (Lin *et al.*, 2017).

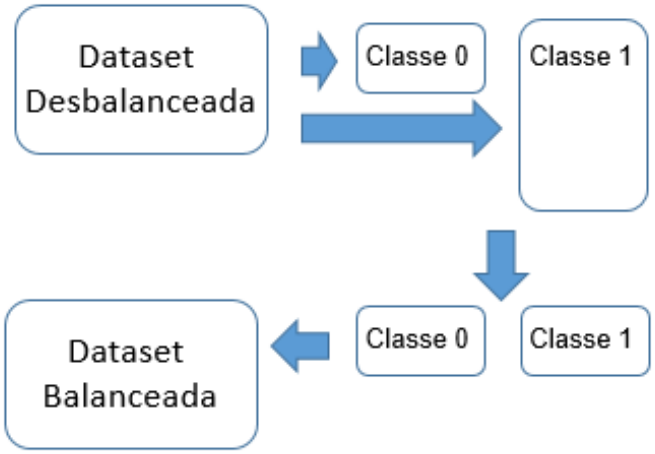

Figura 1 – Representação de *undersample* da classe majoritária.
Fonte**:** Leite-Filho; Melo (2022).

2.3.5. Sobreamostragem

A Sobreamostragem *(i.e. Oversample) é* uma técnica que ajusta as proporções dos dados no conjunto da classe minoritária, pela geração de dados artificiais. Igualando as classes e balanceando os grupos conforme pode ser observado na Figura 2. Uma das técnicas mais utilizadas na geração de dados artificiais é o *Synthetic Minority Oversampling Technique* (*SMOTE)* que foi utilizada nesta Tese (Woteki; Kineman, 2003).

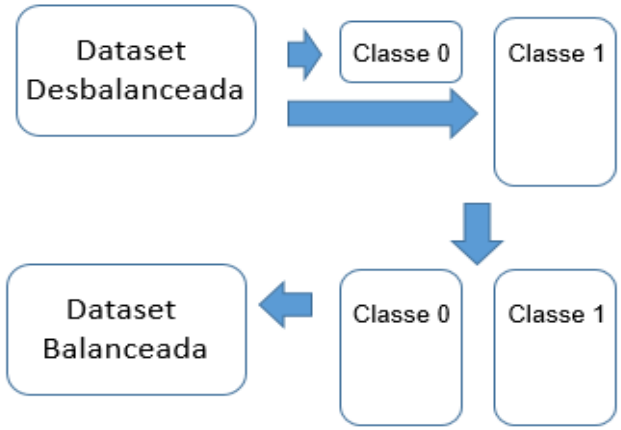

Figura 2 – Representação de *oversample* da classe minoritária.
Fonte**:** Leite-Filho; Melo (2022).

Na seção seguinte será explicado o funcionamento do algoritmo *SMOTE* e aplicações em soluções utilizando agrupamentos de dados.

2.3.6. Synthetic Minority Oversampling Technique (*SMOTE*)

Diferenciar os níveis de complexidade de amostras balanceadas ou não balanceadas é uma tarefa que exige entendimento dos dados amostrados. O não balanceamento dos dados torna complexa a decisão dos usuários em confiar nos agrupamentos formados por sistemas que utilizem a IA.

O *SMOTE* é um método de Sobreamostragem de registros da classe minoritária em que há a criação artificial de novos registros pela interpolação dos dados na classe minoritária. Esses registros são criados selecionando aleatoriamente um ou mais *k*-vizinhos mais próximos dos registros da classe minoritária do *dataset* (Lin *et al.*, 2017). *SMOTE* é um método utilizado para gerar dados artificiais para abordagem de clustering. *SMOTE* foi combinado com outros algoritmos, gerando alguns variantes, como *SMOTEBoost*, onde a Sobreamostragem inclui um procedimento de *boosting*; *Borderline-SMOTE*, no qual as amostras de dados minoritários são geradas em regiões inseguras; *safe-Level-SMOTE*, que proporciona amostras de dados mais seguras; *LN-SMOTE*, que explora as informações locais sobre as vizinhanças de amostras de dados superamostradas; e *MWMOTE*, que otimiza a geração de dados sintéticos (Nnamoko; Korkontzelos, 2020).

Para cada uma destes métodos de Subamostragem ou Sobreamostragem de dados há desafios próprios a serem considerados, os principais sendo: Quando a classe majoritária é reduzida pode acarretar perda de informação (Zhou *et al.*, 2020), ou ao gerar dados artificiais na classe minoritária, pode ocorrer a alteração da localização dos centróides dos *clusters* (Douzas; Rauch; Bacao, 2021). Reduzindo ou acrescendo novos elementos no *input* dos dados de algoritmos de *clustering* proporciona dificuldades para a formação dos agrupamentos. Estas variações de *inputs* foram avaliadas nos experimentos desta tese. Além do desbalanceamento dos tipos de dados, a dimensionalidade dos dados também influencia nas formações dos agrupamentos, e se constitui num fator que causa grande impacto e complexidade na obtenção de bons modelos computacionais.

2.3.7. Seleção de dimensionalidade dos dados

A criação contínua de distintos conjuntos de dados multidimensionais é cada vez mais comum numa vasta gama de campos de investigação, como saúde, demografia ou economia. Estes conjuntos de dados geralmente se constituem de um grande número de observações e atributos (Mohedano-Munoz et al., 2021).

A dimensionalidade ou as características dos dados dos *datasets* estão associadas ao número de atributos, e a partir destas características são calculados os componentes que determinam as localizações de cada item e, por fim, o agrupamento correspondente. A alteração no quantitativo das características tem um impacto relativamente considerável, seja na redução dos atributos, *feature selection*, ou criação de novos atributos, *feature extraction* (Khalid; Khalil; Nasreen, 2014). Cada um destes métodos tem suas características e técnicas em particular, porém estas técnicas são estudadas para: Avaliar melhorias nos algoritmos, reduzir o custo computacional e propor algoritmos mais eficientes, mesmo que as melhorias sejam significativamente mínimas. Após a formação dos agrupamentos podem ocorrer pontos de indeterminação de pertinência dos elementos em seus respectivos agrupamentos, devido ao posicionamento dos elementos nos subespaços formados, incorrendo em dificuldades em setores dos agrupamentos denominadas regiões de fronteira.

2.3.8. Problemas de fronteira

O espaço *n*-dimensional no qual os dados estão inseridos pode ser caracterizado como sendo o conjunto de todos os vetores possíveis com n atributos ordenados e arbitrários (características). Um dataset é um subconjunto finito desse espaço, possivelmente formado por diversos grupos interpenetrantes. Muitas vezes os elementos dos agrupamentos tornam-se sobrepostos em áreas de fronteira ou limítrofe (*borderline*). A dificuldade de avaliar a pertinência dos elementos nestas áreas pode proporcionar erros de agrupamento. Este problema pode ocorrer quando há desbalanceamento de classes (Pratap; Singh, 2023).

Os agrupamentos podem ser categorizados de forma mais rigorosa com respeito à pertinência dos elementos como as partições *Hard* (Beleites; Salzer; Sergo, 2013), como CRISP ou *Fuzzy*, em que a pertinência dos elementos nos subconjuntos é e não discreta, cujos valores estão em [0,1] (Brouwer, 2009). Para permitir a comparação destes métodos de avaliação de agrupamento, é necessária uma medida da qualidade do agrupamento, principalmente em questões específicas de saúde (*i.e.* genoma) (Wartelle *et al*., 2021). Estas avaliações de validação dos elementos nos respectivos clusters podem ser realizadas pelas métricas citadas na seção 2.4 desta Tese.

A probabilidade ou incerteza da pertinência dos elementos nos agrupamentos determinados pelos algoritmos de *clustering* proporciona casos recorrentes de incertezas para previsões, como probabilidades posteriores, mas também pode ser o caso para rótulos de referência. As misturas ou sobreposições das classes subjacentes a grupos homogêneos ou heterogêneos, em que a heterogeneidade não é resolvida pela medição, proporcionam indeterminações em casos como de aplicações biomédicas, onde surgem referências incertas (*i.e.* a incerteza de um patologista determinar células saudáveis ou patológicas) (Beleites; Salzer; Sergo, 2013).

As associações parciais dos agrupamentos determinados em classes podem ser contextualizadas em três níveis, como: (*i*) as *previsões suaves*, que são frequentemente consideradas um resultado intermediário, e são então reforçadas por limites, presentes nas aplicações utilizando discriminante linear ou quadrática ou regressão logística, *proporção de votos* de k-vizinhos mais próximos no kNN, *random forests e* outras aplicações similares; (*ii*) *as amostras de treinamento*, que raramente são utilizadas e estão presentes nas aplicações utilizando regressão logística, redes neurais artificiais ou análise discriminante de mínimos quadrados parciais *e* outras aplicações similares; e (*iii*) *as amostras de teste suave*, ou seja, amostras com referência ambígua ou incerta, utilizados em diagnóstico como padrão-ouro de qualidade (Beleites; Salzer; Sergo, 2013).

Após as descrições dos fatores que dificultam a determinação de elementos nos agrupamentos em regiões limítrofes, serão abordadas na seção seguinte os métodos utilizados pela computação na avaliação dos resultados obtidos.

2.4. Sistemas de avaliação computacional

Os conceitos, diferenças e aplicações das métricas de avaliações estatísticas utilizadas na saúde humana e nos modelos computacionais estão descritos nesta seção da Tese, assim como a estimação do impacto nas métricas pelas dificuldades dos itens a serem ordenados em agrupamentos.

Nesta Tese foram utilizados algoritmos de *clustering* não-supervisionados, e para avaliar os resultados obtidos pelos agrupamentos foram utilizadas técnicas de mensuração dos grupos formados. Os métodos mais comuns na literatura para a avaliação de agrupamentos são: *Silhouette Score* (Wang *et al.*, 2017), *Rand Index* (Santos; Embrechts, 2009), *Normalized Mutual Information* (Estévez *et al*., 2009), *Internal Cluster Validation Index* (Rendón *et al.,* 2011), *Davies-Bouldin Index* (Wijaya *et al*., 2021) e *Calinski-Harabasz Index* (Lukasik *et al.,* 2016) estes métodos avaliam a eficiência dos agrupamentos formados baseados nas características de cada algoritmo. Nesta Tese também foram utilizadas outras métricas que são comumente utilizadas nas avaliações biológicas como a Similaridade, Especificidade e a Acurácia, como métodos de referências utilizados tradicionalmente na determinação de resultados obtidos em saúde.

2.4.1. Métodos avaliativos e métricas estatísticas de performance

Existem técnicas que avaliam as quantidades de acertos e erros de algoritmos, como a Precisão e outros. Contudo, estes métodos têm como objetivos principais na computação a identificação de grupos ou classes. Para avaliar separadamente os fatores que contribuem para indicar os algoritmos com melhor rendimento estatístico, não são calculando os níveis de importância de cada registro utilizado (Hossin; Sulaiman, 2015). Para os analistas dos resultados obtidos na aplicação dos algoritmos de Aprendizado de Máquina, são observados fatores externos que influenciam os resultados dos testes, como as razões para a formação de grupos, características dos indivíduos, faixa etária, dentre outros (Cattell, 1943). Estes fatores são necessários para validar de forma qualitativa os resultados obtidos para aplicações de pesquisas em seres humanos, medicamentos, técnicas envolvendo tratamentos psicológicos e outras aplicações.

Para a computação as métricas também têm este propósito, e são calculadas pela diferença calculada nos parâmetros de confiabilidade entre Verdadeiro-Positivo (VP), Verdadeiro-Negativo (VN), Falso-Positivo (FN) e Falso-Negativo (FN) das Equações 1 – 3, onde estão representadas respectivamente *Accuracy ou Precisão (Acc), Specificity ou Especificidade (Spe) e Sensitivity ou Sensibilidade (Sen).*

$$\mathrm{ACC} = \frac{VP + VN}{\mathrm{VP} + \mathrm{VN} + \mathrm{FP} + \mathrm{FN}} \quad (1)$$

$$SEN = \frac{\mathrm{VP}}{\mathrm{VP} + \mathrm{FN}} \quad (2)$$

$$SPE = \frac{\mathrm{VN}}{\mathrm{VN} + \mathrm{FP}} \quad (3)$$

As definições de Precisão, Sensibilidade e Especificidade foram amplamente discutidas (Powers, 2020)(Beale, 2015)(Van Stralen *et al.*, 2009), sendo consideradas nas métricas como: Precisão, assim como em medicina diagnóstica, sendo a quantidade de concordância entre os resultados do teste de diagnóstico *em relação ao teste de referência; Especificidade, sendo a proporção de casos* negativos preditos como positivos (tnr), os quais são negativos previstos; *e* Sensibilidade, sendo a proporção de casos positivos que são corretamente preditos como os positivos (+P).

Dentre as métricas citadas no parágrafo anterior, o ponto comum em todas são os exemplos dos dados obtidos dos conjuntos de dados reais ou simulados, e avaliados pelas métricas nos resultados dos testes. Contudo, a proposta a ser analisada deve ser desenhada para determinar a atribuição de pertinência dos elementos nos agrupamentos complexos, avaliando posteriormente o quanto o algoritmo é eficiente para resolver determinada tarefa ou objetivo nos correspondentes níveis de dificuldade, mediante a métrica utilizada. Para avaliar a qualidade dos agrupamentos formados serão utilizados os índices de medidas de validação dos *clusters*

2.4.2. Índices de validação de cluster

Os dados podem ser organizados pelos algoritmos de *clustering* em conjuntos distintos ou não (*i.e.* Fuzzy). E esta organização está contida nas

partições dos agrupamentos obtidos pelos distintos algoritmos de *clustering*. Para avaliar esta formação de grupos de elementos distintos ocorre a validação destes elementos nos *clusters*. Existem dois tipos de medidas de validação de *cluster,* os quais estão baseados em critérios de formação dos elementos externos e internos aos agrupamentos (Halkidi *et al.,* 2002a)(Jain; Dubes, 1988). Nas medidas de validação de *cluster* externa, as medidas avaliam até que ponto a estrutura de agrupamento descoberta por um algoritmo de agrupamento corresponde a alguma estrutura externa (*i.e*. distâncias entre elementos dos diferentes *clusters*). Para a medidas de validação de *cluster* interna, contudo, a avaliação do *cluster* baseia-se apenas no próprio agrupamento, sem qualquer informação adicional proveniente dos dados (Wu *et al*., 2009).

Há um conjunto de equações conhecidas na literatura (Desgraupes, 2017) para calcular os índices de medidas de validação de cluster internos e externos. Destas equações foram selecionados os seguintes índices internos: Silhouette (Wang *et al.*, 2017), Dunns (Jain *et al.*, 2022), Davies-Bouldin (Thomas; Peñas; Mora, 2013) e PBM (Saini *et al.*, 2021); e os índices externos: Foulkes-Mallows (Jain *et al.*, 2022), Rand (Brouwer, 2009) e Jaccard (Brouwer, 2009).

Os índices de qualidade são denotados pela mesma variável *C*, onde *d* representa a função distância entre dois pontos conhecidos. Os índices avaliam a medida interna, externa e de estabilidade para o desempenho entre os *clusters* reais e clusters/grupos previstos, a partir das medidas de correspondência entre duas partições dos mesmos dados, e baseiam-se na forma como pares de objetos são organizados em uma tabela de contingência (Anand; Kumar, 2023)(Santos; Embrechts, 2009).

Considere um conjunto de *n* objetos S = {O_1, O_2, ..., O_n} e suponhamos que U = {u_1, u_2, ..., u_R} e V = {v_1, v_2, ..., v_C} representem duas partições distintas de S, tal que $S = U_{i=1}^{R} u_i = U_{j=1}^{C} v_j$ e $u_i \cap u_{i'} = \emptyset = v_j \cap v_j, para\ 1 \leq i \neq i' \leq R$ e $1 \leq j \neq j' \leq C$. Para essas partições de *S* é construída uma Tabela 1 de contingência, que exibe alguns indicadores sobre a configuração dos grupos entre U e V. Considerando o número possível de combinações de pares $\binom{n}{2}$ de elementos de *S*, definem-se: C_a é o número de pares de objetos em que, para cada par, ambos os elementos são inseridos no mesmo grupo, tanto em U quanto em V; C_b é o número de pares de objetos em que, para cada par, ambos

os elementos são inseridos no mesmo grupo em U, mas que são inseridos em grupos diferentes em V; C_c é o número de pares de objetos em que, para cada par, ambos os elementos são inseridos no mesmo grupo em V , mas em grupos diferentes em U; e C_d é o número de pares de objetos em que, para cada par, seus elementos são inseridos em grupos diferentes tanto de U quanto de V. As combinações dos pares podem ser organizadas na Tabela 1, relacionando as partições U e V (Desgraupes, 2017)(Rendón *et al*., 2011).

Tabela 1 – Tabela de contingência para comparar as partições U e V.

Partição	V	
U	Pares do mesmo grupo	Pares de diferentes grupos
Pares do mesmo grupo	C_a	C_b
Pares de grupos distintos	C_c	C_d

Os valores da Tabela 1 podem ser calculados pelas Equações 4 – 7:

$$C_a = \sum_{r=1}^{R}\sum_{c=1}^{C}\binom{t_{rc}}{2} = \langle \sum_{r=1}^{R}\sum_{c=1}^{C} t_{rc}^2 - n \rangle / 2 \qquad (4)$$

$$C_b = \sum_{r=1}^{R}\binom{t_r}{2} - C_a = \langle \sum_{r=1}^{R} t_r^2 - \sum_{r=1}^{R}\sum_{c=1}^{C} t_{rc}^2 \rangle / 2 \qquad (5)$$

$$C_c = \sum_{c=1}^{C}\binom{t_c}{2} - C_a = \langle \sum_{c=1}^{C} t_c^2 - \sum_{r=1}^{R}\sum_{c=1}^{C} t_{rc}^2 \rangle / 2 \qquad (6)$$

$$C_d = \langle \sum_{r=1}^{R}\sum_{c=1}^{C} t^2{}_{rc} + n^2 - \sum_{r=1}^{R}\sum_{c=1}^{C} t_c^2 \rangle / 2 \qquad (7)$$

onde t_r representa o número de objetos classificados no *r*-ésimo subconjunto da partição *R*, t_c representa o número de objetos classificados no *c*-ésimo subconjunto da partição *C*, t_{rc} representa o número de objetos classificados no *r*-ésimo subconjunto da partição *R* e no *c*-ésimo subconjunto da partição *C.*

São classificados por Verdadeiro-Positivo (VP), Verdadeiro-Negativo (VN), Falso-Positivo (FN) e Falso-Negativo (FN) os pontos pertencentes aos agrupamentos formados, sendo NT correspondente a população avaliada, ou seja, o somatório de todos os pontos, onde são aplicados nas equações dos índices correspondentes as Equações de 8 – 14.

Os índices externos utilizados nesta Teses foram calculados pelas Equações de 8 – 10, representadas respectivamente por *Rand, Foulkes-Mallows* e *Jaccard*, e descritos individualmente a seguir.

William M Rand propôs um indicador denominado *Rand* ou *Rand* indexado para medir a qualidade de um agrupamento. O índice Rand, calculado pela Equação 8, é usado para encontrar o valor da Precisão quando a variável dependente não está presente no conjunto de dados. O valor de Rand assume valores entre [0,1], em que 0 representa o pior nível de concordância entre os agrupamentos e 1 o melhor nível, representando uma medição da qualidade de um agrupamento, quando o outro agrupamento comparado é o de referência (i.e. rotulado) (Rand,1971)(Rendón *et al*., 2011), equação adaptada de Desgraupes (2017) e Anand; Kumar (2023).

$$Rand = \frac{VP+VN}{NT} \quad (8)$$

O índice Foulkes-Mallows, calculado pela Equação 9, é utilizado para encontrar similaridade entre dois agrupamentos, utilizando uma matriz de confusão para determinar o valor de similaridade. O índice Foulkes-Mallow usa uma taxa preditiva positiva e uma taxa VP. Este índice é preferencialmente usado com medida de qualidade de agrupamentos do tipo hierárquico (Fowlkes *et al*., 1983)(Rendón *et al.*, 2011), equação adaptada de Desgraupes (2017) e Anand; Kumar (2023).

$$Foulkes\text{-}Mallows = \frac{VP}{\sqrt{(VP+VN).(VP+FP)}} \quad (9)$$

Paul Jaccard propôs o índice Jaccard como medida do coeficiente de similaridade para os dados. O índice Jaccard, calculado pela Equação 10, é popularmente usado para medir a qualidade do agrupamento formado para dados binários ou categóricos (Jaccard, 1912)(Rendón *et al*., 2011), equação adaptada de Desgraupes (2017) e Anand; Kumar (2023).

$$Jaccard = \frac{VP}{(VP+VN+FP)} \quad (10)$$

A medida de validação externa é a entropia, que avalia a “pureza” dos *clusters* com base nos rótulos de classe fornecidos. As medidas de validação

externa quantificam quão bem um agrupamento corresponde a uma partição de verdade (Liu *et al.*, 2013). As Equações de 8 – 10 utilizam os mesmos parâmetros de avaliação das Equações de 1 – 3, utilizando Verdadeiro-Positivo (VP), Verdadeiro-Negativo (VN), Falso-Positivo (FN) e Falso-Negativo (FN) como referência nas respectivas equações, diferentemente das Equações de 11 a 14 que utilizam outros métodos para avaliar os agrupamentos formados.

Os índices internos utilizados nesta Tese foram calculados pelas Equações de 11 – 14, representadas respectivamente por *Silhouette*, *Dunns*, *Daves-Bouldin* e *PBM*. Eles são apresentados individualmente a seguir.

O índice *Silhouette*, calculado pela Equação 11, é utilizado para testar o desempenho do algoritmo de *clustering*. O objetivo do índice *Silhouette* é encontrar a distância e expressar a proximidade dos dados no agrupamento em comparação com outros agrupamentos, medindo a proporção de distâncias inter e *intracluster*. A referência para o índice *Silhouette* é o intervalo [-1,1], que representa uma avaliação do agrupamento de ruim a bom, e a média das larguras do agrupamento (C_k) determina o agrupamento (S_k) da Equação 11 (Rendón *et al.*, 2011), equação adaptada de Desgraupes (2017) e Anand; Kumar (2023).

$$Silhouette = \frac{1}{K} \sum_{K=1}^{K} S_k \quad (11)$$

O índice Dunns, calculado pela Equação 12, indica que quanto maior for o resultado obtido, melhor é o agrupamento formado. Para obter um bom resultado, o índice Dunns depende de um agrupamento compacto e de menor variância, indicando a razão entre a separação mínima e o diâmetro máximo. Na Equação 12, d_{min} é a distância mínima entre pontos de diferentes agrupamentos e d_{max} a maior distância dentro do agrupamento (Rendón *et al*., 2011; Benabdellah; Benghabrit; Bouhaddou, 2019), equação adaptada de Desgraupes (2017).

$$Dunn = \frac{d_{min}}{d_{max}} \quad (12)$$

O índice Davies-Bouldin, calculado pela Equação 13, identifica conjuntos de agrupamentos compactos e bem separados, onde a distância média dos

pontos pertencentes ao cluster C_k ao seu baricentro G{k} é representada por δ_k. E para cada agrupamento k calcula-se o máximo M_k dos quocientes $\frac{\delta_k+\delta_{k\prime}}{\Delta_{kk\prime}}$ para todos os índices $k' \neq k$, onde $\Delta_{kk\prime}$ é a distância entre os baricentros de C_k e $C_{k'}$. O índice de Davies-Bouldin é o valor médio entre todos os agrupamentos das quantidades M_k (Rendón *et a*l., 2011), equação adaptada de Desgraupes (2017).

$$Davies\text{-}Bouldin = \frac{1}{K}\sum_{k=1}^{K} \underset{k' \neq 1}{max} \left(\frac{\delta_k+\delta_{k\prime}}{\Delta_{kk\prime}}\right) \quad (13)$$

O índice PBM (Pakhira, Bandyopadhyay e Maulik), calculado pela Equação 14, utiliza as distâncias entre os pontos, seus baricentros e as distâncias entre os próprios baricentros, onde E_W é a soma das distâncias de cada agrupamento ao seu baricentro e E_T a soma das distâncias de todos os pontos ao baricentro *G* de todo o conjunto de dados e *D* são os dados utilizados no agrupamento (Rendón *et al*. , 2011), equação adaptada de Desgraupes (2017).

$$PBM = \left(\frac{1}{K} \cdot \frac{E_T}{E_w} \cdot D_n\right)^2 \quad (14)$$

As medidas internas avaliam a qualidade de uma estrutura de agrupamento sem respeitar informações externas. Este tipo de avaliação de agrupamentos *CRISP* tem como dificuldades o ruído nos dados que podem ter um impacto significativo no desempenho de uma medida de validação interna. Provavelmente as medidas existentes tem um bom desempenho apenas em agrupamentos em forma de esfera (Liu *et a*l., 2013).

Para uma avaliação mais objetiva utilizam-se as referências dos acertos e erros nas atribuições dos elementos nos respectivos agrupamentos. Como os algoritmos de *clustering* não utilizam rótulos previamente atribuídos (i.e. não-supervisionado) se faz necessário determinar um método comparativo entre as atribuições dos elementos nos agrupamentos e a significância destas atribuições. Na seção seguinte serão apresentados métodos utilizados para realizar esta validação dos algoritmos de *clustering*.

2.4.3. *Avaliação da validação de elementos agrupados*

Os algoritmos de *clustering* são não-supervisionados, ou seja, não necessitam de um rótulo para aprendizado. As métricas citadas na seção anterior são utilizadas para referenciar o quanto um agrupamento é ideal, ou seja, calcula valores de referência para indicar quanto um algoritmo de *clustering* atingiu seu objetivo. O objetivo dos algoritmos de *clustering* é organizar os dados em conjuntos de elementos com características mais homogêneas possíveis.

Os algoritmos de *clustering* utilizam diferentes técnicas para atingir estes objetivos. A técnica tradicional é a não supervisionada, porém a técnica semi-supervisionadas também é utilizada amplamente em agrupamento de dados. Esta técnica avalia as determinação dos medoides em aplicações Fuzzy em algoritmos *multi-view* (Diogo; Francisco de, 2023), em classificações de imagens (Feng *et al*., 2021), em técnicas utilizando agrupamentos por grafos e *multi-view* (Qiang *et al.*, 2023), na avaliação da informação das partições dos conjuntos obtidos (Mei, 2019) e outras aplicações evolvendo a associação do conhecimento prévio de elementos previamente classificados e a atribuição dos não rotulados.

Há trabalhos sobre técnicas semi-supervisionadas aplicadas em algoritmos de *clustering* (*i.e. Multi-view Clustering*) em para soluções aplicadas nas características de dados e visualização única de dados relacionais (Diogo; Francisco de, 2023). Tomar as classes de dados rotulados como verdade é uma abordagem amplamente utilizada para classificar técnicas de agrupamentos em conjuntos de dados de *benchmark* (Jeon *et al*., 2022). O uso de rótulos previamente conhecidos é simples, e é a forma clássica de aplicar técnicas semi-supervisionadas com a abordagem de restrições pareadas foi usada em alguns trabalhos (Maraziotis, 2012; YIN *et al.*, 2019). As técnicas semi-supervisionadas foram utilizadas inicialmente na estrutura do algoritmo de aglomeração competitiva restrita aos pares (PCCA) (Grira; Crucianu; Boujemaa, 2005), usando restrições *must-link* e *can-link* para orientar o agrupamento. Estes processos foram aplicados em algoritmos de *clustering* do tipo *Fuzzy*, onde o processo indica quais pares de objetos devem ser colocados no mesmo agrupamento e quais pares devem ser colocados em agrupamentos distintos (Diogo; Francisco de, 2023).

Após a etapa de agrupamento dos elementos, é empregada uma função objetiva para tratar das violações das restrições, permitindo a correção das informações referentes às pertinências dos dados agrupados. Isso ocorre em qualquer abordagem de supervisão indireta. O uso desse nível de adesão possibilita a predefinição ou reutilização de adesões de elementos aos clusters estabelecidos durante o processo de agrupamento.

Estes métodos proporcionam ao algoritmo de *clustering* um *input* de dados para um *output* de agrupamentos a serem interpretados pelos especialistas humanos. Aplicações em áreas de reconhecimento de padrões, processamento de imagens, mineração de dados e outras aplicações de aprendizado de máquina necessitam que a matriz obtida nos métodos de agrupamento sejam interpretáveis, ou seja, ocorram uma associação objetiva com o problema abordado. Nestas matrizes podem ocorrerem problemas que exijam a fatoração de matrizes não negativas ou positivas (Gillis, 2020). Particularmente a a fatoração de matrizes não negativas ou *Nonnegative Matrix Factorization* (NMF) é um campo onde há dificuldades para a significância na modelagem de processos envolvendo diversas áreas, particularmente a de saúde. Outros métodos envolvendo problemas de NMF utilizam em suas soluções técnicas de agrupamento semi-supervisionado, *co-clustering*, agrupamento de consenso e agrupamento de grafos (Aggarwal; Reddy, 2014).

A significância das atribuições dos elementos agrupados e suas correspondências com os a esperança dos especialista humanos com os rótulos previamente conhecidos ou determinados pelos algoritmos de *clustering*, é de grande importância para a avaliação e determinação de algoritmos ótimos em obter bons resultados a partir dos dados amostrados (Leite-Filho; Melo, 2023). A atribuição de agrupamentos correspondentes às classes esperadas pelos especialistas humanos pode ser atribuída utilizando método de votação em *Ensemble*, esta técnica pode ser associadas com a TRI para avaliar as dificuldades na atribuição dos elementos aos agrupamentos realizados pelos algoritmos de *clustering* (Chen; Ahn, 2020)(Kandanaarachchi, 2022a).

Para avaliar objetivamente os comportamentos das pertinências dos elementos nos grupos formados, será admitida que os agrupamentos formados correspondam exatamente aos rótulos previamente conhecidos nos *datasets* (Dimitriadou; Weingessel; Hornik, 2001). Este processo utilizando o método de

votação em *Ensemble* associado com a TRI para avaliar as dificuldades dos elementos agrupados possibilita para esta Tese as condições de avaliar as dificuldades na formação de grupos. Onde o vetor de rótulos utilizados possibilita de forma simples a confirmação das atribuições dos elementos aos respectivos agrupamentos binários previamente definidos nos repositórios. Ajustar o método não supervisionado para condicionalmente supervisionado proporciona um método avaliativo para informar as quantidades de acertos e erros nas atribuições dos elementos nos agrupamentos. Na seção seguinte serão descritos os métodos de utilização de Ensemble referenciando algoritmos de c*lustering* e o método para determinar as possíveis classes alvo.

2.4.4. *Atribuição de elementos nos clusters ou classes*

Os algoritmos supervisionados têm como um dos seus objetivos a classificação. Para obter resultados mais precisos é utilizada a estratégias de combinação de resultados, conhecida por *Ensemble*. Esta técnica tem como melhorias nas aplicações utilizando estratégias na combinação de múltiplos classificadores, fusão de classificadores, comitês de redes neurais e diversas combinações desses esquemas (Dimitriadou; Weingessel; Hornik, 2001). A técnica de *Ensembles* é utilizada na computação para a ponderação da tomada de decisões em conjuntos de classes (Mohedano-Munoz *et al.*, 2021). Esta técnica apresenta uma série de motivações para o uso de conjuntos de algoritmos de *clustering*, proporcionando solução mais significativas, das que são utilizadas em conjuntos de classificação ou regressão, onde se está principalmente interessado em melhorar a precisão preditiva dos elementos nos respectivos grupos (Aggarwal; Reddy, 2014). Por estas razões a técnica pode ser ajustada para selecionar um método de determinação das classes, mediante os agrupamentos dos resultados dos algoritmos de *clustering*, obtenção de uma decisão mais próxima do esperando pelos analistas humanos (Dimitriadou; Weingessel; Hornik, 2001), melhor qualidade da solução, cluster robusto e seleção de modelo. reutilização de conhecimento, *clustering Multiview*, Computação Distribuída (Aggarwal; Reddy, 2014) e outras aplicações relacionadas.

Os métodos de algoritmos de cluster Ensemble são representados por: i) abordagens probabilísticas, ii) abordagens baseadas em co-associação e iii) métodos diretos e outros métodos heurísticos. Como exemplos destes métodos destacam-se o *Mixture Model for Cluster Ensembles (MMCE)*, *Bayesian Cluster Ensembles (BCE), Nonparametric Bayesian Cluster Ensembles (NPBCE), Pairwise Similarity-Based Approaches, Direct Approaches Using Cluster Labels, Graph Partitioning, Cumulative Voting, Applications of Consensus Clustering* e outros métodos aplicáveis aos algoritmos de *clustering* (Aggarwal; Reddy, 2014).

Dentre os métodos mencionados no parágrafo anterior, *Cumulative Voting* é uma aplicação importante para esta Tese, pois cada determinação de elemento nos respectivos agrupamentos, contribui como parte proporcional em uma votação direta ou probabilística para avaliar nos agrupamentos formados a solução de consenso em seus pontos de dados. Contabilizando os votos nas soluções básicas e limitados para determinar a adesão de cada objeto aos agrupamentos de consenso (Ayad; Kamel, 2008).

Nos trabalhos de Dubois; Prade (1980), Kacprzyk (1976) e Dimitriadou; Weingessel; Hornik (2001) foi utilizado o seguinte artificio utilizando *labels* conhecidos para a comparação das pertinências em agrupamentos e atribuições de classes. Onde, para a determinação de uma partição *P* de um determinado conjunto de dados {x_1,..., x_N} em k classes que representam idealmente um conjunto de M partições do mesmo conjunto. Definiu-se que cada uma dessas M partições é representada por uma matriz de membros $N \times k$ de uma matriz $U^{(m)}$, m = 1, ..., M, onde o elemento $u_{ij}^{(m)}$ da matriz $U^{(m)}$ é o grau de pertinência de x_i para a j-ésima classe da m-ésima partição. Denota-se que a i-ésima linha de $U^{(m)}$como $u_{ij}^{(m)}$, ou seja, $u_{ij}^{(m)}$ é o vetor de pertinência ou função de pertinência do ponto de dados x_i para a partição de $U^{(m)}$. A partição final P é codificada como uma matriz $N \times k$ com elementos p_{ij} e linhas p_i. Ao definir que os rótulos dos *clusters* são conhecidos, ou seja, este é um caso correspondente à classificação. Como função de dissimilaridade $h(U^{(m)}, P)$ entre $U^{(m)}$ e P usamos a distância quadrada média entre as funções de pertinência $u_i^{(m)}$ e p_i para todo x_i descritos na Equação 15.

$$h(U^{(m)}, P) = \frac{1}{N} \sum_{i=1}^{N} ||u_i^{(m)} - p_i||^2 \quad (15)$$

A partir da Equação 15 é observado que caso um elemento de $u_i^{(m)}$ e p_i são iguais a 1, por consequência os complementares serão iguais a 0. Desta forma é calculado o número relativo de pontos de dados x_i que pertencem a diferentes classes em $U^{(m)}$ e P. Desta forma a Equação 15 mede exatamente a classificação esperada (Dimitriadou; Weingessel; Hornik, 2001), proporcionando uma adaptação da Equação 15 na interpretação das pertinências dos elementos agrupados e a validação das correspondências com as classes originais nos *datasets*. Na seção seguinte as características da TRI serão apresentadas.

2.5. Teoria da Resposta ao Item

A TRI avalia as habilidades latentes dos estudantes e a proficiência nas avaliações, e é cada vez mais utilizada em diversos países. A TRI tem substituído a Teoria Clássica dos Testes (TCT) desde 2009, pelo fato de que a TCT apenas avalia os acertos e erros das questões. E o diferencial da TRI para a TCT é que a TRI possibilita avaliar fatores associados ao intelecto dos estudantes. Este fator é tão importante que a TRI está sendo aplicada há muitos anos no Brasil no Exame Nacional do Ensino Médio (ENEM) e em outros países como *Graduate Registration Examination* (GRE), no *Graduate Management Admission Test* (GMAT), no *Test of English as a Foreign Language* (TOEFL) e o *Scholastic Assessment Test* (SAT). No caso do SAT, o propósito é selecionar estudantes no exame nacional do ensino médio realizado nos Estados Unidos para ingresso nas universidades norte-americanas (El-Alfy; Abdel-Aal, 2008)

O método utilizado na TRI evoluiu do método Bayesiano, que realiza os cálculos pela probabilidade condicional, avaliando os eventos probabilísticos consecutivos baseados nos eventos já ocorridos, utilizado em experimento para a determinação dos ajustes dos melhores resultados dos parâmetros da TRI (Zhu; Gao; Zhang, 2021). Outras pesquisas com a TRI contribuíram para ajustes na busca de melhores modelos dos parâmetros da TRI, e está evoluindo com novos métodos propostos como a análise de dimensionalidades e amostragem pelo método de Gibbs (Fu *et al.*, 2021), ajustes do modelo de Parâmetros Logísticos da TRI pelo *Markov Chain Monte Carlo* (*MCMC*) (Chang; Yanyan, 2017), aplicações de avaliação de métodos estatísticos pelo Critério de

Informação Akaike (AIC) e Critério de Informação Bayesiano (BIC) (Lumley; Scott, 2015), análise utilizando *Marginal Maximum Likelihood Estimation* (MMLE) (Kieftenbeld; Natesan, 2012), e outros métodos que procuram ajustes de melhorias de modelos da TRI.

O emprego da TRI contribuí para ajustes de modelos estatísticos confiáveis, tornando a otimização do método Bayesiano mais preciso para o uso na computação, como nos exemplos de ajuste de pesos em redes neurais pelos parâmetros da TRI (Benitez Rochel *et al.,* 2000), análise de dificuldade e classificação de parâmetros logísticos da TRI (Martínez-Plumed *et al.*, 2019), análise de dimensionalidades em D*eep Learning* (Urban; Bauer, 2021) e outras aplicações. Portanto, a TRI tem proporcionado melhorias significativas na otimização das avaliações de métodos computacionais, tendo em vista que os parâmetros da TRI são versáteis e oferecem informações importantes no momento de avaliação de metodologias que analisam conjuntos de dados, avaliando os variados métodos de rendimento de *performance* computacional.

Na computação são utilizados procedimentos na fase de modelagem cujo objetivo é reduzir problemas para a obtenção de modelos ótimos. Durante esta fase são realizados ajustes nos dados, balanceamento de classes, transformações de variáveis, redução de características e outros procedimentos. Estas rotinas são necessárias para a redução do custo computacional, do tempo de processamento e para evitar perda no desempenho dos modelos de IA obtidos. Isto torna os modelos obtidos sensíveis a novas alterações, exigindo novo processo de modelagem. Estas limitações quando não são resolvidas causam dificuldades no processo das classificações por algoritmos de *Machine Learning.*

2.5.1. Equação da Teoria da Resposta ao Item

Nesta seção será apresentada a equação da TRI para dados dicotômicos, informando como são calculados os parâmetros da Discriminação do item (a), Dificuldade do item (b), Probabilidade de acerto ao acaso (c), o valor de θ das respostas dos itens e as probabilidades condicionais pelo método *Bayesiano*. Assim como as representações gráficas com os comportamentos dos

parâmetros da TRI, os Modelos Logísticos e aplicações da TRI nas avaliações de diversas áreas e principalmente na computação (Baker, 2001).

A Teoria da Resposta ao Item é um método estatístico de teste baseada em modelos probabilísticos para determinar a relação entre os itens de teste e a capacidade dos indivíduos. A TRI define a probabilidade de resposta de um examinado a um item de teste como uma função da capacidade latente do examinado e das características do item. Para isto foi desenvolvida uma escala de pontuação que leva em conta as diferenças na dificuldade dos testes e calculada pela Equação 16, onde, seja U_{ij} uma resposta binária de um respondente *j* ao item *i*, em que $U_{ij} = 1$ para uma resposta correta e $U_{ij} = 0$ caso contrário, e θ_j a habilidade ou proficiência de *j* (Martínez-Plumed *et. al*., 2016).

$$P(U_i = 1|\theta_j) = c_{i+} \frac{1 - c_i}{1 + \exp(-a_i\,(\theta_j - b_j))} \quad (16)$$

$$i = 1, 2, 3, \ldots, n$$

onde:

i – Item classificado;

j – Classificador;

θ – Habilidade do classificador;

a_i – Discriminante do item;

b_i – Dificuldade do item;

c_j – Probabilidade de acerto ao acaso.

Os parâmetros da TRI estão presentes nos três Modelos Logísticos (ML). Estes modelos combinam os parâmetros da Discriminação (a), da Dificuldade (b) e da Probabilidade de acerto ao acaso (c).

2.5.2. Modelos de Parâmetros Logísticos da TRI

Existem três modelos logísticos na TRI para dados dicotômicos a serem observados na Tabela 2. Os Parâmetros Logísticos (PL) estão representados como podemos observar na Equação 16 para cada um dos Modelos de Parâmetros Logísticos.

Cada modelo tem um papel importante na avaliação do item em relação ao conjunto a ser classificado pelos algoritmos (Martínez-Plumed *et al*., 2019). Estes modelos seguem os três princípios da TRI, que são i) o foco na resposta, ii) o reconhecimento da natureza randômica das respostas e a iii) necessidade probabilística para explicar as distribuições e a presença de parâmetros separados para os efeitos das habilidades ou classificações dos algoritmos e as prioridades dos itens (Moraes *et al.*, 2022).

Tabela 2 – Modelo de Parâmetros Logísticos da TRI.

Modelo PL da TRI	***a*** **(Discriminação)**	***b*** **(Dificuldade)**	***c*** **(Probabilidade de acerto ao acaso)**
1 PL	Constante igual a 1	Valor calculado entre [-4,+4]	Constante igual a 0
2 PL	Valor calculado entre [-4,+4]	Valor calculado entre [-4,+4]	Constante igual a 0
3 PL	Valor calculado entre [-4,+4]	Valor calculado entre [-4,+4]	Valor calculado entre [0,1]

Fonte: Baker, Kim (2017).

2.5.3. Gráficos dos modelos dicotômicos da TRI

Uma forma objetiva de avaliar as respostas dos modelos da TRI é analisando as Curvas Características do Item (CCI), representadas na Figura 3. Para a variável Discriminação (a), representada na Figura 3(a), o grau de inclinação indica o quão discriminativo o item seja, o gráfico da esquerda da Figura 3(a) representa um item com baixa Discriminação, o gráfico da direita da Figura 3(a) representa um item com maior Discriminação. Para a variável Dificuldade (b), representada na Figura 3(b) o eixo *X* é a referência, o gráfico da esquerda da Figura 3(b) representa um item com menor Dificuldade, ou seja, o item é considerado fácil, o gráfico da direita da Figura 3(b) representa um item com maior Dificuldade, ou seja, quando a curva é deslocada para a direita o item é considerado difícil.

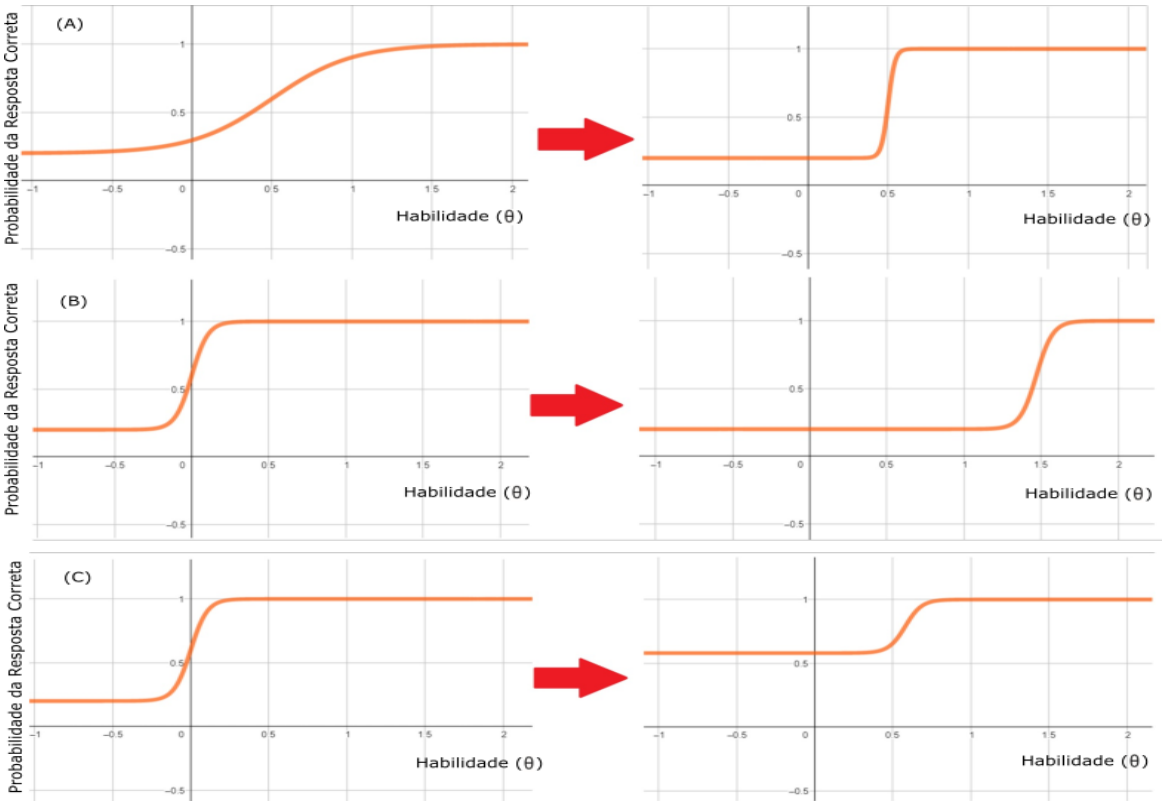

Figura 3 – Parâmetro de Discriminação (*a*), Parâmetro de Dificuldade (*b*) e Parâmetro de Probabilidade de acerto ao acaso (*c*) na Curva Característica na TRI. Fonte: Leite-Filho; Melo (2023).

Por outro lado, na Figura 3(c) a Probabilidade de acerto ao acaso *(c)*, o eixo Y é a referência para a Probabilidade de acerto ao acaso ou “chute”, o gráfico da esquerda da Figura 3(c) representa um item com menor Probabilidade de acerto ao acaso quando a curva é deslocada para baixo. O gráfico da direita da Figura 3(c) representa um item com maior Probabilidade de acerto ao acaso quando a curva é deslocada para cima. Cada um destes gráficos representam uma informação importante para a avaliação do item em relação ao conjunto de dados. Estes itens são considerados importantes na avaliação dos indivíduos (Setiawati; Izzaty; Hidayat, 2018).

A TRI também foi aplicada no rastreamento de conhecimento baseado em *Deep Learning* (Yeung, 2019)*;* assim também como na utilização da Curva Característica do Item (CCI) para ajustar as arquiteturas das redes neurais e obter respostas de como melhor avaliar os rendimentos de conhecimentos de estudantes (Benitez Rochel *et al.,* 2000) e em outro trabalho na utilização da TRI para avaliar os ajustes dos parâmetros e arquiteturas de *Autoencoders* em soluções de otimização de resultados (Chang; Vattikuti; Chow, 2019). Os

trabalhos realizados por Chen *et al.* (2019) e de Martínez-Plumed *et al.* (2019) serviram como referências do uso da TRI em computação. Com o objetivo e implicâncias do uso dos modelos estatísticos da TRI nos algoritmos de *Machine Learning,* o que proporcionou uma motivação mais acentuada a esta Tese.

2.5.4. Aplicações da TRI em avaliações

A TRI avalia nos estudantes as habilidades latentes e a proficiência dos mesmos nas avaliações, sendo cada vez mais empregada também em outros países. A TRI é um método psicométrico utilizado para avaliar as habilidades de resposta aos itens, diferentemente do método padrão de avaliações, Teoria Clássica dos Testes (TCT), a qual ainda é muito utilizada em provas e questionários, a TCT, como foi informado anteriormente, avalia apenas a pontuação obtida no número de acertos das questões propostas.

No trabalho de Setiawati; Izzaty; Hidayat (2018) foi realizada uma avaliação de valores dicotômicos com 1.046 indivíduos, sendo esta avaliação composta por 60 questões a serem respondidas em 30 minutos. As respostas ao serem analisadas utilizando o TRI, usou como referência os parâmetros do modelo de ajuste do item, dificuldade do item, discriminação do item, curvas de informação do item e função de informação do teste. Estas respostas foram avaliadas utilizando os programas *BILOG-MG* e *Matlab* para mensurar os resultados em uma análise de correspondência. Estes parâmetros serviram como base para avaliar os itens individualmente e aferir diferentes graus de correlacionar do item em si, ordenados em uma a matriz de resultados de todas as questões. Este tipo de ação possibilita associar os acertos das questões com a inteligência do respondente, diferenciando estudantes inteligentes pelo método da TRI, o que não seria possível utilizando o TCT.

A inteligência está associada a capacidade de resolver diferentes questões inéditas baseadas no conhecimento prévio do indivíduo. Esta capacidade chamada de inteligência, foi motivo de pesquisas no campo da Psicometria. Alfred Binet foi o pioneiro neste campo de pesquisa, e idealizou os primeiros testes de inteligência a fim de avaliar crianças e identificar indícios de traços latentes ou habilidades distintas, e esta pesquisa foi denominada de Teoria da Resposta ao Item (Gardner, 2006).

A Teoria da Resposta ao Item é um método de teste baseado em modelos probabilísticos para determinar a relação entre os itens de teste e a capacidade dos indivíduos, na qual define a probabilidade de resposta de um item de teste como uma função da capacidade latente do examinado, assim como das características do item. Para isto foi desenvolvida uma escala de pontuação que leva em conta as diferenças na dificuldade dos testes (Chang; Vattikuti; Chow, 2019), esta escala é baseada no item a ser respondido, considerando uma classificação do item no intervalo compreendido do mais fácil ao mais difícil.

2.5.5. Aplicações da TRI em Machine Learning

Os modelos computacionais usando a TRI proporcionam a avaliação da aprendizagem de algoritmos, diferentemente da forma utilizada no TCT. O uso da TRI na área de *Machine Learning* pode ser encontrada na literatura nos trabalhos de Chen; Ahn (2020); kandanaarachchi (2022a); Martínez-Plumed *et al.*(2016); Paganin *et al.* (2022).

Ao comparar estudantes e algoritmos, identifica-se que os métodos baseados nas métricas tradicionais assemelham-se ao TCT durante a modelagem em *Machine Learning*, os quais utilizam grupos os dados para o treinamento ou não de seus modelos, possibilita uma quantidade considerável de divergências nas avaliações realizadas pelas métricas ou pelo TCT. Pois há divergências na escolha da melhor métrica a ser aplicada, devido as diferentes formas de avaliar os algoritmos a serem considerados ideais classificadores, assim como a escolha de qual o objetivo final a ser considerado como padrão de qualidade da aplicação computacional (Baldwin, 2011).

Para utilizar a TRI na seleção dos modelos ideais, previamente deve ser realizada uma análise de quais parâmetros da TRI são considerados como referência. Esta análise deve constar como solução ideal para determinar quais os registros são classificados como difíceis ou não, possibilitando uma forma de avaliar a complexidades dos dados a serem triados no aprendizado dos algoritmos. Ou seja, avaliar as classificações de algoritmos não supervisionados pela TRI é uma oportunidade de entender como a complexidade dos dados impacta na seleção de modelos computacionais ideais, devido a característica ou habilidade deste algoritmo em realizar a classificação baseada

exclusivamente na natureza da distribuição dos dados (Chang; Vattikuti; Chow, 2019).

2.5.6. *Níveis de complexidade dos modelos da TRI e as métricas*

A determinação de qual métrica deve ser utilizada como método avaliativo dos resultados após as simulações computacionais é um fator a ser considerado, pois a métrica é utilizada como parâmetro que determinada evoluções qualitativa na seleção de modelos computacionais (Ferri; Hernández-Orallo; Modroiu, 2009). Tradicionalmente é utilizada uma métrica como método de seleção de qualidade, porém esta ação é um processo exaustivo na busca de soluções. Na heurística estas métricas também são utilizadas como função de *fitness*, proporcionando neste processo ajustes na seleção de modelos ideais para o objetivo que deve ser alcançado, possibilitando a maximização de uma solução subótima para resultados esperados.

Usar uma métrica como função de referência para seleção do modelo ideal associada a aplicação da TRI, necessitará a especificação prévia de qual métrica será utilizada, como no caso da Precisão (Uto; Ueno, 2018), da Sensibilidade ou da Especificidade (Zhu *et al.*, 2021). Estes trabalhos avaliaram as relações dos parâmetros da TRI com as métricas de formas independentes, sem realizarem uma comparação dos resultados obtidos de forma conjunta com todas as métricas citadas.

As métricas auxiliam nas decisões de seleção dos algoritmos, porém, a decisão baseada na complexidade dos dados e a definição de regras para a conduta na tomada de decisão, necessita que regras de decisões sejam estabelecidas para padronização das ações a serem executadas em conjunto com a TRI. Além da complexidade dos dados que proporcionam uma dificuldade para a formação dos agrupamentos, serão relacionados na seção seguinte os fatores que influenciam a decisão humana em situações complexas.

2.6. Avaliação de probabilidades complexas

Os dados compõem a principal parcela para a modelagem na computação, pois a partir dos dados de treinamento serão obtidos os modelos de *Machine Learning*. Estes modelos são avaliados por diferentes métricas para ajustes e

seleção dos melhores candidatos para a solução de problemas previamente definidos (Leite-Filho; Melo; Cassia-Moura, 2021b). Os dados são compostos por valores numéricos, textos, captados por sensores, imagens ou outros formatos, os quais são transformados posteriormente em dados numéricos para uso nos algoritmos de *Machine Learning* (Aggarwal, 2007; Ha *et al.*, 2011).

Diferentes procedimentos são utilizadas na literatura para avaliar os dados em suas distribuições, como por exemplo a variância e o desvio padrão que representam a dispersão dos dados em torno da média (Ehsaninia; Ghavi Hossein-Zadeh; Shadparvar, 2016), a entropia mede a incerteza ou irregularidade em conjuntos de dados (*i.e. clusters*) (Rendón *et al.*, 2011) e o coeficiente de variação que calcula uma medida de variação em torno da média. Estas métricas são bastante utilizadas na literatura para avaliar a variabilidade de populações e amostras (Aerts; Haesbroeck; Huwet, 2015), estas métricas são utilizados para avaliar a homogeneidade dos dados em um conjunto, onde uma baixa homogeneidade pode representar uma dificuldade na formação dos agrupamentos realizada pelos algoritmos de *clustering*.

Os dados representados matematicamente por valores numéricos estão contidos em diferentes conjuntos como: os naturais, inteiros, racionais, reais, imaginários, complexos e outros. O tipo do conjunto em que os dados estejam contidos e a forma como estão representados em seu nível de organização implicam no grau de homogeneidade dos dados a serem analisados (Ehsaninia; Ghavi Hossein-Zadeh; Shadparvar, 2016). A pertinência dos dados em determinados conjuntos e a organização dos mesmos proporcionam uma complexidade na determinação de soluções computacionais em relação ao tempo, pois tradicionalmente a obtenção de soluções computacionais está condicionada ao espaço de busca da solução e o tempo de processamento do algoritmo (Dumontier; Venugopal; Kumar, 2020).

Os modelos de *Machine Learning* têm como objetivo principal na área da computação a obtenção de uma solução baseada, em sua maioria, em um polinômio HA uma solução linearmente separável (*i.e.* na classificação e no *cluster*) (HA *et al.*, 2011). Este polinômio resulta em uma solução algébrica, assim como a obtenção de valores transcendentes (Minati, 2021), os quais são bastante complexos e difíceis de serem obtidos, mesmo utilizando ferramentas computacionais. Portanto, a partir do uso da computação para obter soluções

complexas, a busca de resultados numéricos está contida no conjunto de números computáveis por algoritmos finitos e com tempo de execução determinístico (Chaitin, 2011).

O conjunto de números computáveis está contido no conjunto de números não computáveis de forma a completar a abrangência das soluções. Esta abrangência no domínio dos conjuntos numéricos implica no crescimento do espaço de busca de soluções, referenciando o problema dos grandes números. Este conjunto mais abrangente determina uma dificuldade crescente da complexidade na obtenção de resultados na computação, sendo esta complexidade associada a Ordem de Grandeza (O-grande). Encontrar a melhor solução para problemas que tenham um gigantesco espaço de busca e uma diversidade de possíveis combinações de soluções, implica em uma ação que o tempo tende ao infinito (Hasanin *et al.*, 2019). Allan Turing demonstrou que existem formas de obter estes resultados numéricos, porém a computação convencional não encontrou uma solução viável para resolver os cálculos em tempo polinomial, mesmo utilizando a heurística como método de otimização do tempo de busca de resultados ótimos (Brener, 2010).

A atribuição da dificuldade para obter determinados resultados em soluções computacionais também pode ser observada nos problemas Polinomiais (P) e Não-Polinomial-Completo (NPC), onde a Ordem de Grandeza (O-grande) utilizada para calcular estas dificuldades é o tempo de busca da solução, estando o O-grande relacionado ao comportamento da solução algorítmica com o tempo tendendo ao infinito para problemas muito complexos (Mateo *et al.*, 2016).

2.6.1. Complexidade de tempo (O-grande)

O tempo calculado no O-grande é o fator predominante para selecionar algoritmos ótimos, onde a busca para a obtenção do melhor método de implementação algorítmica resulta comumente em soluções de busca exaustiva, devido ao vasto domínio de soluções. Os métodos Polinomiais e o O-grande são baseados na matemática implementada no algoritmo, assim como na capacidade do algoritmo em resolver problemas de questões complexas com severas limitações de tempo e na assertividade na resolução. Embora a

computação esteja evoluindo com a computação quântica, o problema ainda persiste e não há uma solução para todos os casos de tempos polinomiais. Para resolver esta classe de problemas de maior complexidade, foram definidos os problemas NP-difíceis, os quais são desafios a serem superados pela computação (Chatterjee; Bourreau; Rančić, 2023).

Os algoritmos de *clustering* avaliados na literatura apresentam distintas equações para o cálculo da complexidade de tempo. As equações baseadas no O-grande de cada algoritmo demonstram suas particularidades, ao estabelecer os agrupamentos. Podem ser citados como exemplo os algoritmos de *k-means* (O(nkd)), *PAM* (O(k$(n-k)^2$)), *CLARA* (O(k$(40+k)^2$ + k(n-k))), *SOM* (O(n^2m)) (Benabdellah; Benghabrit; Bouhaddou, 2019), nos quais é possível observar que cada algoritmo tem características diferenciadas na resolução das dificuldades para a formação dos agrupamentos.

A TRI proporciona dois fatores importantes na proposta de identificar os graus de dificuldade em problemas computáveis de classificação e outros objetivos inclusos na computação. O primeiro fator considera que a TRI utiliza o método Bayesiano para determinar as habilidades dos respondentes ou classificadores. O método Bayesiano é bastante útil quando estão envolvidas grandes quantidades de dados para que uma solução seja encontrada. O segundo fator a ser considerado é que a TRI está gradativamente melhorando suas técnicas de otimização. A TRI utilizou o método de Monte Carlo para a amostragem e cálculo de seus PLs, evoluindo com a implantação da cadeia de *Markov* e, posteriormente, a Estimativa de Máxima Verossimilhança (do inglês *Maximum-Likelihood Estimation*, MLE) (Kieftenbeld; Natesan, 2012). O método de *Gibbs* também foi implementado para refinar o método de busca de soluções ótimas em grandes volumes de dados (Lu; Zhang; Tao, 2018). A aplicação da Máxima Expectativa (do inglês, *Expectation Maximization*, EM) foi inserido como um processo iterativo para determinação da MLE de parâmetros de modelos de probabilidade, na presença de variáveis aleatórias latentes (Andrade; Valle, 2000), e outros métodos que contribuem para a eficiência dos resultados na amostragem dos dados e o cálculo de seus parâmetros (Chalmers, 2012).

A complexidade dos dados é um algo de grande relevância a ser considerado durante e posteriormente a formação dos grupos. Todavia, a

interpretação destes agrupamentos pelos especialistas humanos é um fator que também deve ser compreendido e discutido na próxima seção.

2.6.2. Interpretação humana na tomada de decisão crítica

Encontrar uma solução viável nem sempre corresponde a uma solução factível. A TRI é um método bastante utilizado nas análises psicométricas e seu propósito é avaliar o comportamento cognitivo dos indivíduos. Ao aplicar a TRI na avaliação dos objetivos dos algoritmos de *Machine Learning* são avaliadas as decisões dos algoritmos de forma análoga aos seres humanos. Porém a decisão ao responder ou classificar um item não corresponde a uma atitude simples aos seres humanos, ocasionando vieses nas seleções das respostas. Um exemplo desta ação é o Efeito Allais, que ocorre quando as preferências das pessoas por determinadas opções contradizem suas preferências em situações de tomadas de decisões críticas (Van de Kuilen; Wakker, 2006).

O Efeito Allais é comumente observado quando há um viés cognitivo em seres humanos, de modo que este viés ativa gatilhos mentais associados as habilidades dos especialistas humanos. Os gatilhos mentais são baseados em evidências e percepções naturais, e influenciam a tomada de decisões, podendo haver a formação de gatilho que pode ser interpretado como Viés de comportamento.

O Viés poderá ocorrer caso exista uma situação de decisão ao agrupar ou classificar um novo elemento. O especialista poderá decidir ignorar o novo elemento na classe para a qual foi agrupado, ou generalizar e incluir de forma errônea em outro grupo. Estas decisões são interpretadas no agrupamento formado para a tomada de decisões em momentos críticos. A Teoria da Utilidade Esperada geralmente ocorre em ambientes humanos de tomada de decisões (Newman, 1990), como no mercado de ações e negócios, descrevendo como as pessoas tomam decisões em situações de incerteza, ponderando as possíveis recompensas e riscos envolvidos de perdas ou lucros. Para avaliar estas decisões, ocorre a ponderação por condições distintas como a Teoria do Prospecto (Karmarkar, 1979).

Na Teoria do Prospecto há outro fator humano que também deve ser considerado, pois consiste no método baseado em como as pessoas tomam

decisões em situações de incerteza e riscos. Esta situação é bastante comum na área de saúde, onde a decisão de salvar vidas proporciona tendências na tomada de decisões, contrariando a probabilidade e a avaliação de todo o conjunto de situações em que o paciente e a equipe de saúde estejam inseridos (Treadwell; Lenert, 1999). Esta teoria consiste nos métodos do Efeito Certeza e Efeito Possibilidade.

O Efeito Certeza está relacionado com a ação humana baseada em comportamento psicológico diante de um evento muito provável, distorcendo desta forma as probabilidades a serem ponderadas na tomada de decisão (Newman, 1990). O Efeito Possibilidade está relacionado com um evento cognitivo humano referente à tendência de uma tomada de decisão baseada na maior probabilidade, onde o resultado probabilístico obtido consiste na realização de uma tarefa de acreditar que há alguma possibilidade de sucesso (Cerreia-vioglio; Dillenberger; Ortoleva, 2015; Karmarkar, 1979).

O Efeito Certeza e o Efeito Possibilidade são amplamente empregados por profissionais que tomam decisões em mercados de ações, onde são decididos quando será realizado determinado movimento baseado em respostas de modelos computacionais para tendências de mercado. O profissional decide se confia nos dados probabilísticos ou na escolha de correr menor risco e abortar um investimento baseado na decisão de perda ou ganho monetário (Van de Kuilen; Wakker, 2006).

Em seres humanos há fatores psicológicos que podem contribuir para a tomada de decisão, não baseada na lógica. Há evidências de que além dos fatores cognitivos, existem as interpretações associadas aos ajustes na tomada de decisão, e estas decisões podem ocorrer parcialmente na forma de ações em sistemas artificiais.

A tomada de decisão baseada em habilidades e contextos adversos a decisão humana são condições muitas vezes consideradas como antagônicas, devido aos critérios da complexidade *versus* tempo. Os algoritmos também estão inseridos nestas condições, pois a amostragem de novos dados a serem agrupados por algoritmos de *clustering* previamente utilizados são inseridos em situações nas quais novos dados são amostrados, e cabe ao algoritmo a tomada da decisão de atribuição correta do novo elemento no grupo específico. Tradicionalmente, os agrupamentos são obtidos de bases de dados

selecionadas em ambientes controlados, e cabe ao algoritmo de *clustering* atribuir uma base de confiança para a tomar as decisões dos profissionais e validação final, e muitas destas decisões dos algoritmos são consideradas difíceis. Para estas decisões, a TRI tem sido aplicada em diversos experimentos com algoritmos supervisionados e não supervisionados, com bons resultados (Chen *et al.*, 2019)(Chen; Ahn, 2020)(Kandanaarachchi, 2022a)(Li; Zheng, 2016)(Paganin *et al.*, 2022)(Silva *et al.*, 2021).

O método implícito na TRI avalia o desempenho de seres humanos ou de algoritmos em seus objetivos, com base em suas respectivas habilidades e tomadas de decisões ao responder questões em um exame ou na capacidade de resolver questões difíceis em ambientes de dados complexos. Esta escolha na TRI pode ser mensurada pela Probabilidade de acerto ao acaso, que é o Parâmetro (*c*) da TRI (Kandanaarachchi, 2022a)(Silva *et al.*, 2021). As questões das avaliações são respondidas com base na capacidade do aluno de resolver questões conhecidas ou desconhecidas, e às vezes sem conhecimento prévio do assunto. Analogamente, o algoritmo representado pelos agrupamentos, utiliza a dissimilaridade para a formação de grupos, agrupando novos elementos à medida que são inseridos. A abordagem apresentada nesta seção reflete a complexidade em ações particularmente consideradas simples, porém decisivamente condicionadas aos vieses cognitivos dos indivíduos em casos críticos.

2.6.3. Sistemas de apoio a decisões complexas na saúde

O agrupamento de dados ou clusters é um dos métodos mais utilizados por profissionais analistas que atuam na área de saúde, devido a aplicação em dados transversais entre as unidades de atendimento em saúde, avaliação objetiva dos vários dados decorrentes de cada atendimento de cada paciente, análise de múltiplas variáveis potencializando a interpretação da relação entre as mesmas para as decisões de gestão dos estabelecimentos de saúde e outras aplicações (Mello et al., 2021)(Tanaka et al., 2015).

Há diversos sistemas de apoio à decisão no Ministério da Saúde (MS) no Brasil, os quais proporcionam a coleta de dados relacionados à saúde e suporte a decisões dos gestores do MS. O MS do Brasil coordena ações de gestão,

regulação do funcionamento dos atendimentos e outras atividades associadas à saúde da população brasileira, com atribuições que estão contidas na Constituição da República Federativa do Brasil (1988), Lei 8.080/1990, Lei 8.142/1990, dentre outras legislações referentes à saúde humana. O MS utiliza distintos Sistemas Informatizados em Saúde (SIS) com o objetivo de exercer uma gestão mais efetiva. Estes sistemas são responsáveis pelo processamento dos dados coletados em serviços de saúde, proporcionando suporte à tomada de decisão no âmbito das políticas e do cuidado em saúde (Wartelle et al., 2021).

Os SIS compreendem sistemas distintos como o Sistema de Informação Ambulatorial (SIA), Sistema de Informações Hospitalares (SIH), Sistema de Informações sobre Nascidos Vivos (SINASC), Sistema de Informação de Agravos de Notificação (SINAN) e Cadastro Nacional de Estabelecimentos de Saúde (CNES), dentre outros. A partir dos dados obtidos destes sistemas são elaborados um conjunto de decisões e planejamentos das ações em saúde (Coelho Neto; Chioro, 2021).

As análises dos dados coletados pelos diversos SIS podem ser realizadas pela estatística descritiva, estatística inferencial ou sendo utilizado algoritmos de Inteligência Artificial (IA). O algoritmo de IA empregado para avaliar os dados coletados dependerá do objetivo especificado pelos analistas do MS, estes objetivos compreendem o agrupamento de dados para avaliar os sintomas de pacientes com câncer (Mello *et al*., 2021), previsões epidemiológicas a partir de regressão logística (Silva Abreu; Siqueira; Caiaffa, 2009), redes neurais no diagnóstico de COVID-19 (Fernandes Figueiredo *et al.*, 2021), classificação de estresse (Sobral; Santos; Sousa, 2019), dentre outras aplicações em saúde.

2.6.4. Avaliação psicométrica por analistas humanos em sistemas de saúde

A aprendizagem é o ato de conhecer a partir de experiências, e o processo de aquisição deste conhecimento é chamado de cognição (Piaget, 1952). No artigo de Turing, datado de 1950, há a proposta de um teste que avalia a capacidade da máquina de equiparar-se ao ser humano na precognição de respostas. O teste consistiu na comunicação entre um computador dotado de IA e um ser humano. Ambos estavam escondidos atrás de uma tela. O desafio foi

identificar se o humano conseguiria distinguir se estava se comunicando com outro ser humano ou com um computador, este teste persiste como referência clássica de avaliação para IA (Korukonda, 2003).

No Método Psicométrico, após a construção da base de conhecimento e da seleção do método avaliativo, constata-se que persiste a complexidade de avaliar os estudantes ou os algoritmos pelas suas distintas habilidades inerentes. Para avaliar estas habilidades foi proposto por Alfred Binet o método da Teoria da Resposta ao Item (TRI) que considera os fatores psicométricos e a medição da inteligência nas avaliações, utilizando respostas dicotômicas, baseado na escala *Likert*, proporcionando uma alternativa à TCT (Linden, 2016).

A TRI evoluiu com o desenvolvimento do método e foi ajustada por Luise Thurstone com novas características, como as curvas de proporção de resposta e a Função de Distribuição Acumulada (FDA), posteriormente agrupando métodos mais modernos com as aplicações de Frederic Lord, George Rasch e Samejima como a avaliação de dados contínuos, proposta de modelos com dois tipos de parâmetros como discriminação dos itens e habilidade dos indivíduos, modelo de Poisson para erros e gama para o tempo (Uto; Ueno, 2018; Zhou; Guo, 2020).

A TRI é um método estatístico que possibilita ordenar respostas em uma escala psicométrica. Esta escala foi aplicada em diversas áreas além da educacional, podendo ser aplicada por exemplo como uma escala utilizada na avaliação da ansiedade e depressão em pacientes hospitalizados, medindo os sintomas depressivos no instrumento, representando uma escala chamada "*Center for Epidemiological Studies Depression Scale*" (Giusti *et al.*, 2020). Nos trabalhos como em Perera *et al*. (2020), foram utilizadas outras escalas como a comparação da eficácia na marcha modificada (*mGES*). Esta escala é uma medida de autorrelato usada para examinar a confiança dos pacientes ao caminhar. No trabalho de Woudstra *et al*. (2019) foi testado um método para validar um instrumento baseado em computação e desempenho, avaliando as habilidades de alfabetização em saúde para a tomada de decisão no rastreamento do câncer colorretal. Assim como no trabalho de Paige *et al.*(2017) que comparou a TCT com a TRI em *escores* da *eHealth Literacy Scale* (eHEALS) de pacientes com doenças crônicas. A partir das informações apresentadas nesta seção, infere-se que mensurar em escalas as decisões é trivial na saúde.

2.6.5. Escalas de avaliação utilizadas em saúde

Para avaliar consistentemente o desempenho do aprendizado da IA, se faz necessário compreender suas capacidades de aprendizagem. A IA está em evolução e categorizada em diferentes níveis de cognição, como *Artificial Narrow Intelligence (ANI), Artificial General Intelligence (AGI)* e *Artificial Super Intelligence (ASI).* As diferenças entre cada nível estão na capacidade da IA tornar-se capaz de resolver problemas específicos (*ANI)*; evoluindo para a capacidade semelhante a humana (*AGI)* e por fim uma inteligência superior à humana (*ASI)* (Toxirjonovich; Fozilovich, 2022),

Para resolver questões identificadas como difíceis, é fundamental considerar a inteligência do respondente e como ser realizada a avaliação, podendo serem consideradas aplicações como *ACE Test for College Freshmen, Alexander Performance Scale, Arthur Point Scale,* dentre 44 escalas, que foram citadas por Cattell (1943) para medir a inteligência em seres humanos adultos, com ênfase nas avaliações verbal e não verbal, perceptiva, individual e em grupo. Nestas escalas, são consideradas a medição da inteligência, mas também predisposições fisiológicas do indivíduo. Nos serviços de saúde, rotineiramente são empregadas escalas específicas que avaliam fisiologicamente o ser humano em determinados ambientes, diante das situações de adaptação, destacando-se as seguintes escalas:

- Escala de *Glasgow* (*Glasgow Coma Scale,* GCS): avalia níveis de consciência de pacientes na Unidade de Terapia Intensiva (UTI), sendo considerada a abertura ocular (na escala, variando de 1 até 4), resposta verbal (na escala, variando de 1 até 5), resposta motora (na escala, variando de 1 até 6) (Muniz *et al.*, 1997).
- Escala de *Apgar*: avalia o recém-nascido no ambiente fora do útero, nos primeiros minutos após o nascimento. A escala avalia a frequência cardíaca, esforço respiratório, tônus muscular, coloração da pele e a irritabilidade reflexa. A partir da pontuação na escala, identifica-se anoxia grave (variando de 0 até 3), anoxia moderada (variando de 4 até 7) e sendo considerada adequada (variando de 8 até 10) (Schardosim; Rodrigues; Rattner, 2018).

- Escala de *RASS* (*Richmond Agitation Sedation Scale*): utilizada para medir a consciência dos pacientes no ambiente de UTI, em resposta aos níveis de sedação, cujas pontuações na escala são avaliadas pelos seguintes valores: -4 como estando combatível, -3 com muita agitação, -2 como agitado, -1 como inquieto, 0 como alerta e calmo, +1 como sonolento, +2 como sedação leve, +3 como sedação moderada, +4 como sedação profunda e +5 como inalterável (Sessler *et al.,* 2002).

Nestas escalas utilizadas em serviços de saúde, são avaliados categoricamente os distintos estados fisiológicos do indivíduo, os quais podem comprometer a cognição dos analistas humanos, permitindo identificar antecipadamente riscos neurológicos e psiquiátricos no ser humano, decorrentes de situações adversas que os pacientes estejam vivenciando.

Há doenças que podem acarretar a perda da consciência humana, levando ao comprometimento generalizado do indivíduo com déficits cognitivos. Este fato pode ser simulado computacionalmente quando o algoritmo em fase de treinamento, tem bom desempenho nas respostas, mas começa a ter perda de *performance* e a desaprender (*i.e.* perda da capacidade de cognição). A variação de *performance* está associada aos dados a serem agrupados por algoritmos de *clustering,* proporcionando um comparativo nas escalas cognitivas, no comportamento dos seres humanos, quando ocorrem variações de ganho ou perda de *performance*.

3.METODOLOGIA

Os algoritmos implementados nos experimentos realizados nesta Tese foram codificados na linguagem R versão 4.2.2. Os dados contínuos dos *datasets* foram transformados em dicotômicos utilizando a técnica *Simulate Response Patterns* do pacote "*mirt*" do R para as aplicações da TRI (Chalmers, 2012). Os experimentos foram realizados em um microcomputador i3-8100 com *HD SSD* e 8*GB* de *RAM* com o Sistema Operacional Windows 10 de 64 *bits*.

Para os experimentos foram consideradas restrições nos experimentos para avaliação das dificuldades calculadas pelos Parâmetros de Dificuldade b dos três Modelo PL da TRI, nas atribuições dos elementos nos respectivos agrupamentos realizados pelos algoritmos de *clustering*. Foram realizadas as seguintes restrições:

- Todos os algoritmos testados são de *clustering*, correspondendo aos seres humanos respondentes.
- Os *datasets* avaliados são binários, determinando que o número de *clusters* (*k*) é igual a dois. Os datasets correspondem aos itens e as classes originais correspondem às respostas.
- Os agrupamentos obtidos para cada algoritmo de *cluster* foi a resposta. O agrupamento positivo foi igual a classe zero e o agrupamentos negativo equivale a classe um. Esta atribuição de classificação simula uma validação do agrupamento realizado por um especialista humano. Contudo, esta validação pode ser realizada de forma automatizada pelo método de votação das atribuições dos elementos, nos respectivos agrupamentos ou classes, citada na seção 2.4.4 desta Tese, porém não utilizada devido a veracidade de pertinência dos elementos nos grupos avaliados nos experimentos.
- Foram realizadas as medições dos Parâmetros de Dificuldade b dos três Modelo PL da TRI, em cada uma das situações adversas, para as corretas atribuições dos elementos nos agrupamentos correspondentes às classes previamente conhecidas.

- O algoritmo foi considerado ótimo ao falhar menos nas atribuições dos itens nos agrupamentos, em relação às classes originais de cada conjunto de dados.
- Existiu uma correspondência de cada classe identificada corretamente com o valor determinado pela dificuldade da TRI. O valor de cada registro no conjunto de dados foi determinado no intervalo graduado de -4 como o mais fácil, a +4 como o mais difícil, como podemos observar na Tabela 2.

3.1. Algoritmos de *clustering*

Os 34 algoritmos utilizados nos experimentos desta Tese estão descritos na Tabela 3, divididos em 12 grupos, e diferenciados em cada grupo pelas características de ajustes do centróide no agrupamento e pelas dissimilaridades.

Tabela 3 – Algoritmos não supervisionados utilizados nos experimentos.

N	Algoritmo	Método/ dissimilaridade	N	Algoritmo	Método/ dissimilaridade
1	*Fanny*	*Euclidean*	21	*Fuzzy C-means*	*Euclidean*
2		*Manhattan*	22		*Manhattan*
3		*SqEuclidean*	23	*Cluster Medoids*	*Euclidean*
4	*PAM*	*Euclidean*	24		*Manhattan*
5		*Manhattan*	25		*Chebyshev*
6	*Clara*	*Euclidean*	26		*Canberra*
7		*Manhattan*	27		*Braycurtis*
8		*Jaccard*	28		*Pearson correlation*
9	*Sota*	*Euclidean*	29		*Simple matching coefficient*
10	*Kmeans*	*Hartigan-Wong*	30		*Minkowski*
11		*Lloyd*	31		*Hamming*
12		*Forgy*	32		*Jaccard coefficient*
13		*MacQueen*	33		*Rao coefficient*
14	*Diana*	*Euclidean*	34		*Mahalanobis*
15	*Kmeans*	*arma/Euclidean*	35		*Cosine*
16		*rcpp/Euclidean*	36	*Gustafson-Kessel*	*Mahalanobis*
17	*Mini Batch Kmeans*	*kmeans++/Euclidean.*	37	*MOCCA*	*Kmeans*
18		*random/Euclidean.*	38		*Neuralgas*
19		*optimal init/Euclidean*			
20		*quantile init/Euclidean*			

Fonte: Autor.

O grupo 1 foi composto pelos algoritmos de 1 a 3, representados pelo algoritmo *Fuzzy Analysis Clustering (fanny).* O grupo 2 foi composto pelos

algoritmos de 4 a 5, representados pelo algoritmo *Partitioning Around Medoids (pam). O g*rupo 3 foi composto pelos algoritmos de 6 a 8, representados pelo algoritmo *Clustering Large Application (clara*). O grupo 4 foi composto pelo algoritmo 9, representado pelo algoritmo *Self-Organization Tree Algorithm (sota).* O grupo 5 foi composto pelos algoritmos de 10 a 13, representado pelo algoritmo *K-means. O* grupo 6 foi composto pelo algoritmo 14, representado pelo algoritmo *Divisive Analysis* (Diana). O grupo 7 foi composto pelos algoritmos de 15 a 16, representados pelos algoritmos KMeans_arma e KMeans_rcpp. O grupo 8 foi composto pelos algoritmos de 17 a 20, representado pelo algoritmo Mini Batch Kmeans. O grupo 9 foi composto pelos algoritmos 21 e 22 e representados pelo Fuzzy C-means. O grupo 10 foi composto pelos algoritmos de 23 a 35, representado pelo algoritmo Cluster Medoids. O grupo 11 foi comporto pelo algoritmo Gustafson-Kessel. O grupo 12 foi composto pelo algoritmo *Multi-objective Optimization for Collecting Cluster Alternatives* (MOCCA).

As dissimilaridades foram as características utilizadas em cada grupo para diferenciar os métodos de determinação dos distanciamentos dos elementos nos agrupamentos, proporcionando uma visão mais ampla da formação dos agrupamentos formados por cada algoritmo específico nos 12 grupos analisados. Para os algoritmos *fanny, pam* e *clara,* dos grupos 1 a 4, e diana do grupo 6, foram aplicadas as medidas de dissimilaridade *Euclidean, Manhattan, Square Euclidean e Jaccard*.

Para o algoritmo *K-means* do grupo 5, foram utilizadas atribuições de posicionamento de centróides, pelos métodos de *HartiganWong, Lloyd, Forgy e MacQueen*. Estes algoritmos foram testados no trabalho de Anand; Kumar (2023). Para os *K-means* do grupo 7, foram também utilizadas atribuições de posicionamento de centróides, porém os algoritmos do grupo 7 diferem dos algoritmos do grupo 5 pelo objetivo da otimização de ajustes dos centróides a partir de bibliotecas específicas (*i.e.* linguagem C). Para os algoritmos de *Mini Batch Kmeans* do grupo 8, foi utilizado como característica destes algoritmos o posicionamento inicializado do centróide como objetivo, utilizando técnicas de inicializações dos centróides representadas por *Kmeans* ++, *Randômico*, Inicialização otimizada e Inicialização qualitativa (Feizollah *et al.*, 2015).

Para os algoritmos de Fuzzy C-means do grupo 9, foram avaliados elementos contidos em múltiplos agrupamentos, como no trabalho de *Alomari et*

al. (2016). Para os algoritmos de *cluster* medóides do grupo 10, o objetivo foi avaliar os elementos definidos como centroides ou medóides nos agrupamentos, proporcionando um referencial nas distâncias calculadas pelas dissimilaridades dos elementos dos grupos obtidos. Este método foi utilizado no trabalho de Sisodia; Verma; Vyas (2017) para avaliar a formação de agrupamentos de dados obtidos da *internet*. Os algoritmos utilizados nos grupos 10 e 11 são mais atuais e objetivaram uma comparação na evolução com os algoritmos dos demais grupos. Os algoritmos 9 e 38 destacaram-se na formação dos agrupamentos o método de Mapa de Auto-Organização (*Self Organizing Map* – SOM). Neste método há a formação de grupos mapeados para a identificação de padrões sutis, para uma análise mais criteriosa.

Os algoritmos utilizados nos experimentos desta Tese estão contidos nos pacotes *clValid, ClusterR, clusterSim e cluster* do R *Statistics* e não foram otimizados para obtenção dos melhores agrupamentos. A seleção destes algoritmos para os experimentos desta Tese foi baseada na aplicação da formação dos agrupamentos por cada um dos algoritmos de *clustering*, considerando as diferentes técnicas utilizadas nos 12 grupos. E a partir destas diferentes técnicas de formação de agrupamento, foram avaliadas as dificuldades na determinação dos elementos pertinentes aos *clusters* formados por cada algoritmo. As dificuldades na formação dos agrupamentos consistiram nos ajustes dos elementos nos determinados grupos pelos métodos de dissimilaridade, considerando a forma de cálculo das distâncias entre a centróide e os elementos constituintes do agrupamento. Estes algoritmos foram selecionados pelos objetivos semelhantes a proposta desta Tese, em avaliar as dificuldades nas amostragens de dados, dimensionalidade e outros fatores que impactam na formação dos agrupamentos pelos algoritmos de *clustering*. Estes algoritmos foram também avaliados em trabalhos na literatura envolvendo formação de agrupamentos de dados complexos em saúde (k-means) como no trabalho de Hassan; Mollick; Yasmin (2022) e de Tanaka *et al*. (2015); redução de dimensionalidade *(i.e*. Fuzzy C-means, Fanny e K-means), como no trabalho de Maltauro *et al*. (2023); representação dos dados nos grupos (*i.e.* K-means, clara, pam, diana, Sota e Fanny), como no trabalho de Anand; Kumar (2023); pertinência de elementos nos agrupamentos formado (*i.e.* Cluster Medoids), como no trabalho de Sisodia; Verma; Vyas (2017). Assim como os experimentos

com algoritmos mais modernos, utilizados em trabalhos a combinação do algoritmo de agrupamento recursivo de Gustafson-Kessel e o método de mínimos quadrados recursivos Fuzzy (Dovžan; Škrjanc, 2011). O algoritmo *Multi-objective Optimization for Collecting Cluster Alternatives* (MOCCA) teve como foco a avaliação multi-objetivo dos índices de medidas de validação de *cluster* pelo método de Pareto, todavia o MOCCA é um forte candidato a ser avaliado em outros métodos de medidas de validação de *cluster* como descrito no trabalho de Kraus *et al*. (2011).

3.2. *Datasets* avaliados

Para avaliar os algoritmos da Tabela 3 foram selecionados os *datasets* citados na Tabela 4, onde estão representadas as características e instâncias contidas em cada *dataset,* a especificação de classe binária, correspondendo às classes originais de cada *dataset*, as proporções percentuais das classes desbalanceadas e o percentual após o balanceamento por *SMOTE*. Na Tabela 4 estão apresentadas as instâncias de cada *dataset*, as quais representam os elementos de interesse dos experimentos desta Tese. Cada instância equivale a um elemento a ser organizado em um determinado agrupamento, e estes agrupamentos representam as classes contidas nos *datasets* apresentados na Tabela 4. Esta correspondência entre os agrupamentos e os grupos dos *datasets* ocorreu pela validação semi-supervisionada para as classificações indiretas dos *clusters* (Diogo; Francisco de, 2023). As atribuições de cada elemento para cada partição dos agrupamentos gerados, foram realizadas pelo algoritmo de *clustering* e comparadas de forma semi-supervisionada com as classes dos dados em cada *dataset*.

Os *datasets* tratados de D1 até D4 foram obtidos do trabalho de Martínez-Plumed *et al.* (2019) no *link* (https://github.com/nandomp/IRT4ML); os *datasets* correspondentes de D5 até D9 foram obtidos de um segundo *link* e acessado em (https://archive.ics.uci.edu/); e os *datasets* D10 e D11 foram obtidos de um terceiro *link* (https://www.openml.org/). Em todos os *datasets* não estão contidos dados faltantes e o número de *clusters* corresponde a quantidade de classes determinadas nos *datasets* originais. Ou seja, classes binárias em cada um dos *datasets*.

Tabela 4 – Representação dos *datasets* utilizados nos experimentos, com distintos níveis de balanceamento.

ID	Dataset	Característica	Instâncias	Classes	Desbalanceada		Balanceada	
					Classe 0	Classe 1	Classe 0	Classe 1
D1	*Echocardiogram*	10	111	2	69(67,20%)	42(32,80%)	172 (49,57%)	175 (50,43%)
D2	*Heart-stat log*	13	230	2	102(44,40%)	128(55,60%)	260 (56,52%)	200 (43,48%)
D3	*Ionosphere*	33	321	2	206(64,10%)	115(35,90%)	444 (51,51%)	418 (48,49%)
D4	*Parkison*	22	185	2	140(75,40%)	45(24,60%)	273 (48,49%)	290 (51,51%)
D5	*Blood Transfusion*	5	748	2	570(76,20%)	178(23,80%)	1140 (51,63%)	1068 (48,37%)
D6	*Diabetes*	8	768	2	268(34,90%)	500(65,10%)	1072 (51,74%)	1000 (48,26%)
D7	*Madelon*	500	2600	2	1300(50,00%)	1300(50,00%)	1300 (50,00%)	1300 (50,00%)
D8	*Vinnie*	2	380	2	185(48,68%)	195(51,32%)	390 (51,32%)	370 (48,68%)
D9	*Glass*	10	214	2	138(64,49%)	76(35,51%)	276 (47,59%)	304 (52,41%)
D10	*Banana(18,87% do original)*	2	2000	2	1000(50,00%)	1000(50,00%)	1000(50,00%)	1000(50,00%)
D11	*Breast câncer*	10	683	2	317(100,00%)	366(100,00%)	317(34,99%)	366(65,01%)

Fonte: Autor.

Os *datasets* foram normalizados com média igual a zero e variância igual a 1, os *datasets* utilizados nos experimentos estão representados na Tabela 4. Estes *datasets* foram selecionados para os experimentos nesta Tese devido as seguintes características: para os *datasets* D1, D2, D4, D5, D6 e D11 há a representação de dados de saúde.

O *dataset* D3 é composto apenas por dados contínuos, ou seja, não há uma variabilidade dos tipos dos dados nos atributos do *dataset*, os quais precisem obrigatoriamente de transformações ou ajustes nos tipos de dados para a redução de variabilidade de dados (*i.e* normalização de dados), favorecendo uma homogeneidade para o agrupamento.

Os *datasets* D7 e D10 representam conjuntos de dados com grande quantidade de instâncias, ou seja, grande volume de elementos a serem agrupados, onde o D7 contém uma elevada quantidade de dimensionalidade, proporcionando uma grande variabilidade de características para os algoritmos de *clustering* selecionar os agrupamentos, podendo ocasionar formações de maior quantidade de agrupamentos, diferindo do que foram determinados nos *datasets* informados nos repositórios referenciados anteriormente nesta seção.

O dataset D8 possui baixa quantidade de características e um considerado número de instâncias, proporcionando um *dataset* de interesse para os experimentos, pois a quantidade de características é baixa, proporcionando um fator de possível dificuldade para a formação dos agrupamentos, considerando a quantidade de instancias a serem agrupadas. O dataset D9 contém numericamente as quantidades de atributos e característica semelhantes a D1 e D2, proporcionando uma base de comparação do método proposto nesta Tese com aplicações relacionadas com outras áreas que exijam agrupamento de dados, além dos propostos para a saúde em D1, D2, D4, D5, D6 e D11. Em todos os datasets foram considerados os diferentes tipos de características dos dados (*i.e.* multivariados) e normalizados.

Os *datasets* foram balanceados pelo método de *SMOTE*, conforme pode ser verificado na Tabela 3. Após o balanceamento e transformação dos dados foi possível realizar a extração de amostras estratificadas para análise. Nestes experimentos foi considerado os três níveis dos Parâmetro de Dificuldade b dos três Modelos PL, as respectivas atribuições das pertinências dos elementos aos respectivos agrupamentos por cada algoritmo de *clustering*, comparando

posteriormente com as classificações já atribuídas a cada *dataset* nos repositórios citados. Desta forma, foi proporcionada uma atribuição de classificação como se fosse uma validação de um especialista humano. O fluxo utilizado nos experimentos está representado na Figura 4, na qual estão descritos os experimentos realizados nesta Tese.

3.3. Algoritmos e fluxo da proposta da Tese

Inicialmente a população foi estratificada de maneira proporcional a 172 registros, equivalente à menor classe minoritária dos *datasets* (*i.e. Echocardiogram)*, constituindo o tamanho padrão para análises e comportamentos na determinação dos dados nos respectivos agrupamento pelos algoritmos de *clustering* nesta Tese.

No fluxo apresentado na Figura 4 é possível observar que, após as formações dos agrupamentos para identificar a correspondência com as classificações da amostra, estas classificações estão contidas nos *datasets* originais, que foram identificados neste capítulo, e posteriormente calculados os respectivos valores do Parâmetro de Dificuldade b por registro. Foram calculadas as métricas pelas comparações dos vetores que correspondem às possíveis classificações obtidas e as classificações-alvo de cada *dataset*.

Para cada um dos *datasets* (D_n) foram calculados os três Modelos Logísticos (ML) correspondendo a ML_1, ML_2 e ML_3 da TRI. E destes modelos foram obtidos os respectivos valores do Parâmetro de Dificuldade b de cada *dataset*. Foi calculado um vetor VLM_x, onde x corresponde a cada um dos três parâmetros do ML. Assim como também foram representados os valores da classe original do *dataset* por (OD) e um vetor (VCo) com as respectivas classificações de cada algoritmo no *dataset*. Em seguida, foram obtidos os vetores (VD) representados por $VLM_x(ID_x, VLM_x, VCo)$, que foram inseridos em matrizes (UD_x) correspondentes a cada *dataset* e seu respectivo ML.

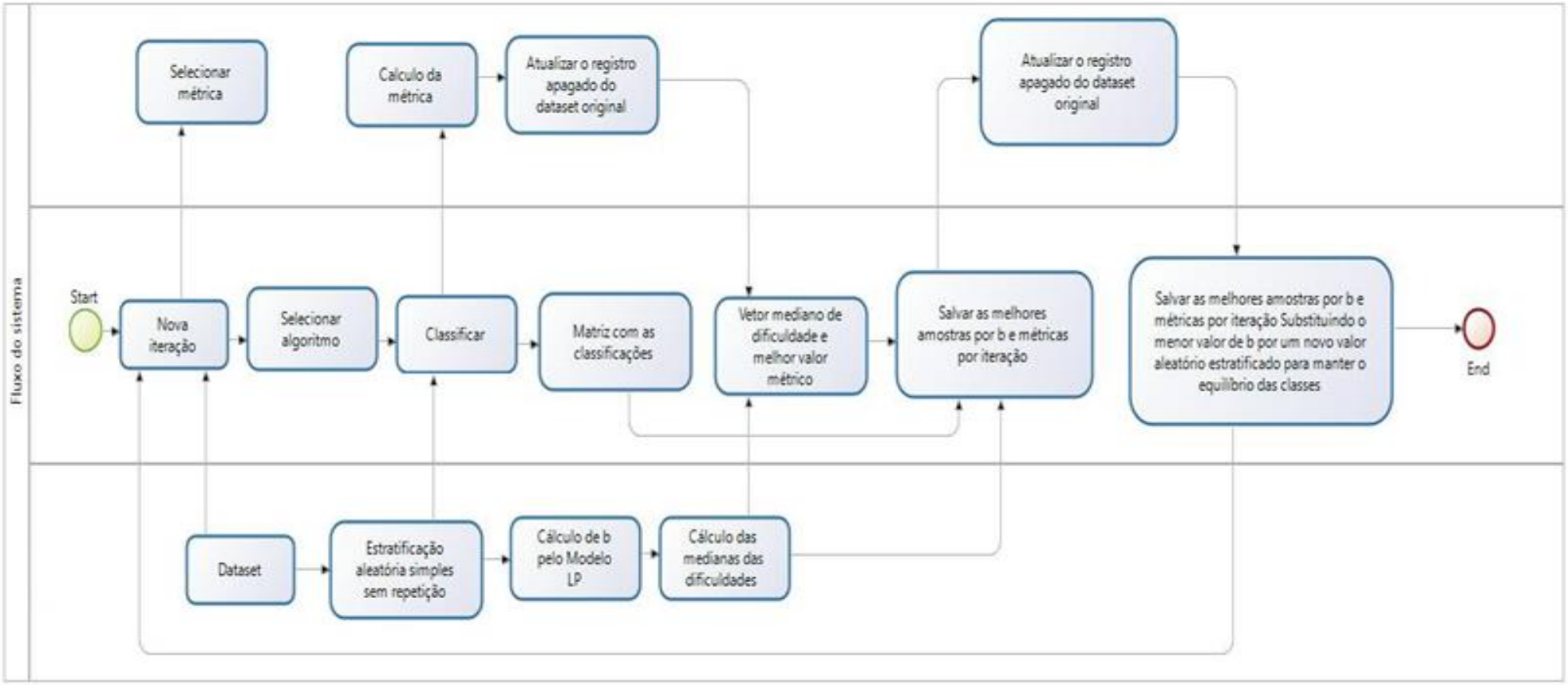

Figura 4 - Fluxograma do método proposto. Fonte**:** Adaptado de Leite-Filho; Melo (2023).

As variáveis estão apresentadas nos pseudocódigos *OrderResults* e *Notification of Easier - exits and stratified - enters* (Not2ESE), onde estão representadas as variações dos ML no pseudocódigo dependendo de cada ML representadas pelos PLs, como: *Not2ESE-1PL, Not2ESE-2PL e Not2ESE-3PL.*

Algoritmo *OrderResults* Identificando *b* em cada item

```
1: Função Set_b (dataset, algoritmo, PL)
2: U <- 0 # Matriz do Acumulador.
3: D <- dataset # Cada dataset.
4: I <- 0 # Cada item por dataset.
5: LP <- 0 # Cada b em (n) do PL [item] (Equação da TRI).
6: A<- Algoritmo #Os agrupamentos dos itens pelos Algoritmos.
7: C <- Situação # Condição do agrupamento pelo algoritmo.
8:
9: Faça D[i] até
9:        Faça LP até
10:       I <- dataset [item]
11:       nLP[i]<- b [item] #Calculo de b para cada LP
12:
13:       Se A[i] == Classificação[i] faça #Associação do
          agrupamento com a classificação existente.
13:               C[i]<- Verdadeira
14:       Se então
15:               C[I] <- Falsa
16:       Fim se
17:       U <- (I, C[i],1LP[i],2LP[i],3LP[i])
18:       Fim faça
19: Fim faça
20: Fim da função
```

Algoritmo Not2ESE

```
# selecionando amostras mais complexas de agrupar/classificar
1: Função Not2ESE
2: Faça Selecione a métrica (MKx) até
3: D <- dataset # Cada dataset.
4: I <- 0 # Cada item por dataset.
5: nLP<- 0 # Cada b em (n) do PL [item] (Equação da TRI)
6: Vetor de medianas de b<-0
7: A <- Algoritmo # Os agrupamentos dos Algoritmos por item.
8: Matriz agrupamento/classificações por b<-0.
9: selecionar o dataset e extrair a amostra estratificada
10: Selecionar o algoritmo
11: Faça OrderResults até # Executar as 500 iterações
12: Comparação/atualização da métrica (MKx) selecionada
13:       Cálculo do Vetor de medianas de b <- atualizado
14:       Matriz de b agrupamento/classificações atualizadas.
15:       Salvar a amostra com os itens.
16:       Seleção do valor de b mais fácil.
17:       Seleção de novo I em D de forma estratificada
18: Fim faça
19: Salvar U e a amostra final de I em cada D
20: Selecione nova métrica
21: Fim da Função
```

As métricas são representadas por MK_y, onde y={1,2,3,4}, correspondendo a cada uma das quatro métricas avaliadas neste estudo, e representadas pelas Equações de 1 a 3. As métricas correspondentes de 8 a 14 foram calculadas em paralelo, em cada uma das métricas de 1 a 3, determinando desta forma os valores correspondentes aos índices de validação dos agrupamentos formados. Cada métrica tem seu objetivo, e a determinação da métrica impacta na decisão final da escolha do melhor algoritmo, na aplicação pelos especialista humanos (Liu *et al.*, 2014). Para uma avaliação mais aprofundada, foram comparadas as métricas de 1 a 3 e suas relações das correspondências dos elementos agrupados com as classificações, e os valores do Parâmetro de Dificuldade b nos três modelos PL.

Os valores obtidos foram armazenados em vetores e comparados a cada iteração, onde os melhores resultados foram atualizados até obter o ponto de parada, ou seja, 500 iterações. Foram obtidas as amostras mais difíceis pelo cálculo do Parâmetro de Dificuldade b da TRI e com elementos distintos. Assim como também os algoritmos com os melhores resultados nas respectivas métricas, e posteriormente salvos em arquivos com as atualizações dos valores calculados. Os resultados foram comparados estatisticamente pelo método de Kruskal Wallis e ajustados pelo método de Bonferroni, para identificar se as

distribuições das classificações e vetor de cada Parâmetro de Dificuldade b seguiram a mesma distribuição ou relação.

3.4. Escala psicométrica baseada na TRI

Foram categorizadas seis faixas para a referência dos Níveis de Dificuldades (ND) dos valores do Parâmetro de Dificuldade (b) em níveis de complexidade baseadas na Tabela 2 da TRI, descritas abaixo:

- de [- infinito a -4] é extremamente fácil (XE);
- de [-4,-2] é muito fácil (VE);
- de [-2,0] é fácil (E);
- de [0,+2] é difícil (H);
- de [+2,+4] é muito difícil (VH); e
- de [+4,+ infinito] é extremamente difícil (XH).

Estas faixas foram adaptadas do trabalho de Andrade; Valle (2000), onde os níveis do Parâmetro de Dificuldade b foram divididos da seguinte maneira: MFC (Muito fácil e classificado corretamente) [-4,-2], FC (Fácil e classificado corretamente) [-2,0], MDC (Muito difícil e classificado corretamente) [0,+2], DC (Difícil e classificado corretamente) [+2,+4], MFE (Muito fácil e classificado erroneamente) [-4,-2], FE (Fácil e classificado erroneamente) [-2,0], MDE (Muito Difícil e classificado erroneamente) [0,+2], DE (Difícil e classificado erroneamente) [+2,+4]. As faixas menores que -4 e maiores que +4 foram ajustadas para adequar valores que extrapolem os possíveis resultados registrados.

Estas faixas foram utilizadas para avaliar os resultados das correspondências dos elementos agrupados pelos algoritmos de *clustering* e as classificações, determinando desta forma os acertos e erros dos algoritmos, possibilitando a aplicação da TRI nesta Tese. Este método foi também utilizado nos trabalhos de Leite-Filho; Melo; Cassia-Moura (2021a) e Leite-Filho; Melo; Cassia-Moura (2021b), adaptado do trabalho com dados supervisionados de Martínez-Plumed *et al*. (2019) para dados não supervisionados, utilizando a Equação 15.

4. RESULTADOS E DISCUSSÃO

Os resultados estão divididos em duas etapas de experimentos conforme informado no método, na primeira etapa as classes não foram balanceadas e as amostras correspondem a população de cada *dataset*. Na segunda etapa houve o balanceamento das classes, a criação de dados artificiais e extração de amostras, para que as métricas da Precisão, Sensibilidade e Especificidades fossem avaliadas em seus comportamentos em relação aos valores do Parâmetro de Dificuldade b da TRI. Nas duas fases foram avaliados os comportamentos do Parâmetro de Dificuldade b nas condições que influenciem as classificações dos algoritmos não supervisionados descritas nesta Tese.

4.1. Comportamento dos algoritmos *versus* dados

Os algoritmos de *clustering* da Tabela 3 foram selecionados pela principal característica de trabalharem os dados dos *datasets* da Tabela 4 para: determinação da formação dos *clusters*, distância *inter-cluster, intre-cluster* e pertinência dos elementos (*borderlines);* ou seja, sem a necessidade de criar elementos artificiais para referências de centróides. Estas características possibilitaram aos algoritmos de *clustering* um aprofundamento para avaliação posterior da aplicando da TRI. Este aprofundamento consiste na avaliação das dificuldades adversas na obtenção do objetivo determinado, a formação dos agrupamentos.

Para a aplicação da TRI, ao compararmos os algoritmos de *clustering* dos experimentos desta Tese com estudantes, foi avaliada a capacidade de responder questões objetivas, e sem necessidade de extrapolar os itens ou novas questões ali contidas. Ou seja, sem questões adicionais criadas pela combinação de outras questões (*i.e.* dados). Os algoritmos de *clustering* dependem exclusivamente de suas características naturais para resolverem os agrupamentos dos itens, sem qualquer tratamento significativo dos dados ou otimização dos algoritmos. Os experimentos foram realizados para que tornem independentes cada um dos tópicos apresentados, e em cada resultado foram realizadas análises da Dificuldade do Parâmetro b da TRI dos algoritmos de *clustering* nos *datasets* selecionados. Nas subseções a seguir foram avaliadas as situações adversas.

4.1.1. Pertinência dos elementos nas áreas de conflitos nos clusters

Conforme abordado na seção da revisão da literatura sobre dimensionalidade, a decomposição das dimensões e suas representações gráficas, podem induzir erros de interpretação, assim como problemas de pertinência dos elementos no subespaço onde os *clusters* são formados. O objetivo de visualizar os *clusters* formados pela separação de elementos com características semelhantes, proporciona um dos fatores associados a dimensionalidade dos dados e a decomposição da dimensionalidade em gráficos bidimensionais. Este problema em particular exprime a dificuldade de identificar as pertinências dos elementos em áreas de fronteira. Ou seja, as áreas onde os limites entre os grupos estão sobrepostos, muito próximos ou muito distantes do centro do *cluster*. Destacando que os *datasets* desbalanceados proporcionam uma maior dificuldade devido as classes majoritárias, causando problemas nas classificações de *clusters* (Liu, 2021). Nos experimentos iniciais foram gerados 418 gráficos de dispersão de dados proporcionais aos 38 algoritmos nos 11 *datasets* desbalanceados, conforme Figura 5 para exemplificar os problemas de pertinência dos elementos nos respectivos agrupamentos.

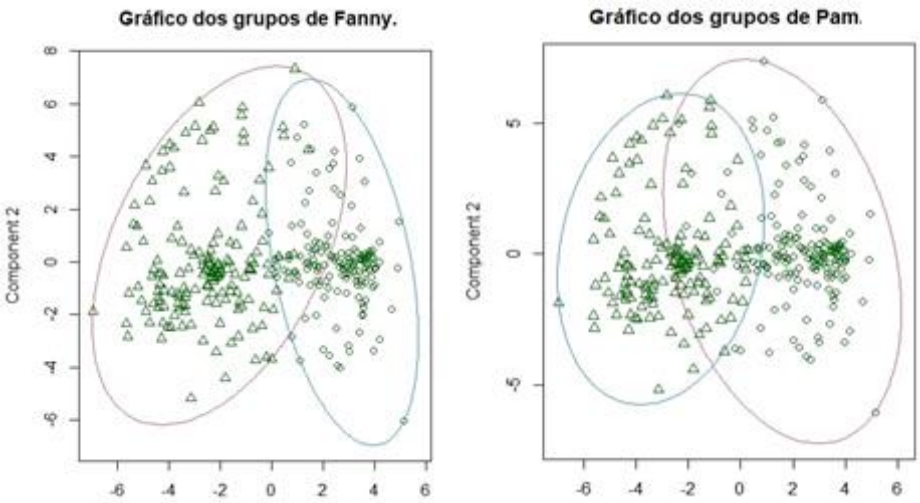

Figura 5 – Algoritmos fanny(SqEuclidean) (E) e pam(manhattan) (D) no d*ataset* 7, estando representados os distintos grupos formados. Fonte: Autor.

Na Figura 5 estão representados dois gráficos correspondente aos algoritmos fanny(SqEuclidean) e pam(manhattan) no d*ataset* 7, onde é possível observar que cada algoritmo criou suas respectivas partições utilizando técnicas diferentes utilizadas em cada algoritmo de *clustering*, assim como pelo uso de

diferentes dissimilaridades. Este problema é característico da aplicação de cada algoritmo e método de cálculo de distâncias, além do problema do desbalanceamento. Estes fatores podem ser suavizados ao resolver questões como o balanceamento de classes. Estes fatores podem ocasionar problemas de decisão como relacionado no trabalho de Liu (2021), onde o autor avaliou os ruídos ocasionados pelo desbalanceamento, o impacto causado pelas classes majoritárias nas classificações e um fator que tornam complexas as decisões de classificação pelos algoritmos. Visualmente é perceptível a separação de elementos em grupos nos gráficos da Figura 5. Da mesma forma ocorrem as dificuldades na determinação das pertinências destes elementos nos respectivos grupos de *cluster*. Esta contextualização é importante para a compreensão de que as dissimilaridades não determinam um método matemático preciso para a identificação dos elementos nos *clusters*, proporcionando uma particularidade a ser tratada posteriormente à formação dos grupos pelos algoritmos de *clusters*.

Na Figura 6 é possível observar o conflito de identificar corretamente os elementos nos respectivos agrupamentos, considerando a área de fronteira dos elementos corretamente identificados nos respectivos grupos e os identificados de forma incorreta (*i.e.* os ruídos).

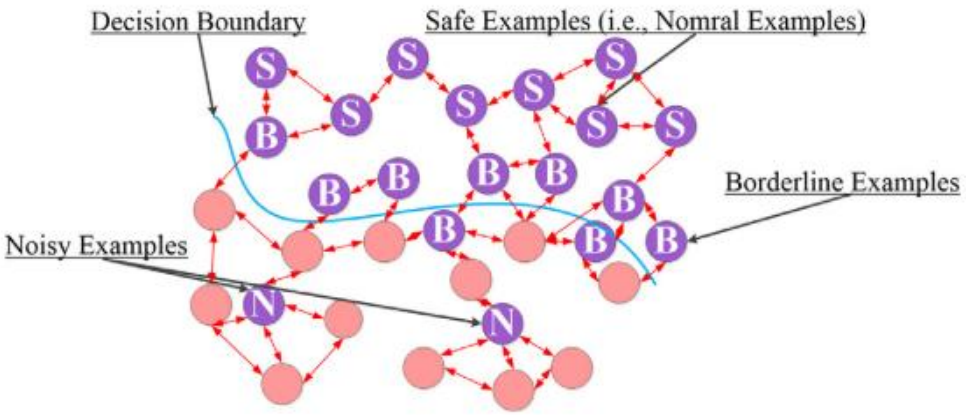

Figura 6 – Representação de conflito de pertinência de elementos em área limítrofe. Fonte: Adaptado de Li *et al.* (2021).

Este comportamento de incerteza na atribuição de pertinência de elementos aos agrupamento correspondentes, é avaliada na TRI em aplicações em estudos humanos de indivíduos que apresentem comportamento limítrofe entre grupos de pessoas contidas em determinados agrupamentos ou

classificadas entre saudáveis e doentes, como demonstrado no trabalho de Mcmahon *et al.* (2019). Assim como a avaliação destas dificuldade na área de saúde, há trabalhos como os de Jia; Zhang; He (2014) e Liu (2021) avaliando as dificuldades na identificação de elementos em áreas de *borderline* dos *clusters*, registrados nos experimentos desta Tese, assim como *datasets* de grande dimensionalidade observados na Tabela 4.

Os dados expressam apenas o momento em que os algoritmos agrupam os itens em classes, mas não considera esta avaliação quando ocorrem dificuldades com o desequilíbrio de classe ou alta dimensionalidade dos itens, conforme outros estudos (Tahir; Kittler; Yan, 2012). Os datasets originais estão representados em classes binárias e seus elementos identificados nas suas respectivas classes. Partindo desta classificação já existente, para estes experimentos iniciais foram determinados dois grupos correspondendo a novas amostragens de classes. O primeiro grupo representou amostras de cada classe com subamostragem de 50% da classe minoritária e reduzindo na amostra a quantidade de elementos da classe majoritária, balanceando as duas classes, mesmo com perda de informação. Para o segundo grupo, foi realizado o mesmo procedimento, porém com a equivalência de 100% dos elementos da classe minoritária. Este balanceamento indireto proporcionou as condições para os primeiros experimentos, descritos a seguir.

4.1.2. Tempo de ajuste na formação dos agrupamentos

Para a determinação de algoritmos ótimos dentre os analisados, deve-se considerar o tempo na atribuição dos itens em agrupamentos distintos. No entanto, podem ocorrer problemas relacionados ao método de seleção do melhor algoritmo, e estes problemas são tradicionalmente atribuídos aos tempos que estes agrupamentos são formados. Os testes realizados nos experimentos desta seção consideram como parâmetro o tempo de resposta dos algoritmos de *clustering*, sendo o tempo de resposta uma estimativa de tempo de avaliação de complexidade, assim como as medições para Ordem de Grandeza (O-grande). O O-grande utiliza como fator avaliativo a implementação de cada algoritmo, o tempo medido de obtenção de uma solução válida e configurações dos computadores onde foram realizados os testes. No trabalho de Beleites; Salzer;

Sergo (2013) foi relatado que a complexidade nas dimensões pode ocasionar uma ambiguidade da referência de elementos limítrofes, comprometendo os resultados de métricas consideradas como referência em qualidade dos analistas humanos (*i.e.* padrão-ouro), necessitando-se de uma avaliação dos elementos limítrofes como pior caso, melhor caso e o desempenho esperado para a aplicação dos resultados. Os tempos de formação dos agrupamentos definitivos foram calculados considerando o número de itens em cada conjunto de dados, os grupos de subamostragem de 50% e 100%, como pode ser visto na Tabela 4 e Tabela 6.

Na Tabela 5 estão representados em negrito os melhores tempos calculados em unidades de milissegundos, e em vermelho os piores resultados, para o grupo de subamostragem de 50% dos elementos das classes minoritárias.

Os algoritmos estão representados na legenda por Fanny(Euclidean) (Alg1), Fanny(Manhattan) (Alg2), Fanny(SqEuclidean) (Alg3), PAM(Euclidean) (Alg4), AM(Manhattan) (Alg5), Clara(Euclidean) (Alg6), Clara(Manhattan) (Alg7), Clara(Jaccard) (Alg8), Sota(Euclidean) (Alg9), K-means(Hartigan-Wong) (Alg10), K-means(Lloyd) (Alg11), K-means(Forgy) (Alg12), K-means(MacQueen) (Alg13), Diana(Euclidean) (Alg14), K-means(arma/Euclidean) (Alg15), K-means(rcpp/Euclidean) (Alg16), Mini Batch K-means(k-means++/Euclidean) (Alg17), Mini Batch K-means(random/Euclidean) (Alg18), Mini Batch K-means(optimal init/Euclidean) (Alg19), Mini Batch K-means(quantile init/Euclidean) (Alg20), Fuzzy C-means(Euclidean) (Alg21), Fuzzy C-means(Manhattan) (Alg22), Cluster Medoids(Euclidean) (Alg23), Cluster Medoids(Manhattan) (Alg24), Cluster Medoids(Chebyshev) (Alg25),Cluster Medoids(Canberra) (Alg26), Cluster Medoids(Braycurtis) (Alg27), Cluster Medoids(Pearson correlation) (Alg28), Cluster Medoids(Simple matching coefficient) (Alg29), Cluster Medoids(Minkowski) (Alg30), Cluster Medoids(Hamming) (Alg31), Cluster Medoids(Jaccard coefficient) (Alg32), Cluster Medoids(Rao coefficient) (Alg33), Cluster Medoids(Mahalanobis) (Alg34), Cluster Medoids(Cosine) (Alg35), Gustafson-Kessel(Mahalanobis) (Alg36), MOCCA(k-means) (Alg37) e MOCCA(neuralgas) (Alg38).

Tabela 5 – Valores do tempo em milissegundos, do grupo das amostras de 50%.

Algoritmo	*Dataset 1*	*Dataset 2*	*Dataset 3*	*Dataset 4*	*Dataset 5*	*Dataset 6*	*Dataset 7*	*Dataset 8*	*Dataset 9*	*Dataset 10*	*Dataset 11*
Alg1	1,10E-06	3,14E-08	4,61E-08	3,64E-08	6,38E-08	4,63E-08	3,37E-08	2,81E-07	4,58E-08	1,51E-07	5,48E-08
Alg2	3,15E-08	6,56E-08	5,12E-08	3,37E-08	7,93E-08	5,09E-08	2,12E-07	6,12E-08	4,15E-08	2,33E-07	5,23E-08
Alg3	2,97E-08	6,80E-08	5,48E-08	2,96E-08	6,05E-08	2,23E-07	3,74E-08	1,52E-07	3,91E-08	2,40E-07	5,39E-08
Alg4	3,54E-08	2,31E-08	2,89E-08	2,91E-08	1,58E-08	1,87E-08	2,03E-08	1,53E-08	1,84E-08	5,95E-08	2,40E-08
Alg5	3,40E-08	2,32E-08	2,32E-08	2,91E-08	2,34E-08	3,17E-08	2,54E-08	1,64E-08	1,79E-08	5,90E-08	2,37E-08
Alg6	6,07E-08	1,58E-08	1,95E-08	2,55E-08	1,11E-08	4,53E-08	2,19E-08	1,09E-08	1,90E-08	8,18E-09	1,02E-08
Alg7	2,83E-08	1,68E-08	2,10E-08	2,64E-08	1,09E-08	1,08E-08	1,85E-08	1,01E-08	2,00E-08	8,75E-09	1,06E-08
Alg8	2,87E-08	1,90E-08	2,56E-08	2,67E-08	9,92E-09	9,81E-09	2,07E-08	1,03E-08	1,74E-08	7,87E-09	1,08E-08
Alg9	2,20E-06	6,56E-07	6,99E-07	2,39E-06	4,08E-07	4,13E-07	1,18E-06	4,76E-07	1,01E-06	3,49E-07	4,34E-07
Alg10	5,75E-08	1,31E-08	1,51E-08	2,11E-08	8,42E-09	8,43E-09	1,60E-08	8,15E-09	1,20E-08	8,42E-09	8,13E-09
Alg11	2,56E-08	1,13E-08	2,08E-08	2,07E-08	8,17E-09	8,34E-09	1,59E-08	**8,02E-09**	1,19E-08	2,25E-08	1,02E-08
Alg12	2,59E-08	1,34E-08	1,32E-08	2,07E-08	8,97E-09	8,01E-09	1,61E-08	8,52E-09	1,30E-08	7,53E-09	7,98E-09
Alg13	**2,51E-08**	1,51E-08	1,52E-08	2,06E-08	8,14E-09	8,06E-09	1,57E-08	8,53E-09	1,24E-08	7,79E-09	8,07E-09
Alg14	8,57E-08	2,71E-08	3,35E-08	3,25E-08	3,73E-08	6,51E-08	2,33E-08	4,30E-08	2,35E-08	2,88E-07	5,63E-08
Alg15	5,86E-07	2,18E-08	1,67E-08	2,01E-08	9,11E-09	1,30E-08	1,83E-08	9,42E-09	2,10E-08	7,54E-09	1,00E-08
Alg16	1,58E-07	**8,59E-09**	**9,51E-09**	**1,16E-08**	**6,85E-09**	**7,91E-09**	**9,86E-09**	8,37E-09	**1,01E-08**	**6,83E-09**	**7,71E-09**
Alg17	1,08E-05	2,74E-08	2,91E-08	6,30E-08	1,80E-08	1,51E-08	4,33E-08	2,00E-08	3,41E-08	1,38E-08	2,09E-08
Alg18	7,45E-08	2,81E-08	4,33E-08	6,78E-08	1,84E-08	1,70E-08	4,41E-08	1,97E-08	3,13E-08	1,31E-08	1,70E-08
Alg19	8,22E-08	2,85E-08	2,93E-08	6,37E-08	1,79E-08	1,56E-08	4,49E-08	1,87E-08	3,06E-08	1,33E-08	1,78E-08
Alg20	2,04E-07	5,32E-08	4,95E-08	1,04E-07	2,73E-08	2,36E-08	8,52E-08	2,97E-08	4,96E-08	1,76E-08	2,52E-08
Alg21	9,49E-08	2,29E-08	1,74E-08	2,40E-08	1,66E-08	1,97E-08	2,40E-08	1,77E-08	1,84E-08	1,15E-08	1,16E-08
Alg22	4,52E-08	2,55E-08	7,19E-08	3,13E-08	1,28E-08	2,37E-08	2,29E-08	1,50E-08	2,34E-08	1,29E-08	1,74E-08
Alg23	4,81E-06	3,89E-08	3,61E-08	5,66E-08	3,40E-08	3,61E-08	5,35E-08	3,52E-08	3,84E-08	5,44E-08	3,68E-08
Alg24	9,22E-08	3,78E-08	3,64E-08	5,87E-08	3,25E-08	3,76E-08	5,31E-08	1,06E-07	3,85E-08	5,53E-08	3,79E-08
Alg25	1,04E-07	3,51E-08	4,24E-08	5,78E-08	3,31E-08	4,04E-08	5,28E-08	3,57E-08	3,99E-08	8,76E-08	3,90E-08
Alg26	9,03E-08	3,64E-08	4,49E-08	5,84E-08	5,08E-08	4,20E-08	5,30E-08	3,60E-08	3,92E-08	6,11E-08	4,51E-08
Alg27	9,15E-08	3,91E-08	4,64E-08	5,84E-08	3,65E-08	4,56E-08	5,35E-08	3,88E-08	4,07E-08	6,83E-08	4,46E-08
Alg28	8,79E-08	5,26E-08	8,00E-08	6,66E-08	5,51E-08	7,80E-08	5,95E-08	5,60E-08	4,90E-08	1,19E-07	7,17E-08
Alg29	1,12E-07	4,04E-08	5,03E-08	5,89E-08	3,69E-08	4,66E-08	5,84E-08	3,70E-08	4,26E-08	6,57E-08	4,67E-08
Alg30	8,87E-08	5,50E-08	7,19E-08	6,43E-08	4,81E-08	7,02E-08	5,66E-08	4,39E-08	5,09E-08	8,61E-08	6,81E-08
Alg31	9,61E-08	4,70E-08	3,99E-08	5,82E-08	4,20E-08	5,02E-08	5,62E-08	4,01E-08	4,25E-08	6,72E-08	4,95E-08
Alg32	9,15E-08	4,61E-08	5,51E-08	6,09E-08	4,27E-08	5,72E-08	5,84E-08	4,17E-08	4,51E-08	7,66E-08	5,10E-08
Alg33	8,77E-08	4,01E-08	4,50E-08	6,00E-08	4,28E-08	5,18E-08	5,40E-08	4,09E-08	4,31E-08	7,62E-08	4,77E-08
Alg34	1,98E-07	1,41E-07	1,05E-07	7,07E-08	9,54E-08	6,83E-08	5,71E-08	4,75E-08	9,10E-08	9,32E-08	6,07E-08
Alg35	9,05E-08	6,14E-08	1,04E-07	7,20E-08	5,33E-08	8,97E-08	5,93E-08	4,55E-08	2,86E-07	9,05E-08	7,81E-08
Alg36	**2,48E-05**	**2,35E-04**	1,20E-05	7,88E-06	1,33E-06	**6,45E-05**	**2,27E-05**	**1,32E-05**	**1,69E-05**	**5,11E-05**	5,43E-06
Alg37	2,44E-05	1,38E-05	**1,74E-05**	**1,90E-05**	**1,04E-05**	1,14E-05	1,61E-05	9,68E-06	1,25E-05	1,18E-05	**1,06E-05**
Alg38	2,39E-05	1,42E-05	1,70E-05	**1,90E-05**	9,54E-06	1,15E-05	1,56E-05	8,99E-06	1,23E-05	1,22E-05	**1,06E-05**

Fonte: Autor

Na Tabela 5, é possível observar que os algoritmos *K-means* (Lloyd) (Alg11) no *dataset* 8, *K-means*(MacQueen) (Alg13) no *dataset* 1 e K-means(rcpp/Euclidean) (Alg16) nos *datasets* de 2 a 7 e de 9 a 11 registraram os menores tempos para a formação dos agrupamentos, ajustando o centróide dos grupos pelas inicializações e dissimilaridades. Na Tabela 5, também é possível observar que os algoritmos Gustafson-Kessel(Mahalanobis) (Alg36) nos *datasets* 1, 2 e de 6 a 10, o algoritmo MOCCA(k-means) (Alg37) nos *datasets* 3 a 5 e 11 e

o algoritmo MOCCA(neuralgas) (Alg38) nos *datasets* 4 e 11 registraram os piores tempos para a formação dos agrupamentos, ajustando o centróide dos grupos pelas inicializações e dissimilaridades. Os demais algoritmos registraram tempos intermediários nos tempos para a formação dos agrupamentos, ajustando o centróide dos grupos pelas inicializações e dissimilaridades, para os *datasets* avaliados nos experimentos desta Tese.

Tabela 6 – Valores do tempo em milissegundos, do grupo das amostras de 100%.

Algoritmo	*Dataset 1*	*Dataset 2*	*Dataset 3*	*Dataset 4*	*Dataset 5*	*Dataset 6*	*Dataset 7*	*Dataset 8*	*Dataset 9*	*Dataset 10*	*Dataset 11*
Alg1	5,37E-07	4,08E-08	6,52E-08	1,52E-07	9,05E-08	8,47E-08	4,08E-08	1,96E-07	6,13E-08	3,43E-07	1,02E-07
Alg2	4,19E-08	8,34E-08	9,40E-08	3,83E-08	1,14E-07	9,16E-08	1,83E-07	1,96E-07	7,93E-08	5,52E-07	1,20E-07
Alg3	5,14E-08	1,81E-07	8,22E-08	3,06E-08	1,30E-07	2,42E-06	4,20E-08	2,13E-07	9,62E-08	4,09E-07	1,00E-07
Alg4	2,54E-08	2,92E-08	2,72E-08	2,38E-08	2,76E-08	5,46E-08	1,78E-08	4,57E-08	1,55E-08	1,77E-07	3,96E-08
Alg5	2,50E-08	2,82E-08	2,82E-08	2,55E-08	4,34E-08	4,28E-08	2,05E-08	2,65E-08	1,93E-08	1,31E-07	4,95E-08
Alg6	2,55E-08	1,14E-08	1,32E-08	2,31E-08	8,88E-09	7,69E-09	4,25E-08	8,58E-09	1,17E-08	7,42E-09	9,69E-09
Alg7	2,72E-08	1,23E-08	1,36E-08	2,35E-08	8,93E-09	7,86E-09	1,76E-08	8,10E-09	1,22E-08	1,20E-08	8,87E-09
Alg8	2,62E-08	1,34E-08	1,78E-08	2,35E-08	7,99E-09	8,17E-09	1,65E-08	8,00E-09	1,20E-08	1,46E-08	9,82E-09
Alg9	1,07E-06	5,29E-07	4,39E-07	1,49E-06	4,05E-07	2,05E-07	7,56E-07	2,50E-07	6,34E-07	3,27E-07	3,02E-07
Alg10	1,57E-08	8,73E-09	1,04E-08	1,39E-08	7,33E-09	7,29E-09	1,14E-08	7,25E-09	9,10E-09	9,95E-09	7,35E-09
Alg11	1,56E-08	1,07E-08	1,02E-08	1,34E-08	7,63E-09	7,81E-09	1,11E-08	7,01E-09	8,78E-09	8,52E-09	**7,15E-09**
Alg12	1,82E-08	9,80E-09	1,03E-08	1,43E-08	7,55E-09	9,64E-09	1,16E-08	7,01E-09	9,01E-09	7,00E-09	7,64E-09
Alg13	1,78E-08	9,21E-09	1,03E-08	1,36E-08	7,15E-09	7,43E-09	1,10E-08	7,05E-09	1,01E-08	**6,98E-09**	7,27E-09
Alg14	2,43E-08	4,41E-08	1,34E-07	2,68E-08	1,05E-07	2,74E-07	2,20E-08	1,35E-07	3,08E-08	1,08E-06	1,98E-07
Alg15	2,41E-08	1,09E-08	**9,46E-09**	1,41E-08	8,56E-09	8,90E-09	1,69E-08	8,00E-09	9,13E-09	2,89E-08	7,92E-09
Alg16	**9,76E-09**	**8,36E-09**	1,16E-08	**9,22E-09**	**6,99E-09**	**6,61E-09**	**8,23E-09**	**7,00E-09**	**7,72E-09**	1,00E-08	**7,15E-09**
Alg17	4,87E-08	1,84E-08	2,39E-08	3,51E-08	1,32E-08	1,48E-08	2,71E-08	1,50E-08	2,11E-08	1,47E-08	1,59E-08
Alg18	4,31E-08	1,97E-08	2,00E-08	3,69E-08	1,44E-08	1,28E-08	2,63E-08	1,55E-08	2,18E-08	1,26E-08	1,46E-08
Alg19	5,64E-08	2,14E-08	2,41E-08	3,51E-08	1,36E-08	1,46E-08	2,64E-08	1,30E-08	1,98E-08	1,30E-08	1,49E-08
Alg20	9,08E-08	2,76E-08	2,79E-08	5,61E-08	2,06E-08	1,68E-08	4,55E-08	1,98E-08	3,18E-08	1,73E-08	1,78E-08
Alg21	3,76E-08	2,41E-08	1,29E-08	1,62E-08	1,15E-08	1,52E-08	1,74E-08	1,27E-08	1,51E-08	1,57E-08	1,08E-08
Alg22	2,29E-08	1,57E-08	6,39E-08	4,84E-08	1,61E-08	2,31E-08	1,72E-08	1,01E-08	1,85E-08	1,62E-08	1,71E-08
Alg23	5,58E-08	3,67E-08	3,52E-08	3,90E-08	4,37E-08	5,54E-08	5,05E-08	4,22E-08	3,07E-08	9,85E-08	5,50E-08
Alg24	5,06E-08	3,59E-08	3,77E-08	3,74E-08	4,69E-08	6,00E-08	3,70E-08	4,63E-08	3,18E-08	1,05E-07	6,05E-08
Alg25	4,91E-08	4,01E-08	3,79E-08	3,89E-08	4,86E-08	6,49E-08	3,76E-08	4,80E-08	3,39E-08	1,06E-07	6,27E-08
Alg26	5,63E-08	3,96E-08	4,11E-08	3,89E-08	5,06E-08	6,76E-08	4,20E-08	4,87E-08	3,84E-08	1,14E-07	6,34E-08
Alg27	6,30E-08	2,30E-07	4,56E-08	4,12E-08	5,31E-08	7,31E-08	3,98E-08	5,25E-08	3,73E-08	1,25E-07	7,04E-08
Alg28	5,89E-08	6,91E-08	1,05E-07	5,90E-08	9,07E-08	1,35E-07	4,81E-08	8,77E-08	5,58E-08	2,22E-07	1,24E-07
Alg29	5,28E-08	4,59E-08	5,51E-08	4,31E-08	5,56E-08	7,58E-08	3,82E-08	5,14E-08	3,77E-08	1,20E-07	7,36E-08
Alg30	5,96E-08	6,99E-08	1,02E-07	5,32E-08	7,49E-08	1,15E-07	5,42E-08	6,64E-08	5,28E-08	1,61E-07	1,15E-07
Alg31	5,49E-08	4,45E-08	4,82E-08	4,19E-08	6,15E-08	8,24E-08	4,00E-08	6,03E-08	4,07E-08	1,39E-07	8,08E-08
Alg32	5,06E-08	5,39E-08	6,93E-08	4,61E-08	6,22E-08	9,35E-08	4,06E-08	5,98E-08	4,57E-08	1,77E-07	8,84E-08
Alg33	5,20E-08	4,67E-08	5,42E-08	4,42E-08	6,34E-08	8,68E-08	4,18E-08	6,01E-08	4,05E-08	1,43E-07	8,04E-08
Alg34	5,36E-08	9,43E-08	1,66E-07	6,34E-08	7,25E-08	1,13E-07	4,52E-08	7,18E-08	5,08E-08	2,00E-07	1,15E-07
Alg35	6,30E-08	8,82E-08	1,69E-07	6,91E-08	8,24E-08	1,64E-07	4,78E-08	9,73E-08	6,44E-08	1,87E-07	1,47E-07
Alg36	**3,79E-05**	**2,73E-05**	1,43E-05	**3,67E-04**	8,22E-07	**5,40E-05**	**2,88E-05**	**1,46E-05**	8,54E-06	4,45E-07	5,46E-06
Alg37	1,50E-05	1,16E-05	1,91E-05	1,43E-05	**1,04E-05**	1,75E-05	1,15E-05	1,01E-05	**1,07E-05**	**2,28E-05**	1,56E-05
Alg38	1,50E-05	1,18E-05	**1,93E-05**	1,44E-05	1,02E-05	1,70E-05	1,14E-05	1,01E-05	**1,07E-05**	2,24E-05	**1,59E-05**

Fonte: Autor

Na Tabela 6 estão representados em negrito os melhores tempos calculados em unidades de milissegundos, e em vermelho os piores resultados, para o grupo de subamostragem de 100% dos elementos das classes minoritárias.

Na Tabela 6, é possível observar que os algoritmos *K-means* (MacQueen) (Alg13) no *dataset* 10, *K-means*(arma/Euclidean) (Alg15) no *dataset* 3 e K-means(rcpp/Euclidean) (Alg16) nos *datasets* 1, 24 a 9 e 11 registraram os menores tempos para a formação dos agrupamentos, ajustando o centóride dos grupos pelas inicializações e dissimilaridades.

Na Tabela 6, também é possível observar que os algoritmos Gustafson-Kessel(Mahalanobis) (Alg36) nos *datasets* 1, 2, 4 e de 6 a 8, o algoritmo MOCCA(*k-means*) (Alg37) nos *datasets* 5, 9 e 10 e o algoritmo MOCCA(neuralgas) (Alg38) nos *datasets* 3, 9 e 11 registraram os piores tempos para a formação dos agrupamentos, ajustando o centróide dos grupos pelas inicializações e dissimilaridades. Os demais algoritmos registraram tempos intermediários nos tempos para a formação dos agrupamentos, ajustando o centróide dos grupos pelas inicializações e dissimilaridades, para os *datasets* avaliados nos experimentos desta Tese.

Com os ajustes nas quantidades de 50% e 100% dos elementos dos grupos minoritários para balanceamento com subamostragem dos grupos analisados, o resultado obtido para os algoritmos baseados em k-means apresentaram bons resultados ao registrar os menores tempos. Os algoritmos de k-means apresentam O(nkd) como ordem de grandeza de complexidade (Benabdellah; Benghabrit; Bouhaddou, 2019), esta ordem de grandeza foi diferenciada dos demais algoritmos baseado em k-means dos experimentos realizados pelos métodos de inicialização dos centroides. Algoritmos que apresentam ordens de grandeza como PAM (O(k$(n-k)^2$)), CLARA (O(k$(40+k)^2$ + k(n-k))) e SOM (O(n^2m)) utilizam métodos complexos em suas implementações e processamento dos dados. O crescimento da complexidade de processamentos também está associado ao grau do polinômio do O-Grande. No trabalho de Tang *et al.* (2019) foram avaliações os tempos de medição de *clusters* onde o resultado do objetivo do algoritmo foi a regressão, onde foi observado que O($mn^{1,65}$) – O($mn^{1,70}$) está associado a complexidade média de tempo do método proposto, considerando ser muito mais rápido que o método O(mn^3), o tempo como unidade de medição para a complexidade não considerando as demais dificuldades para as formações dos agrupamentos. Para uma comparação mais objetiva, foram gerados os gráficos de barras dos grupos de 50% e 100% na Figura 7 com os valores de máximo e mínimo do tempo de formação dos agrupamentos.

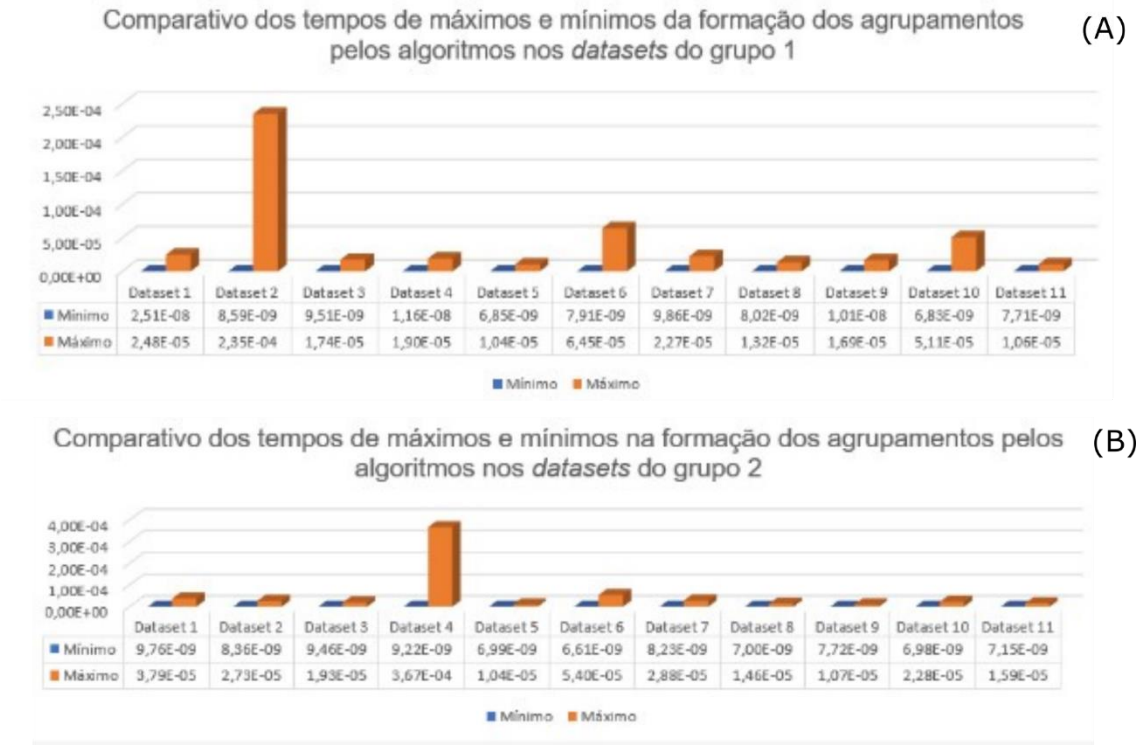

Figura 7 – Gráficos do tempo de formação do agrupamento nos grupos de 50% (a) e 100% (b) de cada *dataset*. Fonte: Autor.

Na Figura 7(a) é possível observar que no grupo 1 equivalente a 50% da classe minoritária o *Dataset* 2 é otimizado com novas amostras do grupo 2 na Figura 7(b) , acrescendo mais dados e reduzindo o tempo geral para a aplicação odos demais algoritmos. O comportamento ocorrido no *dataset 2* pode ser observado nos demais *datasets*, exceto no *dataset* 4.

Para uma avaliação numérica, foram ordenados os valores da Figura 7(a) e (b), onde estão ordenados os valores de máximos e mínimos dos dois grupos. Para uma visualização geral do tempo de formação dos agrupamentos foram gerados gráficos de barras na Figura 8.

Na Figura 8 (a) e (b) estão representadas as médias dos tempos dos dois grupos de amostragens, onde os *datasets* estão representados os *datasets* de 1 a 11 da Tabela 4 de baixo para cima do gráfico. Na Figura 8(a) é possível observar que o tempo calculado pelo algoritmo Gustafson-Kessel(Mahalanobis) (Alg 36) no *dataset* 2 é otimizado na Figura 8(b) e que há uma inversão de qualidade de tempo para o memo algoritmo no *dataset* 4.

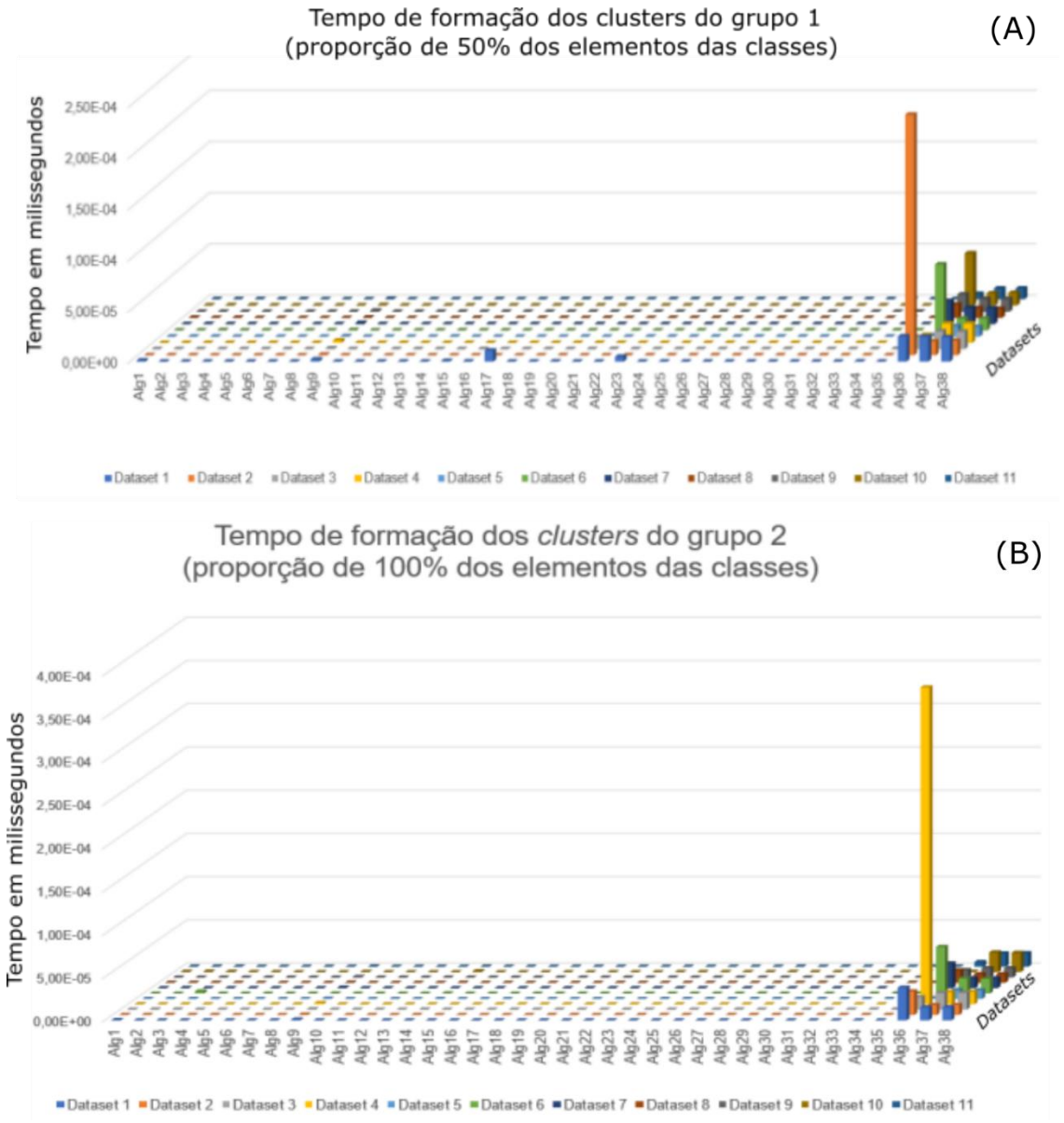

Figura 8 – Representação geral das médias do tempo de formação dos agrupamentos nos grupos de amostras de 50% (a) e 100% (b) de cada *dataset*. Fonte: Autor.

Nos resultados obtidos nos experimentos desta Tese, como pode ser observado na Tabela 7, onde há uma pressuposta comprovação da existência de dificuldades para a formação dos agrupamentos dos itens nos referidos *clusters*. Esta dificuldade deve ser considerada como fator, com destaque para que os algoritmos baseados em *k-means* apresentem os melhores resultados, diferentemente de outros trabalhos que não incluam a dificuldade nesta determinação.

Tabela 7 – Registros do tempo de máximos e mínimos.

Datasets	Grupo dos 50%		Grupo dos 100%	
	Mínimo	Máximo	Mínimo	Máximo
Dataset 1	2,51E-08	2,48E-05	9,76E-09	3,79E-05
Dataset 2	8,59E-09	**2,35E-04**	8,36E-09	2,73E-05
Dataset 3	9,51E-09	1,74E-05	9,46E-09	1,93E-05
Dataset 4	1,16E-08	1,90E-05	9,22E-09	**3,67E-04**
Dataset 5	6,85E-09	1,04E-05	6,99E-09	1,04E-05
Dataset 6	7,91E-09	6,45E-05	**6,61E-09**	5,40E-05
Dataset 7	9,86E-09	2,27E-05	8,23E-09	2,88E-05
Dataset 8	8,02E-09	1,32E-05	7,00E-09	1,46E-05
Dataset 9	1,01E-08	1,69E-05	7,72E-09	1,07E-05
Dataset 10	**6,83E-09**	5,11E-05	6,98E-09	2,28E-05
Dataset 11	7,71E-09	1,06E-05	7,15E-09	1,59E-05

Fonte: Autor.

Para identificar os fatores que tornam complexas as decisões, e posteriormente comparar com os itens previamente agrupados nos respectivos grupos, é necessário calcular o Parâmetro de Dificuldade b dos três Modelos Logísticos da TRI, e determinar qual Parâmetro de Dificuldade b é ideal para expressar numericamente a dificuldade que impacta nos objetivos dos algoritmos. Inclusive, a variabilidade dos dados é outro fator que também agrega complexidade aos algoritmos (Cunha; Mendes; Vilela, 2021).

Existem dificuldades relacionadas à obtenção do tempo gasto pelos algoritmos nos agrupamentos, classificações, na correlação dos elementos nos respectivos grupos, na variabilidade dos tipos de dados, na dimensionalidade e desequilíbrio das classes nos conjuntos de dados, o que impacta diretamente na formação das classes dos itens e proporcionando dificuldades aos algoritmos de *clustering* devido a prevalência das classes majoritárias (Bergner; Halpin; Vie, 2022). Nos próximos experimentos foram realizadas avaliações de correlações estatísticas para identificar o impacto das variações de elementos contidos nos grupos no momento das atribuições dos elementos nos respectivos agrupamentos. Na etapa seguinte dos experimentos ainda foram utilizados todos os algoritmos da *Tabela 3* e os *datasets* D1 a D11 da Tabela 4.

4.1.3. Método comparativo entre as pertinências dos elementos nos agrupamentos e as dificuldades pela estatística de correlação

Foi realizado o teste de Shapiro-Wilk com α igual a 0,05, testando cada grupo separadamente por *dataset*, onde foram obtidos *p-value* < α, indicando que há sinais de normalidade para os grupos, possibilitando realizar o teste de correlação de Spearman. A avaliação pela estatística de correlação indica maior proporcionalidade entre os resultados obtidos e a população a ser estimada. No trabalho de Powers (2020) foram avaliados os resultados baseados em uma compreensão objetiva das condições primárias para a análise dos dados, onde a identificação de correspondência direta ou indireta destes casos com base na estatística de correlação, proporciona um entendimento de quais elementos possam apresentar uma correlação mútua e quais não. Devido a grande quantidade de algoritmos utilizados nos experimentos desta Tese, foram organizados em três grupos os algoritmos de características semelhantes. No grupo A estão contidos os algoritmos de *clustering* que utilizam diferentes técnicas de inicialização de centroides, no grupo B estão contidos os algoritmos que utilizam o k-means como base e no grupo C estão contidos os algoritmos que utilizam como característica o método de *Fuzzy*, os grupos estão representados na seguinte forma:

i) Grupo A de algoritmos correlacionados por Spearman: Fanny(Euclidean) (Alg1), Fanny(Manhattan) (Alg2), Fanny(SqEuclidean) (Alg3), PAM(Euclidean) (Alg4), PAM(Manhattan) (Alg5), Clara(Euclidean) (Alg6), Clara(Manhattan) (Alg7), Clara(Jaccard)(Alg8), Sota(Euclidean) (Alg9), K-means(Hartigan-Wong) (Alg10), K-means(Lloyd) (Alg11), K-means(Forgy) (Alg12), K-means(MacQueen) (Alg13), Diana(Euclidean) (Alg14) e Gustafson-Kessel(Mahalanobis) (Alg36).

ii) Grupo B de algoritmos correlacionados por Spearman: K-means (Hartigan-Wong) (Alg10), K-means(Lloyd) (Alg11), K-means(Forgy) (Alg12), K-means(MacQueen) (Alg13), Diana(Euclidean) (Alg14), K-means(arma/Euclidean) (Alg15), K-means(rcpp/Euclidean) (Alg16), Mini Batch K-means(k-means++/Euclidean) (Alg17), Mini Batch K-means(random/Euclidean) (Alg18), Mini Batch K-means(optimal init/Euclidean) (Alg19), Mini Batch K-means(quantile init/Euclidean) (Alg20), MOCCA(k-means) (Alg37) e MOCCA(neuralgas) (Alg38).

iii) Grupo C de algoritmos correlacionados por Spearman: Fuzzy C-means(Euclidean) (Alg21), Fuzzy C-means(Manhattan) (Alg22), Cluster Medoids(Euclidean) (Alg23), Cluster Medoids(Manhattan) (Alg24), Cluster Medoids(Chebyshev) (Alg25),Cluster Medoids(Canberra) (Alg26), Cluster Medoids(Braycurtis) (Alg27), Cluster Medoids(Pearson correlation) (Alg28), Cluster Medoids(Simple matching coefficient) (Alg29), Cluster Medoids(Minkowski) (Alg30), Cluster Medoids(Hamming) (Alg31), Cluster Medoids(Jaccard coefficient) (Alg32), Cluster Medoids(Rao coefficient) (Alg33), Cluster Medoids(Mahalanobis) (Alg34) e Cluster Medoids(Cosine) (Alg35).

Os conjuntos estatísticos foram calculados indicando os tipos de correlações contidos nos gráficos das Figura 9 a Figura 19. Foi observado que as três primeiras colunas da esquerda para a direita em cada gráfico representam os três Modelos PL da TRI, as demais colunas, de Alg1 a Alg38, representam os 38 algoritmos de *clustering*.

Estes grupos foram avaliados nos dois grupos de concentrações distintas de elementos, o grupo 1 de 50% e o grupo 2 de 100% dos elementos das classes minoritárias. Estes dois grupos avaliaram na seção anterior o comportamento das formações dos agrupamentos mediante variação da quantidade de elementos. Nesta seção Avaliação Estatística foi realizada para quantificar estas representações estatisticamente, representando em cada um dos gráficos das Figura 9 a Figura 19 as seguintes correspondências: para as legendas (a) e (b) para o grupo A de algoritmos, para as legendas (b) e (c) para o grupo B de algoritmos e para as legendas (d) e (f) para o grupo C de algoritmos. Foram agrupadas as legendas (a), (c) e (e) para representar o grupo 1 de 50% dos elementos das classes minoritárias e as legendas (b), (d) e (f) para representar o grupo 2 de 100% dos elementos das classes minoritárias. Para cada Figura 9 a Figura 19 estão representados um dataset diferente da Tabela 4. Esta avaliação estatística tem a finalidade de comparar as atribuições dos agrupamentos/classificações e os valores de *b*, e não seleciona os melhores algoritmos.

Nos gráficos da Figura 9 é possível observar sutis alteração no grupo A representado pelo Figura 9(A) e (B) onde há uma leve alteração de correlação negativa com tendência para neutra ou positiva na maioria dos algoritmos do grupo A nos parâmetros b da TRI nos três Modelos PL. Para o grupo B a mesma

tendência ocorre na maioria dos algoritmos para os parâmetros *b* da TRI nos três Modelos PL, observável na Figura 9(C) e (D). Para o grupo C a tendência de leve alteração de correlação negativa com tendência para neutra ou positiva ocorre na maioria dos algoritmos, apenas nos modelos 1PL e 2PL.

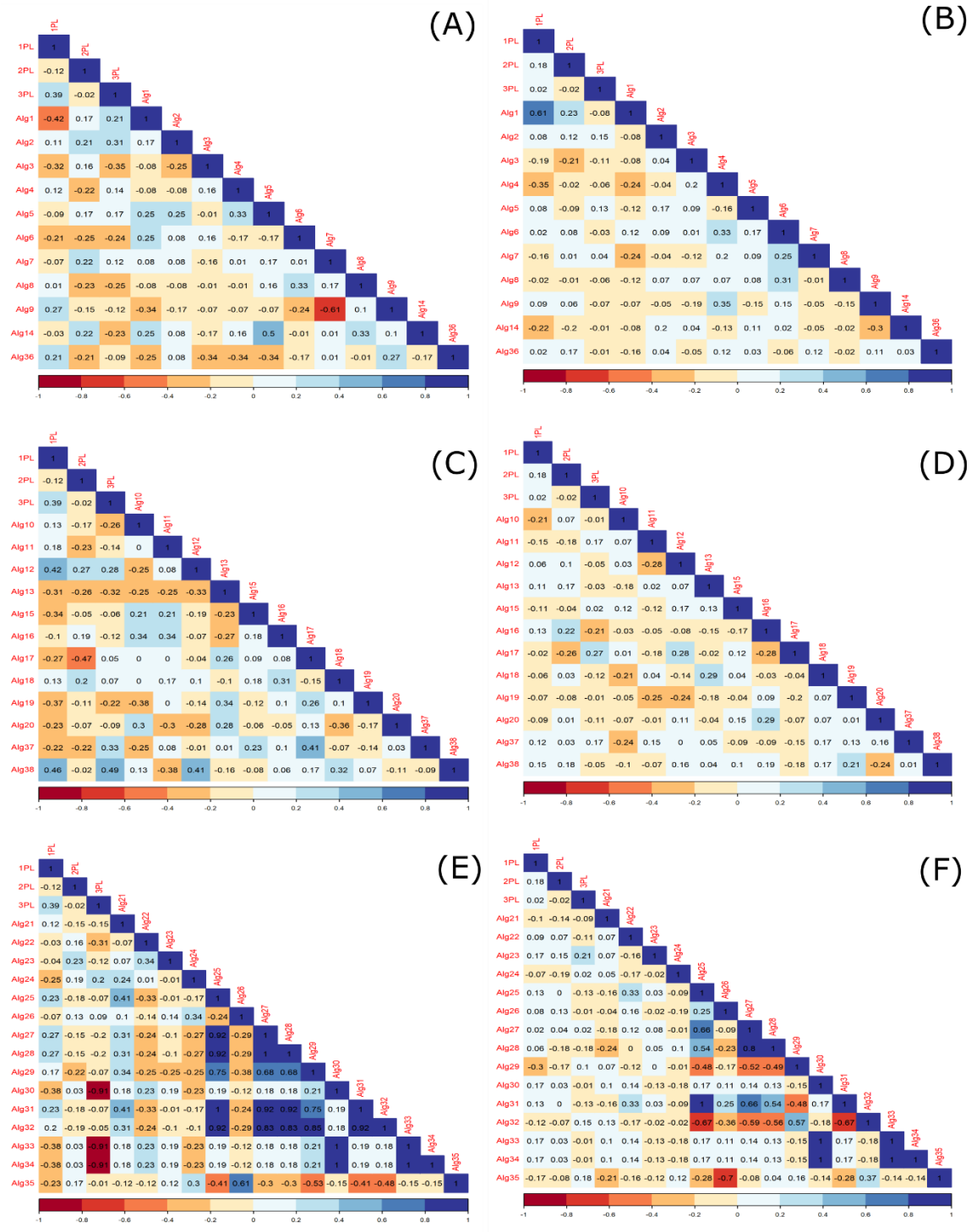

Figura 9 – Conjunto de correlação de Parâmetro de Dificuldade *b* dos grupos 1 e 2: pelo grupo A (A-B), pelo grupo B (C-D) e pelo grupo C (E-F) no *dataset* 1. Fonte: Autor.

Para o 3PL os algoritmos ALG 30, 33 e 34 a tendência de correlação positiva foi mais intensa, observável na Figura 9(E) e (F). Destacando que este grupo se destaca pela técnica de Fuzzy, indicando que os elementos podem estar com diferentes graus de pertinência em ajustes, devido a adição de itens a serem agrupados e aumentando a dificuldade desta classe de algoritmos em determinar as partições definitivas.

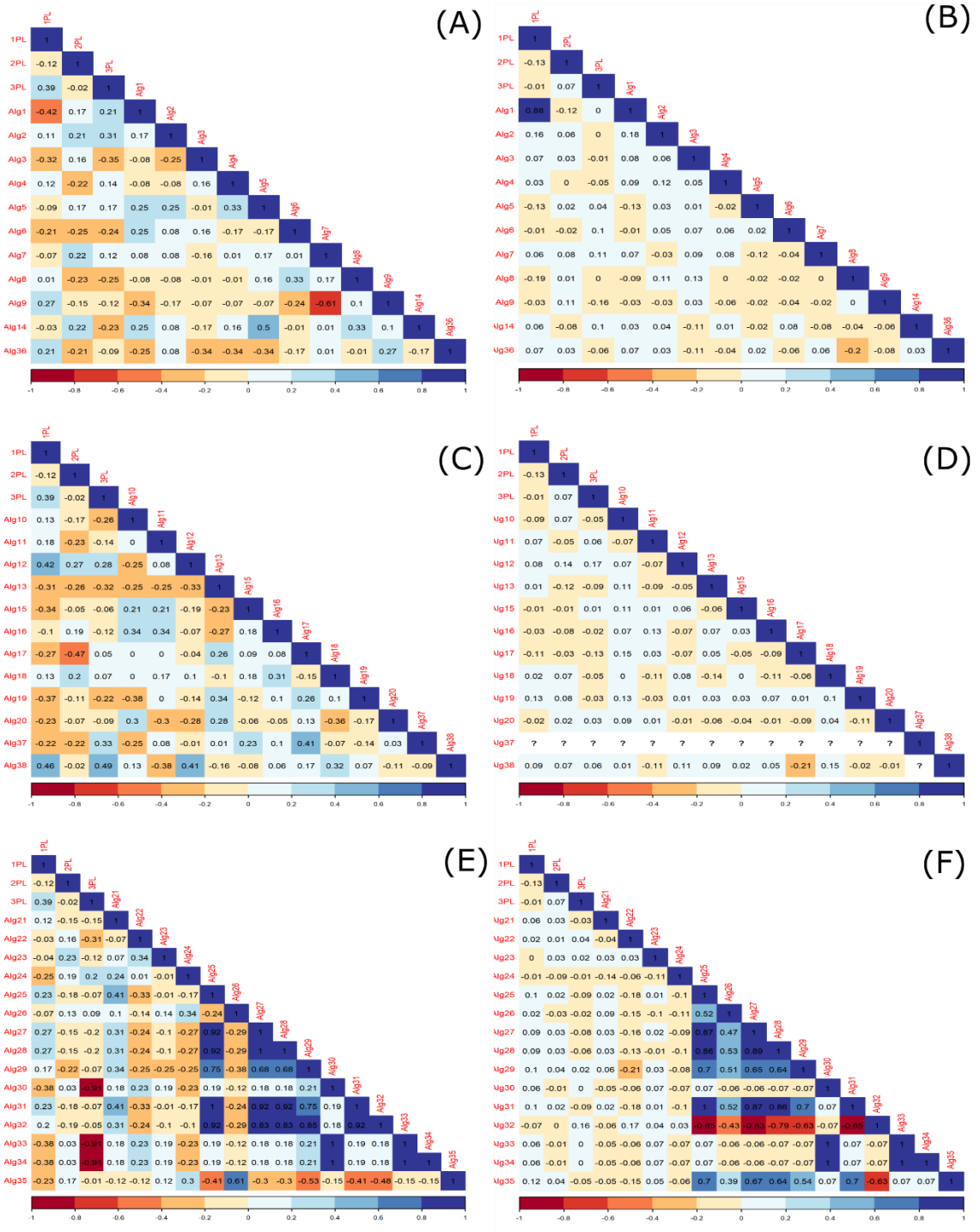

Figura 10 – Conjunto de correlação de Parâmetro de Dificuldade *b* dos grupos 1 e 2: pelo grupo A (A-B), pelo grupo B (C-D) e pelo grupo C (E-F) no *dataset* 2. Fonte: Autor.

Nos gráficos da Figura 10 o mesmo padrão se repete, onde é possível observar sutis alterações no grupo A, representada pela Figura 10(A) e (B). Há uma leve alteração de correlação negativa com tendência para neutra ou positiva na maioria dos algoritmos do grupo A nos parâmetros b da TRI nos três Modelos PL. Destaca-se que o Alg1 apresentou uma forte inversão de correlação aos demais algoritmos do grupo A.

Para o grupo B a mesma tendência ocorre na maioria dos algoritmos para os parâmetros b da TRI nos três Modelos PL, observável em Figura 10(C) e (D). Porém, o Alg37 não registrou resultado devido a problemas de matriz inversiva. Para o grupo C a tendência foi de neutralização das correlações, como pode ser observado nos gráficos da Figura 10(E) e (F).

Nos gráficos da Figura 11 o padrão predominante é de neutralidade das correlações, onde é possível observar no grupo A representado pelo Figura 11(A) e (B) onde há uma alteração significativa de correlação de negativa para positiva no Alg1 no parâmetro b da TRI de Modelos 1PL.

Para o grupo B a mesma tendência ocorre na maioria dos algoritmos para os parâmetros b da TRI nos três Modelos PL, observável em Figura 11(C) e (D). Porém, o Alg10 registrou resultado significativo em relação aos demais algoritmos, e pra o Alg37 não foram registrados os valores das correlações devido a problemas de matriz inversiva. Para o grupo C a tendência foi manter a neutralização das correlações observado nos gráficos da Figura 11(E) e (F).

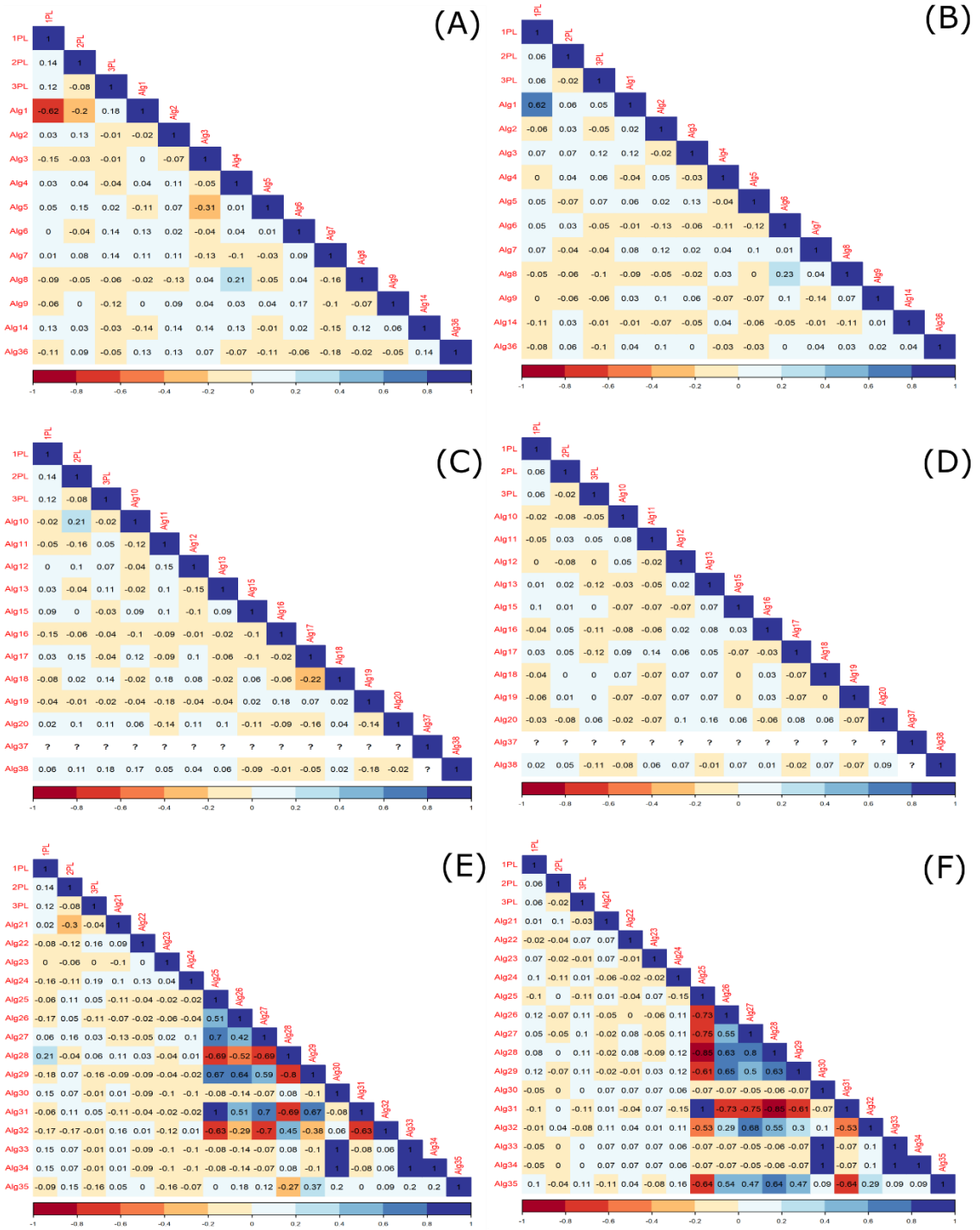

Figura 11 – Conjunto de correlação de Parâmetro de Dificuldade b dos grupos 1 e 2: pelo grupo A (A-B), pelo grupo B (C-D) e pelo grupo C (E-F) no *dataset* 3. Fonte: Autor.

Nos gráficos da Figura 12 o padrão predominante é tender para a neutralidade das correlações, onde é possível observar no grupo A representado pelo Figura 12(A) e (B). A correlação positiva no Alg1 no parâmetro b da TRI de Modelos 1PL se manteve, mesmo decrescendo numericamente. Para o grupo B a tendência de neutralidade na maioria dos algoritmos para os parâmetros b da

TRI nos três Modelos PL, é observável na Figura 12(C) e Figura 12(D). Para o grupo C esta mesma tendência ocorreu, tendendo a neutralização das correlações dos parâmetros b da TRI nos três Modelos PL, como pode ser observado nos gráficos da Figura 12(E) e Figura 12(F).

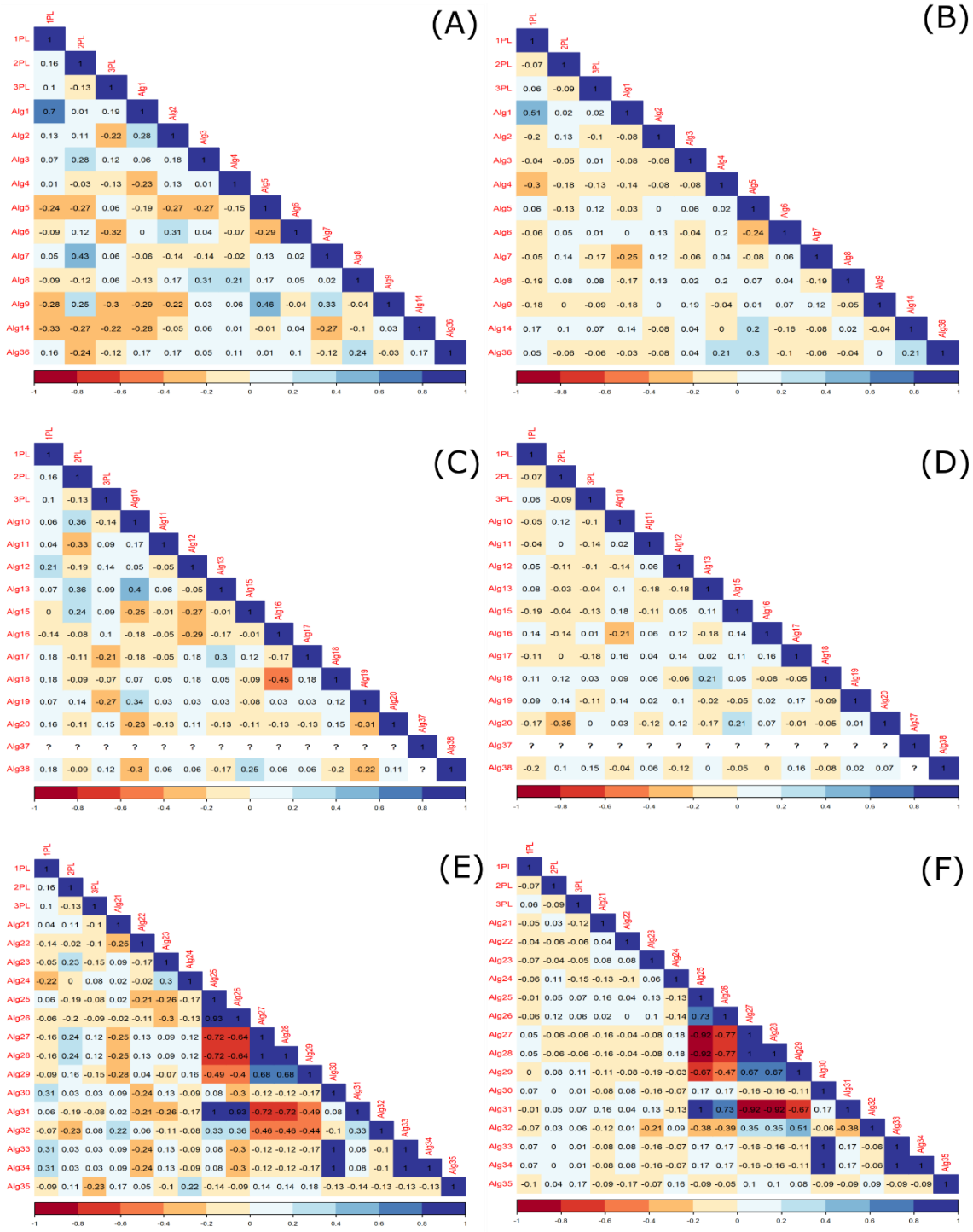

Figura 12 – Conjunto de correlação de Parâmetro de Dificuldade b dos grupos 1 e 2: pelo grupo A (A-B), pelo grupo B (C-D) e pelo grupo C (E-F) no *dataset* 4. Fonte: Autor.

Nos gráficos da Figura 13 o padrão predominante é manter a neutralidade das correlações dos parâmetros b da TRI nos três Modelos PL, onde é possível observar no grupo A representado pelo Figura 13(A) e Figura 13(B).

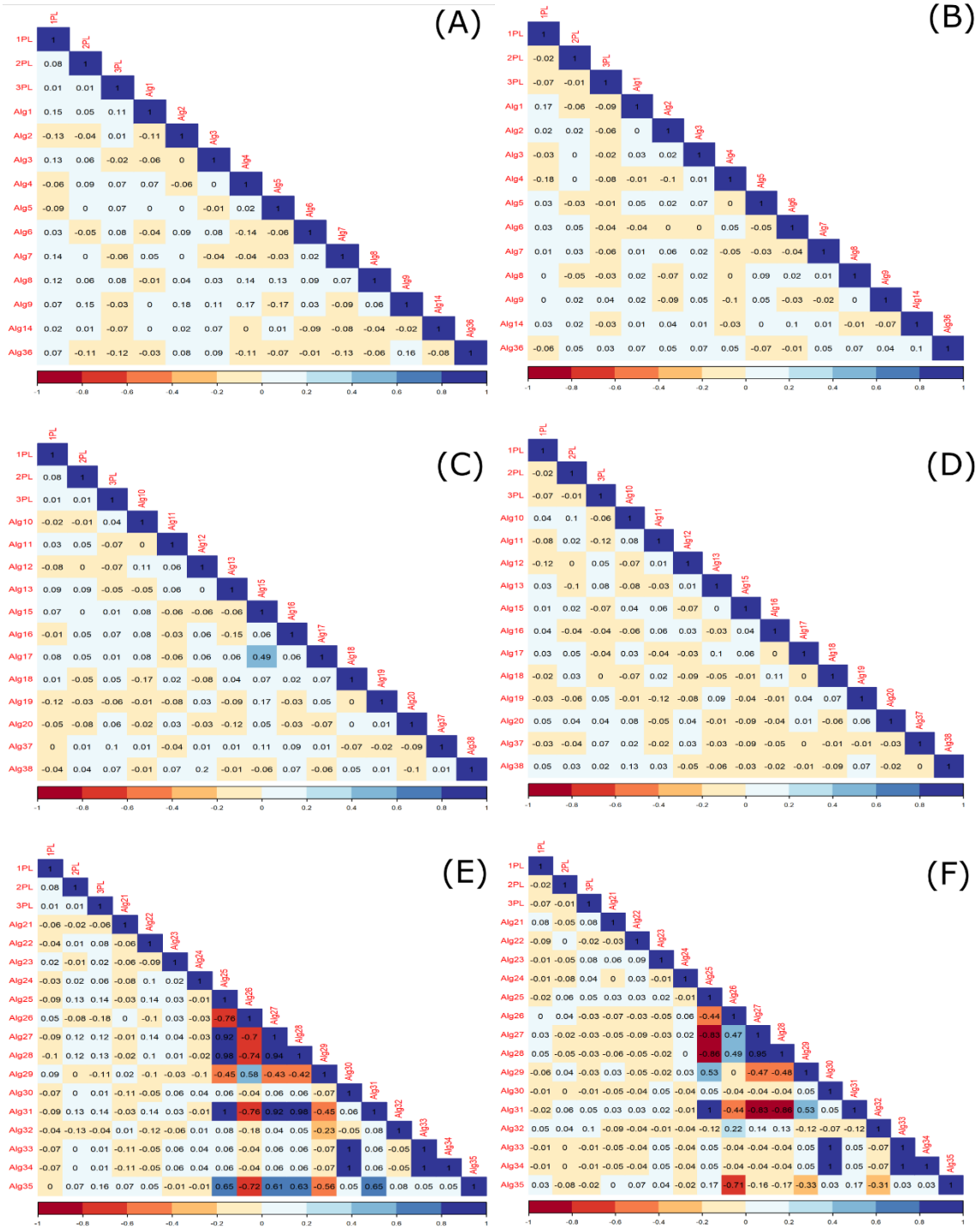

Figura 13 – Conjunto de correlação de Parâmetro de Dificuldade b dos grupos 1 e 2: pelo grupo A (A-B), pelo grupo B (C-D) e pelo grupo C (E-F) no *dataset* 5. Fonte: Autor.

Para o grupo B ocorreu a mesma tendência de neutralidade na maioria dos algoritmos para os parâmetros b da TRI nos três Modelos PL, é observável

na Figura 13(C) e (D). Assim como ocorreu para o grupo C para as correlações dos parâmetros b da TRI nos três Modelos PL, como pode ser observado nos gráficos da Figura 13(E) e Figura 13(F).

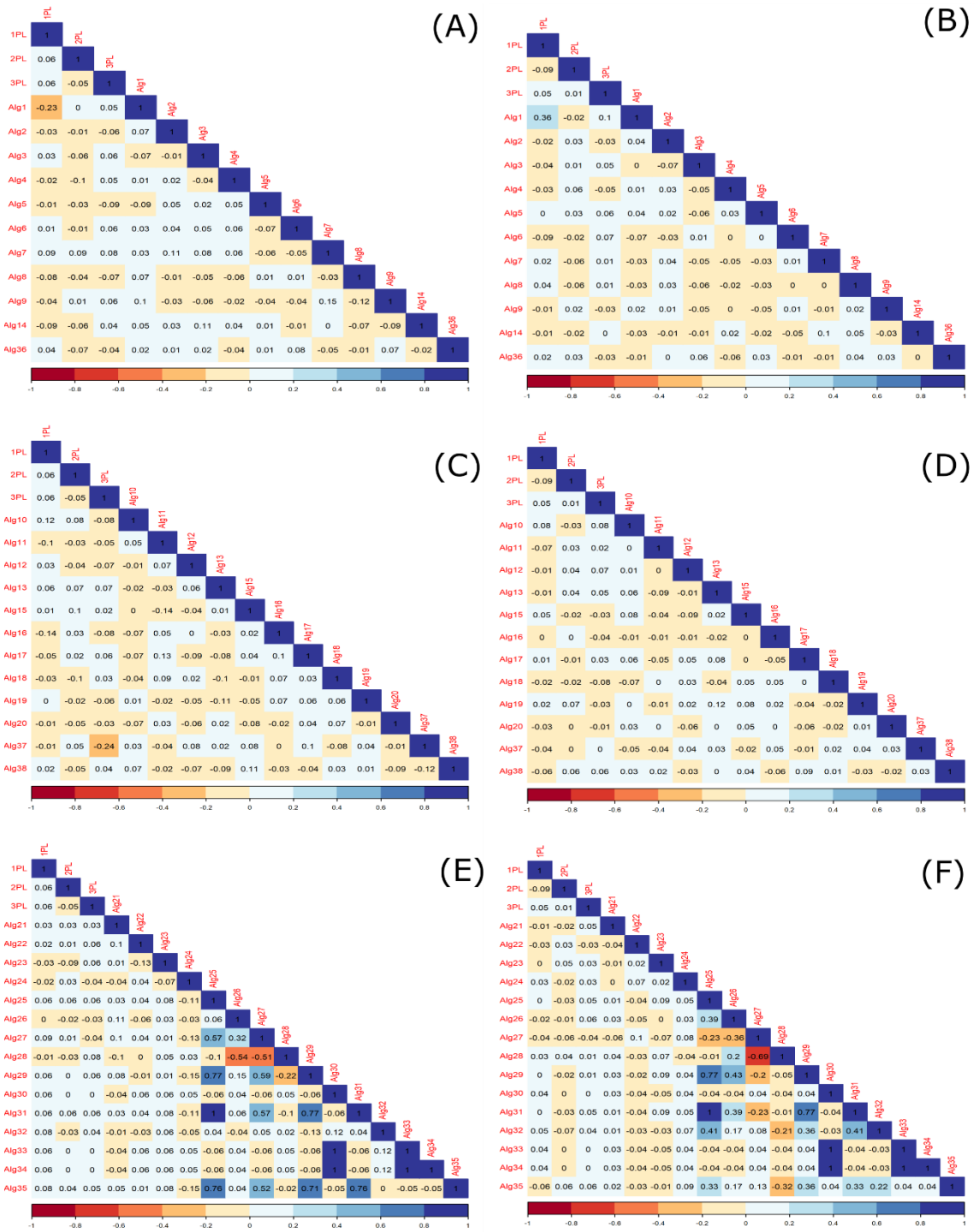

Figura 14 – Conjunto de correlação de Parâmetro de Dificuldade b dos grupos 1 e 2: pelo grupo A (A-B), pelo grupo B (C-D) e pelo grupo C (E-F) no *dataset* 6. Fonte: Autor.

Nos gráficos da Figura 14 o padrão predominante é manter a neutralidade das correlações dos parâmetros b da TRI nos três Modelos PL, onde é possível

observar no grupo A representado pelo Figura 14(A) e (B). Destacando o Alg1 que apresentou inversão de correlação de negativa para positiva. Para o grupo B ocorreu a mesma tendência de neutralidade na maioria dos algoritmos para os parâmetros b da TRI nos três Modelos PL, é observável na Figura 14(C) e Figura 14(D). Assim como ocorreu para o grupo C para as correlações dos parâmetros b da TRI nos três Modelos PL, como pode ser observado nos gráficos da Figura 14(E) e Figura 14(F).

Nos gráficos da Figura 15 o padrão predominante é manter a neutralidade das correlações dos parâmetros b da TRI nos três Modelos PL, onde é possível observar no grupo A representado pela Figura 15(A) e Figura 15(B). Destacando o Alg1 que apresentou inversão de correlação de positiva para negativa.

Os algoritmos Alg4, Alg5, Alg7 e Alg14 apresentaram alterações inversivas significativas nos três Modelos PL. Para o grupo B ocorreu a mesma tendência de neutralidade na maioria dos algoritmos para os parâmetros b da TRI nos três Modelos PL, é observável em Figura 15(C) e Figura 15(D).

O algoritmo Alg15 para o Modelo 3PL. Para o grupo C as correlações dos parâmetros b da TRI nos três Modelos PL tendem a neutralidade na maioria dos algoritmos, como pode ser observado nos gráficos da Figura 15(E) e Figura 15(F), destacando os algoritmos Alg23, Alg24, Alg28, Alg31 e Alg35 que modificaram suas correlações para positivas dentre os três Modelos PL.

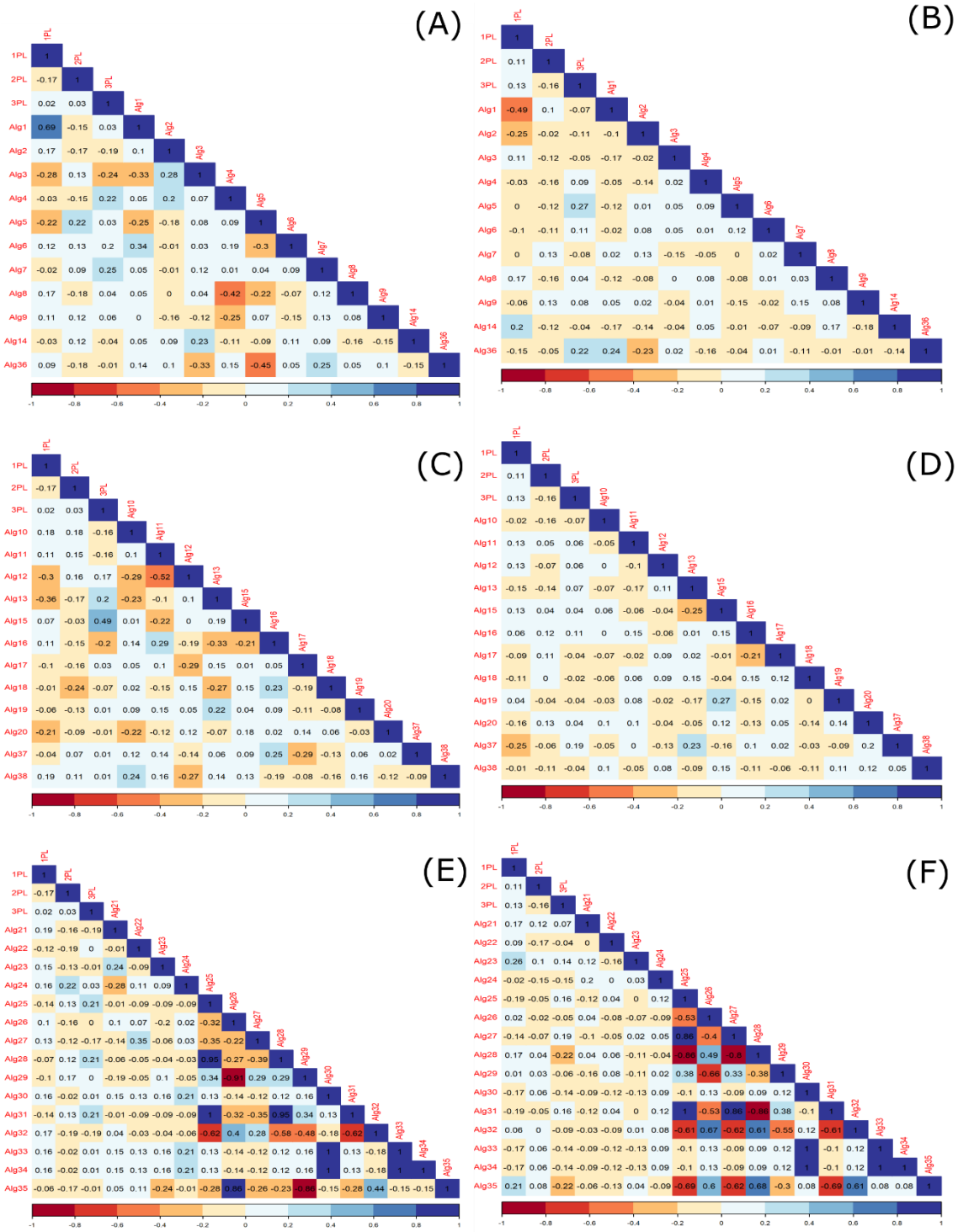

Figura 15 – Conjunto de correlação de Parâmetro de Dificuldade b dos grupos 1 e 2: pelo grupo A (A-B), pelo grupo B (C-D) e pelo grupo C (E-F) no *dataset* 7. Fonte: Autor.

Nos gráficos da Figura 16 o padrão predominante é manter a neutralidade das correlações dos parâmetros b da TRI nos três Modelos PL, onde é possível observar no grupo A representado pela Figura 16(A) e (B). Destacando mais uma vez o Alg1 que apresentou forte tendência de correlação de positiva para neutra.

Para o grupo B ocorreu a mesma tendência de manter a neutralidade na maioria dos algoritmos para os parâmetros b da TRI nos três Modelos PL, é observável em Figura 16(C) e (D). Este mesmo comportamento ocorreu para o grupo C com as correlações dos parâmetros b da TRI nos três Modelos PL, como pode ser observado nos gráficos da Figura 16(E) e (F).

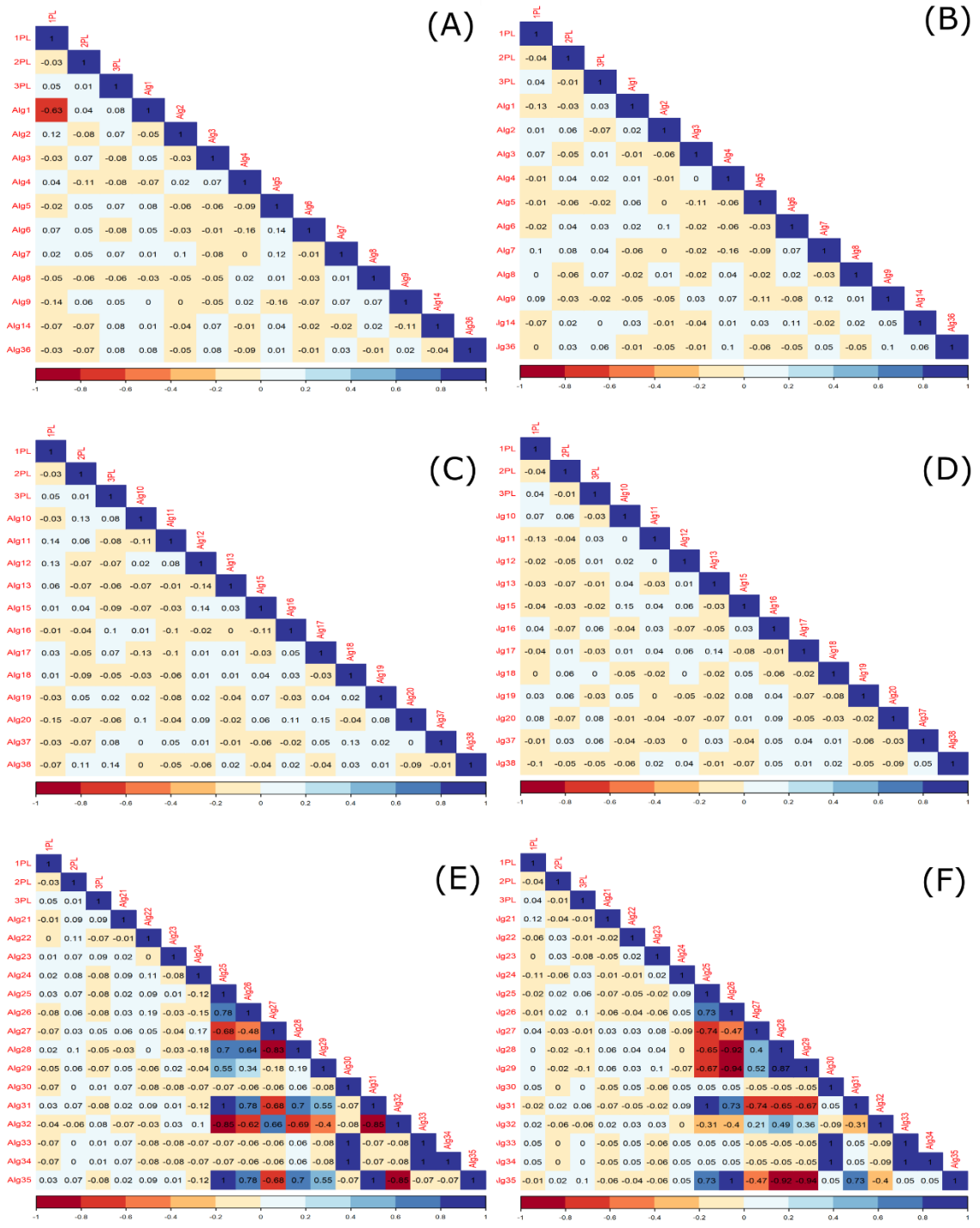

Figura 16 – Conjunto de correlação de Parâmetro de Dificuldade b dos grupos 1 e 2: pelo grupo A (A-B), pelo grupo B (C-D) e pelo grupo C (E-F) no *dataset* 8. Fonte: Autor.

Nos gráficos da Figura 17 o padrão predominante é manter a neutralidade das correlações dos parâmetros b da TRI nos três Modelos PL, onde é possível observar no grupo A representado pela Figura 17(A) e (B). Destacando mais uma vez o Alg1 que apresentou tendência para correlação de positiva.

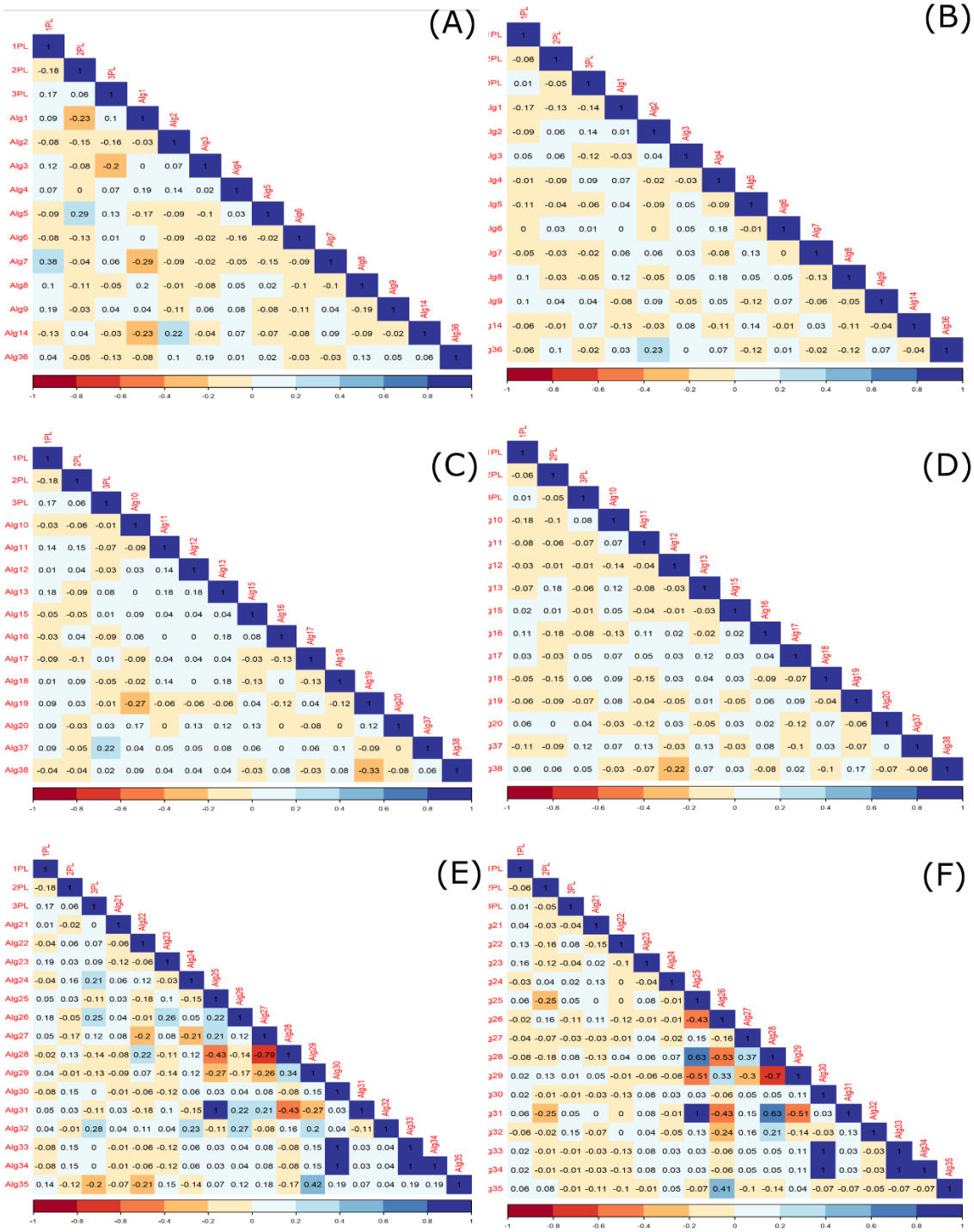

Figura 17 – Conjunto de correlação de Parâmetro de Dificuldade *b* dos grupos 1 e 2: pelo grupo A (A-B), pelo grupo B (C-D) e pelo grupo C (E-F) no *dataset* 9. Fonte: Autor.

Para o grupo B ocorreu a mesma tendência de manter a neutralidade na maioria dos algoritmos para os parâmetros b da TRI nos três Modelos PL, é observável em Figura 17(C) e (D). Este mesmo comportamento ocorreu para o grupo C com as correlações dos parâmetros b da TRI nos três Modelos PL, como pode ser observado nos gráficos da Figura 17(E) e (F).

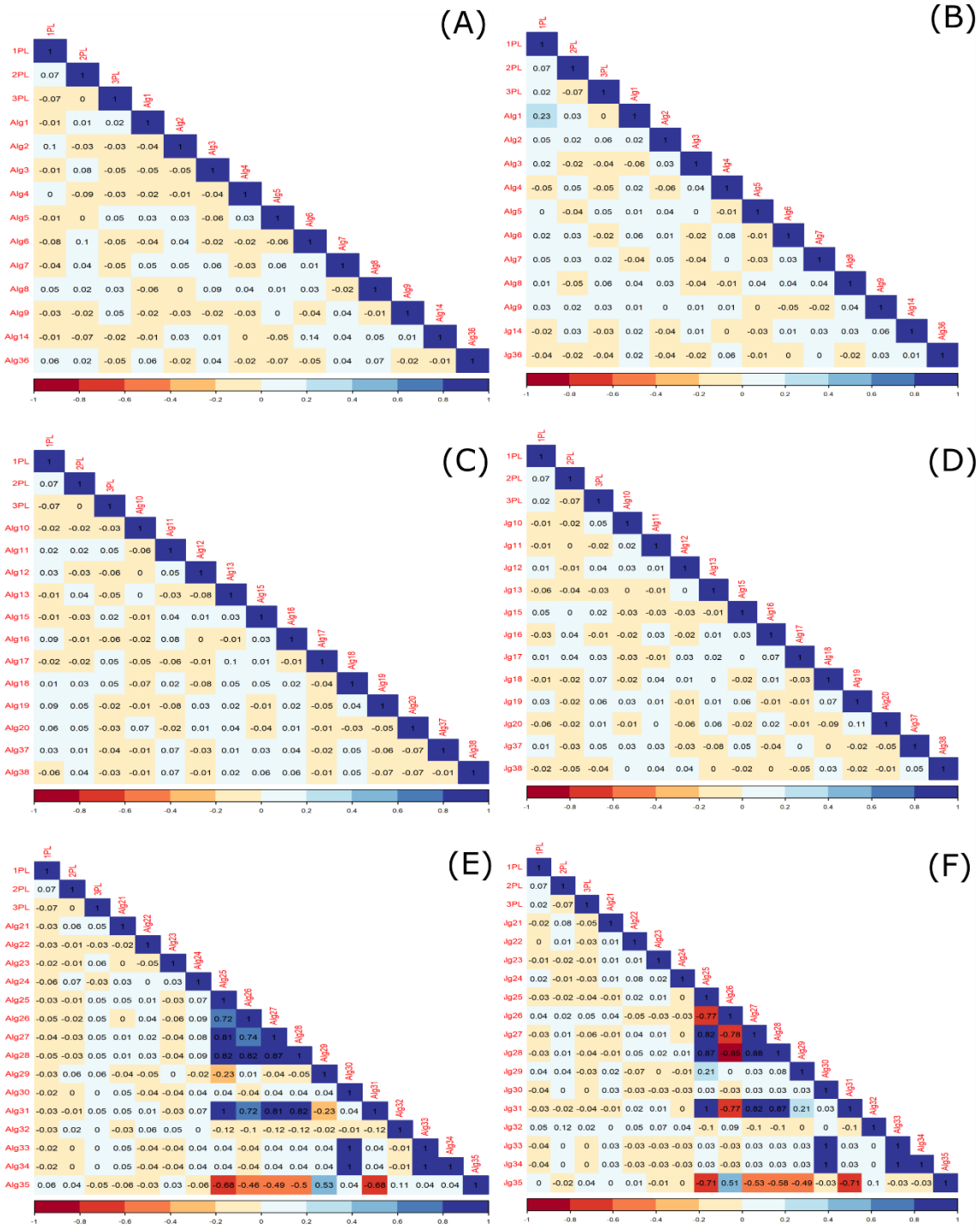

Figura 18 – Conjunto de correlação de Parâmetro de Dificuldade b dos grupos 1 e 2: pelo grupo A (A-B), pelo grupo B (C-D) e pelo grupo C (E-F) no *dataset* 10. Fonte: Autor.

Nos gráficos da Figura 18 o padrão predominante foi manter a neutralidade das correlações dos parâmetros b da TRI nos três Modelos PL, onde é possível observar no grupo A representado pela Figura 18(A) e (B), no grupo B em Figura 18(C) e (D) e no grupo como pode ser observado nos gráficos da Figura 18(E) e (F).

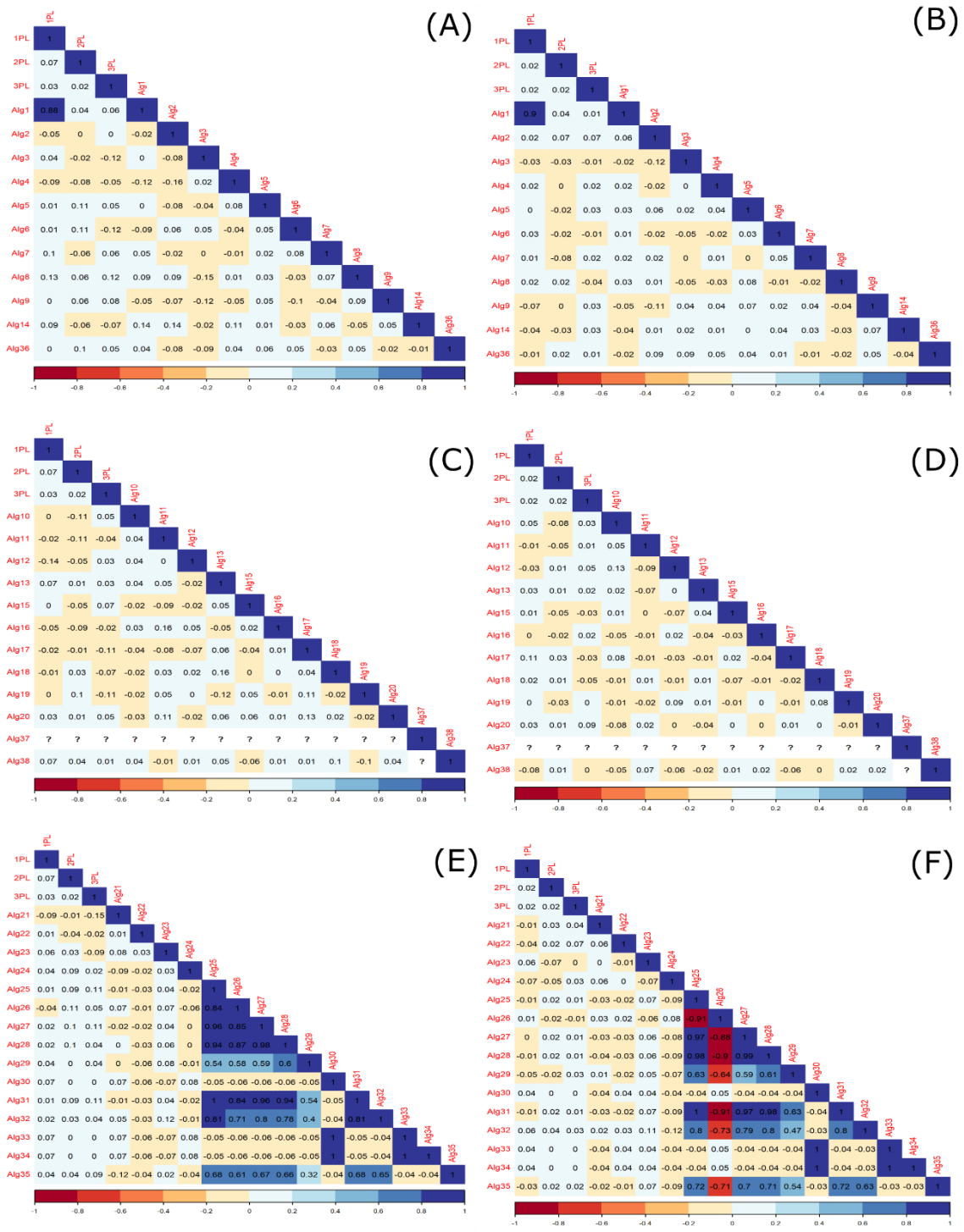

Figura 19 – Conjunto de correlação de Parâmetro de Dificuldade b dos grupos 1 e 2: pelo grupo A (A-B), pelo grupo B (C-D) e pelo grupo C (E-F) no *dataset* 11. Fonte: Autor.

Nos gráficos da Figura 19 o padrão predominante é manter a neutralidade das correlações dos parâmetros b da TRI nos três Modelos PL, onde é possível observar no grupo A representado pela Figura 19(A) e (B).

Destacando mais uma vez o Alg1 que apresentou tendência para manter a correlação de positiva em 1PL. Para o grupo B ocorreu a mesma tendência de manter a neutralidade na maioria dos algoritmos para os parâmetros b da TRI nos três Modelos PL, é observável na Figura 19(C) e (D). Destacando mais uma vez a não obtenção odos valores das correlações para os três modelos PL do algoritmo Alg37. Este mesmo comportamento registrado no grupo A ocorreu para o grupo C com as correlações dos parâmetros b da TRI nos três Modelos PL, como pode ser observado nos gráficos da Figura 19(E) e (F).

O *dataset* 7 (D7) registrou o maior número de amostras, sendo igual a 2.600, e alta dimensionalidade igual a 500, das quantidades entre o grupo 1 de 50% e grupo 2 de 100% foram utilizadas amostras equivalentes a 650 e 1.300 amostras, respectivamente aos grupos 1 e 2 para cada agrupamento. Considerando a aplicação da sub-amostragem de dados, onde o diferencial do dataset 7 (D7) para o dataset 10 (D10) foi a dimensionalidade, para o D7 houve registros de alterações de correlações entre positivas e negativas, diferentemente do D10, onde a neutralidade foi mantida para os três grupos A, B e C nos três modelos de PL.

Para o grupo A destaca-se o algoritmo do tipo *Hard* Fanny(Euclidean) (Alg1) (Brock *et al.*, 2008)como algoritmo que sensível aos ajustes de quantidades de amostras na formação dos agrupamentos, outros algoritmos deste mesmo grupo também registraram ajustes em suas correlações em relação aos parâmetros b da TRI nos três Modelos PL, os algoritmos *PAM(Euclidean)* (Alg4), *PAM(Manhattan)* (Alg5), *Clara(Manhattan)* (Alg7) e *Diana(Euclidean)* (Alg14) também registraram alterações consideráveis no D7 nos parâmetros b da TRI nos três Modelos PL.

Como caraterísticas importantes destes algoritmos destacam-se que PAM tem como particularidade o particionamento em torno de medóides semelhante ao K-means, fixando antecipadamente o *k* dos agrupamentos e um conjunto inicial de centros para iniciar o algoritmo. Clara é um algoritmo baseado em amostragem que implementa PAM em vários subconjuntos de dados, permitindo tempos de execução mais rápidos quando um número de observações é

relativamente grande. *Funny* é um algoritmo *hard* e realiza agrupamento difuso, onde cada observação pode ter participação parcial em cada agrupamento, possibilitando um vetor que dá a pertinência parcial a cada um dos *clusters*. O Diana é um algoritmo hierárquico divisivo que inicialmente começa com todas as observações em um único *cluster* e divide sucessivamente os clusters até que cada cluster contenha uma única observação (Brock *et al.*, 2008), estas características possibilitaram evoluções e melhorias como no caso do *Funny* que foi utilizado como base de algoritmos mais modernos como o algoritmo *CARD-F* (Carvalho; Melo; Lechevallier, 2015) utilizado em técnicas *Mult-view* em avaliações utilizando métricas de avaliação de agrupamento, obtendo resultados consideráveis em *datasets* de grande quantidade de amostras e baixa dimensionalidade como em *Abalone* (4177 exemplos) e *Image Segmentation* (2310 exemplos) obtendo resultados como: *FRHR* index (0,567), *CR Index* (0,132), *F-measure* (0,517) e *OERC* (0,486) e *FRHR index* (0,617), *CR Index* (0,538), *F-measure* (0,711) e *OERC* (0,288) respectivamente, estes resultados indicaram no trabalho de Carvalho; Melo; Lechevallier (2015) um desempenho mediano quando se trata de grande volume de dados para agrupar. O algoritmo de PAM e Clara destacam-se pelas funções de complexidade iguais a O(tkN) e $O(ks^2 + k(N - k))$ respectivamente (Anand; Kumar, 2023), indicando que mesmo utilizando funções de diferentes graus de complexidade são capazes de obter bons resultados em espaços de grande volume de dados. E que o Diana apresenta um bom desempenho calculado pelas métricas de validação de agrupamento como *Average Proportion of Non-overlap Measure e Average Distance Between Means Measure produce,* obtendo resultados melhores do que *K-means* e Fanny (Datta; Datta, 2003) como Alg14 das Tabela 5 e Tabela 6, onde foram calculados os tempos de cada algoritmo.

Para o grupo B baseados no *k-means* os comportamentos dos algoritmos não registraram fortes alterações e correlação dos atributos das atribuições dos agrupamentos dos elementos com os parâmetros b da TRI nos três Modelos PL, indicando um fator positivo para os algoritmos baseados em *k-means*, pois não foram fortemente impactados pelas alterações de acréscimo de novos elementos para agrupar. O algoritmo *k-means* é considerado o algoritmo de aprendizado de máquina mais simples e é amplamente utilizado para resolver problemas de *cluster* simples ou complexo (Zhuo *et al*., 2020), destacando-se em trabalhos de

classificação (Puri; Gupta, 2022), mas é pouco explorado na avaliação de agrupamento/classificações em conjunto com a TRI. Portanto, a estatística de correlação proporciona um método que evidencia quais algoritmos estão agrupando/classificando corretamente e sua relação com os Parâmetros de Dificuldade b de cada ML da TRI.

Para o grupo C baseados em *Fuzzy* os comportamentos de pertinência compartilhada dos elementos não impactaram em registros de dificuldades em agrupar/classificar os elementos nos respectivos grupos, diferentemente da partição *hard* do grupo A. Destacando os comportamentos dos elementos do grupo C nos datasets D1, D2, D4, D7 e D9, os quais apresentaram ajustes consideráveis de correlação e dimensões abaixo de 230, exceto para D7, e amostragens com quantidades médias de 50 elementos para o grupo 1 de 50% e grupo 2 de 100%, o que proporciona um fator a ser estudado mais aprofundado.

Mediante os resultados obtidos nos experimentos desta seção identificando um nível de significância de correlações entre as atribuições dos elementos nos respectivos agrupamentos e os valores do parâmetro b da TRI nos três Modelos PL, aplicamos a escala psicométrica baseada na TRI para avaliar os acertos e erros nos diferentes níveis de dificuldades nas atribuições dos elementos nos respectivos agrupamentos realizados pelos algoritmos de *clustering*.

4.1.4. Comparação das pertinências dos elementos com os valores do Parâmetro b da TRI

A utilização do método de determinação do Parâmetro de Dificuldade *b* da TRI possibilita identificar quais itens podem ser identificados com atribuições aos agrupamentos de forma complexa. A variável do Parâmetro de Dificuldade b está presente nos três ML da TRI (Wauters, Desmet, & Van Den Noortgate, 2012). Para determinar uma correlação entre o valor do Parâmetro de Dificuldade b que tem maior significância com os resultados das pertinências dos elementos nos respectivos agrupamentos foram implementados no algoritmo *OrderResults*, disponível em https://github.com/pauloflf/TRI. O algoritmo *OrderResults* foi implementado nas seguintes etapas do pseudocódigo:

Inicialmente, os valores do Parâmetro de Dificuldade b de cada ML foram calculados pela Equação 16 em cada um dos *datasets* da Tabela 4,

representados pelo vetor (nLP), onde *n* é cada um dos três parâmetros ML da TRI. Paralelamente, o vetor de determinação dos agrupamentos dos elementos de cada algoritmo (A) foi associado aos respectivos itens (I) de cada conjunto de dados (D) e ordenado na matriz *U*. Dessa forma, a associação entre a classificação de cada algoritmo e os valores do Parâmetro de Dificuldade b, em cada um dos três ML, cada registro do conjunto de dados possibilita o entendimento da atribuição do agrupamento com a correspondência com a classificação e dos níveis de dificuldade.

Após os cálculos para obtenção da matriz *U* em cada item dos *datasets,* foram comparados os valores obtidos para o Parâmetro de Dificuldade b em cada ML com cada item, individualmente. Para agrupar/classificar os valores do Parâmetro de Dificuldade b nos respectivos níveis de complexidade foram categorizadas as seguintes faixas, baseadas nos valores da Tabela 2 e descritas no método para os seis Níveis de Dificuldade (ND) representados por: XE, VE, E, H, VH e XH, descritas na metodologia desta Tese, aplicando o método de votação para calcular as quantidades e proporções de compatibilidade dos itens agrupados e suas correspondências com as classes originais de cada *dataset* (*i.e.* pela validação supervisionada). As questões extremamente difíceis muitas vezes estão correlacionadas com o Parâmetro de probabilidade de acerto por acaso (c) (Moraes *et al.*, 2022), pois a complexidade é muito alta e a tendência é "adivinhar" ou "chutar" uma resposta. Esse recurso de ajuste de agrupamento/classificação dos algoritmos está associado ao viés da interpretação do analista humano.

No trabalho de Leite-Filho; Melo (2022) estes resultados estão relacionados com algoritmos não supervisionados, e no trabalho de Martínez-Plumed *et al.* (2019) são utilizados majoritariamente algoritmos supervisionados. Nos dois trabalhos os Parâmetros de Dificuldade *b* da TRI são empregados como referência para a dificuldade da classificação dos itens certos ou errados. Porém, no trabalho de Leite-Filho; Melo (2022) foram tratadas as classes distintas, determinadas pelo Parâmetro da Dificuldade b nos diferentes *datasets* analisados, proporcionando desta forma uma associação cognitiva mensurada por uma escala psicométrica. Possibilitando a determinação de quais itens estão contidos em determinadas classes de dificuldades e relacionando estas classes com os acertos nos agrupamentos/classificações dos algoritmos de *clustering*,

proporcionando testar o método de Martínez-Plumed *et al.* (2019) em algoritmos não supervisionados. Considerando que os algoritmos não supervisionados são avaliados de forma cognitiva pelas sua capacidade de trabalhos com dados não utilizados para a formação do modelo computacional, portanto no trabalho de Leite-Filho; Melo (2022) foi avaliado de forma complementar as habilidades dos algoritmos em identificar registros em diferentes níveis de dificuldade de dados não amostrados previamente aos modelos.

Os experimentos desta seção utilizaram os dois grupos de subamostragem utilizada nos experimentos anteriores. O primeiro grupo representou amostras de cada classe com subamostragem de 50% da classe minoritária e reduzindo na amostra a quantidade de elementos da classe majoritária, balanceando as duas classes, mesmo com perda de informação. Para o segundo grupo, foi realizado o mesmo procedimento, porém com a equivalência de 100% dos elementos da classe minoritária. A escala utilizadas nos resultados destes experimentos seguem os Níveis do Parâmetro de Dificuldade b foram divididos da seguinte maneira: MFC (Muito fácil e classificado corretamente) [-4,-2], FC (Fácil e classificado corretamente) [-2,0], MDC (Muito difícil e classificado corretamente) [0,+2], DC (Difícil e classificado corretamente) [+2,+4], MFE (Muito fácil e classificado erroneamente) [-4,-2], FE (Fácil e classificado erroneamente) [-2,0], MDE (Muito Difícil e classificado erroneamente) [0,+2], DE (Difícil e classificado erroneamente) [+2,+4], previamente adaptado do trabalho de Andrade; Valle (2000) para o trabalho de (Leite-Filho; Melo; Cassia-Moura, 2021a), ajustando os valores fora do intervalo de [-4,+4] para os limites de mínimos e máximos destes intervalos e as atribuições de certo e errado para a pertinência dos elementos nos respectivos agrupamentos. Diferenciando a proposta da aplicação da escala psicométrica da TRI apresentada nesta Tese com a e o trabalho de Andrade; Valle (2000) e refinando o para o trabalho de (Leite-Filho; Melo; Cassia-Moura, 2021a), foram estendidas as classes de dificuldades da TRI, equivalendo MFC ou MFE com VE e valores abaixo de -4 com XE, FC ou FE com E, MDC ou MDE com H e DC ou DE com VH e valores acima de +4 com XH.

Os resultados dos grupos do Parâmetro de Dificuldade b estão representados na Figura 14, onde estão contidos os gráficos dos somatórios de acertos e erros de classificações em todos os *datasets* da seguinte maneira: o

gráfico da esquerda representa o grupo de 24 amostras (*undersample*) na Figura 14 (E) e no da direita o de SMOTE (*oversample*) na Figura 14 (D) nos três Modelos PLs.

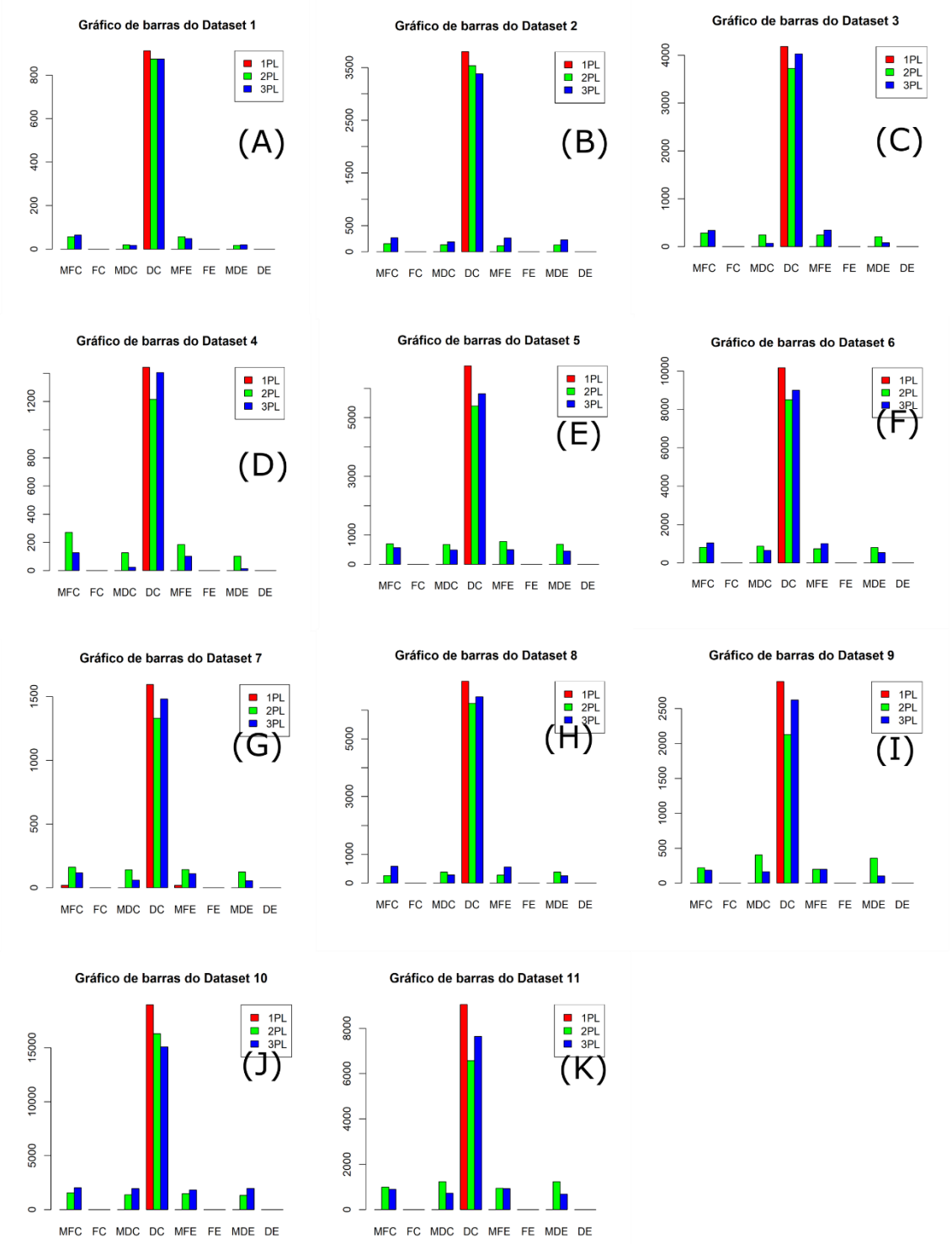

Figura 20 – Gráficos dos valores obtidos pela escala psicométrica do *dataset* 1(A) ao *dataset* 11(K) do grupo 1. Fonte: Autor.

O eixo X as categorias de acerto ou erro informadas no parágrafo anterior, e no eixo Y dos gráficos informa o somatório de itens classificados de todos os *datasets*.

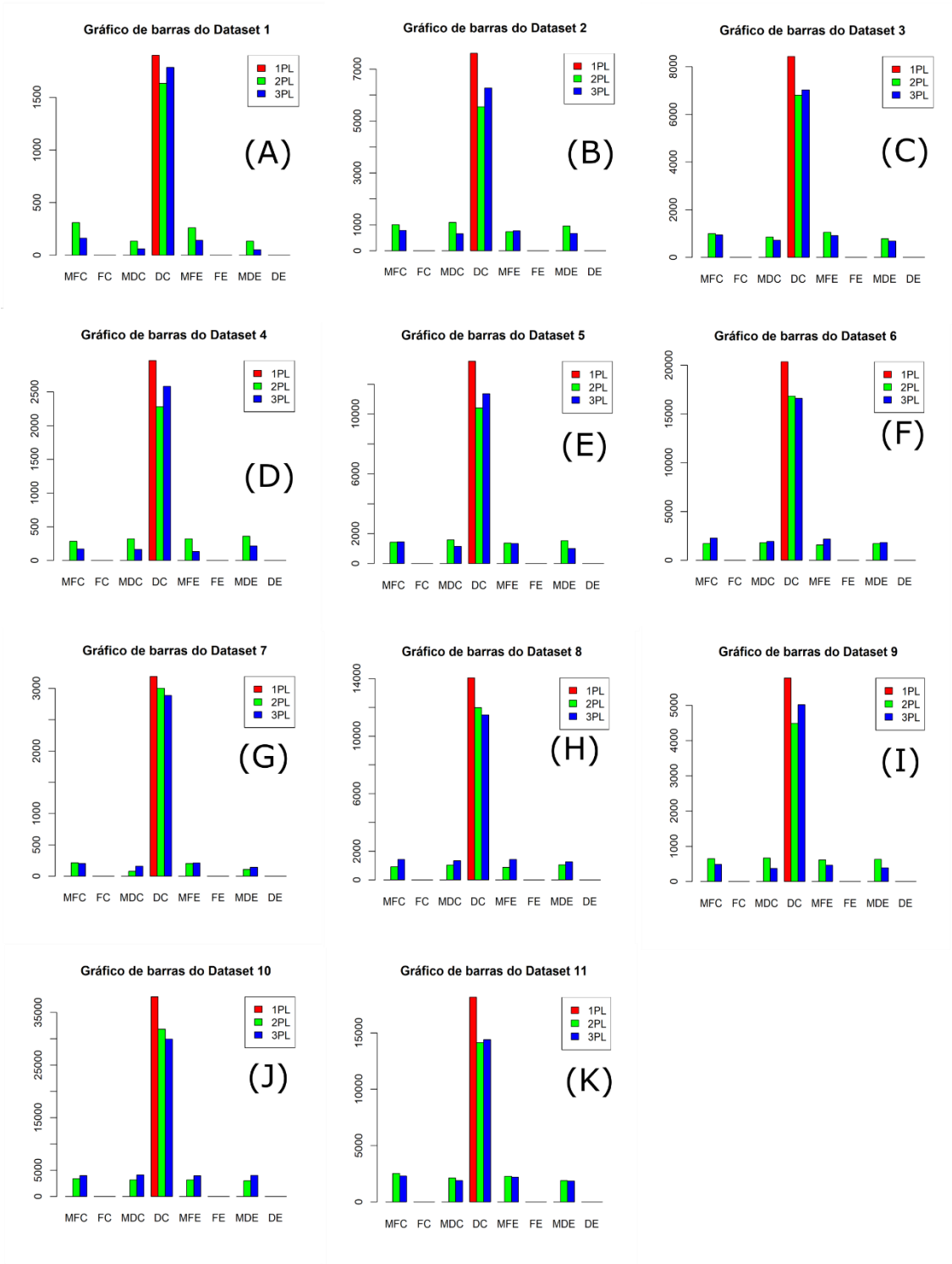

Figura 21– Gráficos dos valores obtidos pela escala psicométrica do *dataset* 1(A) ao *dataset* 11(K) do grupo 2. Fonte: Autor.

É possível observar que ao realizar a sobreamostragem pelo SMOTE o quantitativo de erros reduz e o nível de dificuldade também é reduzido em relação aos três Modelos PLs. Na Figura 20 estão registrados os resultados do grupo 1 com subamostragem de 50% da classe minoritária em cada *dataset*, considerando os intervalos citados na escala psicométrica definida no parágrafo anterior.

Na Figura 21 estão registrados os resultados do grupo 2 com subamostragem de 100% da classe minoritária em cada dataset, considerando os intervalos citados na escala psicométrica definida no parágrafo anterior

Nos trabalhos de Martínez-Plumed *et al.* (2019) e Leite-Filho; Melo (2022) as escalas de complexidade são relacionadas com o Parâmetro de Dificuldade b da TRI e demonstrado que podem ocorrer erros nas classificações mediante complexidade dos dados. Nos experimentos desta Tese os algoritmos não supervisionados foram avaliados com os *datasets* desbalanceados e balanceamento com subamostragem. No trabalho de Martínez-Plumed *et al.* (2019) foram avaliados algoritmos majoritariamente supervisionados e *datasets* balanceados, e parte dos *datasets* testados nos experimentos foram utilizados nos testes desta seção, onde os algoritmos baseados em *k-means* obtiveram novamente bons resultados.

Nas Figura 20 e Figura 21 é observável que há uma concentração de itens em atribuições de agrupamentos em relação às classificações previamente conhecidas na categoria Difícil e Correta (DC) em relação às demais categorias. Entretanto, ao adicionar mais elementos do grupo 1 para o grupo 2 é possível observar uma distribuição entre as demais categorias, como pode ser observado na Figura 20(A) em relação a Figura 21(A), este comportamento também pode ser observado nos demais *datasets* dos experimentos realizados nesta Tese.

Avaliar os comportamentos dos algoritmos em relação a variação das concentrações dos elementos em relação aos grupos de teste é uma prática conhecida por *minority oversampling (ROS)* e *random majority undersampling (RUS),* assim como técnicas de geração de dados artificiais por SMOTE. No trabalhos de Van Hulse; Khoshgoftaar; Napolitano (2007) foram realizados experimentos com quatro níveis de RUS com proporções iguais a 5%, 10%, 25% e 75% da classe majoritária, onde a métrica Precisão foi utilizada para avaliar estas e outras métricas de subamostragem de dados. Identificando RUS como

uma técnica promissora para identificar os bons rendimentos de algoritmos de *Machine Learning* nos experimentos realizados pelo autor.

O desbalanceamento de grupos é um fator que penaliza o desempenho de algoritmos de *clustering*, e como informado no parágrafo anterior, balancear por subamostragem do grupo majoritário é um método promissor e testado nos experimentos desta seção. Aplicar a subamostragem em soluções envolvendo algoritmos de *clustering* apresentam limitação de informações importantes com significantes perdidas durante este processo. As técnicas de subamostragem baseadas em *cluster* resolvam problemas de instâncias majoritárias e conflitos em regiões de sobreposição no conjunto de dados, reduzindo as dificuldades e distribuindo os resultados em situações consideradas anteriormente difíceis, torando-as fáceis de identificar as pertinências dos elementos nos respectivos agrupamentos (Chakraborty *et al*., 2021).

Nas Figura 20 e Figura 21 é notável que o Modelo 2PL apresenta comportamento sensível em relação as mudanças das concentrações de elementos dos grupos 1 e 2. Indicando crescimento nas categorias MFC, MDC, MFE e MDE. Reduzindo a proporção em DC e indicando o crescimento de elementos calculados pela TRI como MFC e MFE. Esta atribuição de dificuldade representada nos novos quantitativos a serem agrupados pelos algoritmos de *clustering* indica que novos elementos forma adicionados em regiões de conflito e difíceis de atribuir aos respectivos grupos, mas também houve um distribuição de elementos mais fáceis de atribuição, proporcionando ao comprar estas atribuições nas partições um modo avaliativo para os analistas humanos de atribuir um controle de qualidade aos itens devidamente agrupados, assim como a identificação de algoritmos ótimos em resolver os objetivos de agrupamento, proporcionando um método de avaliar a tomada de decisão dos analistas humanos, em relação ao desempenho dos algoritmos de *clustering*.

Categorizar as classes, determinadas pelo Parâmetro de Dificuldade b da TRI, em níveis de complexidade, possibilita uma identificação de que existe uma gradação associada a dificuldade dos itens a serem agrupados/classificados, proporcionando segurança na aplicação da escala psicométrica nos agrupamentos/classificações de algoritmos de *Machine Learning*. Para uma análise mais objetiva foram gerados gráficos de boxplot para a avaliação das medianas das distribuições representados na Figura 22 para o grupo 1 de 50%

das classes minoritárias e na Figura 23 do grupo 2 de 100% das classes minoritárias.

Na Figura 22 é possível observar a distribuição dos dados equivalente a DC, onde o comportamento desta distribuição apresenta o mesmo padrão em todos os *datasets* da Figura 22(A-K).

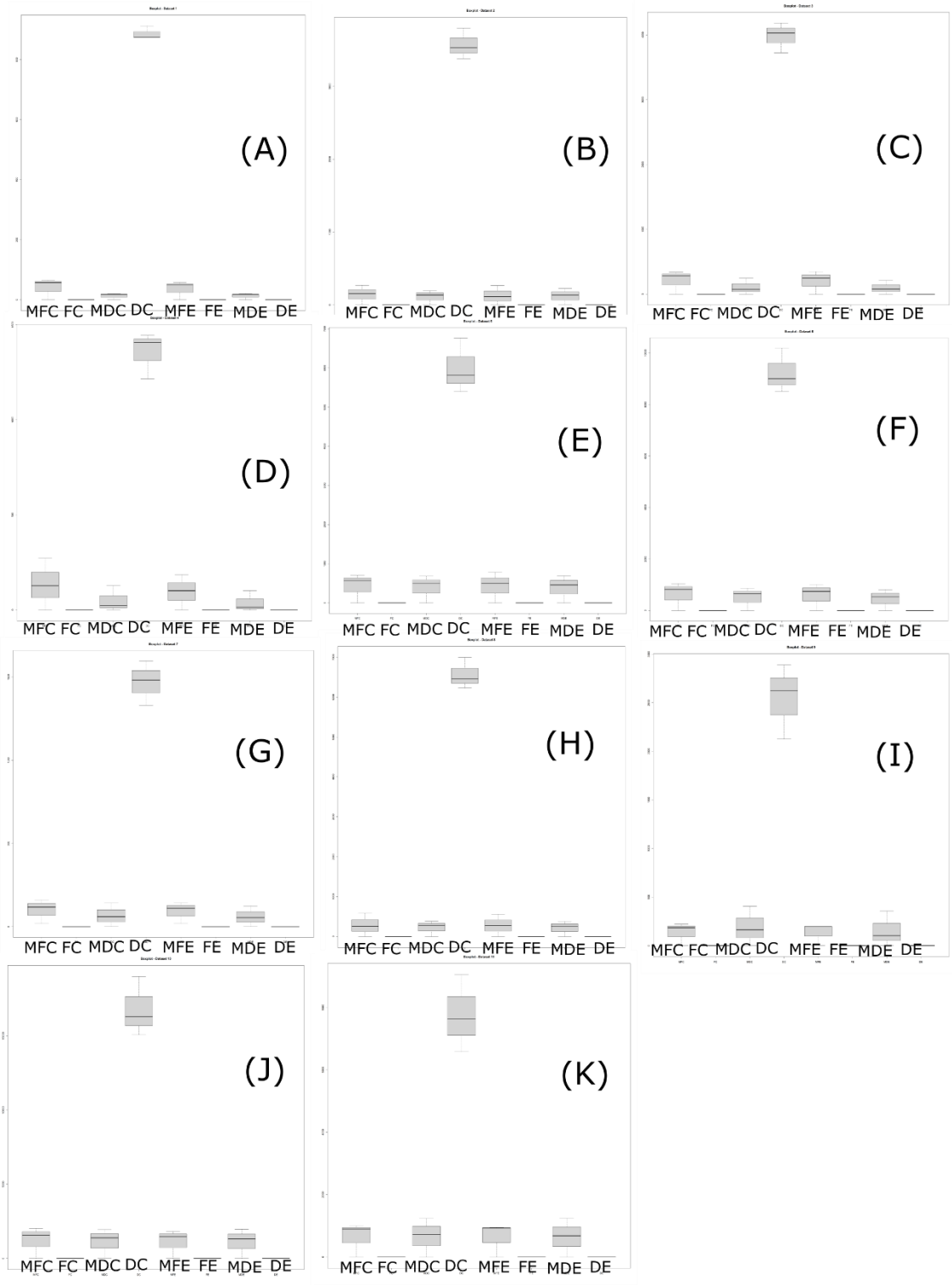

Figura 22 – Gráficos de *boxplot* das categorias de b no grupo 1. Fonte: Autor.

Na Figura 23 as distribuições de DC apresentam uma distribuição interquartil em relação ao grupo 2, demonstrando que mesmo com o acréscimo de novos dados não houve uma variação de mediana em todos os *datasets*.

.

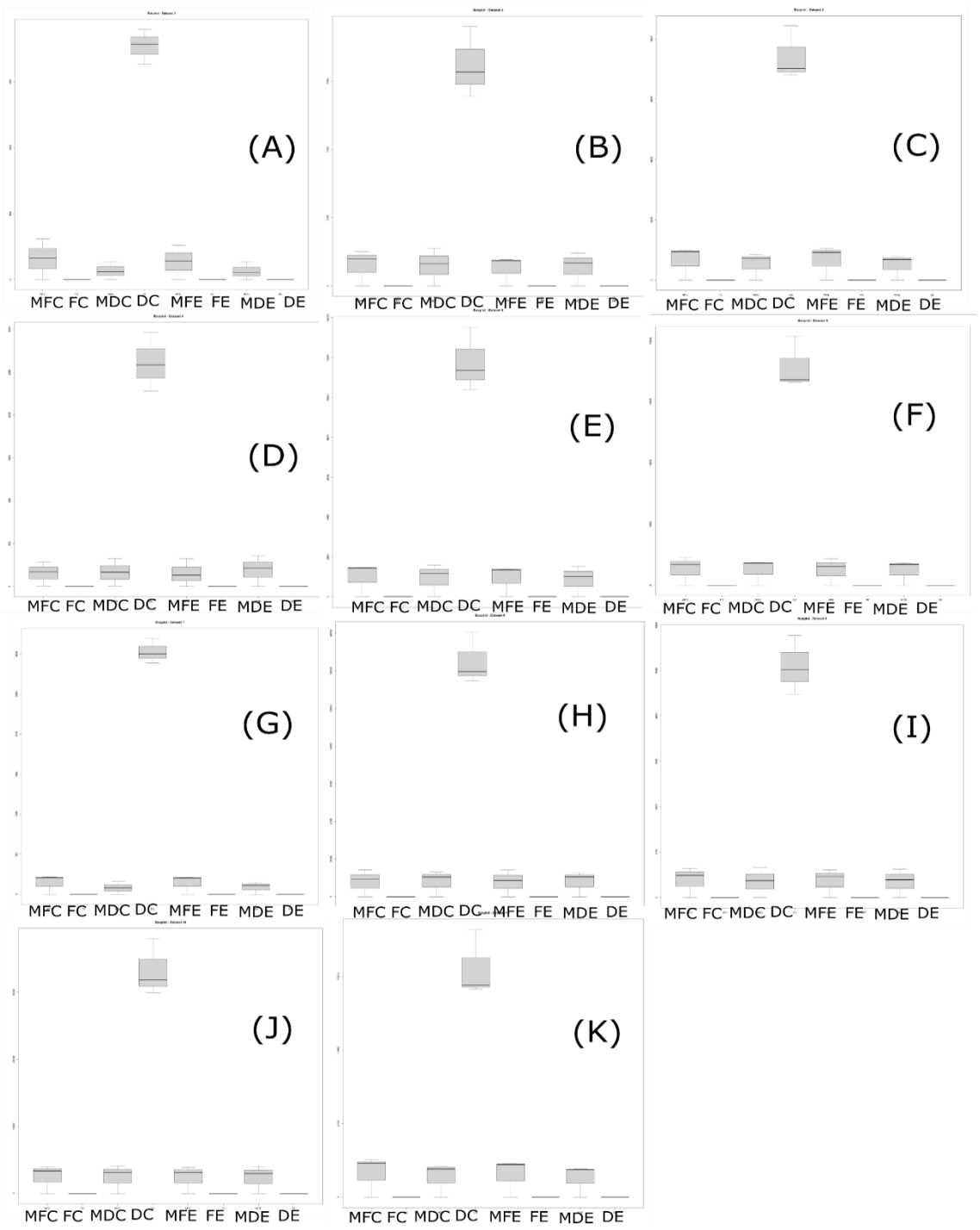

Figura 23 – Gráficos de *boxplot* das categorias de b no grupo 2. Fonte: Autor.

É possível observar também que as distribuições FC, FE e DE não obtiveram registros. Este resultado também pode ser constatado nos gráficos das Figura 20 e Figura 21, representando desta forma um intervalo em que os dados são registrados nos limites da escala psicométrica entre Muito Fácil e Muito Difícil de acertar ou errar as atribuições dos elementos nos agrupamentos específicos

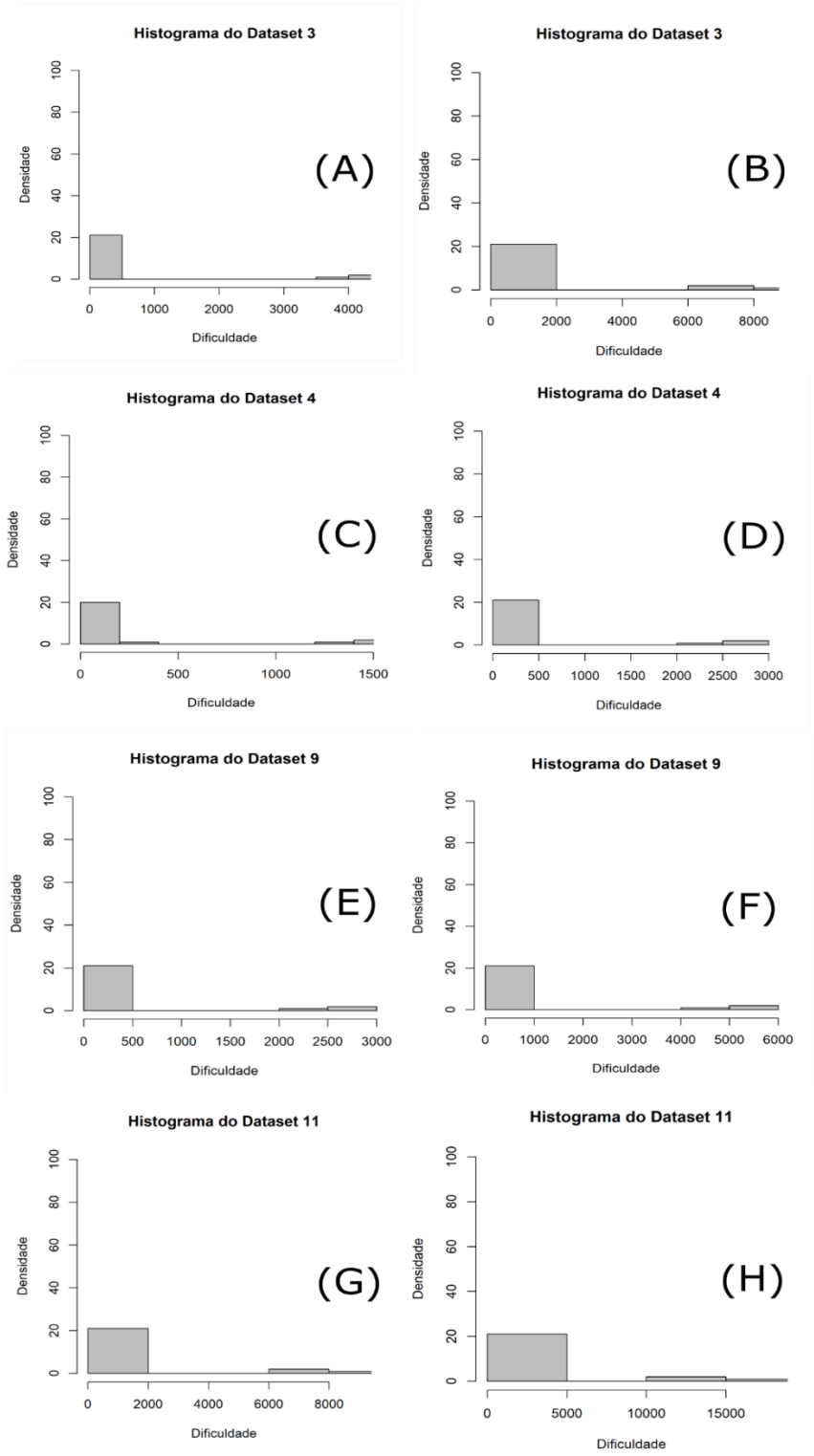

Figura 24 – Gráficos de frequência, *datasets* 3, 4, 9 e 11. Fonte: Autor.

Para compreender estas variações nas concentrações foram gerados gráficos de frequência, representados na Figura 24 onde foram comparados os *datasets* 3, 4, 9 e 11.

Nos gráficos da Figura 24(A) para o grupo 1 e da Figura 24(B) para o grupo 2+ foram comparadas as distribuições dos elementos no *dataset* 3, nesta comparação é possível observar que houve um crescimento na distribuição de determinada quantidade de elementos considerados difíceis, assim como os outros itens em suas respectivas categorias de níveis de dificuldade. Este comportamento se repete para o *dataset* 4 na Figura 24(C-D), para o *dataset* 9 na Figura 24(E-F) e no dataset 11 na Figura 24(G-H).

Não é possível determinar um Modelo PL de consenso entre as variações de dificuldade nas alterações de itens nas amostras. Porém, é notório que o balanceamento de classes pelo método de *oversample* favorece a redução da dificuldade para o agrupamento/classificação dos algoritmos de *clustering*, destacando o Modelo 2PL da Figura 20 e da Figura 21 como referência para categorizar os exemplos na faixa de itens para agrupamento, classificados corretamente e fáceis.

4.1.5. Comportamento dos algoritmos versus inserção de novos dados

Nos experimentos realizados nesta seção foram utilizados todos os algoritmos da Tabela 3 e todos os *datasets* D1 a D11 da Tabela 4, avaliados nos tópicos a seguir. Novos elementos foram adicionados aos grupos já existentes nos datasets originais da Tabela 4 utilizando a técnica de geração de dados artificias pelo método de SMOTE. Destes novos dados foram extraídas amostras estratificadas dos *datasets*, e nestas amostras há reposições de elementos pela medição de cada valor do Parâmetro de Dificuldade b nos modelos PLs, proporcionando novas amostras mais difíceis, avaliando separadamente as capacidades dos algoritmos de *clustering* em realizarem seus objetivos, pelas métricas e correspondências dos agrupamentos com as classificações previamente conhecidas.

Desta forma, os algoritmos foram avaliados pela TRI como a capacidade dos estudantes compreenderem novas informações inseridas em uma prova,

cada vez mais difícil de responder. Este método seguirá o fluxo da Figura 4, utilizando os algoritmos descritos nos pseudocódigos *OrderResults e Notification of Easier - exits and stratified - enters* (Not2ESE), e as variações dos ML no pseudocódigo de *Not2ESE-1PL, Not2ESE-2PL e Not2ESE-3PL,* onde estes algoritmos derivam do Not2ESE, variando apenas pelos PL. Nesta etapa foram avaliados separadamente os seguintes pontos:

- *datasets* balanceados e a avaliação das métricas aplicando as Equações de 1 a 3 como referência de aplicações avaliativas dos especialistas humanos e paralelamente foram calculadas as métricas de validação interna e externas das Equações de 8 a 14;
- avaliação comparativa na reamostragem de dados entre o Parâmetro de Dificuldade b e as métricas de dados balanceados.

Os experimentos foram realizados de forma a tornar independentes em relação aos algoritmos, *datasets* e métricas, e para cada resultado foram realizadas análises das dificuldades dos algoritmos nos *datasets* selecionados.

4.2. Datasets balanceados e a avaliação das métricas

Para avaliar os valores do Parâmetro de Dificuldade b durante o processo de inserção de novos dados e as habilidades dos algoritmos nas classificações, foram aplicados os pseudocódigos *de OrderResults e Notification of Easier - exits and stratified - enters* (Not2ESE), e as variações dos ML no pseudocódigo de *Not2ESE-1PL, Not2ESE-2PL e Not2ESE-3PL,* foram executados os seguintes procedimentos referentes ao fluxo na Figura 4:

1) Obtenção de amostras estratificadas de cada *dataset* com tamanho fixo igual a 137 registros por classe, totalizando 274 registros por amostra e de cada amostra foram realizadas 500 iterações consecutivas. A preferência por amostras pequenas deve ser observada em relação a conjuntos de dados muito grandes, devendo ser utilizado amostras com no máximo 500 registros para testes. A razão para isso é que ao trabalhar com um número muito grande de itens, os pacotes que estimam os valores da TRI podem ficar presos em mínimos locais ou não convergirem. Para conjuntos de dados muito grandes é recomendado usar menos de mil instâncias para gerar os parâmetros. Este fator

influenciou na redução de instâncias nesta Tese, selecionando um conjunto de dados com menos de 500 instâncias, conforme realizado no trabalho de Liu *et al.* (2014).

2) A cada iteração foi removido o menor valor do Parâmetro de Dificuldade *b* (*i.e.* itens mais fáceis) e substituídos aleatoriamente por novo item de mesma classe dos *datasets* originais, objetivando tornar as amostras relativamente mais difíceis de classificar a cada atualização.
3) A cada iteração foram calculadas as respectivas classificações e dos valores dos resultados de cada algoritmo foram aplicadas as métricas das Equações de 1 a 3, armazenando os resultados gerados.
4) Os dados obtidos foram armazenados em 1.496 matrizes em formato.csv, correspondentes aos valores das quatro métricas extraídas da execução de 38 algoritmos em 11 conjuntos de dados. Cada matriz contém os resultados de 500 iterações dos algoritmos considerando cada um dos três Modelos PL.
5) As métricas foram calculadas a partir dos resultados da amostra final obtida nas iterações de cada algoritmo. Para cada amostra foram avaliados os níveis do Parâmetro de Dificuldade (*b*) de cada item.

Estes resultados foram organizados pela ordenação dos algoritmos da Tabela 3, de acordo com os percentuais quantitativos dos itens dos níveis do Parâmetro de Dificuldade b da TRI. Ao executas os algoritmos *OrderResults e Notification of Easier - exits and stratified - enters* (Not2ESE) foram obtidos os seguintes resultados.

A partir dos dados gerais, observou-se que a Precisão tem um comportamento interessante, pois permanece na faixa intermediária de itens agrupados/classificados correta e incorretamente. Destacando-se que a Equação 2, referente a Precisão, na qual o denominador é o somatório de todas as questões classificadas pertencentes a classe alvo (TP), não pertencente (TN) e classificações incorretas (FP e FN). Em comparação probabilística, seria o espaço amostral. E o numerador é o somatório das classes alvo (TP) e não-alvo (TN). Portanto, o denominador não faz distinção de questões corretas ou não, e o numerador é o responsável por indicar a qualidade da avaliação para o teste realizado.

Outras observações a serem consideradas é que a curva ROC composta pelas métricas de Especificidade e Sensibilidade apresentam uma questão a ser discutida, pois a área abaixo da curva ROC é considerada como referência de qualidade, identificando como bons resultados a maximização da área, entre os eixos da Especificidade e a Sensibilidade. A partir das Equações 2 e 3 são utilizados respectivamente as variáveis que estão presentes nos numeradores (TP) quando nos denominadores (TN), diferenciando as métricas qualitativamente pelos índices (FN e FP) em cada equação. A métrica da Sensibilidades calculadas pela Equação 2, considera que a quantidade de elementos (FN) proporciona um decrescimento do resultado esperado, ou seja, a quantidade de elemento pertencentes a classe Alvo, que foi agrupado/classificado de forma incorretamente na classe não-alvo (FN) contribuindo para a penalização do método avaliado. *N*a área de saúde ocorre na avaliação de uma medicação para detecção de antígeno, anticorpo ou um reagente que precise de pouca amostra para um determinado diagnóstico.

Para os valores calculados da Especificidade pela Equação 3, a dificuldade para os algoritmos ocorre ao agrupar/classificar elementos pertencentes a classe não-alvo, incorretamente na classe alvo (FP), contribuindo para a penalização do método avaliado, ou seja, para avaliar a eficiência no uso de reagentes que não proporcionem reações cruzadas e causem um falso diagnóstico. Os fatores citados para problemas de atribuições de elementos por algoritmos de *clustering* em grupos errados é definido por ruido. No trabalho de Liu (2021) foi observado que funções *Fuzzy* podem ser penalizadas pelo ruído limítrofe entre duas classes. A ruído em regiões limítrofe entre *clusters* podem ser atribuídos altos valores de pertinência difusa pelas soluções de algoritmos baseados no método *Fuzzy*.

Nos experimentos da seção 4.1.3 os algoritmos foram agrupados por categorias, onde o grupo A estão contidos os algoritmos de *clustering* que utilizam diferentes técnicas de inicialização de centroides, no grupo B estão contidos os algoritmos que utilizam o *k-means* como base e no grupo C estão contidos os algoritmos que utilizam como característica o método de *Fuzzy.* Para avaliar os algoritmos com base nas observações citadas no parágrafo anterior sobre o impacto das atribuições incorretas dos elementos nos agrupamentos incorretos, foi avaliado este impacto nas métricas de Precisão, Sensibilidade e

Especificidade. Para avaliar este impacto foram avaliadas estatisticamente se os resultados das métricas dos algoritmos e os modelos PL nos *datasets,* onde foram avaliadas para identificar se há indícios de semelhança entre estes grupos de resultados.

Foi utilizada a estatística de Kruskal-Wallis com α igual a 0,05, testando cada grupo separadamente por *dataset*, onde foram obtidos *p-value* < α significativos. Após calcular a estatística de Kruskal-Wallis foram obtidos os seguintes resultados. Para a métrica Precisão, o *dataset* 1 destacou-se como referência principal para os algoritmos *Fuzzy C-means(Manhattan)* o ML 1PL e *Cluster Medoids(Hamming) os MLs 1PL, 2PL e 3PL.* Para a métrica Sensibilidade, no *dataset* 1 destacou-se como referência principal para o algoritmo *Fuzzy C-means(Manhattan) o* ML 1PL e no *dataset* 4 destacou-se como referência principal para o algoritmo *Cluster Medoids(Mahalanobis)* os MLs 2PL e 3PL e para a métrica Especificidade, no *dataset* 1 destacou-se como referência principal para o algoritmo *Cluster Medoids(Hamming)* os MLs 1PL, 2PL e 3PL. Estes resultados foram organizados pela ordenação dos algoritmos da Tabela 3, de acordo com os percentuais quantitativos dos itens dos níveis do Parâmetro de Dificuldade *b* da TRI, e organizados na Figura 7 pelas atribuições dos itens agrupados nas classificações originais corretas e na Figura 8 pelas atribuições dos itens agrupados nas classificações originais incorretas.

Na Tabela 8 estão apresentadas as porcentagens dos algoritmos do grupo C com melhores resultados, indicando que para o *Cluster.Medoids (Pearson_correlation)* registrou melhor desempenho em “Fácil” (E) com 99,15% na Precisão, Cluster_Medoids(Hamming) com melhor desempenho em “Difícil” (H) com 84,98% na Precisão, Cluster_Medoids(Canberra) com melhor desempenho em “Muito difícil” (VH) com 5,99% na Especificidade da Tabela 8.

Tabela 8 – Percentuais relativos de atribuições corretas dos melhores algoritmos por níveis do Parâmetro de Dificuldade b e a métrica avaliativa correspondente.

Atribuição CORRETA dos elementos agrupados com as classes originais								
Faixa	XE	VE	E	H	VH	XH	Métrica	Algoritmo
Melhor E	0	0	99,15	0,85	0	0	Precisão	Cluster Medoides (Pearson Correlation)
Melhor H	0	0	15,2	84,98	0	0	Precisão	Cluster Medoids (Hamming)
Melhor VH	4,32	3,63	60,80	21,08	5,99	4,19	Especificidade	Cluster Medoides (Canberra)

Fonte: Autor.

É possível observar que os algoritmos Cluster_Medoids(Canberra) obtiveram bons resultados na classificação correta dos itens VH e XH mesmo em métricas diferentes. Indicando estes algoritmos como ideais para resolver classificações consideradas mais complexas.

Na Tabela 9 são apresentados os algoritmos do grupo C com “piores” desempenho nas atribuições para as classificações por dificuldades para os referidos algoritmos, foram registrados os seguintes resultados, o algoritmo *Fuzzy C-means(Euclidean)* obteve em VE o percentual de 4,77% na Especificidade, o Cluster_Medoids(Pearson_correlation) com 99,48% na Precisão, Cluster_Medoids(Hamming) com 79,77% na Precisão.

Tabela 9 – Percentuais relativos de atribuições erradas dos piores algoritmos por níveis do Parâmetro de Dificuldade b e a métrica avaliativa correspondente.

Atribuição INCORRETA dos elementos agrupados com as classes originais								
Faixa	XE	VE	E	H	VH	XH	Métrica	Algoritmo
Pior VE	3,25	4,77	69,28	16,42	3,37	2,92	Especificidade	Fuzzy C-means (Euclidean)
Pior E	0	0	99,49	0,51	0	0	Precisão	Cluster Medoides (Pearson Correlation)
Pior H	0	0	20,23	79,77	0	0	Precisão	Cluster Medoids (Hamming)

Fonte: Autor.

Curiosamente, o algoritmo Cluster_Medoids(Pearson_correlation) e o Cluster.Medoids(Hamming) apresentaram um comportamento peculiar, pois representam a situação ambígua no *ranking,* como bons e piores algoritmos de *clustering* nas Dificuldades “Fácil” (E) e “Difícil”(H). Na Figura 7, o percentual de Precisão está presente em questões intermediárias em E e H, porém não pontuando em agrupamento/classificações extremas definidas por XE, VE, VH e XH das Tabelas 8 e 9. Na Figura 8, a Especificidade demonstra um indicador métrico tanto para a avaliação do VE quanto para o XH. A precisão tem um comportamento interessante, pois permanece na faixa intermediária de itens agrupados/classificados correta e incorretamente.

Nos grupos A e B, foram obtidos os seguintes resultados. para os algoritmos do grupo B foram registrados os seguintes resultados para o algoritmo k.means(MacQueen) com melhor desempenho geral, e o k.means(Hartigan-Wong) de menor resultado com 5,07% na Sensibilidade. Para o grupo A o algoritmo Fanny(SqEuclidean) obtive bons resultados na atribuição/classificação correta dos itens VH e XH. E destaca-se a interseção entre os grupos A e B,

onde os algoritmos PAM(Euclidean) e k.means(Hartigan-Wong) como os menos adequados para resolver classificações com nível de Dificuldade mais complexas (VH e XH).

Nos experimentos desta seção, houve a extração de amostras estratificadas, das quais foram calculados os valores dos Parâmetros de Dificuldade b em cada ML. Durante o processo de 500 iterações com reposição de novos dados, os valores das métricas e agrupamento/classificações de cada algoritmo foram calculados, com a finalidade de avaliar as habilidades dos algoritmos de *clustering*. Estas amostragens foram atualizadas pela substituição do menor valor do Parâmetro de Dificuldade b das amostras. Ou seja, itens mais fáceis são substituídos com reposição estratificada por novos itens, tornando as amostras relativamente mais difíceis de atribuir a classe especificada a cada atualização de amostragem balanceada, onde a cada iteração há reposição estratificada de elementos pelo método Tabu, com checagem gradativa nas amostras a cada iteração.

Os resultados dos experimentos foram armazenados em 1.496 matrizes em formato .csv, correspondendo aos valores de cada uma das quatro métricas extraídas da execução de 38 algoritmos em cada um dos 11 *datasets*. Cada matriz contém os resultados de 500 execuções dos algoritmos, considerando cada um dos três modelos PL, ilustrados graficamente na Figura 25, Figura 26 e Figura 27. No eixo das abscissas, o número de execuções dos algoritmos, nos gráficos das figuras estão representadas as amostras de traçados de comportamentos de algoritmos distintos. O eixo das ordenadas representa os valores extraídos das métricas, considerando cada Modelo PL, no qual o vermelho corresponde ao 1PL, o verde ao 2PL e o azul ao 3PL. Esta representação demonstra como os algoritmos apresentam capacidade em agrupar novos elementos corretamente nos agrupamentos específicos em relação a cada uma das métricas calculadas, onde são visualizados os traçados de evolução de melhores resultados mediante nova amostragem de dados.

Os traçados das iterações dos algoritmos estão representados nos gráficos das figuras para os algoritmos do grupo C, obtidos da estatística de Kruskal-Wallis na Figura 26, indicando o comportamento suave do traçado e demonstrando que as métricas não apresentaram alterações consideráveis nas 500 iterações. Para a Figura 27 estão representados os gráficos dos grupos A,

B e C com os melhores resultados e para os gráficos dos algoritmos PAM(Euclidean) e k.means(Hartigan-Wong) foi gerada a Figura 25 onde estão representados os comportamentos bastante semelhantes dos dois algoritmos no *dataset* e das três métricas.

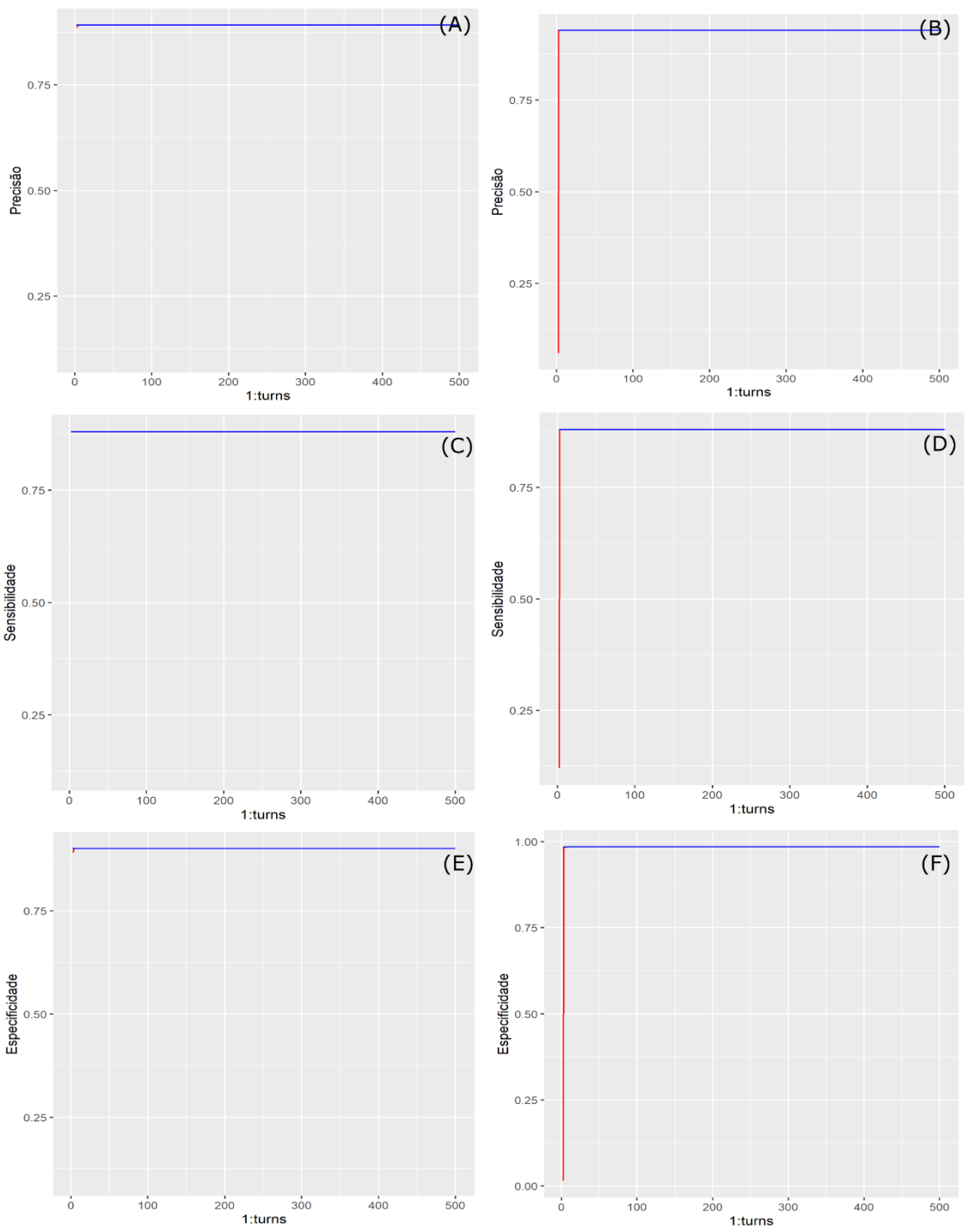

Figura 25 – Gráficos dos comportamentos no *dataset* 2 de PAM(Euclidean) (A) e k.means(Hartigan-Wong) (B) na Precisão, de PAM(Euclidean) (C) e k.means(Hartigan-Wong) (D) na Sensibilidade e de PAM(Euclidean) (E) e k.means(Hartigan-Wong) (F) na Especificidade.

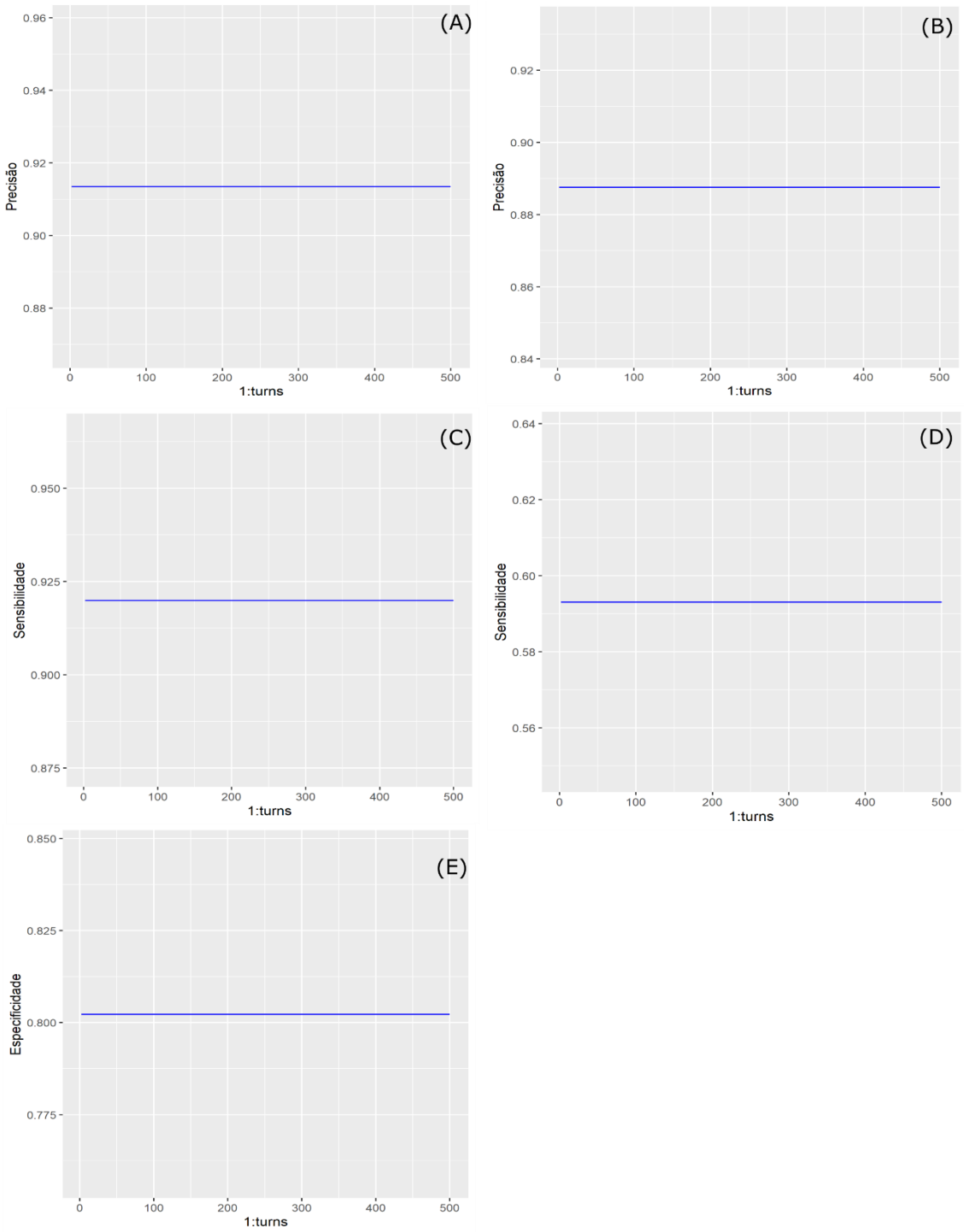

Figura 26 – Comportamento dos traçados dos algoritmos do grupo C na Precisão no dataset 1 para *Fuzzy C-means(Manhattan)* (A), *Cluster Medoids(Hamming) (B), n*a Sensibilidade, no *dataset* 1 para *Fuzzy C-means(Manhattan) (C),* no *Dataset* 4, o *Cluster Medoids(Mahalanobis)* (D), Especificidade, no *dataset* 1 (E) *Cluster Medoids(Hamming)*.

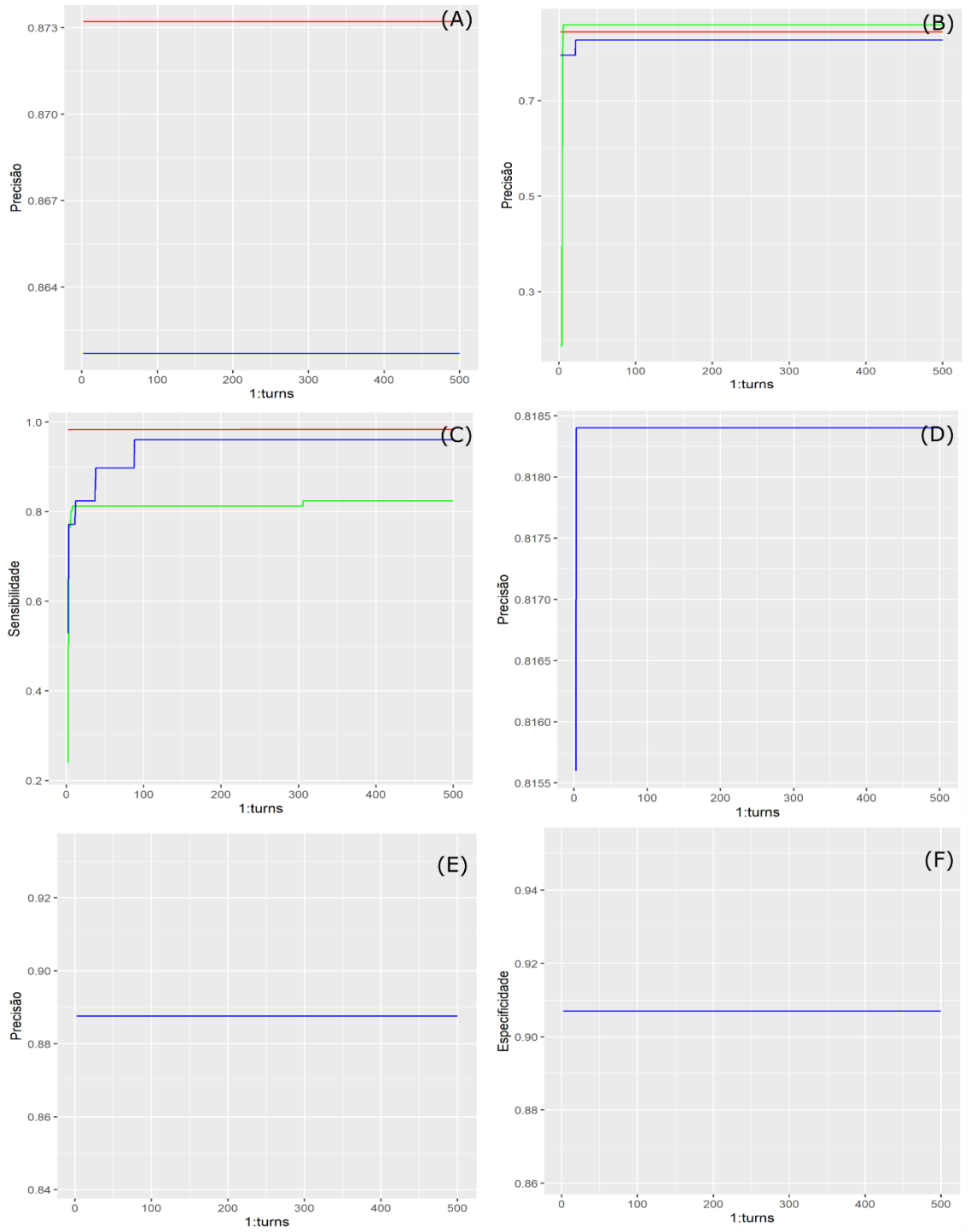

Figura 27 – Gráficos dos grupos A, B e C nas métricas de Precisão Fanny(SqEuclidean)(A), Precisão do k.means(MacQueen)(B), da Sensibilidade do k.means(Hartigan-Wong)(C), da precisão de Cluster.Medoids (Pearson_correlation)(D), Cluster_Medoids(Hamming)(E) e da Sensibilidade do Cluster_Medoids(Canberra) (F).

Nos gráficos das Figura 25, Figura 26 e Figura 27 é possível observar o comportamento diferenciado entre os traçados de cada algoritmo nas respectivas métricas, assim como o os Modelos PL podem ser relacionados a

avalição dos algoritmos pelas amostras gradativamente mais complexas, para avaliar estes comportamentos é necessário um aprofundamento nas variáveis das Equações de 1 a 3.

Ao observarmos as Equações de rendimento de 1 a 3 e as de validação de índices externos, constata-se que a diferença entre as equações de Precisão e Rand é a variável do numerador VN para Precisão e FN para Rand. A única Equação que não possui VP no numerador é a Especificidade, e em todas as Equações há as variáveis FP, FN ou as duas. Isto indica que a atribuição errada de uma considerável fração dos elementos em agrupamentos distintos impacta nas métricas citadas de forma importante para a decisão do especialista humano. Muitas vezes o questionamento de uma determinada pertinência de um elemento pode ser questionada pelos especialistas da área, principalmente quando o tempo é um fator importante na ponderação dos especialista (Klein, 2005).

Comparando os comportamentos das métricas pelas avaliações das Equações, destaca-se que a Equação 1, referente a Precisão, na qual o denominador é o somatório de todas as questões classificadas pertencentes a classe alvo (TP), não pertencente (TN) e classificações incorretas (FP e FN). Em comparação probabilística, seria o espaço amostral. E o numerador é o somatório das classes alvo (TP) e não-alvo (TN). Portanto o denominador não faz distinção de questões corretas ou não, e o numerador é o responsável por indicar a qualidade da avaliação para o teste realizado. Nas Tabelas 8 e 9 foi constatado que a Precisão para os melhores ou piores algoritmos não são avaliados em condições em que os itens são muito fáceis ou difíceis de acertar ou errar, proporcionando uma avaliação meramente quantitativa devido os valores percentuais (99,15%, 84,98%, 5,99% e 4,19%) apresentados nas Tabelas 8.

Para a Sensibilidade não houve registro a ser considerado na Figura 7, porém na Figura 8 a Sensibilidade registra maior penalização para itens XH, possibilitando cogitar que houve alguma falha na atribuição do agrupamento para a classificação de elementos FN. O trabalho de Chen; Ahn (2020) realizou uma análise de métricas com TRI a habilidade da TRI. Chen; Ahn (2020) realizaram uma análise do comportamento de 12 algoritmos em 33 conjuntos de dados com muitas dimensões. As métricas analisadas por Chen; Ahn (2020) foram Precisão,

pontuação F1, pontuação *Brier*, perda *Llog* e *AUC*. Comparando o trabalho de Chen; Ahn (2020) com esta Tese, a Precisão também tem um bom desempenho nos classificadores quando relacionada ao Parâmetro de Dificuldade *b*. Outro fator a ser considerado é que os algoritmos testados foram diferentes entre os dois trabalhos, com resultados muito semelhantes em relação a Precisão como a métrica em destaque. No trabalho de Beleites; Salzer; Sergo (2013) foram avaliadas as métricas de Sensibilidade, Especificidade, Valores preditivos e Taxas de acerto ou erro, onde foram observadas ambiguidades no desempenho medido na Precisão utilizando principalmente algoritmos de SVM e *Fuzzy c-means* com o objetivo final de atribuir classes, onde foram avaliados os critérios de desempenho do melhor caso, do esperado e do pior caso para as atribuir classes, mediante as métricas citadas.

Um fator a ser destacado nos experimentos desta seção é que a Precisão apresentou um comportamento distinto como pode ser observado nos intervalos de E e H nas Tabelas 8 e 9, concentrando com os maiores percentuais de classificação dos melhores ou piores algoritmos de agrupamento/classificadores. Este tipo de comportamento pode apresentar uma indeterminação de confiança para esta métrica, dificultando decisões mais complexas e críticas como na saúde, que utilizam a Precisão como um dos referenciais para o padrão-ouro (Beleites; Salzer; Sergo, 2013). Para avaliar cognitivamente um indivíduo, as alterações de memória devem ser consideradas como referência (Dourado, 2016; Klein, 2005). Ou seja, acertar ou errar a mesma informação é uma ação inconstante e de grande impacto, e de difícil avaliação para os profissionais da saúde. Portanto, a Dificuldade *b* da TRI avaliada nos experimentos desta Tese, proporciona um diferencial que pode ser utilizado em saúde e que viabiliza aplicações que podem ser utilizadas em diferentes métricas.

Recentemente foi realizado um experimento em uma métrica de saúde, que tem por objetivo avaliar distintos níveis de cognição nos índices de fragilidade de pessoas idosas (Pua *et al*., 2024). Esta forma de avaliar a métrica apresenta uma precisão variada devido aos diferentes quadros de debilidades mentais dos pacientes. Para esta avaliação foi utilizado o método da TRI, proporcionando uma melhor interpretação na contextualização dos pontos de corte da métrica. E fornecendo um significado qualitativo às medidas quantitativas deste método, conduzindo a limiares consensuais e a conclusões mais harmônicas entre os

profissionais desta área. Recentemente foi também empregada a TRI como solução para um problema de precisão de escala cognitiva (Dubbelman *et al.*, 2023). Portanto, é notável que a TRI é uma solução utilizada como forma de decisão para diferentes níveis de classificações, assim como foi utilizada nos experimentos desta Tese. Diante do exposto, é importante a determinação de regras que avaliem os valores obtidos do Parâmetro de Dificuldade b relacionado às métricas, tornando os resultados mais significativos.

Alguns trabalhos têm realizado críticas nas metodologias utilizadas nas avaliações de especialistas, dentre os quais podemos destacar que os trabalhos de Ferri; Hernández-Orallo; Modroiu (2009); Hossin; Sulaiman (2015); Liu *et al.* (2014), os quais indicam uma variabilidade de métricas, aplicações e particularidades em seus resultados. O critério de seleção da métrica tem sido dependente da escolha do analista, mas não da dificuldade nos tipos de dados. Porém, há o uso de equipamentos e sistemas de apoio a decisão que também devem ser considerados. No trabalho de Feuerstahler (2022) foi realizada uma análise sobre os tamanhos mínimos de amostras necessários para estimar com precisão os parâmetros do item, visando uma avaliação no uso da métrica. Esta abordagem é muito importante, pois com a redução do tamanho das amostras, o tempo de processamento computacional é reduzido para obter respostas mais rápidas; contudo, nem sempre ocorre pelo método exaustivo de busca. A amostra deve representar estatisticamente a população, e os itens da amostra devem indicar também sua complexidade dos dados para uma melhor avaliação dos algoritmos não supervisionados (*i.e.* c*lusters*).

Com a redução dos dados pelo tamanho das amostras e a reposição de novos dados, as métricas apresentaram resultados promissores. Alguns algoritmos não apresentaram em suas habilidades a capacidade de classificar corretamente os novos elementos, e com isto melhorar os resultados das métricas, como pode-se observar no trabalho de Leite-Filho; Melo (2023). Os algoritmos apresentam diferentes comportamentos na progressão do traçado de forma uniforme e sem evoluções consideráveis em relação a complexidades e aumento da dificuldade nos grupos de formação de classes. Este tipo de comportamento demonstra o desempenho dos algoritmos em processar novos dados, onde o traçado do gráfico relaciona ao tempo e a forma de resolver

agrupamentos cada vez mais complexos nas diferentes situações vivenciadas por analistas humanos, no decorrer da atuação profissional.

As métricas ao estarem relacionadas com o Parâmetro da Dificuldade b apresentam comportamentos peculiares, demostrando que dependendo de como foi determinada a forma avaliativa nas classificações dos algoritmos, pode-se observar que as métricas não seguem um comportamento idêntico. Após a obtenção destes resultados foi proposto um método de avaliar inicialmente os dados antes da aplicação de cada métrica, identificando desta forma quais métricas são mais sensíveis aos níveis de dificuldade dos dados. Observando o comportamento diferenciado em cada um do três Modelos PL, infere-se que para o modelo 3PL apresenta maior generalização do que o modelo 2PL, onde o Modelo 2PL é frequentemente preferido na prática. Tais preferências podem ser atribuídas ao fato de que o parâmetro de determinação de o Parâmetro Logístico b nos modelos 3PL é geralmente difícil de estimar devido ao uso dos três Parâmetros Logisticos (i.e. *a*, *b* e *c*) como apresentando nos trabalhos de McLeod, Swygert, & Thissen (2001) e Stone & Yeh (2006).

Para uma decisão mais objetiva, pode ser aplicado o método proposto no trabalho de Zhang & Stone (2004), onde recorrem a um procedimento de duas etapas, no qual o Parâmetro Logístico b é primeiro estimado com um modelo 3PL unidimensional e posteriormente comparado com a estimativa dos parâmetros restantes do modelo. Esse procedimento de estimativa em duas etapas, no entanto, tem demonstrou ser inferior à estimativa simultânea de todos os parâmetros no modelo 3PL através da estimativa de *Markov Chain Monte Carlo* (MCMC). Embora ao empregar o algoritmo MCMC todos os parâmetros do modelo possem a ser estimados simultaneamente, ocorrendo muitas vezes problemas de convergência do modelo com o modelo 3PL estimado com o algoritmo MCMC e recorrem ao modelo 2PL (Eckes, 2014). No trabalho de Wainer, Bradlow e Wang (2007) a dificuldade de estimativa do modelo 3PL ocorre devido à fraca identificabilidade, devido a presença de múltiplos trigêmeos diferentes dos quais os ICCs estão próximos. Consequentemente, eles recomendam que o modelo 3PL seja usado com cautela (LUO; WOLF, 2019).

5. CONCLUSÕES

Nesta Tese foi apresentada uma associação quantitativa e qualitativa dos fatores que impactam na tarefa de agrupamento por algoritmos, bem como uma estimativa da dificuldade dessa tarefa através do parâmetro de dificuldade da TRI, validando com métricas de performance. A abordagem utilizada partiu da já conhecida aplicação da TRI como método para determinar as dificuldades de classificação dos algoritmos supervisionados, e foi ajustada para avaliar a qualidade na pertinência de elementos nos agrupamentos dos algoritmos de clustering, utilizando métodos semi-supervisionados.

A qualidade da pertinência está associada às dificuldades desses algoritmos na atribuição correta dos elementos nos agrupamentos, e foi utilizada uma escala psicométrica baseada na TRI. Esse procedimento foi validado através das comparações das métricas de rendimentos estatísticos e de validações internas e externas dos agrupamentos, avaliadas nos dados obtidos dos testes de três modelos de parametrização da TRI: 1PL, 2PL e 3PL, com foco no parâmetro de dificuldade.

Durante os experimentos foram realizadas análises comparativas entre os níveis de dificuldades obtidos pela TRI nos elementos agrupados, e as correspondências dos agrupamentos criados pelos algoritmos de *clustering* com as rotulações já existentes nos *datasets*. Nestes experimentos foram consideradas situações adversas de agrupamento, como desbalanceamento nos grupos e problemas de dimensionalidade. As análises realizadas com os experimentos demonstraram que, ao avaliar essa abordagem em três categorias de algoritmos baseados em *k-means*, método *Fuzzy* e outros algoritmos clássicos, o modelo 2PL da TRI mostrou-se o mais preciso. Neste cenário, observou-se que os algoritmos baseados em *k-means* são menos influenciados pelas situações adversas para a formação dos agrupamentos, e que os algoritmos baseados em Fuzzy, como o C-means, são mais precisos nas atribuições corretas dos elementos agrupados com respeito às classes identificadas pela escala psicométrica como mais difíceis de acerto.

Conclui-se que esta Tese confirma a hipótese de que o parâmetro de dificuldade da TRI pode ser exitosamente aplicado como medida de escala psicométrica, a ser utilizada na avaliação de algoritmos de *clustering* em

ambientes complexos para suporte à decisão dos analistas humanos, bem como na avaliação de métricas no contexto de seleção de qualidade de algoritmos ideais para formação de classes. Além disso, mostrou-se que é possível utilizar a TRI para determinar algoritmos de *clustering* ótimos na determinação da pertinência dos elementos nos respectivos agrupamentos. Mas esta tese também possibilitou a criação de um método que expõe certas restrições no uso da TRI para avaliar algoritmos de *clustering* na identificação de itens corretos do agrupamento, restrições estas que devem ser analisadas com maior cautela pelos analistas humanos em dados da área de saúde.

6. CONTRIBUIÇÕES

Com base nos resultados e conclusões obtidas nesta Tese, abaixo estão enumeradas as seguintes contribuições:

- A aplicação do Parâmetro de Dificuldade da TRI nas avaliações das atribuições dos agrupamentos/classificações dos algoritmos de *clustering* em ambientes desfavoráveis, como desbalanceamento de classes, sobreposição de grupos, amostragem de dados, e outros fatores apresentados nesta Tese.
- Uma escala numérica de avaliação cognitiva baseada nos resultados do Parâmetro de Dificuldade da TRI, a serem utilizadas por algoritmos não supervisionados e possivelmente por algoritmos supervisionados de *Machine Learning*, como referências e indicadores de qualidade nas classificações, em ambientes desfavoráveis apresentados nesta Tese.
- Uma análise de como a dificuldade nas atribuições dos elementos em agrupamentos/classificações impactam nos resultados das métricas abordadas nesta Tese.
- Uma visão diferenciada de como avaliar novos modelos cognitivos para os analistas humanos, baseada nos experimentos realizados nesta Tese, com algoritmos tradicionais.

7. LIMITAÇÕES DOS RESULTADOS

O trabalho desenvolvido nesta Tese aplicando a avaliação do Parâmetro de Dificuldade (b) da TRI como fator comparativo na obtenção de bons resultados nas atribuições/classificações de algoritmos de *clustering*, apresentou os seguintes fatores de limitação dos resultados:

- a não aplicação em algoritmos supervisionados o método proposto, e
- não ter avaliado *datasets* com diferentes números de *clusters* com o método proposto.

8. TRABALHOS FUTUROS

A partir dos resultados obtidos, alguns pontos são promissores para a continuidade do embasamento teórico e dos experimentais apresentados, visando complementar dados e avaliar outros fatores que possam contribuir com melhorias nos resultados obtidos pelo método proposto nesta Tese, conforme abaixo:

- Realizar testes adicionais em outros *datasets e* algoritmos não supervisionados.
- Aplicar em algoritmos supervisionados e semi-supervisionados.
- Testar em agrupamentos de objetos nas imagens.
- Testar em dados contínuos.
- Testar o método proposto nesta Tese em outras métricas: Índice de Variação Normalizada (*V-Measure*), Homogeneidade, *Average Proportion of Non-overlap (APN), Average Distance (AD), Average Distance Between Means (ADM)* e *Figure Of Merit (FOM).*
- Propor inovação em sistemas automatizados de e-*Learning.*

Os resultados obtidos nos experimentos desta Tese proporcionarão referências para outros estudos e comparações com outras métricas, algoritmos e *datasets*.

REFERÊNCIAS BIBLIOGRÁFICAS

ACHARYA, Sudipta; SAHA, Sriparna; MORENO, Jose G.; DIAS, Gael. Multi-Objective Search Results Clustering. *In*: PROCEEDINGS OFCOLING 2014, THE 25TH INTERNATIONAL CONFERENCE ON COMPUTATIONAL LINGUISTICS: TECHNICAL PAPERS 2014, Dublin. Dublin p. 99–108. 2014.

AERTS, S.; HAESBROECK, G.; RUWET, C. Multivariate coefficients of variation: Comparison and influence functions. **Journal of Multivariate Analysis**, v. 142, p. 183–198, 2015. DOI: 10.1016/j.jmva.2015.08.006. Disponível em: http://dx.doi.org/10.1016/j.jmva.2015.08.006.

AGGARWAL, Cham C. **Data Streams: Models and Algorithms**. 1. ed. Boston/Dordrecht/London: DATA STREAMS: MODELS AND ALGORITHMS Edited by CHARU C. AGGARWAL IBM T. J. Watson Research Center, Yorktown Heights, NY 10598 Kluwer Academic Publishers, 2007. v. 31 DOI: 10.1007/978-0-387-47534-9. Disponível em: http://dx.doi.org/10.1007/978-0-387-47534-9_2.

ALOMARI, Yazan M.; ABDULLAH, Siti Norul Huda Sheikh; ZIN, Reena Rahayu Md; OMAR, Khairuddin. Iterative randomized irregular circular algorithm for proliferation rate estimation in brain tumor Ki-67 histology images. **Expert Systems with Applications**,v. 48, p. 111–129, 2016. DOI: 10.1016/j.eswa.2015.11.012.

ANAND, Sanjay Kumar; KUMAR, Suresh. Experimental Comparisons of Clustering Approaches for Data Representation. **ACM Computing Surveys**, v. 55, n. 3, p. 1–33, 2023. DOI: 10.1145/3490384.

ANDRADE, Dalton Francisco De; VALLE, Cunha. **Teoria da Resposta ao Item - Conceitos e Aplicações**. 1 ED. ed. UNICAMP, 2000. v. 1 Disponível em: https://docs.ufpr.br/~aanjos/CE095/LivroTRI_DALTON.pdf.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 10520: Informação e documentação — Citações em documentos – Apresentação. Rio de Janeiro, 2023.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 14724: Informação e documentação — Trabalhos acadêmicos — Apresentação. Rio de Janeiro, 2011. Disponível em: <https://www.normasabnt.org/abnt-nbr-14724/>. Acesso em: 4 jan. 2024.

AYAD, Hanan G.; KAMEL, Mohamed S. Cumulative voting consensus method for partitions with variable number of clusters. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, *[S. l.]*, v. 30, n. 1, p. 160–173, 2008. DOI: 10.1109/TPAMI.2007.1138.

AYO, Femi Emmanuel; FOLORUNSO, Olusegun; IBHARALU, Friday Thomas; OSINUGA, Idowu Ademola; ABAYOMI-ALLI, Adebayo. A probabilistic clustering model for hate speech classification in twitter. **Expert Systems with Applications**, v. 173, n. February, p. 114762, 2021. DOI: 10.1016/j.eswa.2021.114762. Disponível em: https://doi.org/10.1016/j.eswa.2021.114762.

BALDWIN, Peter. A Strategy for developing a common metric in item response theory when parameter posterior distributions are known. **Journal of Educational Measurement**, *[S. l.]*, v. 48, n. 1, p. 1–11, 2011. DOI: 10.1111/j.1745-3984.2010.00127.x.

BEALE, Mark Hudson; HAGAN, Martin T.; DEMUTH, Howard B. Neural Network Toolbox User ' s Guide How to Contact MathWorks. **MathWorks**, v. 1, n. June, p. 1–846, 2015. DOI: 10.1002/0471221546.

BEHERA, Rajat Kumar; BALA, Pradip Kumar; DHIR, Amandeep. The emerging role of cognitive computing in healthcare**: A systematic literature review. International Journal of Medical Informatics**, v. 129, n. February, p. 154–166, 2019. DOI: 10.1016/j.ijmedinf.2019.04.024.

BELEITES, Claudia; SALZER, Reiner; SERGO, Valter. Validation of soft classification models using partial class memberships: An extended concept of sensitivity & co. applied to grading of astrocytoma tissues. **Chemometrics and Intelligent Laboratory Systems**, v. 122, p. 12–22, 2013. DOI: 10.1016/j.chemolab.2012.12.003.

BENABDELLAH, Abla Chouni; BENGHABRIT, Asmaa; BOUHADDOU, Imane. A survey of clustering algorithms for an industrial context. **Procedia Computer Science**, v. 148, p. 291–302, 2019. DOI: 10.1016/j.procs.2019.01.022.

BENITEZ ROCHEL, R.; TRELLA LOPEZ, M.; CONEJO MUÑOZ, R.; BENITEZ ROCHEL, R.; TRELLA LOPEZ, M.; TRELLA LOPEZ, M.; CONEJO MUÑOZ, R.; CONEJO MUÑOZ, R. Neural networks applied to Item Response Theory. **Proceeding of the ICSC Symposia on Neural Computation (NC'2000) May 23-26, 2000 in Berlin, Germany**, p. 23–26, 2000.

BERGNER, Yoav; DRÖSCHLER, Stefan; KORTEMEYER, Gerd; RAYYAN, Saif; SEATON, Daniel; PRITCHARD, David E. Model-based collaborative filtering analysis of student response data: Machine-learning item response theory. **Proceedings of the 5th International Conference on Educational Data Mining, EDM 2012**, p. 95–102, 2012.

BERGNER, Yoav; HALPIN, Peter; VIE, Jill Jênn. Multidimensional Item Response Theory in the Style of Collaborative Filtering. **Psychometrika**, v. 87, n. 1, p. 266–288, 2022. DOI: 10.1007/s11336-021-09788-9.

BIBLIOTECA DIGITAL DE TESES E DISSERTAÇÕES. Disponível em: <https://repositorio.ufpe.br/handle/123456789/50>. Acesso em: 4 jan. 2024.

BOONGOEN, Tossapon; IAM-ON, Natthakan. Cluster ensembles: A survey of approaches with recent extensions and applications. **Computer Science Review**, v. 28, p. 1–25, 2018. DOI: 10.1016/j.cosrev.2018.01.003.

BRENER, Nicolas. A definable number which cannot be approximated algorithmically. **Imsart-generic**, v. 1, p. 1–6, 2010.

BROCK., Guy; VASYL PIHUR; SUSMITA DATTA; SOMNATH DATTA. Brillouin Scattering From Thermal Fluctuations in Superionic Conductors. **Journal of Statistical Software**, v. 25, n. 4, p. 371–372, 2008. DOI: 10.1016/0038-1098(77)91248-0.

BROUWER, Roelof K. Extending the rand, adjusted rand and jaccard indices to fuzzy partitions. **Journal of Intelligent Information Systems**, v. 32, n. 3, p. 213–235, 2009. DOI: 10.1007/s10844-008-0054-7.

CARVALHO, Francisco De A. T. De; MELO, Filipe M. De; LECHEVALLIER, Yves. Neurocomputing A multi-view relational fuzzy c-medoid vectors clustering algorithm. **Neurocomputing**, v. 163, p. 115–123, 2015. DOI: 10.1016/j.neucom.2014.11.083.

CATTELL, R. B. The measurement of adult intelligence. **Psychological Bulletin**, v. 40, n. 3, p. 153–193, 1943. DOI: 10.1037/h0059973.

CERREIA-VIOGLIO, Simone; DILLENBERGER, David; ORTOLEVA, Pietro. Cautious Expected Utility and the Certainty Effect. **Econometrica**, v. 83, n. 2, p. 693–728, 2015. DOI: 10.3982/ecta11733.

CHAITIN, Gregory. How real are real numbers? **Manuscrito - Rev. Int. Fil**, v. 34, n. 1, p. 115–141, 2011. DOI: 10.1590/s0100-60452011000100006.

CHAKRABORTY, Anuran; GHOSH, Kushal Kanti; DE, Rajonya; CUEVAS, Erik; SARKAR, Ram. Learning automata based particle swarm optimization for solving class imbalance problem. **Applied Soft Computing**, v. 113, p. 1–9, 2021. DOI: 10.1016/j.asoc.2021.107959.

CHALMERS, R. Philip. Mirt: A multidimensional item response theory package for the R environment. **Journal of Statistical Software**, v. 48, n. 6, p. 1–29, 2012. DOI: 10.18637/jss.v048.i06.

CHANG, Joshua C.; PORCINO, Julia; RASCH, Elizabeth K.; ID, Larry Tang. Regularized Bayesian calibration and scoring of the WD-FAB IRT model improves predictive performance over marginal maximum likelihood. **PLOS ONE**, v. 17, p. 1–17, 2022. DOI: 10.1371/journal.pone.0266350.

CHANG, Joshua C.; VATTIKUTI, Shashaank; CHOW, Carson C. Probabilistically-autoencoded horseshoe-disentangled multidomain item-response theory models. **4th Workshop on Bayesian Deep Learning (NeurIPS 2019), Vancouver, Canada.**,n. NeurIPS, p. 1–11, 2019. DOI: https://doi.org/10.48550/arXiv.1912.02351.

CHANG, Meng-I.; SHENG, Yanyan. A Comparison of Two MCMC Algorithms for the 2PL IRT Model BT - Quantitative Psychology. *In*: (L. Andries van der Ark, Marie Wiberg, Steven A. Culpepper, Jeffrey A. Douglas, Wen-Chung Wang, Org.) 2017, Cham. Cham: Springer International Publishing, 2017. p. 71–79. 2017.

AGGARWAL, Charu C.; REDDY, Chandan K. **Data Clustering: Algorithms and Applications**. 1. ed. London: CRC Press, v. 1, p. 1-648. 2014.

CHATTERJEE, Yagnik; BOURREAU, Eric; RANČIĆ, Marko J. Solving various NP-Hard problems using exponentially fewer qubits on a Quantum Computer. **arXiv**, v. 1, p. 1–11, 2023.

CHEN, Yu; FILHO, Telmo Silva; PRUDÊNCIO, Ricardo B. C.; DIETHE, Tom; FLACH, Peter. β3-IRT: A new item response model and its applications. **arXiv**, v. 89, n. March, p. 1–9, 2019.

CHEN, Ziheng; AHN, Hongshik. Item Response Theory Based Ensemble in Machine Learning. **International Journal of Automation and Computing**, v. 17, n. 5, p. 621–636, 2020. DOI: 10.1007/s11633-020-1239-y.

COELHO NETO, Giliate Cardoso; CHIORO, Arthur. After all, how many nationwide Health Information Systems are there in Brazil? **Cadernos de Saude Publica**, v. 37, n. 7, 2021. DOI: 10.1590/0102-311X00182119.

CUNHA, Mariana; MENDES, Ricardo; VILELA, João P. A survey of privacy-preserving mechanisms for heterogeneous data types. **Computer Science Review**, v. 41, p. 1–15, 2021. DOI: 10.1016/j.cosrev.2021.100403.

DAS, Amit Kumar; BHUYAN, Prasanta Kumar. Self-Organizing Tree Algorithm (SOTA) Clustering for Defining Level of Service (LOS) Criteria of Urban Streets. v. 1, p. 1–9, 2017.

DATTA, Susmita; DATTA, Somnath. Comparisons and validation of statistical clustering techniques for microarray gene expression data. **Bioinformatics**, v. 19, n. 4, p. 459–466, 2003. DOI: 10.1093/bioinformatics/btg025.

DESGRAUPES, Bernard. Clustering Indices. **CRAN Package**, v. 1, n. April, p. 1–34, 2013. Disponível em: cran.r-project.org/web/packages/clusterCrit.

DICKSON, Tara; KRUMWIEDE, Kim Hoggatt. Item response theory as a method to evaluate the goodness of fit for attitudes towards interprofessional healthcare teams. **Journal of Interprofessional Education and Practice**, v. 18, n. April 2019, p. 1–7, 2020. DOI: 10.1016/j.xjep.2019.04.002.

DIMITRIADOU, Evgenia; WEINGESSEL, Andreas; HORNIK, Kurt. Voting-merging: An ensemble method for clustering. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 2130, p. 217–224, 2001. DOI: 10.1007/3-540-44668-0_31.

DING, Ling; LI, Chao; JIN, Di; DING, Shifei. Survey of spectral clustering based on graph theory. **Pattern Recognition**, v. 151, n. January, p. 1-12, 2024. DOI: 10.1016/j.patcog.2024.110366.

DIOGO, Diogo P.; FRANCISCO DE, Francisco de A. Medoid based semi-supervised fuzzy clustering algorithms for multi-view relational data. **Fuzzy Sets and Systems**, v. 469, p. 1–27, 2023. DOI: 10.1016/j.fss.2023.108630.

DORN, M.; GRISCI, B. I.; NARLOCH, P. H.; FELTES, B. C.; AVILA, E.; KAHMANN, A.; ALHO, C. S. Comparison of machine learning techniques to handle imbalanced COVID-19 CBC datasets. **PeerJ Computer Science**, v. 7, p. 1–34, 2021. DOI: 10.7717/peerj-cs.670.

DOURADO, Marcia; LAKS, Jerson; LEIBING, Annette; ENGELHARDT, Eliasz. Consciência da doença na demência. **Revista de Psiquiatria Clínica**, v. 33, n. 21, p. 313–321, 2006.

DOUZAS, Georgios; RAUCH, Rene; BACAO, Fernando. G-SOMO: An oversampling approach based on self-organized maps and geometric SMOTE. **Expert Systems with Applications**, v. 183, n. March, p. 115230, 2021. DOI: 10.1016/j.eswa.2021.115230.

DOVŽAN, Dejan; ŠKRJANC, Igor. Recursive clustering based on a Gustafson-Kessel algorithm. **Evolving Systems**, v. 2, n. 1, p. 15–24, 2011. DOI: 10.1007/s12530-010-9025-7.

DUBBELMAN, Mark A; POSTEMA, Merel C.; JUTTEN, Roos J.; HARRISON, John E.; RITCHIE, Craig W.; ALEMAN, André; DE JONG, Frank Jan; SCHALET, Benjamin D.; TERWEE, Caroline B.; VAN DER; Flier, WIESJE M.; SCHELTENS, Philip; SIKKES, Sietske A. M.. What's in a score: A longitudinal investigation of scores based on item response theory and classical test theory for the Amsterdam Instrumental Activities of Daily Living Questionnaire in cognitively normal and impaired older adults. **Neuropsychology**, v. 38, n. 1, p. 96–105, 2023. DOI: 10.1037/neu0000914.

DUBOIS, Didier., PRADE, Henri. Fuzzy Sets and Systems: Theory and Applications, chap. Fuzzy Sets, v. 1. p. 1-393. **Academic Press,** Harcourt Brace Jovanovich.1980.

DUMONTIER, Michel; VENUGOPAL, Vinu E.; KUMAR, P. Sreenivasa. Difficulty-level modeling of ontology-based factual questions. **Semantic Web**, v. 11, n. 6, p. 1023–1036, 2020. DOI: 10.3233/SW-200381.

EHSANINIA, Jamshid; GHAVI HOSSEIN-ZADEH, Navid; SHADPARVAR, Abdol Ahad. Homogeneity and heterogeneity of variance components for milk and protein yield at different cluster sizes in Iranian Holsteins. **Livestock Science**, v. 188, p. 174–181, 2016. DOI: 10.1016/j.livsci.2016.04.020.

EL-ALFY, El Sayed M.; ABDEL-AAL, Radwan E. Construction and analysis of educational tests using abductive machine learning. **Computers and Education**, v. 51, n. 1, p. 1–16, 2008. DOI: 10.1016/j.compedu.2007.03.003.

FEIZOLLAH, Ali; ANUAR, Nor Badrul; SALLEH, Rosli; AMALINA, Fairuz. Comparative study of k-means and mini batch k-means clustering algorithms in android malware detection using network traffic analysis. **Proceedings - 2014 International Symposium on Biometrics and Security Technologies, ISBAST 2014**, p. 193–197, 2015. DOI: 10.1109/ISBAST.2014.7013120.

FENG, Zhixi; YANG, Shuyuan; WANG, Min; JIAO, Lichen. Learning Dual Geometric Low-Rank Structure for Semisupervised Hyperspectral Image Classification. **IEEE Transactions on Cybernetics**, v. 51, n. 1, p. 346–358, 2021. DOI: 10.1109/TCYB.2018.2883472.

FERNANDES FIGUEIREDO, Alex; SOUSA DE OLIVEIRA, Hygo; JAMES, Eduardo; SOUTO, Pereira. Redes Neurais Densas para Classificação de Estresse. **Journal of health informations**, v. 1, p. 202–208, 2021.

FERRI, C.; HERNÁNDEZ-ORALLO, J.; MODROIU, R. An experimental comparison of performance measures for classification. **Pattern Recognition Letters**, v. 30, n. 1, p. 27–38, 2009. DOI: 10.1016/j.patrec.2008.08.010.

FEUERSTAHLER, Leah M. Metric Stability in Item Response Models. **Multivariate Behavioral Research**, v. 57, n. 1, p. 94–111, 2022. DOI: 10.1080/00273171.2020.1809980.

FU, Zhihui; ZHANG, Susu; SU, Ya Hui; SHI, Ningzhong; TAO, Jian. A Gibbs sampler for the multidimensional four-parameter logistic item response model via a data augmentation scheme. **British Journal of Mathematical and Statistical Psychology**, v. 74, n. 3, p. 427–464, 2021. DOI: 10.1111/bmsp.12234.

GARDNER, Howard. **Frames of Mind: the theory of multiple intelligences**. New York. Ed.7. v. 34, n. 5, p. 1-529, 1983. DOI: 10.1016/j.intell.2006.04.002.

GHESMOUNE, Mohammed; LEBBAH, Mustapha; AZZAG, Hanene. State-of-the-art on clustering data streams. **Big Data Analytics**, v. 1, n. 1, p. 1–13, 2016. DOI: 10.1186/s41044-016-0011-3.

GILLIS, Nicolas. **Nonnegative Matrix Factorization**. Mons: Data Science Book Series, 2020. v. 1 DOI: 10.1137/1.9781611976410.

GIUSTI, Emanuele Maria; JONKMAN, Annelies; MANZONI, Gian Mauro; CASTELNUOVO, Gianluca; TERWEE, Caroline B.; ROORDA, Leo D.; CHIAROTTO, Alessandro. Proposal for Improvement of the Hospital Anxiety and Depression Scale for the Assessment of Emotional Distress in Patients With Chronic Musculoskeletal Pain: A Bifactor and Item Response Theory Analysis. **Journal of Pain**, v. 21, n. 3–4, p. 375–389, 2020. DOI: 10.1016/j.jpain.2019.08.003.

GOMES, Diego Eller; GUEDES DOS SANTOS, José Luís; PEREIRA BORGES, José Wicto; PEDROSO ALVES, Murilo; DE ANDRADE, Dalton Francisco; ERDMANN, Alacoque Lorenzini. Theory of the Response To the Item in Research in Public Health. **Journal of Nursing UFPE / Revista de Enfermagem UFPE**, v. 12, n. 6, p. 1800–1812, 2018.

GONZALEZ, Oscar. Psychometric and machine learning approaches for diagnostic assessment and tests of individual classification. **Psychological Methods**, *[S. l.]*, v. 26, n. 2, p. 236–254, 2021. DOI: 10.1037/met0000317.

GRIRA, Nizar; CRUCIANU, Michel; BOUJEMAA, Nozha. Semi-supervised fuzzy clustering with pairwise-constrained competitive agglomeration. **IEEE International Conference on Fuzzy Systems**, v. 2, n. 1, p. 867–872, 2005. DOI: 10.1109/fuzzy.2005.1452508.

GULTOM, Syawal; SRIADHI, S.; MARTIANO, M.; SIMARMATA, Janner. Comparison analysis of K-Means and K-Medoid with Ecluidience Distance Algorithm, Chanberra Distance, and Chebyshev Distance for Big Data Clustering. **IOP Conference Series: Materials Science and Engineering**, v. 420, n. 1, p. 1–7, 2018. DOI: 10.1088/1757-899X/420/1/012092.

HAHSLER, Michael; PIEKENBROCK, Matthew; DORAN, Derek. **dbscan : Fast Density-based Clustering with R**. v. 1, n. 1990, p. 1–28, 2016.

HASSAN, Md. Mehedi; MOLLICK, Swarnali; YASMIN, Farhana. An unsupervised cluster-based feature grouping model for early diabetes detection. **Healthcare Analytics**, v. 2, n. September, p. 100112, 2022. DOI: 10.1016/j.health.2022.100112.

HATHALIYA, Jigna J.; TANWAR, Sudeep. An exhaustive survey on security and privacy issues in Healthcare 4.0. **Computer Communications**, v. 153, n. September 2019, p. 311–335, 2020. DOI: 10.1016/j.comcom.2020.02.018.

HOSSIN, M.; SULAIMAN, M.. A Review on Evaluation Metrics for Data Classification Evaluations. **International Journal of Data Mining & Knowledge Management Process**, v. 5, n. 2, p. 01–11, 2015. DOI: 10.5121/ijdkp.2015.5201.

HUANG, Jinhong; YU, Zhu Liang; GU, Zhenghui. A clustering method based on extreme learning machine. **Neurocomputing**, v. 277, p. 108–119, 2017. DOI: 10.1016/j.neucom.2017.02.100.

JAIN, Preeti; SHRIVASTAVA, Gyanesh ; PANDEY, Umesh Kumar. Faculty Performance and Clusteirng – A Critical Review. **Journal od Algebraic Statistics**, v. 13, n. 1, p. 111–120, 2022.

JAYADEVA; PANT, Himanshu; SHARMA, Mayank; SOMAN, Sumit. Twin Neural Networks for the classification of large unbalanced datasets. **Neurocomputing**, v. 343, p. 34–49, 2019. DOI: 10.1016/j.neucom.2018.07.089.

JEON, Hyeon; AUPETIT, Michael; SHIN, DongHwa; CHO, Aeri; PARK, Seokhyeon; SEO, Jinwook. Sanity Check for External Clustering Validation Benchmarks using Internal Validation Measures., n. Clm, p. 1–13, 2022.

JIA, Pengfei; ZHANG, Chunkai; HE, Zhenyu. A new sampling approach for classification of imbalanced data sets with high density. **2014 International Conference on Big Data and Smart Computing, BIGCOMP 2014**, p. 217–222, 2014. DOI: 10.1109/BIGCOMP.2014.6741439.

HA, Jiawei; KAMBE, Micheline; PE, Jian. **Data Mining: Concepts and Techniques**. 3. ed. Waltham. v. 1. p. 1-703. 2011. DOI: 10.1016/C2009-0-61819-5.

KANDANAARACHCHI, Sevvandi. Unsupervised anomaly detection ensembles using item response theory. **Information Sciences**, v. 587, p. 142–163, 2022. a. DOI: 10.1016/j.ins.2021.12.042.

KARMARKAR, Uday S. Subjectively weighted utility and the Allais Paradox. **Organizational Behavior and Human Performance**, v. 24, n. 1, p. 67–72, 1979. DOI: 10.1016/0030-5073(79)90016-3

KHABSA, Madian; ELMAGARMID, Ahmed; ILYAS, Ihab; HAMMADY, Hossam; OUZZANI, Mourad. Learning to identify relevant studies for systematic reviews using random forest and external information. **Machine Learning**, v. 102, n. 3, p. 465–482, 2016. DOI: 10.1007/s10994-015-5535-7.

KHALID, Samina; KHALIL, Tehmina; NASREEN, Shamila. A survey of feature selection and feature extraction techniques in machine learning. **2014 Science and Information Conference**, p. 372–378, 2014. DOI: 10.1109/SAI.2014.6918213.

KIEFTENBELD, Vincent; NATESAN, Prathiba. Recovery of Graded Response Model Parameters: A Comparison of Marginal Maximum Likelihood and Markov Chain Monte Carlo Estimation. **Applied Psychological Measurement**, v. 36, n. 5, p. 399–419, 2012. DOI: 10.1177/0146621612446170.

KITCHENHAM, Barbara; CHARTERS, Shiart M. **Guidelines for performing Systematic Literature Reviews in Software EngineeringTechnical Report EBSE 2007- 001. Keele University and Durham University Joint Report**. Durham. v.1. p. 1-66. 2007.

KLEIN, CAROLINE GUGISCH. Avaliação da arquitetura óssea trabecular por meio de processamento de imagem digital em radiografias panorâmicas. 2005. 74 f. **Dissertação (Mestrado Graduação em Engenharia Elétrica e Informática Industrial) – Universidade Tecnológica Federal do Paraná, Curitiba**, 2005.

KLINE, A.; KLINE, T.; ABAD, Z. S. H.; LEE, J. Using item response theory for explainable machine learning in predicting mortality in the intensive care unit: Case-based approach. **Journal of Medical Internet Research**, v. 22, n. 9, 2020. DOI: 10.2196/20268.

KORUKONDA, APPA RAOTAKING STOCK OF TURING TEST: A REVIEW, ANALYSIS, And appraisal of issues surrounding thinking machines. Takingstock of Turingtest: a review, analysis, and appraisal of issues surroundingthinkingmachines. **International Journal of Human Computer Studies**, v. 58, n. 2, p. 240–257, 2003. DOI: 10.1016/S1071-5819(02)00139-8.

KRAUS, Johann M.; MÜSSEL, Christoph; PALM, Günther; KESTLER, Hans A. Multi-objective selection for collecting cluster alternatives. **Computational Statistics**, v. 26, n. 2, p. 341–353, 2011. DOI: 10.1007/s00180-011-0244-6.

KUMAR, Saran; D., Chandrakala. A Survey on Customer Churn Prediction using Machine Learning Techniques**. International Journal of Computer Applications**, v. 154, n. 10, p. 13–16, 2016. DOI: 10.5120/ijca2016912237.

LEITE-FILHO, P. F.; MELO, S. B.; CASSIA-MOURA, R. Teoria de resposta ao item e o paradoxo de Moravec na obtenção de algoritmos ótimos. In: Mostra de Extensão, Inovação e Pesquisa POLI/UPE 2021, 2021, Recife. **Anais da Mostra de Extensão, Inovação e Pesquisa POLI/UPE 2021**. v. 1. p. 165-166, 2021a.

LEITE-FILHO, P. F.; MELO, S. B.; CASSIA-MOURA, R. Métricas de rendimentos estatísticos em uma visão ampla dos dados. In: Semana Universitária UPE 2021 - Democratizando a ciência do litoral ao sertão, 2021, Recife. **Anais da Semana Universitária UPE 2021 - Democratizando a ciência do litoral ao sertão.** p. 1002, 2021b.

LEITE-FILHO, P. F.; MELO, S. B.. Tomada de decisões baseada no parâmetro *b* da teoria da resposta ao item nas classificações de ensemble. **Revista dos Mestrados Profissionais**, v. 11, n. 2, p. 235–249, 2022. DOI: 10.51359/2317-0115.2022.256869.

LEITE-FILHO, P. F.; MELO, S. B.. Significance of Item Response Theory in Cluster Evaluation Metrics. **2023 XLIX Latin American Computer Conference (CLEI),** pp. 1-9, 2023, DOI: 10.1109/CLEI60451.2023.10346171.

LI, Junnan; ZHU, Qingsheng; WU, Quanwang; ZHANG, Zhiyong; GONG, Yanlu; HE, Ziqing; ZHU, Fan. SMOTE-NaN-DE: Addressing the noisy and borderline examples problem in imbalanced classification by natural neighbors and differential evolution. **Knowledge-Based Systems**, v. 223, p. 1–16, 2021. DOI: 10.1016/j.knosys.2021.107056.

LI, Xiaohua.; ZHENG, Jian.. Joint machine learning and human learning design with sequential active learning and outlier detection for linear regression problems. In: **ANNUAL CONFERENCE ON INFORMATION SCIENCE AND SYSTEMS (CISS). Princeton, NJ, USA, 2016**. v. 1. p. 407-411, 2016. DOI: 10.1109/CISS.2016.7460537.

LIN, Wei Chao; TSAI, Chih Fong; HU, Ya Han; JHANG, Jing Shang. Clustering-based undersampling in class-imbalanced data. **Information Sciences,** v. 409–410, p. 17–26, 2017. DOI: 10.1016/j.ins.2017.05.008.

LINDEN, Wim J. Van der. **Habdbook of Item Response Theory**. 3. ed. Boca Raton: CRC Press,. v. 7. p. 1-1688. 2016.

LIU, Jie. Fuzzy support vector machine for imbalanced data with borderline noise. Fuzzy Sets and Systems, v. 413, p. 64–73, 2021. DOI: 10.1016/j.fss.2020.07.018

LIU, Yanchi; LI, Zhongmou; XIONG, Hui; GAO, Xuedong; WU, Junjie; WU, Sen. Understanding and enhancement of internal clustering validation measures. **IEEE Transactions on Cybernetics**, , v. 43, n. 3, p. 982–994, 2013. DOI: 10.1109/TSMCB.2012.2220543.

LIU, Yangguang; ZHOU, Yangming; WEN, Shiting; TANG, Chaogang. A Strategy on Selecting Performance Metrics for Classifier Evaluation. **International Journal of Mobile Computing and Multimedia Communications**,, v. 6, n. 4, p. 20–35, 2014. DOI: 10.4018/IJMCMC.2014100102.

LU, Jing; ZHANG, Jiwei; TAO, Jian. Slice–Gibbs sampling algorithm for estimating the parameters of a multilevel item response model. **Journal of Mathematical Psychology**, v. 82, p. 12–25, 2018. DOI: 10.1016/j.jmp.2017.10.005.

LUKASIK, Szymon; KOWALSKI, Piotr A.; CHARYTANOWICZ, Malgorzata; KULCZYCKI, Piotr. Clustering using flower pollination algorithm and Calinski-Harabasz index. 2016 **IEEE Congress on Evolutionary Computation, CEC 2016.** v. 1. p. 2724–2728, 2016. DOI: 10.1109/CEC.2016.7744132.

LUMLEY, Thomas; SCOTT, Alastair. AIC and BIC for modeling with complex survey data. Journal of Survey **Statistics and Methodology**, v. 3, n. 1, p. 1–18, 2015. DOI: 10.1093/jssam/smu021.

LUO, Yong; WOLF, Melissa Gordon. Item Parameter Recovery for the Two-Parameter Testlet Model with Different Estimation Methods. **Psychological Test and Assessment Modeling,** v. 61, n. 1, p. 65–89, 2019.

LYNDON, Mataroria P. et al. Hacking Hackathons: Preparing the next generation for the multidisciplinary world of healthcare technology. **International Journal of Medical Informatics**, v. 112, n. December 2017, p. 1–5, 2018. DOI: 10.1016/j.ijmedinf.2017.12.020.

MAECHLER, M.; STRUYF, Anja; HUBERT, Mia; HORNIK, K.; STUDER, Matthias; ROUDIER, Pierre. Package 'cluster': Cluster Analysis Basics and Extensions. **R topics Documented**, v. 1. p. 1–79, 2015.

MALTAURO, Tamara Cantú; GUEDES, Luciana Pagliosa Carvalho; URIBE-OPAZO, Miguel Angel; CANTON, Letícia Ellen Dal. Spatial multivariate optimization for a sampling redesign with a reduced sample size of soil chemical properties. **Revista Brasileira de Ciencia do Solo**, v. 47, p. 1–23, 2023. DOI: 10.36783/18069657rbcs20220072.

MARAZIOTIS, Loannis A. A semi-supervised fuzzy clustering algorithm applied to gene expression data. **Pattern Recognition**, v. 45, n. 1, p. 637–648, 2012. DOI: 10.1016/j.patcog.2011.05.007.

MARTÍNEZ-PLUMED, Fernando; PRUDÊNCIO, Ricardo B. C.; MARTÍNEZ-USÓ, Adolfo; HERNÁNDEZ-ORALLO, José. Item response theory in AI: Analysing machine learning classifiers at the instance level. **Artificial Intelligence**, v. 271, p. 18–42, 2019. DOI: 10.1016/j.artint.2018.09.004.

MARTÍNEZ-PLUMED, Fernando; PRUDÊNCIO, Ricardo B. C.; MARTÍNEZ-USÓ, Adolfo; HERNÁNDEZ-ORALLO, José. Making sense of Item response theory in machine learning. **Frontiers in Artificial Intelligence and Applications**, v. 285, p. 1140–1148, 2016. DOI: 10.3233/978-1-61499-672-9-1140.

MATEO, J.; TORRES, A. M.; APARICIO, A.; SANTOS, J. L. An efficient method for ECG beat classification and correction of ectopic beats. **Computers and Electrical Engineering**, v. 53, p. 219–229, 2016. DOI: 10.1016/j.compeleceng.2015.12.015.

MCMAHON, Kibby; HOERTEL, Nicolas; PEYRE, Hugo; BLANCO, Carlos; FANG, Caitlin; LIMOSIN, Frédéric. Age differences in DSM-IV borderline personality disorder symptom expression: Results from a national study using item response theory (IRT). **Journal of Psychiatric Research**, v. 110, n. December 2018, p. 16–23, 2019. DOI: 10.1016/j.jpsychires.2018.12.019.

MEI, Jian Ping. Semisupervised Fuzzy Clustering with Partition Information of Subsets. **IEEE Transactions on Fuzzy Systems**, v. 27, n. 9, p. 1726–1737, 2019. DOI: 10.1109/TFUZZ.2018.2889010.

MELLO, Iza Rodrigues; GUIMARÃES, Noélly Maura de Jesus; MONTEIRO, Luana Silva; TAETS, Gunnar de Cunto Carelli. Cluster de Sintomas e o Impacto na Qualidade de Saúde Global de Pacientes com Câncer Avançado. **Revista Brasileira de Cancerologia**, v. 67, n. 3, p. 1–21, 2021. DOI: 10.32635/2176-9745.rbc.2021v67n3.1190.

MINATI, Gianfranco. A note on the reality of incomputable real numbers and its systemic significance. **Systems**, v. 9, n. 2, p. 1–15, 2021. DOI: 10.3390/systems9020044.

MITRA, Sayantan; HASANUZZAMAN, Mohammed; SAHA, Sriparna; WAY, Andy. Incorporating deep visual features into multiobjective based multi-view search results clustering. COLING 2018 - **27th International Conference on Computational Linguistics, Proceedings**, v. 1.p. 3793–3805, 2018.

MOHEDANO-MUNOZ, M. A.; ALIQUE-GARCÍA, S.; RUBIO-SÁNCHEZ, M.; RAYA, L.; SANCHEZ, A. Interactive visual clustering and classification based on dimensionality reduction mappings: A case study for analyzing patients with dermatologic conditions. **Expert Systems with Applications**, v. 171, n. January, p. 1–11, 2021. DOI: 10.1016/j.eswa.2021.114605.

MORAES, João V. C.; REINALDO, Jéssica T. S.; FERREIRA-JUNIOR, Manuel; FILHO, Telmo Silva; PRUDÊNCIO, Ricardo B. C. Evaluating regression algorithms at the instance level using item response theory. **Knowledge-Based Systems**, v. 240, p. 1–14, 2022. DOI: 10.1016/j.knosys.2021.108076.

NEWMAN, John Eatwell; MURRAYMILGATE, Peter. **Utility and Probability.** 1. ed. London: THE MACMILLAN PRESS LIMITED, V.1. p.-330.1990. DOI: 10.1007/978-1-349-20568-4.

OLIVEIRA, Chaina S.; MORAES, João V. C.; FILHO, Telmo Silva; PRUDÊNCIO, Ricardo B. C. A two-level Item Response Theory model to evaluate speech synthesis and recognition. **Speech Communication**, v. 137, n. September 2021, p. 19–34, 2022. DOI: 10.1016/j.specom.2021.11.002.

OSINENKO, Pavel; DOBRIBORSCI, Dmitrii; AUMER, Wolfgang. Reinforcement learning with guarantees: a review. **IFAC-PapersOnLine**, v. 55, n. 15, p. 123–128, 2022. DOI: 10.1016/j.ifacol.2022.07.619.

OUZZANI, Mourad; HAMMADY, Hossam; FEDOROWICZ, Zbys; ELMAGARMID, Ahmed. Rayyan-a web and mobile app for systematic reviews. **Systematic Reviews**, v. 5, n. 1, p. 1–10, 2016. DOI: 10.1186/s13643-016-0384-4.

PAGANIN, Sally; PACIOREK, Christopher J.; WEHRHAHN, Claudia; RODRIGUEZ, Abel; RABE-HESKETH, Sophia; DE VALPINE, Perry. Computational methods for Bayesian semiparametric Item Response Theory models. **arXiv**. v. 1, p. 1–43, 2022. DOI: https://doi.org/10.48550/arXiv.2101.11583.

PAGE, M. J.; MCKENZIE, J. E.; BOSSUYT, P. M.; BOUTRON, I.; HOFFMANN, T. C; MULROW, C. D.; SHAMSEER, L.; TETZLAFF, JENNIFER M.; AKL, E. A.; BRENNAN, S. E.; CHOU, R.; GLANVILLE, J.; GRIMSHAW, J. M.; HRÓBJARTSSON, A.; LALU, M; M.; LI, T.; LODER, E. W.; MAYO-WILSON, E.; MCDONALD, S.; MCGUINNESS, L. A.; STEWART, L. A.; THOMAS, J.; TRICCO, A. C.; WELCH, V. A.; WHITING, P.; MOHER, D.. The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. **International Journal of Surgery**, v. 88, n. March, p. 1-9. 2021. DOI: 10.1016/j.ijsu.2021.105906.

PAIGE, Samantha R.; KRIEGER, Janice L.; STELLEFSON, Michael; ALBER, Julia M. eHealth literacy in chronic disease patients: An item response theory analysis of the eHealth literacy scale (eHEALS). **Patient Education and Counseling**, v. 100, n. 2, p. 320–326, 2017. DOI: 10.1016/j.pec.2016.09.008.

PERERA, Subashan; VANSWEARINGEN, Jessie; SHUMAN, Valerie; BRACH, Jennifer S. Assessing gait efficacy in older adults: An analysis using item response theory. **Gait and Posture**, v. 77, n. July 2019, p. 118–124, 2020. DOI: 10.1016/j.gaitpost.2020.01.028.

PIAGET, Jean. **The origins of intelligence in children.** (Margaret Cook, Org.). **The origins of intelligence in children.** New York, NY, USW W Norton & Co, v.1, p. 1-419. 1952. DOI: 10.1037/11494-000.

PLIAKOS, Konstantinos; JOO, Seang Hwane; PARK, Jung Yeon; CORNILLIE, Frederik; VENS, Celine; VAN DEN NOORTGATE, Wim. Integrating machine learning into item response theory for addressing the cold start problem in adaptive learning systems. **Computers and Education**, v. 137, n. April, p. 91–103, 2019. DOI: 10.1016/j.compedu.2019.04.009.

POURBAHRAMI, Shahin; BALAFAR, Mohammad Ali; KHANLI, Leyli Mohammad; KAKARASH, Zana Azeez. A survey of neighborhood construction algorithms for clustering and classifying data points. **Computer Science Review,** v. 38, p. 1-17.2020. DOI: 10.1016/j.cosrev.2020.100315.

POWERS, David M. W. Evaluation: from precision, recall and F-measure to ROC, **Informedness, markedness and correlation**. n. May, v. 1. p. 37–63, 2020. DOI: 10.9735/2229-3981.

PRATAP, Vibha; SINGH, Amit Prakash. Novel fuzzy clustering-based undersampling framework for class imbalance problem. **International Journal of System Assurance Engineering and Management**, v. 14, n. 3, p. 967–976, 2023. DOI: 10.1007/s13198-023-01897-1.

PUA, Y. H.; TERLUIN, B.; TAY, L.; CLARK, R. A.; THUMBOO, J.; TAY, E. L.; MAH, S. M.; N. G, Y. S.. Using item response theory to estimate interpretation threshold values for the Frailty Index in community dwelling older adults. **Archives of Gerontology and Geriatrics**, v. 117, n. November 2023, p. 1–6, 2024. DOI: 10.1016/j.archger.2023.105280.

PURI, Arjun; GUPTA, Manoj Kumar. Improved Hybrid Bag-Boost Ensemble with K-Means-SMOTE-ENN Technique for Handling Noisy Class Imbalanced Data. **Computer Journal,** v. 65, n. 1, p. 124–138, 2022. DOI: 10.1093/comjnl/bxab039.

QIANG, Qianyao; ZHANG, Bin; NIE, Feiping; WANG, Fei. Multi-view semi-supervised learning with adaptive graph fusion. **Neurocomputing**, v. 557, p. 1-13. n. August, 2023. DOI: 10.1016/j.neucom.2023.126685.

RENDÓN, Eréndira; ABUNDEZ, Itzel; ARIZMENDI, Alejandra; QUIROZ, Elvia M. Internal versus External cluster validation indexes. **International Journal of Computers and Communications**,v. 5, n. 1, p. 27--34, 2011.

SAHA, Sriparna; MITRA, Sayantan; KRAMER, Stefan. Exploring multiobjective optimization for multiview clustering. **ACM Transactions on Knowledge Discovery from Data**, 12, n. 4, v. 1. p. 1-31. 2018. DOI: 10.1145/3182181.

SAINI, Naveen; BANSAL, Diksha; SAHA, Sriparna; BHATTACHARYYA, Pushpak. Multi-objective multi-view based search result clustering using differential evolution framework. **Expert Systems with Applications**, v. 168, n. November 2020, p. 1-14, 2021. DOI: 10.1016/j.eswa.2020.114299.

SANCHES, Marcelo Kaminski. **Aprendizado de máquina semi-supervisionado: proposta de um algoritmo para rotular exemplos a partir de poucos exemplos rotulados**. 2003. USP, p. 1-120. 2003.

SANTOS, Jorge M.; EMBRECHTS, M. Artificial Neural Networks – ICANN 2009. *In*: **CONFERENCE: ARTIFICIAL NEURAL NETWORKS - ICANN 2009, 19TH INTERNATIONAL CONFERENCE 2009**, v. 1. p. 416–425. 2009. DOI: 10.1007/978-3-642-04277-5.

SCHAPIRA, Marilyn M.; WALKER, Cindy M.; SEDIVY, Sonya K. Evaluating existing measures of health numeracy using item response theory. **Patient Education and Counseling**, v. 75, n. 3, p. 308–314, 2009. DOI: 10.1016/j.pec.2009.03.035.

SCHARDOSIM, J. M.; RODRIGUES, N. L. de A.; RATTNER, D.. Parâmetros utilizados na avaliação de bem-estar do bebê no nascimento. **Avances en Enfermería**, v. 36, n. 2, p. 187–208, 2018. DOI: 10.15446/av.enferm.v36n2.67809.

SESSLER, C. N.; GOSNELL, M. S.; GRAP, M. J.; BROPHY, G. M.; O'NEAL, P. V.; KEANE, K. A.; TESORO, E. P.; ELSWICK, R. K. The Richmond Agitation-Sedation Scale: validity and reliability in adult intensive care unit patients. **American Journal of Respiratory and Critical Care Medicine**, v. 166, n. 10, p. 1338–1344, 2002. DOI: 10.1164/rccm.2107138.

SETIAWATI, Farida Agus; IZZATY, Rita Eka; HIDAYAT, Veny. Items parameters of the space-relations subtest using item response theory. **Data in Brief,** v. 19, p. 1785–1793, 2018. DOI: 10.1016/j.dib.2018.06.061.

SILVA ABREU, M. N.; SIQUEIRA, A. L.; CAIAFFA, W. T. Ordinal logistic regression in epidemiological studies. **Revista de Saude Publica**, v. 43, n. 1, p. 1–11, 2009. DOI: 10.1590/S0034-89102009000100025.

SISODIA, Dilip Singh; VERMA, Shrish; VYAS, Om Prakash. A subtractive relational fuzzy c-medoids clustering approach to cluster web user sessions from web server logs. **International Journal of Applied Engineering Research**, v. 12, n. 7, p. 1142–1150, 2017.

SOBRAL, David Santos; SANTOS, Tamires Flores Dos; SOUSA, Miguel Angelo de Abreu De. IDENTIFICAÇÃO DE IMAGENS DE RAIO-X COM REDES NEURAIS CONVOLUCIONAIS PARA DETECÇÃO DE PNEUMONIA CAUSADA POR COVID-19. **Angewandte Chemie International Edition**, 6(11), 951–952., v. 2, p. 71–86, 2019.

SOUSA, Arthur F. M.; PRUDÊNCIO, Ricardo B. C.; LUDERMIR, Teresa B.; SOARES, Carlos. Active learning and data manipulation techniques for generating training examples in meta-learning. **Neurocomputing**, v. 194, p. 45–55, 2016. DOI: 10.1016/j.neucom.2016.02.007.

SUKASSINI, M. P.; T.VELMURUGAN. A SURVEY ON THE RESEARCH CHALLENGES OF BIG DATA ANALYTICS. **Journal of Fundamental & Comparative Research,** v. VIII, n. 1, p. 36–42, 2022.

TAHIR, M. A. ; KITTLER, J. ; YAN, F.. Inverse random under sampling for class imbalance problem and its application to multi-label classification. **Pattern Recognition**, v. 45, n. 10, p. 3738–3750, 2012. DOI: 10.1016/j.patcog.2012.03.014.

TANAKA, Oswaldo Yoshimi; DRUMOND, Marcos; CRISTO, Elier Broche; SPEDO, Sandra Maria; DA SILVA PINTO, Nicanor Rodrigues. Uso da análise de clusters como ferramenta de apoio à gestão no SUS. **Saude e Sociedade,** v. 24, n. 1, p. 34–45, 2015. DOI: 10.1590/S0104-12902015000100003.

TANG, Wei; YANG, Yang; ZENG, Lanling; ZHAN, Yongzhao. Optimizing MSE for clustering with balanced size constraints. **Symmetry**, v. 11, n. 3, p. 1–16, 2019. DOI: 10.3390/sym11030338.

TAO, Xinmin; LI, Qing; REN, Chao; GUO, Wenjie; LI, Chenxi; HE, Qing; LIU, Rui; ZOU, Junrong. Real-value negative selection over-sampling for imbalanced data set learning. **Expert Systems with Applications**, v. 129, p. 118–134, 2019. DOI: 10.1016/j.eswa.2019.04.011.

TEZZA, Rafael; BORNIA, Antonio Cezar; ANDRADE, Dalton Francisco De; BARBETTA, Pedro Alberto. Modelo multidimensional para mensurar qualidade em website de e-commerce utilizando a teoria da resposta ao item. **Gestão & Produção**, v. 25, n. 4, p. 916–934, 2018. DOI: 10.1590/0104-530x1875-18.

THOMAS, Juan Carlos Rojas; PEÑAS, Matilde Santos; MORA, Marco. New Version of Davies-Bouldin Index for Clustering Validation Based on Cylindrical Distance. **Proceedings - International Conference of the Chilean Computer Science Society, SCCC,** v. 0, n. 1, p. 49–53, 2013. DOI: 10.1109/SCCC.2013.29.

TOXIRJONOVICH, O'rinov Nodirbek; FOZILOVICH, Yunusov Odiljon. Artificial Intelligence and its Application in Information Security Management. **Central Asian Journal of Theoretical & Applied Sciences**, v. 3, n. 4, p. 90–97, 2022. DOI: 10.13140/RG.2.2.12066.09921.

TREADWELL, Jonathan R.; LENERT, Leslie A. Health values and prospect theory. **Medical Decision Making**, v. 19, n. 3, p. 344–352, 1999. DOI: 10.1177/0272989X9901900313.

UNIVERSIDADE FEDERAL DE PERNAMBUCO. Disponível em: <https://www.ufpe.br/documents/39654/195702/MODELO+DE+DISSERTA%C3%87%C3%83O+OU+TESE+-+VERS%C3%83O+PDF/e599ff4a-726e-42b8-b6cc-eec8da82b802>. Acesso em: 4 jan. 2024.

URBAN, Christopher J.; BAUER, Daniel J. A Deep Learning Algorithm for High-Dimensional Exploratory Item Factor Analysis. **Psychometrika**, v. 86, n. 1, p. 1–29, 2021. DOI: 10.1007/s11336-021-09748-3.

UTO, Masaki; UENO, Maomi. Empirical comparison of item response theory models with rater's parameters. **Heliyon**, v. 4, n. 5, p. 1–32, 2018. DOI: 10.1016/j.heliyon.2018.e00622

VAN DE KUILEN, Gijs; WAKKER, Peter P. Learning in the Allais paradox. **Journal of Risk and Uncertainty**, v. 33, n. 3, p. 155–164, 2006. DOI: 10.1007/s11166-006-0390-3.

VAN HULSE, Jason; KHOSHGOFTAAR, Taghi M.; NAPOLITANO, Amri. Experimental perspectives on learning from imbalanced data. **ACM International Conference Proceeding Series**, v. 227, n. January, p. 935–942, 2007. DOI: 10.1145/1273496.1273614.

VAN STRALEN, Karlijn J.; STEL, Vianda S.; REITSMA, Johannes B.; DEKKER, Friedo W.; ZOCCALI, Carmine; JAGER, Kitty J. Diagnostic methods I: Sensitivity, specificity, and other measures of accuracy. **Kidney International**, v. 75, n. 12, p. 1257–1263, 2009. DOI: 10.1038/ki.2009.92.

VIEIRA, Kelmara Mendes; POTRICH, Ani Caroline Grigion; BRESSAN, Aureliano Angel. A proposal of a financial knowledge scale based on item response theory. **Journal of Behavioral and Experimental Finance**, v. 28, p. 1–11, 2020. DOI: 10.1016/j.jbef.2020.100405.

WANG, Fei; FRANCO-PENYA, Hector Hugo; KELLEHER, John D.; PUGH, John; ROSS, Robert. An analysis of the application of simplified silhouette to the evaluation of k-means clustering validity. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 10358 LNAI, n. July, p. 291–305, 2017. DOI: 10.1007/978-3-319-62416-7_21.

WANG, Pingxin; YANG, Xibei; DING, Weiping; ZHAN, Jianming; YAO, Yiyu. Three-way clustering: Foundations, survey and challenges. **Applied Soft Computing**, v. 151, n. December 2023, p. 1-13. 2024a. DOI: 10.1016/j.asoc.2023.111131.

WANG, Yizhang; QIAN, Jiaxin; HASSAN, Muhammad; ZHANG, Xinyu; ZHANG, Tao; YANG, Chao; ZHOU, Xingxing; JIA, Fengjin. Density peak clustering algorithms: A review on the decade 2014–2023. **Expert Systems with Applications**, v. 238, n. PA, p. 1-16, 2024b. DOI: 10.1016/j.eswa.2023.121860.

WARTELLE, Adrien; MOURAD-CHEHADE, Farah; YALAOUI, Farouk; CHRUSCIEL, Jan; LAPLANCHE, David; SANCHEZ, Stéphane. Clustering of a health dataset using diagnosis co-occurrences. **Applied Sciences (Switzerland)**, v. 11, n. 5, p. 1–20, 2021. DOI: 10.3390/app11052373.

WEISS, Karl; KHOSHGOFTAAR, Taghi M.; WANG, Ding Ding. A survey of transfer learning. **Springer International Publishing**, v. 3, p. 1-40. 2016.DOI: 10.1186/s40537-016-0043-6.

WIJAYA, Yudhistira Arie; KURNIADY, Dedy Achmad; SETYANTO, Eddy; TARIHORAN, Wahdan Sanur; RUSMANA, Dadan; RAHIM, Robbi. Davies Bouldin Index Algorithm for Optimizing Clustering Case Studies Mapping School Facilities. **TEM Journal**, 10, n. 3, p. 1099–1103, 2021. DOI: 10.18421/TEM103-13.

WOTEKI, Catherine E.; KINEMAN, Brian D. C Hallenges and a Pproaches To R Educing. **Review Literature And Arts Of The Americas**, v. 1. n. June, p. 177–182, 2003.

WOUDSTRA, Anke J.; SMETS, Ellen M. A.; GALENKAMP, Henrike; FRANSEN, Mirjam P. Validation of health literacy domains for informed decision making about colorectal cancer screening using classical test theory and item response theory. **Patient Education and Counseling,** v. 102, n. 12, p. 2335–2343, 2019. DOI: 10.1016/j.pec.2019.09.016.

WU, Junjie; CHEN, Jian; XIONG, Hui; XIE, Ming. External validation measures for K-means clustering: A data distribution perspective. **Expert Systems with Applications**, v. 36, n. 3 PART 2, p. 6050–6061, 2009. DOI: 10.1016/j.eswa.2008.06.093

YEUNG, Chun Kit. Deep-IRT: Make deep learning based knowledge tracing explainable using item response theory. EDM 2019 - **Proceedings of the 12th International Conference on Educational Data Mining**, v. 1. p. 683–686, 2019. DOI: https://doi.org/10.48550/arXiv.1904.11738.

YIN, Qiyue; ZHANG, Junge; WU, Shu; LI, Hexi. Multi-view clustering via joint feature selection and partially constrained cluster label learning. **Pattern Recognition**, v. 93, p. 380–391, 2019. DOI: 10.1016/j.patcog.2019.04.024.

YUAN, Chao; YANG, Liming. Knowledge-Based Systems Large margin projection-based multi-metric learning for classification. **Knowledge-Based Systems**, v. 243, p. 1-15, 2022. DOI: 10.1016/j.knosys.2022.108481.

ZHANG, Cha; MA, Yunqian. **Ensemble Machine Learning**. 1. ed. New York: Spring, 2012. v. 1. p; 1-321. 2012. DOI: 10.1007/978-1-4419-9326-7.

ZHENG, Yi; CHEON, Hyunjung; KATZ, Charles M. Using Machine Learning Methods to Develop a Short Tree-Based Adaptive Classification Test: Case Study With a High-Dimensional Item Pool and Imbalanced Data. **Applied Psychological Measurement**, v. 44, n. 7–8, p. 499–514, 2020. DOI: 10.1177/0146621620931198.

ZHOU, Qian; SUN, Bo; SONG, Yunsheng; LI, Shuang. K-means Clustering Based Undersampling for Lower Back Pain Data. In: **PROCEEDINGS OF THE 2020 3RD INTERNATIONAL CONFERENCE ON BIG DATA TECHNOLOGIES 2020,** New York, NY, USA. New York, NY, USA: ACM, v. 1. p. 53–57. 2020. DOI: 10.1145/3422713.3422725.

ZHOU, Zhipeng.; GUO, Wenya.. Applications of item response theory to measuring the safety response competency of workers in subway construction projects.**Safety Science**, v. 127, n. Oct.r 2019, p. 1-11, 2020. DOI: 10.1016/j.ssci.2020.1047041

ZHU, Fan. SMOTE-NaN-DE: Addressing the noisy and borderline examples problem in imbalanced classification by natural neighbors and differential evolution. **Knowledge-Based Systems**, . 223, p. 1–16, 2021. DOI: 10.1016/j.knosys.2021.107056.

ZHU, Gancheng; ZHOU, Yuci; ZHOU, Fengfeng; WU, Min; ZHAN, Xiangping; SI, Yingdong; WANG, Peng; WANG, Jun. Proactive Personality Measurement Using Item Response Theory and Social Media Text Mining. **Frontiers in Psychology,** v. 12, n. July, p. 1–14, 2021. DOI: 10.3389/fpsyg.2021.705005.

ZHU, Hongyue; GAO, Wei; ZHANG, Xue. Bayesian Analysis of a Quantile Multilevel Item Response Theory Model. **Frontiers in Psychology**, v. 11, n. January, p. 1–15, 2021. DOI: 10.3389/fpsyg.2020.607731.

APÊNDICE A – Código fonte utilizado nos experimentos.

```
rm(list=ls())

InstallPackages<-function(){
  # Instasll packages

  .lib<- c("ltm","devtools", "ggplot2","stats", "ggrepel", "wordcloud", "Hmisc",
"ggfortify","grid","mirt", "dplyr", "reshape2") #"mirt"
  .lib<-
c("RSNNS","clusterSim","cluster","MASS","caret","kohonen","clValid","mclust",
"beepr", "pROC","tictoc",
          "PMCMR","pROC","ROCR","xlsx",
"EvaluationMeasures","pcIRT","eRm","latticeExtra","corrplot","tm","SnowballC"
,

"RColorBrewer","irr","nortest","pspearman","Kendall","gridExtra","Metrics","plo
t3D","readxl","pacman","rstatix","irr",

"DescTools","rcompanion","PMCMRplus","dunn.test","effsize","MOCCA","FSA",
"smotefamily","ppclust","pvec","clusterCrit") #rcompanion
  .inst <- .lib %in% installed.packages()
  if (length(.lib[!.inst])>0) install.packages(.lib[!.inst],
repos=c("http://rstudio.org/_packages", "http://cran.rstudio.com"))
  lapply(.lib, require, character.only=TRUE)

}

LoadLibrary<-function(){

  library(MOCCA)
  library(ppclust)
  library(clusterCrit)
  #library("EBImage")
  library(jpeg)
  library(ClusterR)
  library(smotefamily)
  library(RSNNS) ;
  library(clusterSim,cluster);
  library(MASS);
  library(caret);
  library(kohonen);
  library(clValid);
  library(mclust);
  library(ggplot2);
  library(beepr);
  library(PMCMR)
  library(pROC)
  library(ROCR)
  library(mirt)
  library(xlsx)
  library(readxl)
  library(EvaluationMeasures)
  library(ltm, mvtnorm)
```

```
  library(pcIRT)
  library(eRm,latticeExtra)
  library(nortest)
  library(pspearman)
  library(Kendall)
  library(gridExtra)
  library(Metrics)
  library(plot3D)
  library(corrplot)
  library(Hmisc)
  library(PerformanceAnalytics)
  library(pacman)
  library(rstatix)
  library(DescTools)
  library(irr)
  library(tictoc)
  library(rcompanion)
  library(PMCMRplus)
  library(dunn.test)
  library(FSA)
  library(parallel)

  print("------------------");
  print("Libraries OK");
  print("------------------");
}

SlicesStartMin<-function(DatasetSelected){
  dataSplit1<-c();dataSplit2<-c();

  rawdata<-DatasetSelected

  # data
  # dataAnaliseClass

  #-----------------------------------------
  cls<-summary(dataAnaliseClass)
  if(cls[1]<cls[2]){mini<-"0"}
  if(cls[1]>=cls[2]){mini<-"1"}

  library(dplyr)

  if(mini=="0"){
    mirority<-min(cls);
    dataAnaliseOrder<-arrange(dataAnalise, dataAnalise[,dataAnaliseCol],);
    splitMinority<-dataAnaliseOrder[1:mirority,];
    splitMajority<-dataAnaliseOrder[(mirority+1):(dim(dataAnaliseOrder)[1]),]
  }

  if(mini=="1"){
    mirority<-min(cls);
    dataAnaliseOrder<-arrange(dataAnalise, desc(dataAnalise[,dataAnaliseCol]),)
    splitMinority<-dataAnaliseOrder[1:mirority,];
    splitMajority<-dataAnaliseOrder[(mirority+1):(dim(dataAnaliseOrder)[1]),]
  }
```

```
  szLineMinority<-floor(dim(splitMinority)[1])
  szColMinority<-floor(dim(splitMinority)[2])

  szLineMajority<-floor(dim(splitMajority)[1])
  szColMajority<-floor(dim(splitMajority)[2])

  return(splitMinority)
}

SlicesStartMaj<-function(DatasetSelected) {

  dataSplit1<-c();dataSplit2<-c();

  rawdata<-DatasetSelected

  # data
  # dataAnaliseClass

  #-----------------------------------------
  cls<-summary(dataAnaliseClass)
  if(cls[1]<cls[2]){mini<-"0"}
  if(cls[1]>=cls[2]){mini<-"1"}

  library(dplyr)

  if(mini=="0"){
    mirority<-min(cls);
    dataAnaliseOrder<-arrange(dataAnalise, dataAnalise[,dataAnaliseCol],);
    splitMinority<-dataAnaliseOrder[1:mirority,];
    splitMajority<-dataAnaliseOrder[(mirority+1):(dim(dataAnaliseOrder)[1]),]
  }

  if(mini=="1"){
    mirority<-min(cls);
    dataAnaliseOrder<-arrange(dataAnalise, desc(dataAnalise[,dataAnaliseCol]),)
    splitMinority<-dataAnaliseOrder[1:mirority,];
    splitMajority<-dataAnaliseOrder[(mirority+1):(dim(dataAnaliseOrder)[1]),]
  }

  szLineMinority<-floor(dim(splitMinority)[1])
  szColMinority<-floor(dim(splitMinority)[2])

  szLineMajority<-floor(dim(splitMajority)[1])
  szColMajority<-floor(dim(splitMajority)[2])

  return(splitMajority)

}

SlicesJoin<-function(splitMinority,splitMajority){

  splitMinority<-splitMinority[1:nrow(splitMinority),]
  splitMajority<-splitMajority[1:nrow(splitMajority),]
  szLineMinority<-floor(dim(splitMinority)[1])
```

```
  int0<-1; int1<-(floor(szLineMinority/2));
  int2<-(int1+1); int3<-nrow(splitMinority)

  dataSplit1<-rbind(splitMinority[int0:int3,])
  dataSplit2temp<-splitMajority[sample(nrow(splitMajority),replace=F),]
  dataSplit2<-dataSplit2temp[1:int3,]

  dataSplit1<-dataSplit1[sample(nrow(dataSplit1),replace=F),]
  dataSplit2<-dataSplit2[sample(nrow(dataSplit2),replace=F),]

  joinSplits<-rbind(dataSplit1,dataSplit2)

  joinSplits<-joinSplits[sample(nrow(joinSplits),replace=F),]

  return(joinSplits)


}

OutEasyData<-function(dataTemp,vetDificultXPL){
  mtxOutPutData<-cbind(dataTemp,vetDificultXPL)
  zsmtxOutPutData<-ncol(mtxOutPutData)
  dataOrder<-arrange(mtxOutPutData,mtxOutPutData[,zsmtxOutPutData],);
  return(dataOrder)
}

SlicesChange<-function(classOuputdata, dataOrder,rowsOff,mini){
  myxInputNew<-c()

  if(classOuputdata != mini){
    splitMajorityMix<-splitMajority[sample(nrow(splitMajority),replace=F),]
    myxInputNew<-splitMajorityMix[1:rowsOff,]
  }

  if(classOuputdata == mini){
    splitMinorityMix<-splitMinority[sample(nrow(splitMinority),replace=F),]
    myxInputNew<-splitMinorityMix[1:rowsOff,]
  }


  return(myxInputNew)


}

amostragem<-function(){
  classtarget<-c()
  vetPos<-sample(1:dataAnaliseRow, 1, replace=T)
  classtarget<-dataAnalise[vetPos,dataAnaliseCol]
  amostra<-dataAnalise[vetPos,]
  return(amostra)
}

AdjustClass<-function(TempLevel){
```

```
  empLevel<-as.matrix(TempLevel)
  #TempLevel<-as.matrix.data.frame(TempLevel)

  for (i in 1:length(TempLevel)){
    #print(TempLevel[i])
    if(TempLevel[i]=="1"){TempLevel[i]<-"0"}
    if(TempLevel[i]=="2"){TempLevel[i]<-"1"}
    if(TempLevel[i]=="3"){TempLevel[i]<-"2"}
  }

  TempLevel<-as.numeric(TempLevel)

  for (i in 1:length(TempLevel)){
    #print(TempLevel[i])
    if(TempLevel[i]=="tested_positive"){TempLevel[i]<-0}
    if(TempLevel[i]=="tested_negative"){TempLevel[i]<-1}

  }

  TempLevel<-as.numeric(TempLevel)
  return(TempLevel)

}

classifiers<-function(dataSplitx,algorithm){

  dataSplitx<-dataTemp
  data<-scale(dataSplitx[1:ncol(dataSplitx)-1])

  szdata<-dim(data)[2]
  vetclasse<-dataSplitx[,ncol(dataSplitx)]

  NumeroClusters<-2;
  mtxmse<-c();TempLevel<-c();

  szdataLine<-(1000*dim(data)[1])


  if (algorithm==1){
    time.start<-Sys.time();

    fanny.fit<-fanny(as.data.frame(data), NumeroClusters, memb.exp =
(NumeroClusters),metric = "euclidean",maxit = 500)
    TempLevel<-(fanny.fit$clustering)
    outCluster.fanny.fit<-AdjustClass(TempLevel)
    time.stop<-Sys.time();
    totalTime<-(time.stop - time.start)/szdataLine
    MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))
  }

  if (algorithm==2){
    time.start<-Sys.time();

    # Fanny - metric = c("euclidean", "manhattan", "SqEuclidean")
```

```
   fanny.fit<-fanny(as.data.frame(data), NumeroClusters, memb.exp =
NumeroClusters,metric = "manhattan",maxit = 500)
   TempLevel<-(fanny.fit$clustering)
   outCluster.fanny.fit<-AdjustClass(TempLevel)
   time.stop<-Sys.time();
   totalTime<-(time.stop - time.start)/szdataLine
   MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))

 }

 if (algorithm==3){
   time.start<-Sys.time();
   # Fanny - metric = c("euclidean", "manhattan", "SqEuclidean")
   fanny.fit<-fanny(as.data.frame(data), NumeroClusters, memb.exp =
NumeroClusters,metric = "SqEuclidean",maxit = 500)
   TempLevel<-(fanny.fit$clustering)
   outCluster.fanny.fit<-AdjustClass(TempLevel)
   time.stop<-Sys.time();
   totalTime<-(time.stop - time.start)/szdataLine
   MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))
 }

 if (algorithm==4){
   time.start<-Sys.time();
   #pam - metric = c("euclidean", "manhattan")
   pam.fit<-pam(as.data.frame(data), NumeroClusters, diss = inherits(data,
"dist"),metric = "euclidean")
   TempLevel<-(pam.fit$clustering)
   outCluster.fanny.fit<-AdjustClass(TempLevel)

   time.stop<-Sys.time();
   totalTime<-(time.stop - time.start)/szdataLine
   MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))


 }

 if (algorithm==5){
   time.start<-Sys.time();
   pam.fit<-pam(as.data.frame(data), NumeroClusters, diss = inherits(data,
"dist"),metric = "manhattan")
   TempLevel<-(pam.fit$clustering)
   outCluster.fanny.fit<-AdjustClass(TempLevel)

   time.stop<-Sys.time();
   totalTime<-(time.stop - time.start)/szdataLine
   MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))
 }

 if (algorithm==6){
   time.start<-Sys.time();
   #clara - metric = c("euclidean", "manhattan", "jaccard")
   clara.fit<-clara(as.data.frame(data), NumeroClusters, metric ="euclidean")
   TempLevel<-(clara.fit$clustering)
   outCluster.fanny.fit<-AdjustClass(TempLevel)
```

```
  time.stop<-Sys.time();
  totalTime<-(time.stop - time.start)/szdataLine
  MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))
 }

 if (algorithm==7){
  time.start<-Sys.time();
  clara.fit<-clara(as.data.frame(data), NumeroClusters, metric ="manhattan")
  TempLevel<-(clara.fit$clustering)
  outCluster.fanny.fit<-AdjustClass(TempLevel)
  time.stop<-Sys.time();
  totalTime<-(time.stop - time.start)/szdataLine
  MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))
 }

 if (algorithm==8){
  time.start<-Sys.time();
  clara.fit<-clara(as.data.frame(data), NumeroClusters, metric ="jaccard")
  TempLevel<-(clara.fit$clustering)
  outCluster.fanny.fit<-AdjustClass(TempLevel)
  time.stop<-Sys.time();
  totalTime<-(time.stop - time.start)/szdataLine
  MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))
 }

 if (algorithm==9){
  time.start<-Sys.time();
  #sota
  sota.fit<-sota(as.matrix(data),NumeroClusters-1)
  TempLevel<-(sota.fit$clust)
  outCluster.fanny.fit<-AdjustClass(TempLevel)
  time.stop<-Sys.time();
  totalTime<-(time.stop - time.start)/szdataLine
  MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))
 }

 if (algorithm==10){
  time.start<-Sys.time();
  k.means.fit <- kmeans(as.data.frame(data), NumeroClusters, algorithm =
"Hartigan-Wong")
  TempLevel<-(k.means.fit$cluster)
  outCluster.fanny.fit<-AdjustClass(TempLevel)
  time.stop<-Sys.time();
  totalTime<-(time.stop - time.start)/szdataLine
  MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))
 }

 if (algorithm==11){
  time.start<-Sys.time();
  k.means.fit <- kmeans(as.data.frame(data), NumeroClusters, algorithm =
"Lloyd")
  TempLevel<-(k.means.fit$cluster)
  outCluster.fanny.fit<-AdjustClass(TempLevel)
  time.stop<-Sys.time();
  totalTime<-(time.stop - time.start)/szdataLine
```

```
  MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))
 }

 if (algorithm==12){
  time.start<-Sys.time();
  #clusplot(data, k.means.fit$cluster, main='Gr?fico dos grupos de kmeans',
color=TRUE, shade=FALSE,labels=2, lines=0)
  k.means.fit <- kmeans(as.data.frame(data), NumeroClusters, algorithm =
"Forgy")
  TempLevel<-(k.means.fit$cluster)
  outCluster.fanny.fit<-AdjustClass(TempLevel)
  time.stop<-Sys.time();
  totalTime<-(time.stop - time.start)/szdataLine
  MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))
 }

 if (algorithm==13){
  time.start<-Sys.time();
  k.means.fit <- kmeans(as.data.frame(data), NumeroClusters, algorithm =
"MacQueen")
  TempLevel<-(k.means.fit$cluster)
  outCluster.fanny.fit<-AdjustClass(TempLevel)
  time.stop<-Sys.time();
  totalTime<-(time.stop - time.start)/szdataLine
  MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))
 }

 if (algorithm==14){
  time.start<-Sys.time();
  #Diana
  g14<-diana(data,metric = "euclidean", stand = TRUE) # default: Euclidean
  TempLevel<-cutree(as.hclust(g14), k = 2)
  #source("Adjustlevels.R")
  outCluster.fanny.fit<-AdjustClass(TempLevel)
  time.stop<-Sys.time();
  totalTime<-(time.stop - time.start)/szdataLine
  MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))

 }

 if (algorithm==15){
  # KMeans_arma
  library(ClusterR)
  time.start<-Sys.time();
  g15<-KMeans_arma(data, clusters = 2, n_iter = 10, seed_mode =
"random_subset", verbose = T, CENTROIDS = NULL)
  TempLevel<-predict_KMeans(data , g15)
  #source("Adjustlevels.R")
  outCluster.fanny.fit<-AdjustClass(TempLevel)
  time.stop<-Sys.time();
  totalTime<-(time.stop - time.start)/szdataLine
  MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))

 }
```

```
  if (algorithm==16){
    # KMeans_rcpp
    time.start<-Sys.time();
    g16<-KMeans_rcpp(data, clusters = 2)#, num_init = 5, max_iters = 10, initializer
= 'optimal_init', threads = 1, verbose = F)
    TempLevel<-g16$clusters
    #source("Adjustlevels.R")
    outCluster.fanny.fit<-AdjustClass(TempLevel)
    time.stop<-Sys.time();
    totalTime<-(time.stop - time.start)/szdataLine
    MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))

  }

  if (algorithm==17){
    # MiniBatchKmeans
    time.start<-Sys.time();
    g17<-MiniBatchKmeans(data, clusters = 2, batch_size = 10, num_init = 5,
max_iters = 100, init_fraction = 0.2, initializer = 'kmeans++', early_stop_iter = 10,
verbose = F)
    TempLevel<-predict_MBatchKMeans(data, g17$centroids)
    #source("Adjustlevels.R")
    outCluster.fanny.fit<-AdjustClass(TempLevel)
    time.stop<-Sys.time();
    totalTime<-(time.stop - time.start)/szdataLine
    MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))

  }

  if (algorithm==18){
    time.start<-Sys.time();
    g18<-MiniBatchKmeans(data, clusters = 2, batch_size = 10, num_init = 5,
max_iters = 100, init_fraction = 0.2, initializer = 'random', early_stop_iter = 10,
verbose = F)
    TempLevel<-predict_MBatchKMeans(data, g18$centroids)
    #source("Adjustlevels.R")
    outCluster.fanny.fit<-AdjustClass(TempLevel)
    time.stop<-Sys.time();
    totalTime<-(time.stop - time.start)/szdataLine
    MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))

  }

  if (algorithm==19){
    time.start<-Sys.time();
    g19<-MiniBatchKmeans(data, clusters = 2, batch_size = 10, num_init = 5,
max_iters = 100, init_fraction = 0.2, initializer = 'optimal_init', early_stop_iter = 10,
verbose = F)
    TempLevel<-predict_MBatchKMeans(data, g19$centroids)
    #source("Adjustlevels.R")
    outCluster.fanny.fit<-AdjustClass(TempLevel)
    time.stop<-Sys.time();
    totalTime<-(time.stop - time.start)/szdataLine
    MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))
```

```
 }

 if (algorithm==20){
  time.start<-Sys.time();
  g20<-MiniBatchKmeans(data, clusters = 2, batch_size = 10, num_init = 5,
max_iters = 100, init_fraction = 0.2, initializer = 'quantile_init', early_stop_iter =
10, verbose = F)
  TempLevel<-predict_MBatchKMeans(data, g20$centroids)
  #source("Adjustlevels.R")
  outCluster.fanny.fit<-AdjustClass(TempLevel)
  time.stop<-Sys.time();
  totalTime<-(time.stop - time.start)/szdataLine
  MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))

 }

 if (algorithm==21){
  # Fuzzy C-means
  library("e1071")
  time.start<-Sys.time();
  g21<- cmeans(data, centers = 2,dist='euclidean')
  TempLevel<-g21$cluster
  #source("Adjustlevels.R")
  outCluster.fanny.fit<-AdjustClass(TempLevel)
  time.stop<-Sys.time();
  totalTime<-(time.stop - time.start)/szdataLine
  MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))

 }

 if (algorithm==22){
  # Fuzzy C-means
  library("e1071")
  time.start<-Sys.time();
  g22<- cmeans(data, centers = 2,dist='manhattan')
  TempLevel<-g22$cluster
  #source("Adjustlevels.R")
  outCluster.fanny.fit<-AdjustClass(TempLevel)
  time.stop<-Sys.time();
  totalTime<-(time.stop - time.start)/szdataLine
  MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))

 }

 if (algorithm==23){
  # Cluster_Medoids
  time.start<-Sys.time();
  g23<-Cluster_Medoids(data, clusters = 2, swap_phase = TRUE, verbose =
F,distance_metric='euclidean')
  TempLevel<-g23$clusters
  #source("Adjustlevels.R")
  outCluster.fanny.fit<-AdjustClass(TempLevel)
  time.stop<-Sys.time();
  totalTime<-(time.stop - time.start)/szdataLine
  MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))
```

```
 }

 if (algorithm==24){
   # Cluster_Medoids
   time.start<-Sys.time();
   g24<-Cluster_Medoids(data, clusters = 2, swap_phase = TRUE, verbose =
F,distance_metric='manhattan')
   TempLevel<-g24$clusters
   #source("Adjustlevels.R")
   outCluster.fanny.fit<-AdjustClass(TempLevel)
   time.stop<-Sys.time();
   totalTime<-(time.stop - time.start)/szdataLine
   MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))

 }

 if (algorithm==25){
   # Cluster_Medoids
   time.start<-Sys.time();
   g25<-Cluster_Medoids(data, clusters = 2, swap_phase = TRUE, verbose =
F,distance_metric='chebyshev')
   TempLevel<-g25$clusters
   #source("Adjustlevels.R")
   outCluster.fanny.fit<-AdjustClass(TempLevel)
   time.stop<-Sys.time();
   totalTime<-(time.stop - time.start)/szdataLine
   MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))

 }

 if (algorithm==26){
   # Cluster_Medoids
   time.start<-Sys.time();
   g26<-Cluster_Medoids(data, clusters = 2, swap_phase = TRUE, verbose =
F,distance_metric='canberra')
   TempLevel<-g26$clusters
   #source("Adjustlevels.R")
   outCluster.fanny.fit<-AdjustClass(TempLevel)
   time.stop<-Sys.time();
   totalTime<-(time.stop - time.start)/szdataLine
   MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))

 }

 if (algorithm==27){
   time.start<-Sys.time();
   g27<-Cluster_Medoids(data, clusters = 2, swap_phase = TRUE, verbose =
F,distance_metric='braycurtis')
   TempLevel<-g27$clusters
   #source("Adjustlevels.R")
   outCluster.fanny.fit<-AdjustClass(TempLevel)
   time.stop<-Sys.time();
   totalTime<-(time.stop - time.start)/szdataLine
   MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))
```

```
  }

 if (algorithm==28){
  time.start<-Sys.time();
  g28<-Cluster_Medoids(data, clusters = 2, swap_phase = TRUE, verbose =
F,distance_metric='pearson_correlation')
  TempLevel<-g28$clusters
  #source("Adjustlevels.R")
  outCluster.fanny.fit<-AdjustClass(TempLevel)
  time.stop<-Sys.time();
  totalTime<-(time.stop - time.start)/szdataLine
  MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))

 }

 if (algorithm==29){
  time.start<-Sys.time();
  g29<-Cluster_Medoids(data, clusters = 2, swap_phase = TRUE, verbose =
F,distance_metric='simple_matching_coefficient')
  TempLevel<-g29$clusters
  #source("Adjustlevels.R")
  outCluster.fanny.fit<-AdjustClass(TempLevel)
  time.stop<-Sys.time();
  totalTime<-(time.stop - time.start)/szdataLine
  MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))

 }

 if (algorithm==30){
  time.start<-Sys.time();
  g30<-Cluster_Medoids(data, clusters = 2, swap_phase = TRUE, verbose =
F,distance_metric='minkowski')
  TempLevel<-g30$clusters
  #source("Adjustlevels.R")
  outCluster.fanny.fit<-AdjustClass(TempLevel)
  time.stop<-Sys.time();
  totalTime<-(time.stop - time.start)/szdataLine
  MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))

 }

 if (algorithm==31){
  time.start<-Sys.time();
  g31<-Cluster_Medoids(data, clusters = 2, swap_phase = TRUE, verbose =
F,distance_metric='hamming')
  TempLevel<-g31$clusters
  #source("Adjustlevels.R")
  outCluster.fanny.fit<-AdjustClass(TempLevel)
  time.stop<-Sys.time();
  totalTime<-(time.stop - time.start)/szdataLine
  MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))

 }

 if (algorithm==32){
```

```
   time.start<-Sys.time();
   g32<-Cluster_Medoids(data, clusters = 2, swap_phase = TRUE, verbose =
F,distance_metric='jaccard_coefficient')
   TempLevel<-g32$clusters
   #source("Adjustlevels.R")
   outCluster.fanny.fit<-AdjustClass(TempLevel)
   time.stop<-Sys.time();
   totalTime<-(time.stop - time.start)/szdataLine
   MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))

 }

 if (algorithm==33){
   time.start<-Sys.time();
   g33<-Cluster_Medoids(data, clusters = 2, swap_phase = TRUE, verbose =
F,distance_metric='Rao_coefficient')
   TempLevel<-g33$clusters
   #source("Adjustlevels.R")
   outCluster.fanny.fit<-AdjustClass(TempLevel)
   time.stop<-Sys.time();
   totalTime<-(time.stop - time.start)/szdataLine
   MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))

 }

 if (algorithm==34){
   time.start<-Sys.time();
   g34<-Cluster_Medoids(data, clusters = 2, swap_phase = TRUE, verbose =
F,distance_metric='mahalanobis')
   TempLevel<-g34$clusters
   outCluster.fanny.fit<-AdjustClass(TempLevel)
   time.stop<-Sys.time();
   totalTime<-(time.stop - time.start)/szdataLine
   MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))

 }

 if (algorithm==35){
   time.start<-Sys.time();
   g35<-Cluster_Medoids(data, clusters = 2, swap_phase = TRUE, verbose =
F,distance_metric='cosine')
   TempLevel<-g35$clusters
   #source("Adjustlevels.R")
   outCluster.fanny.fit<-AdjustClass(TempLevel)
   time.stop<-Sys.time();
   totalTime<-(time.stop - time.start)/szdataLine
   MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))

 }

 if (algorithm==36){
   time.start<-Sys.time();
   v <- inaparc::kmpp(data, k=2)$v
   u <- inaparc::imembrand(nrow(data), k=2)$u
   gk.res <- gk(data, centers=v, memberships=u, m=2)
```

```
  head(gk.res$u, 5)
  TempLevel<-gk.res$cluster
  outCluster.fanny.fit<-AdjustClass(TempLevel)
  time.stop<-Sys.time();
  totalTime<-(time.stop - time.start)/szdataLine
  MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))

}

if (algorithm==37){
  time.start<-Sys.time();
  set.seed(1)
  res.mocca<-mocca(data,R=10,K=2, nstart = 100)
  TempLevel<-res.mocca$cluster$kmeans[[2]][[10]] #k-means
  outCluster.fanny.fit<-AdjustClass(TempLevel)
  time.stop<-Sys.time();
  totalTime<-(time.stop - time.start)/szdataLine
  MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))


}

if (algorithm==38){
  time.start<-Sys.time();
  set.seed(1)
  res.mocca<-mocca(data,R=10,K=2, nstart = 100)
  TempLevel<-res.mocca$cluster$neuralgas[[2]][[10]] #neuralgas cluster
  outCluster.fanny.fit<-AdjustClass(TempLevel)
  time.stop<-Sys.time();
  totalTime<-(time.stop - time.start)/szdataLine
  MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))

}

if (algorithm==39){
  time.start<-Sys.time();
  #set.seed(1)
  optics_res <- optics(data, minPts = 2)#,eps = 2)
  TempLevel <- extractDBSCAN(optics_res, eps_cl = .065)
  outCluster.fanny.fit<-TempLevel$cluster
  time.stop<-Sys.time();
  totalTime<-(time.stop - time.start)/szdataLine
  MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))

}

if (algorithm==40){
  time.start<-Sys.time();
  dbs <- dbscan(data, minPts=2, eps=25)
  outCluster.fanny.fit<-dbs$cluster
  time.stop<-Sys.time();
  totalTime<-(time.stop - time.start)/szdataLine
  MtxTimesClustes<-(rbind(MtxTimesClustes,totalTime))

}
```

```
  return(c(outCluster.fanny.fit,totalTime))

}

CalcMSE<-function(vetTarget, vetPredict,metric){

  # Calculate metrics
  if(metric==1){msevalue<-mse(vetTarget, vetPredict)}
  if(metric==2){msevalue<-EvaluationMeasures.Accuracy(vetTarget, vetPredict)}
  if(metric==3){msevalue<-EvaluationMeasures.Sensitivity(vetTarget, vetPredict)}
  if(metric==4){msevalue<-EvaluationMeasures.Specificity(vetTarget, vetPredict)}
  if(metric==5){msevalue<-roc(vetTarget, vetPredict)}
  return(msevalue)
}

CalcMASE<-function(vetTarget2, vetPredict){
  #vetTargetMAPE<-AdjustClass(vetTarget2)
  #vetTargetMAPE2<-as.integer(vetTargetMAPE)
 mapevalue<-mase(vetTarget2, vetPredict)
  return(mapevalue)
}

funcTime<-function(time.start,data){

  szdataLine<-dim(data)[1]

  time.stop<-Sys.time();
  totalTime<-(time.stop - time.start)/szdataLine
  return(totalTime)
}

compareAll<-function(data001){

  countRight<-0;vetRight<-0;
  szdata001L<-dim(data001)[1]
  szdata001C<-dim(data001)[2]

  for(stepdata001L in 1:szdata001L){
    countRight<-0;
    for(stepdata001C in 2:szdata001C){
      classCurrent<-data001[stepdata001L,1]
      if((classCurrent)==data001[stepdata001L,stepdata001C]){countRight<-
countRight+1}
    }
    vetRight<-rbind(vetRight,countRight)


  }
  vetRight<-rbind(vetRight,"end")
  return(vetRight)


}
```

```
funcEstatistic<-function(mtxTimes){
  mx<-max(mtxTimes)
  mi<-min(mtxTimes)
  md<-mean(mtxTimes)
  dp<-sd(mtxTimes)
  cv<-(dp/md)*100
  vettime1<-cbind(mx,mi,dp,md,cv)
  return(vettime1)
}

calcDificult1PL<-function(dataDif){

  vetdata1<-as.data.frame(dataTemp[,ncol(dataTemp)])
  sizevet<-dim(vetdata1)[1]
  dataDif<-vetdata1
  dataset1 <- simdata(dataDif, vetPredict, sizevet, itemtype = 'dich') # change
numeric to categorical

  mod1 <- mirt(dataset1, 1, method = 'EM',itemtype='Rasch',technical =
list(NCYCLES = sizevet), GenRandomPars = TRUE)
  coefModel<-coef(mod1,simplify=TRUE, IRTpars=TRUE)
  coefModel1<-as.data.frame(coefModel[1])
  dificultModel1<-coefModel1[,2]

  TethaResult(mod1,SelectDifficult,algorithm,stepMoment)

  return(dificultModel1)

}

calcDificult2PL<-function(dataDif){

  vetdata1<-as.data.frame(dataTemp[,ncol(dataTemp)])
  sizevet<-dim(vetdata1)[1]
  dataDif<-vetdata1
  dataset1 <- simdata(dataDif, vetPredict, sizevet, itemtype = 'dich') # change
numeric to categorical

  mod2 <- mirt(dataset1, 1, method = 'EM',itemtype='2PL',technical =
list(NCYCLES = sizevet), GenRandomPars = TRUE)
  coefModel<-coef(mod2,simplify=TRUE, IRTpars=TRUE)
  coefModel2<-as.data.frame(coefModel[1])
  dificultModel2<-coefModel2[,2]

  TethaResult(mod2,SelectDifficult,algorithm,stepMoment)

  return(dificultModel2)

}

calcDificult3PL<-function(dataDif){

  vetdata1<-as.data.frame(dataTemp[,ncol(dataTemp)])
  sizevet<-dim(vetdata1)[1]
  dataDif<-vetdata1
```

```
  dataset1 <- simdata(dataDif, vetPredict, sizevet, itemtype = 'dich') # change
numeric to categorical

  mod3 <- mirt(dataset1, 1, method = 'EM',itemtype='3PL',technical =
list(NCYCLES = sizevet))#, GenRandomPars = TRUE)
  coefModel<-coef(mod3,simplify=TRUE, IRTpars=TRUE)
  coefModel3<-as.data.frame(coefModel[1])
  dificultModel3<-coefModel3[,2]
  BCdificultModel3<-coefModel3[,2:3]

  TethaResult(mod3,SelectDifficult,algorithm,stepMoment)

  return(coefModel3)

}

funcTime<-function(time.start,data){

  szdataLine<-dim(data)[1]

  time.stop<-Sys.time();
  totalTime<-(time.stop - time.start)/szdataLine
  return(totalTime)
}

plotG1<-function(vetDifficult,metric){

  legend_title <- "PL?s Models"
  legendsModels<-c("1PL","2PL", "3PL")
  gg<-(vetDifficult)
  gg<-as.data.frame(t(gg))
  turns<-dim(gg)[1]
  mtrik<-c()
  if(metric==1){df<-ggplot(gg,aes(x = 1:turns,y =
Precisão),group=legendsModels)+
     geom_line(aes(y = gg[,1]), color = "red")+
     geom_line(aes(y = gg[,2]), color = "green")+
     geom_line(aes(y = gg[,3]), color = "blue")+ scale_x_continuous(limits = c(2,
turns))#+ scale_y_continuous(limits = c(0, 0.15))

    df1<-df + theme(
     legend.position = c(.95, .95),
     legend.justification = c("right", "top"),
     legend.box.just = "right",
     legend.margin = margin(6, 6, 6, 6)
    )
  }
  if(metric==2){df<-ggplot(gg,aes(x = 1:turns,y =
Sensibilidade),group=legendsModels)+
    geom_line(aes(y = gg[,1]), color = "red")+
    geom_line(aes(y = gg[,2]), color = "green")+
    geom_line(aes(y = gg[,3]), color = "blue")+ scale_x_continuous(limits = c(2,
turns))#+ scale_y_continuous(limits = c(0, 0.15))

  df1<-df + theme(
```

```
    legend.position = c(.95, .95),
    legend.justification = c("right", "top"),
    legend.box.just = "right",
    legend.margin = margin(6, 6, 6, 6)
  )
  }
  if(metric==3){df<-ggplot(gg,aes(x = 1:turns,y =
Especificidade),group=legendsModels)+
    geom_line(aes(y = gg[,1]), color = "red")+
    geom_line(aes(y = gg[,2]), color = "green")+
    geom_line(aes(y = gg[,3]), color = "blue")+ scale_x_continuous(limits = c(2,
turns))#+ scale_y_continuous(limits = c(0, 0.15))

  df1<-df + theme(
    legend.position = c(.95, .95),
    legend.justification = c("right", "top"),
    legend.box.just = "right",
    legend.margin = margin(6, 6, 6, 6)
  )

    mtrik<-""}
  if(metric==4){df<-ggplot(gg,aes(x = 1:turns,y = ""),group=legendsModels)+
    geom_line(aes(y = gg[,1]), color = "red")+
    geom_line(aes(y = gg[,2]), color = "green")+
    geom_line(aes(y = gg[,3]), color = "blue")+ scale_x_continuous(limits = c(2,
turns))#+ scale_y_continuous(limits = c(0, 0.15))

  df1<-df + theme(
    legend.position = c(.95, .95),
    legend.justification = c("right", "top"),
    legend.box.just = "right",
    legend.margin = margin(6, 6, 6, 6)
  )
  }
  if(metric==5){
  df<-ggplot(gg,aes(x = 1:turns,y = Rand),group=legendsModels)+
    #geom_line(aes(y = gg[,1]), color = "red")+
    geom_line(aes(y = gg[,2]), color = "green")+
    geom_line(aes(y = gg[,3]), color = "blue")+ scale_x_continuous(limits = c(2,
turns))#+ scale_y_continuous(limits = c(0, 0.15))

  df1<-df + theme(
    legend.position = c(.95, .95),
    legend.justification = c("right", "top"),
    legend.box.just = "right",
    legend.margin = margin(6, 6, 6, 6)
  )
  }
  if(metric==6){
    df<-ggplot(gg,aes(x = 1:turns,y = Jaccard),group=legendsModels)+
      #geom_line(aes(y = gg[,1]), color = "red")+
      geom_line(aes(y = gg[,2]), color = "green")+
      geom_line(aes(y = gg[,3]), color = "blue")+ scale_x_continuous(limits = c(2,
turns))#+ scale_y_continuous(limits = c(0, 0.15))
```

```
   df1<-df + theme(
     legend.position = c(.95, .95),
     legend.justification = c("right", "top"),
     legend.box.just = "right",
     legend.margin = margin(6, 6, 6, 6)
   )
  }
  if(metric==7){df<-ggplot(gg,aes(x = 1:turns,y =
Folkes_Mallows),group=legendsModels)+
     #geom_line(aes(y = gg[,1]), color = "red")+
     geom_line(aes(y = gg[,2]), color = "green")+
     geom_line(aes(y = gg[,3]), color = "blue")+ scale_x_continuous(limits = c(2,
turns))#+ scale_y_continuous(limits = c(0, 0.15))

   df1<-df + theme(
     legend.position = c(.95, .95),
     legend.justification = c("right", "top"),
     legend.box.just = "right",
     legend.margin = margin(6, 6, 6, 6)
   )
  }
  if(metric==8){df<-ggplot(gg,aes(x = 1:turns,y = Dunn),group=legendsModels)+
     #geom_line(aes(y = gg[,1]), color = "red")+
     geom_line(aes(y = gg[,2]), color = "green")+
     geom_line(aes(y = gg[,3]), color = "blue")+ scale_x_continuous(limits = c(2,
turns))#+ scale_y_continuous(limits = c(0, 0.15))

   df1<-df + theme(
     legend.position = c(.95, .95),
     legend.justification = c("right", "top"),
     legend.box.just = "right",
     legend.margin = margin(6, 6, 6, 6)
   )
  }
  if(metric==9){
   df<-ggplot(gg,aes(x = 1:turns,y = SIlhouette),group=legendsModels)+
    # geom_line(aes(y = gg[,1]), color = "red")+
     geom_line(aes(y = gg[,2]), color = "green")+
     geom_line(aes(y = gg[,3]), color = "blue")+ scale_x_continuous(limits = c(2,
turns))#+ scale_y_continuous(limits = c(0, 0.15))

   df1<-df + theme(
     legend.position = c(.95, .95),
     legend.justification = c("right", "top"),
     legend.box.just = "right",
     legend.margin = margin(6, 6, 6, 6)
   )
  }
  if(metric==10){
   df<-ggplot(gg,aes(x = 1:turns,y = Davies_Bouldin),group=legendsModels)+
     #geom_line(aes(y = gg[,1]), color = "red")+
     geom_line(aes(y = gg[,2]), color = "green")+
     geom_line(aes(y = gg[,3]), color = "blue")+ scale_x_continuous(limits = c(2,
turns))#+ scale_y_continuous(limits = c(0, 0.15))
```

```
    df1<-df + theme(
      legend.position = c(.95, .95),
      legend.justification = c("right", "top"),
      legend.box.just = "right",
      legend.margin = margin(6, 6, 6, 6)
    )
  }
  if(metric==11){
    df<-ggplot(gg,aes(x = 1:turns,y = PBM),group=legendsModels)+
     # geom_line(aes(y = gg[,1]), color = "red")+
      geom_line(aes(y = gg[,2]), color = "green")+
      geom_line(aes(y = gg[,3]), color = "blue")+ scale_x_continuous(limits = c(2,
turns))#+ scale_y_continuous(limits = c(0, 0.15))

    df1<-df + theme(
      legend.position = c(.95, .95),
      legend.justification = c("right", "top"),
      legend.box.just = "right",
      legend.margin = margin(6, 6, 6, 6)
    )
  }

  return(print(df1))

  #scale_fill_discrete(name = "PLs", labels = c("1PL", "2PL", "3PL"))
}

funcGraphic<-function(vetDifficult){
  nameGrafico<-paste0('Dataset',DatasetCorrente,"Alg",algorithm,".tiff");
  tituloGrafico<-paste0('Dataset',DatasetCorrente,"Alg",algorithm,".tiff");

  setwd(path); getwd();

  tiff(nameGrafico, units="in", width=5, height=5, res=300)
  setwd(paste0(path,"/Splits/Graficos")); getwd();
  plotG1(vetDifficult,metric)
  dev.off()

  setwd("c:/TRICluster-out/"); getwd();
}

reduceData<-function(DatasetSelected){

  newExemplesByClass<-24 # 24 exemples by dataset of original class balanced

  #lmitClass<-dim(DatasetSelected)[1]/2
  lmitClass<-newExemplesByClass

  fistSplitTemp<-SlicesStartMin(DatasetSelected)
  fistSplit<-fistSplitTemp[1:lmitClass,]
  secondSplitTemp<-SlicesStartMaj(DatasetSelected)
  secondSplit<-secondSplitTemp[1:lmitClass,]

  dim(fistSplit)
  dim(secondSplit)
```

```
  DatasetSelected<-rbind(fistSplit,secondSplit)

  DatasetSelected<-DatasetSelected[sample(nrow(DatasetSelected),replace=F),]

  return(DatasetSelected)

}

balancedDataX<-function(DatasetSelected,DatasetCorrente){

  if(DatasetCorrente=="1"){multiplo<-5}
  if(DatasetCorrente=="3"){multiplo<-0.5}
  if(DatasetCorrente=="4"){multiplo<-2}
  if(DatasetCorrente=="5"){multiplo<-5}
  if(DatasetCorrente=="13"){multiplo<-4}
  if(DatasetCorrente=="14"){multiplo<-2}
  if(DatasetCorrente=="16"){multiplo<-0.5}
  if(DatasetCorrente=="17"){multiplo<-0.5}
  if(DatasetCorrente=="19"){multiplo<-0.5}
  if(DatasetCorrente=="20"){multiplo<-2}
  if(DatasetCorrente=="21"){multiplo<-0.5}
  if(DatasetCorrente=="24"){multiplo<-0.5}

  dataX<-DatasetSelected[,1:(ncol(DatasetSelected)-1)]
  dataClassX<-as.factor(DatasetSelected[,(ncol(DatasetSelected))])
  Balanceded<-SMOTE(dataX, dataClassX, K=2, dup_size = multiplo)
  cls1<-summary(dataAnaliseClass);cls1
  cls2<-
summary(as.factor(Balanceded$syn_data[,ncol(Balanceded$syn_data)]));cls2

  dim(DatasetSelected)
  dim(Balanceded$data)

  oversampled<-Balanceded$data
  dataY<-oversampled[,1:(ncol(oversampled)-1)]
  #datajoin<-rbind(dataX[1:nrow(dataX)-1,],dataY[1:nrow(dataY)-1,])
  datajoin<-rbind(dataX,dataY)

  dataClassY<-as.factor(oversampled[,(ncol(oversampled))])
  dataClassJoin<-c(dataClassX,dataClassY)

  mode(dataClassY)

  mode(dataClassX)

  newDataBalanced<-cbind(datajoin,dataClassJoin)

  return(newDataBalanced)

}

PercentData<-function(DatasetSelected,Percent){

  # By percentage
```

```
  szsplitMinority<-dim(splitMinority)[1]
  szsplitMajority<-dim(splitMajority)[1]

  newszsplitMinority<-floor(szsplitMinority*(Percent/100))
  newszsplitMajority<-floor(szsplitMinority*(Percent/100))

  #fistSplit<-splitMinority[1:newszsplitMinority,1:(ncol(splitMinority)-1)]
  #secondSplit<-splitMajority[1:newszsplitMinority,1:(ncol(splitMajority)-1)]

  fistSplit<-splitMinority[1:newszsplitMinority,1:(ncol(splitMajority))]
  secondSplit<-splitMajority[1:newszsplitMinority,1:(ncol(splitMajority))]


  dim(fistSplit)
  dim(secondSplit)

  DatasetSelected<-rbind(fistSplit,secondSplit)

  DatasetSelected<-DatasetSelected[sample(nrow(DatasetSelected),replace=F),]

  return(DatasetSelected)

}

SaveParameters<-function(dataTemp,meandificultBest,vetDificultPL){

  setwd(path); getwd();

  namevetDificultPL<-
paste0(path,"/Splits/Mensure/Best/","vetDificult1PL",'Dataset',DatasetCorrente,"a
lgorithm",algorithm,SelectDifficult,"PL",".csv")
  write.csv(vetDificultPL,namevetDificultPL)

  namemeandificultBest<-
paste0(path,"/Splits/Mensure/Best/","meandificultBest",'Dataset',DatasetCorrent
e,"algorithm",algorithm,SelectDifficult,"PL",".xlsx")
  vetDificultX<-vetDificultPL
  vetDificultPLOrd<-sort(as.matrix(vetDificultX));

  vetOut<-
c("min",min(vetDificultPLOrd),"media",meandificultBest,"maximo",max(vetDificu
ltPLOrd),
          "Desvio-padr?o",sd(vetDificultPLOrd),"Coeficiente de
variacao",(sd(vetDificultPLOrd)/meandificultBest)*100)

  write.csv(vetOut,namemeandificultBest)

  nameBestSet<-
paste0(path,"/Splits/Mensure/Best/","MtxBestSet",'Dataset',DatasetCorrente,"alg
orithm",algorithm,SelectDifficult,"PL",".csv")
  write.csv(dataTemp,nameBestSet)

  setwd("C:/TRICluster-out/"); getwd();
}
```

```
metricsByTRI<-function(vetTarget,vetPredict,dataNorm,switchType){

  vetTargetOut<-vetTarget#AdjustClass(vetTarget)
  if(switchType=="3"){vetTargetOut<-AdjustClass(vetTarget)}
  vetPredict<-vetPredict[1:length(vetPredict)-1]

  p1<-as.numeric(vetPredict)
  p2<-as.numeric(vetTargetOut)
  C1<-as.integer(dataNorm)
  C2<-as.integer(vetPredict)
  D1<-as.matrix(dataNorm)
  D2<-as.integer(vetPredict)

  vetMeansure<-c()
  #mse <- mse(p1,p2)
  #prec<- EvaluationMeasures.Precision(p1,p2)
  acc <- EvaluationMeasures.Accuracy(p1,p2)
  sen <- EvaluationMeasures.Sensitivity(p1,p2)
  spe <- EvaluationMeasures.Specificity(p1,p2)
  IRand <- extCriteria(C1,D2,"Rand")
  IJac <- extCriteria(C1,C2, "Jaccard")
  IFM <- extCriteria(C1,C2,"Folkes_Mallows")

  IDunn <- intCriteria(D1,C2,"Dunn")
  ISil <- intCriteria(D1,C2,"Silhouette")
  IDB <-intCriteria(D1,C2,"Davies_Bouldin")
  IPBM <-intCriteria(D1,C2,"PBM")
  #roc <- roc(p1,p2)

  titlevetMeansure<-c("acc","sen","spe", "IRand", "Ijac", "IFM", "IDunn", "ISil",
"IDB", "IPBM")
                  #"roc")
  vetMeansure<-cbind(acc,sen,spe,IRand,IJac,IFM,IDunn,ISil,IDB,IPBM)
  #vetMeansure<-c(mse,prec,acc,sen,spe)

  return(vetMeansure)

}

saveVetDown<-function(vetDown,SelectDifficult,vetDifficult){

  nameVetDown<-
paste0(path,'/Splits/Mensure/PointsToPlot/','vetDownDataset',DatasetCorrente,"A
lg",algorithm,
                  'SelectDifficult',SelectDifficult,".csv");
  write.csv(vetDown,nameVetDown)

}

savevetDifficult<-
function(vetDown,SelectDifficult,vetDifficult,stepMoment,vetMSEAll,vetMASEAll,
vetMetricsStepAll,mtxPredict,mtxvetDificultPL,vetDificultPL){

  setwd(path); getwd();
```

```
  namevetDifficult<-
paste0(path,'/Splits/vetDifficult/','vetDifficult',DatasetCorrente,"D",DatasetCorrent
e,"Alg",algorithm,
                    'All3Difficult',stepMoment,".csv");
  write.csv(vetDifficult,namevetDifficult)


  namevetMSEAll<-
paste0(path,'/Splits/vetMSEAll/','vetMSEAll',"D",DatasetCorrente,"Alg",algorithm,
                  'Difficult',SelectDifficult,"_",stepMoment,".csv");
  write.csv(vetMSEAll,namevetMSEAll)


  namevetMASEAll<-
paste0(path,'/Splits/vetMASEAll/','vetMASEAll',"D",DatasetCorrente,"Alg",algorit
hm,
                  'Difficult',SelectDifficult,"_",stepMoment,".csv");
  write.csv(vetMASEAll,namevetMASEAll)


  namevetMetricsStepAll<-
paste0(path,'/Splits/vetMetricsStepAll/','vetMetricsStepAll',"D",DatasetCorrente,"
Alg",algorithm,
                      'Difficult',SelectDifficult,"_",stepMoment,".csv");
  write.csv(vetMetricsStepAll,namevetMetricsStepAll)


  namemtxPredict<-
paste0(path,'/Splits/Classifications/','Classifications',"D",DatasetCorrente,"Alg",a
lgorithm,
                      'Difficult',SelectDifficult,"_",stepMoment,".csv");
  write.csv(mtxPredict,namemtxPredict)


  namemtxvetDificultPL<-
paste0(path,'/Splits/mtxvetDificult1PL/','mtxvetDificult1PL',"D",DatasetCorrente,"
Alg",algorithm,
                  'Difficult',SelectDifficult,"_",stepMoment,".csv");
  write.csv(mtxvetDificultPL,namemtxvetDificultPL)


  namevetDificultPL<-
paste0(path,'/Splits/vetDificultXPL/','mtxvetDificultPL',"D",DatasetCorrente,"Alg",
algorithm,
                      'Difficult',SelectDifficult,"_",stepMoment,".csv");
  write.csv(vetDificultPL,namevetDificultPL)

  setwd("C:/TRICluster-out/"); getwd();

}

TethaResult<-function(modX,SelectDifficult,algorithm,stepMoment){
```

```
  expected<-modX@Model[7]
  nameexpected<-
paste0(path,'/Splits/Thetas/','vetThetas',"D",DatasetCorrente,"Alg",algorithm,
                  'SelectDifficult',SelectDifficult,"Turn",stepMoment,".csv");
  write.csv(expected,nameexpected)

  # Return vector with tethas in each search
  #return(expected)

}

balancedDataOutput<-function(dataTemp,SelectDifficult,flag1){

  dataOrderNew1<-OutEasyData(dataTemp,SelectDifficult)
  classOuputdata<-dataOrderNew1[1,ncol(dataOrderNew1)-1]
  myxInputNew<-SlicesChange(classOuputdata, dataOrderNew1,rowsOff,mini)
  myxInputNew1<-as.list(myxInputNew)
  szdataOrderNew1<-dim(dataOrderNew1)[1]

  while (flag1!=1){
    for (s1 in 1:szdataOrderNew1){
      currentData<-dataOrderNew1[s1,1:ncol(dataOrderNew1)-1]
      currentData<-as.list(currentData)
      qdeIntersection<-length(intersect(myxInputNew1,currentData))

      #if(myxInputNew1 != currentData){flag1<-"1";break}else{myxInputNew<-
SlicesChange(classOuputdata, dataOrderNew1,rowsOff,mini)}
      if(qdeIntersection>0){flag1<-"1";break}else{myxInputNew<-
SlicesChange(classOuputdata, dataOrderNew1,rowsOff,mini)}



    }
  }
  return(myxInputNew)
}

ET_test<-function(){

  library(stats)

  setwd("I:/backup_Projeto/24/Splits/vetDifficult");
  getwd();

  Y<-1;W<-1;X<-1

  nameMTXDifficults<-
paste0("vetDifficult",Y,"D",W,"Alg",X,"All3Difficult1000.csv")
  MTXsetSelected_raw<-read.csv(file = nameMTXDifficults,header = TRUE, sep =
",")
  MTXsetSelected<-
as.data.frame(MTXsetSelected_raw[,2:ncol(MTXsetSelected_raw)])

  dim(MTXsetSelected)
```

```
  zz<-
pairwise.wilcox.test(MTXsetSelected_raw[1,],MTXsetSelected_raw[3,],p.adjust.m
ethod = "bonferroni")
  zz<-
pairwise.wilcox.test(MTXsetSelected[1,],MTXsetSelected[3,],p.adjust.method =
"bonferroni")

  KK_output<-kruskal.test(MTXsetSelected[1:3,])

}

CalcMetrics<-function(vetTarget,vetPredict,metric,dataNorm){

  #a1<-dataTemp[,ncol(dataTemp)]
  a1<-as.numeric(as.matrix(vetTarget))
  a2<-vetPredict#[1:length(vetPredict)-1]
   #a2<-(vetPredict)
  b1<-as.integer(a1)
  b2<-as.integer(a2)

  # Calculate metrics
  #if(metric==0){msevalue<-recall(vetTarget, vetPredict)}
  #if(metric==1){mvalue<-mse(a1, a2)}
  if(metric==1){mvalue<-EvaluationMeasures.Accuracy(a1, a2)}
  if(metric==2){mvalue<-EvaluationMeasures.Sensitivity(a1, a2)}
  if(metric==3){mvalue<-EvaluationMeasures.Specificity(a1, a2)}
  if(metric==4){mvalue<-extCriteria(b1, b2,"Rand")} #Jaccard
  if(metric==5){mvalue<-extCriteria(b1, b2,"Jaccard")} #Jaccard
  if(metric==6){mvalue<-extCriteria(b1, b2,"Folkes_Mallows")} #Jaccard
  if(metric==7){mvalue<-intCriteria(dataNorm,b2,"Dunn")}
  if(metric==8){mvalue<-intCriteria(dataNorm,b2,"Silhouette")}
  if(metric==9){mvalue<-intCriteria(dataNorm,b2,"Davies_Bouldin")}
  if(metric==10){mvalue<-intCriteria(dataNorm,b2,"PBM")}
  #c("mse","acc","sen","spe", "IRand", "Ijac", "IFM", "IDunn", "ISil", "IDB",
"IPBM")

  return(mvalue)
}

#  Main program ------------------------------

set.seed(1)
LoadLibrary()

DatasetCorrenteVet<-
c(1,3,4,5,13,14,17,19,20,21,24);#(1,3,4,5,13,14,16,17,19,20,21,24);
algorithmList<-
c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,3
1,32,33,34,35,36,37,38)#c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,2
3,24,25,26,27,28,29,30,31,32,33,34,35)
metric<-1 #c(1,2,3,4,5)#(mse,Accuracy,Sensitivity,Specificity,roc)
turns<-500 #(1,n) # random rounds
rowsOff<-1#(1,nrow) # New samples in exchange
SelectDifficult<-c(1,2,3) # Molde PL
szDatasetCorrente<-(length(DatasetCorrenteVet))
```

```
#szalgorithm<-length(algorithm)
#szSelectDifficult<-length(SelectDifficult)
Percent<-24 # Percent of minirity class: 50% and 75% of classes
switchType<-3 # if switchType == 1: size of class is 24 or switchType==2: size of
class is iqual ro percent


for (stepmetric in 1:1){
  metric<-stepmetric

# Mode selection to archive output data
if(switchType=="0"){path<-"c:/backup_Projeto/24"}
if((switchType=="1") && (Percent<-50)){path<-'c:/backup_Projeto/50'}
if((switchType=="2")&&(Percent<-75)){path<-'c:/backup_Projeto/75'}
if(switchType=="3"){path<-'c:/backup_Projeto/SMOTE'}

MtxTimesClustes<-c(); MtxTimesClustesAll<-c()
setwd(path); getwd();

for(stepDatasetCorrente in 2:11){ # all Datasets
  DatasetCorrente<-DatasetCorrenteVet[stepDatasetCorrente]
  mtxDiffcultjoinAlgs<-c()

  for(stepalgorithm in 22:22){ # all algorithm
    algorithm<-algorithmList[stepalgorithm]
    vetDifficult<-c()

   for(stepDifficult in 1:3){ # all Difficults
     SelectDifficult<-stepDifficult

     mtxMetrics<-c();vetMSEAll<-c();vetMASEAll<-c();vetMetricsStepAll<-
c();mtxPredict<-c();mtxvetDificultPL<-c();vetMetrics2<-c()
     mtxMetrics2<-c()
#      }}}

     setwd("c:/TRICluster-out"); getwd();
     source("LoadDatasets.R")
     cls<-summary(dataAnaliseClass)
     mini<-"0"
     mirority<-mini
     numeroClasses<-length(levels(dataAnaliseClass))
     bestSlice<-c();vetMSE<-c();vetBest<-c();meandificultBest<-(-12);vetDown<-c()

     splitMinority<-SlicesStartMin(DatasetSelected)
     splitMajority<-SlicesStartMaj(DatasetSelected)
     dataTemp<-SlicesJoin(splitMinority,splitMajority)
     mtxReference<-c()

     # Redulce exemples in all dataset to 25 exemples balanced - set by smalless
class in datasets
     #if(DatasetCorrente=="16"){}

     if(switchType=="1"){dataTemp<-
reduceData(DatasetSelected);if(metric==1){dataBest<-0};if(metric==2){dataBest<-
0};
```

```
    if(metric==3){dataBest<-0};if(metric==4){dataBest<-
0};if(metric==5){dataBest<-1};if(metric==6){dataBest<-1};if(metric==7){dataBest<-
1};
    if(metric==8){dataBest<-10};if(metric==9){dataBest<-(-
1)};if(metric==10){dataBest<- 10};if(metric==11){dataBest<-0}}#size class = 24

    if(switchType=="2"){dataTemp<-
PercentData(DatasetSelected,Percent);if(metric==1){dataBest<-
0};if(metric==2){dataBest<-0};
    if(metric==3){dataBest<-0};if(metric==4){dataBest<-
0};if(metric==5){dataBest<-1};if(metric==6){dataBest<-1};if(metric==7){dataBest<-
1};
    if(metric==8){dataBest<-10};if(metric==9){dataBest<-(-
1)};if(metric==10){dataBest<- 10};if(metric==11){dataBest<-0}} #size class = 80%
of minority class

    if(switchType=="3"){dataTemp<-
balancedDataX(DatasetSelected,DatasetCorrente);if(metric==1){dataBest<-
0};if(metric==2){dataBest<-0};
    if(metric==3){dataBest<-0};if(metric==4){dataBest<-
0};if(metric==5){dataBest<-1};if(metric==6){dataBest<-1};if(metric==7){dataBest<-
1};
    if(metric==8){dataBest<-10};if(metric==9){dataBest<-(-
1)};if(metric==10){dataBest<- 10};if(metric==11){dataBest<-0}}#size class =
minority class original
    # ************

    allmtxMetrics2<-c()

    for (stepMoment in 1:turns){

      mtxReference<-dataTemp
      vetPredict<-classifiers(dataTemp,algorithm)

      # CPU core

      if(SelectDifficult==1){vetDificultPL<-pvec(1:30, function(calcDificult1PL)
dataTemp,mc.cores = 1);meanDificultPL<-
median(as.matrix(vetDificultPL[1:nrow(vetDificultPL),1:ncol(vetDificultPL)-1]))}
      if(SelectDifficult==2){vetDificultPL<-pvec(1:30, function(calcDificult2PL)
dataTemp,mc.cores = 1);meanDificultPL<-
median(as.matrix(vetDificultPL[1:nrow(vetDificultPL),1:ncol(vetDificultPL)-1]))}
      if(SelectDifficult==3){vetDificultPL<-pvec(1:30, function(calcDificult3PL)
dataTemp,mc.cores = 1);meanDificultPL<-
median(as.matrix(vetDificultPL[1:nrow(vetDificultPL),1:ncol(vetDificultPL)-1]))}

      #meanDificult1PL<-
mean(as.matrix(vetDificult1PL[1:nrow(vetDificult1PL),1:ncol(vetDificult1PL)-1]))
      vetTarget<-dataTemp[,ncol(dataTemp)]
      vetTarget2<-as.numeric(vetTarget)

      if(switchType=="3"){vetTarget2<-AdjustClass(vetTarget2)}
      dataNorm<-scale(dataTemp[,(ncol(dataTemp)-1)])
```

```
      #outPutMetric<-
CalcMetrics(as.integer(dataTemp[,ncol(dataTemp)]),as.integer(vetPredict[1:lengt
h(vetPredict)-1]),l_metric,dataNorm)
      #tempMSE<-
CalcMetrics(as.integer(dataTemp[,ncol(dataTemp)]),as.integer(vetPredict[1:lengt
h(vetPredict)-1]),metric,dataNorm)


      tempMSE<-
CalcMetrics(dataTemp[,ncol(dataTemp)],vetPredict[1:(length(vetPredict)-
1)],metric,dataNorm)

      #tempMSE<-CalcMetrics(vetTarget2, vetPredict,metric)
      tempMASE<-CalcMASE(vetTarget2, vetPredict)

      vetMSEAll<-cbind(vetMSEAll,tempMSE)
      vetMASEAll<-cbind(vetMASEAll,tempMASE)

     vetMs<-c()
      #update dataBest and update meandificultBest
      if((metric==1)&&(as.numeric(tempMSE) >=
as.numeric(dataBest))&&(meanDificultPL >= meandificultBest)){
        dataBest<-tempMSE;
        meandificultBest<-meanDificultPL;
        bestSlice<-dataTemp;
        vetDown<-cbind(vetDown,stepMoment); #check-out
        vetMetrics<-c();
        dataNorm<-scale(dataTemp[,(ncol(dataTemp)-1)])
        vetMetrics<-metricsByTRI(vetTarget, vetPredict,dataNorm,switchType)
        mtxMetrics<-cbind(mtxMetrics,vetMetrics)
        m1<-vetMetrics[1]
        m2<-vetMetrics[2]
        m3<-vetMetrics[3]
        m4<-vetMetrics[4]
        m5<-vetMetrics[5]
        m6<-vetMetrics[6]
        m7<-vetMetrics[7]
        m8<-vetMetrics[8]
        m9<-vetMetrics[9]
        m10<-vetMetrics[10]
        m11<-vetMetrics[11]
        vetMs<-c(m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11)
        mtxMetrics2<-cbind(mtxMetrics2,vetMs)
      }

      if((metric==2)&&(as.numeric(tempMSE) >=
as.numeric(dataBest))&&(meanDificultPL >= meandificultBest)){
        dataBest<-tempMSE;
        meandificultBest<-meanDificultPL;
        bestSlice<-dataTemp;
        vetDown<-cbind(vetDown,stepMoment); #check-out
        vetMetrics<-c();
        dataNorm<-scale(dataTemp[,(ncol(dataTemp)-1)])
        vetMetrics<-metricsByTRI(vetTarget, vetPredict,dataNorm,switchType)
        mtxMetrics<-cbind(mtxMetrics,vetMetrics)
```

```
        m1<-vetMetrics[1]
        m2<-vetMetrics[2]
        m3<-vetMetrics[3]
        m4<-vetMetrics[4]
        m5<-vetMetrics[5]
        m6<-vetMetrics[6]
        m7<-vetMetrics[7]
        m8<-vetMetrics[8]
        m9<-vetMetrics[9]
        m10<-vetMetrics[10]
        m11<-vetMetrics[11]
        vetMs<-c(m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11)
        mtxMetrics2<-cbind(mtxMetrics2,vetMs)
      }

      if((metric==3)&&(as.numeric(tempMSE) >=
as.numeric(dataBest))&&(meanDificultPL >= meandificultBest)){
        dataBest<-tempMSE;
        meandificultBest<-meanDificultPL;
        bestSlice<-dataTemp;
        vetDown<-cbind(vetDown,stepMoment); #check-out
        vetMetrics<-c();
        dataNorm<-scale(dataTemp[,(ncol(dataTemp)-1)])
        vetMetrics<-metricsByTRI(vetTarget, vetPredict,dataNorm,switchType)
        mtxMetrics<-cbind(mtxMetrics,vetMetrics)
        m1<-vetMetrics[1]
        m2<-vetMetrics[2]
        m3<-vetMetrics[3]
        m4<-vetMetrics[4]
        m5<-vetMetrics[5]
        m6<-vetMetrics[6]
        m7<-vetMetrics[7]
        m8<-vetMetrics[8]
        m9<-vetMetrics[9]
        m10<-vetMetrics[10]
        m11<-vetMetrics[11]
        vetMs<-c(m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11)
        mtxMetrics2<-cbind(mtxMetrics2,vetMs)
      }

      if((metric==4)&&(as.numeric(tempMSE) <=
as.numeric(dataBest))&&(meanDificultPL >= meandificultBest)){
        dataBest<-tempMSE;
        meandificultBest<-meanDificultPL;
        bestSlice<-dataTemp;
        vetDown<-cbind(vetDown,stepMoment); #check-out
        vetMetrics<-c();
        dataNorm<-scale(dataTemp[,(ncol(dataTemp)-1)])
        vetMetrics<-metricsByTRI(vetTarget, vetPredict,dataNorm,switchType)
        mtxMetrics<-cbind(mtxMetrics,vetMetrics)
        m1<-vetMetrics[1]
        m2<-vetMetrics[2]
        m3<-vetMetrics[3]
        m4<-vetMetrics[4]
        m5<-vetMetrics[5]
```

```
        m6<-vetMetrics[6]
        m7<-vetMetrics[7]
        m8<-vetMetrics[8]
        m9<-vetMetrics[9]
        m10<-vetMetrics[10]
        m11<-vetMetrics[11]
        vetMs<-c(m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11)
        mtxMetrics2<-cbind(mtxMetrics2,vetMs)
      }

      if((metric==5)&&(as.numeric(tempMSE) <=
as.numeric(dataBest))&&(meanDificultPL >= meandificultBest)){
        dataBest<-tempMSE;
        meandificultBest<-meanDificultPL;
        bestSlice<-dataTemp;
        vetDown<-cbind(vetDown,stepMoment); #check-out
        vetMetrics<-c();
        dataNorm<-scale(dataTemp[,(ncol(dataTemp)-1)])
        vetMetrics<-metricsByTRI(vetTarget, vetPredict,dataNorm,switchType)
        mtxMetrics<-cbind(mtxMetrics,vetMetrics)
        m1<-vetMetrics[1]
        m2<-vetMetrics[2]
        m3<-vetMetrics[3]
        m4<-vetMetrics[4]
        m5<-vetMetrics[5]
        m6<-vetMetrics[6]
        m7<-vetMetrics[7]
        m8<-vetMetrics[8]
        m9<-vetMetrics[9]
        m10<-vetMetrics[10]
        m11<-vetMetrics[11]
        vetMs<-c(m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11)
        mtxMetrics2<-cbind(mtxMetrics2,vetMs)

      }

      if((metric==6)&&(as.numeric(tempMSE) <=
as.numeric(dataBest))&&(meanDificultPL >= meandificultBest)){
        dataBest<-tempMSE;
        meandificultBest<-meanDificultPL;
        bestSlice<-dataTemp;
        vetDown<-cbind(vetDown,stepMoment); #check-out
        vetMetrics<-c();
        dataNorm<-scale(dataTemp[,(ncol(dataTemp)-1)])
        vetMetrics<-metricsByTRI(vetTarget, vetPredict,dataNorm,switchType)
        mtxMetrics<-cbind(mtxMetrics,vetMetrics)
        m1<-vetMetrics[1]
        m2<-vetMetrics[2]
        m3<-vetMetrics[3]
        m4<-vetMetrics[4]
        m5<-vetMetrics[5]
        m6<-vetMetrics[6]
        m7<-vetMetrics[7]
        m8<-vetMetrics[8]
        m9<-vetMetrics[9]
```

```
        m10<-vetMetrics[10]
        m11<-vetMetrics[11]
        vetMs<-c(m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11)
        mtxMetrics2<-cbind(mtxMetrics2,vetMs)
      }

      if((metric==7)&&(as.numeric(tempMSE) <=
as.numeric(dataBest))&&(meanDificultPL >= meandificultBest)){
        dataBest<-tempMSE;
        meandificultBest<-meanDificultPL;
        bestSlice<-dataTemp;
        vetDown<-cbind(vetDown,stepMoment); #check-out
        vetMetrics<-c();
        dataNorm<-scale(dataTemp[,(ncol(dataTemp)-1)])
        vetMetrics<-metricsByTRI(vetTarget, vetPredict,dataNorm,switchType)
        mtxMetrics<-cbind(mtxMetrics,vetMetrics)
        m1<-vetMetrics[1]
        m2<-vetMetrics[2]
        m3<-vetMetrics[3]
        m4<-vetMetrics[4]
        m5<-vetMetrics[5]
        m6<-vetMetrics[6]
        m7<-vetMetrics[7]
        m8<-vetMetrics[8]
        m9<-vetMetrics[9]
        m10<-vetMetrics[10]
        m11<-vetMetrics[11]
        vetMs<-c(m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11)
        mtxMetrics2<-cbind(mtxMetrics2,vetMs)
      }

      if((metric==8)&&(as.numeric(tempMSE) <=
as.numeric(dataBest))&&(meanDificultPL >= meandificultBest)){
        dataBest<-tempMSE;
        meandificultBest<-meanDificultPL;
        bestSlice<-dataTemp;
        vetDown<-cbind(vetDown,stepMoment); #check-out
        vetMetrics<-c();
        dataNorm<-scale(dataTemp[,(ncol(dataTemp)-1)])
        vetMetrics<-metricsByTRI(vetTarget, vetPredict,dataNorm,switchType)
        mtxMetrics<-cbind(mtxMetrics,vetMetrics)
        m1<-vetMetrics[1]
        m2<-vetMetrics[2]
        m3<-vetMetrics[3]
        m4<-vetMetrics[4]
        m5<-vetMetrics[5]
        m6<-vetMetrics[6]
        m7<-vetMetrics[7]
        m8<-vetMetrics[8]
        m9<-vetMetrics[9]
        m10<-vetMetrics[10]
        m11<-vetMetrics[11]
        vetMs<-c(m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11)
        mtxMetrics2<-cbind(mtxMetrics2,vetMs)
      }
```

```
      if((metric==9)&&(as.numeric(tempMSE) <=
as.numeric(dataBest))&&(meanDificultPL >= meandificultBest)){
        dataBest<-tempMSE;
        meandificultBest<-meanDificultPL;
        bestSlice<-dataTemp;
        vetDown<-cbind(vetDown,stepMoment); #check-out
        vetMetrics<-c();
        dataNorm<-scale(dataTemp[,(ncol(dataTemp)-1)])
        vetMetrics<-metricsByTRI(vetTarget, vetPredict,dataNorm,switchType)
        mtxMetrics<-cbind(mtxMetrics,vetMetrics)
        m1<-vetMetrics[1]
        m2<-vetMetrics[2]
        m3<-vetMetrics[3]
        m4<-vetMetrics[4]
        m5<-vetMetrics[5]
        m6<-vetMetrics[6]
        m7<-vetMetrics[7]
        m8<-vetMetrics[8]
        m9<-vetMetrics[9]
        m10<-vetMetrics[10]
        m11<-vetMetrics[11]
        vetMs<-c(m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11)
        mtxMetrics2<-cbind(mtxMetrics2,vetMs)
      }

      if((metric==10)&&(as.numeric(tempMSE) <=
as.numeric(dataBest))&&(meanDificultPL >= meandificultBest)){
        dataBest<-tempMSE;
        meandificultBest<-meanDificultPL;
        bestSlice<-dataTemp;
        vetDown<-cbind(vetDown,stepMoment); #check-out
        vetMetrics<-c();
        dataNorm<-scale(dataTemp[,(ncol(dataTemp)-1)])
        vetMetrics<-metricsByTRI(vetTarget, vetPredict,dataNorm,switchType)
        mtxMetrics<-cbind(mtxMetrics,vetMetrics)
        m1<-vetMetrics[1]
        m2<-vetMetrics[2]
        m3<-vetMetrics[3]
        m4<-vetMetrics[4]
        m5<-vetMetrics[5]
        m6<-vetMetrics[6]
        m7<-vetMetrics[7]
        m8<-vetMetrics[8]
        m9<-vetMetrics[9]
        m10<-vetMetrics[10]
        m11<-vetMetrics[11]
        vetMs<-c(m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11)
        mtxMetrics2<-cbind(mtxMetrics2,vetMs)
      }

      if((metric==11)&&(as.numeric(tempMSE) <=
as.numeric(dataBest))&&(meanDificultPL >= meandificultBest)){
        dataBest<-tempMSE;
        meandificultBest<-meanDificultPL;
```

```
      bestSlice<-dataTemp;
      vetDown<-cbind(vetDown,stepMoment); #check-out
      vetMetrics<-c();
      dataNorm<-scale(dataTemp[,(ncol(dataTemp)-1)])
      vetMetrics<-metricsByTRI(vetTarget, vetPredict,dataNorm,switchType)
      mtxMetrics<-cbind(mtxMetrics,vetMetrics)
      m1<-vetMetrics[1]
      m2<-vetMetrics[2]
      m3<-vetMetrics[3]
      m4<-vetMetrics[4]
      m5<-vetMetrics[5]
      m6<-vetMetrics[6]
      m7<-vetMetrics[7]
      m8<-vetMetrics[8]
      m9<-vetMetrics[9]
      m10<-vetMetrics[10]
      m11<-vetMetrics[11]
      vetMs<-c(m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11)
      mtxMetrics2<-cbind(mtxMetrics2,vetMs)
    }

    vetMetricsStep<-c()
    dataNorm<-scale(dataTemp[,(ncol(dataTemp)-1)])
    vetMetricsStep<-metricsByTRI(vetTarget, vetPredict,dataNorm,switchType)
    vetMetricsStepTranspost<-as.vector(t(vetMetricsStep))
    vetMetricsStepAll<-cbind(vetMetricsStepAll,vetMetricsStepTranspost)

    mtxPredict<-rbind(mtxPredict,vetPredict)
    mtxvetDificultPL<-rbind(mtxvetDificultPL,vetDificultPL)

    vetBest<-cbind(vetBest,dataBest);

    # Tabu mode
    flag1<-0
    dataOrderNew<-OutEasyData(dataTemp,SelectDifficult)
    classOuputdata<-dataOrderNew[1,ncol(dataOrderNew)-1]
    valNew<-balancedDataOutput(dataTemp,SelectDifficult,flag1)
    dataOrderNew[1:rowsOff,]<-valNew
    dataTemp<-dataOrderNew[,1:ncol(dataOrderNew)-1]


    setwd(path); getwd();

    setwd("c:/TRICluster-out"); getwd();

    print("-------------------------------------")

print(c("DatasetCorrente",DatasetCorrente,"algorithm",algorithm,"Moment",step
Moment,"tempMSE",tempMSE,"dataBest",
          dataBest,"tempMASE",tempMASE,"mean Dificult
1PL",meanDificultPL,"meandificultBest",meandificultBest,
          "stepDifficult:",stepDifficult,"Check-out",vetDown))
    print("-------------------------------------")

  }
```

```
    print("########################################")
    print("########################################")

    setwd("c:/TRICluster-out"); getwd();

    # Storage in vetor the three models by classificator
    vetDifficult<-rbind(vetDifficult,vetBest)

    setwd(path); getwd();


    vetDiffcultjoinAlgs<-cbind(stepalgorithm,vetDifficult)
    mtxDiffcultjoinAlgs<-rbind(mtxDiffcultjoinAlgs,vetDiffcultjoinAlgs)


    #nameMTXClusterTime<-
paste0("c:/backup_Projeto/OutputClusters/timeparallel/",'MAtrix',DatasetCorrent
e,"TimesClusters","%",metric,".csv")
    #write.csv(MtxTimesClustes,nameMTXClusterTime)

    namemtxMetrics<-
paste0(path,"/Splits/Metrics/","mtxMetrics",'Dataset',DatasetCorrente,"algorithm"
,algorithm,SelectDifficult,"PL",".csv")
    write.csv(mtxMetrics,namemtxMetrics)

    namemtxMetrics2<-
paste0(path,"/Splits/Metricsall/","mtxMetricsAll",'Dataset',DatasetCorrente,"algor
ithm",algorithm,SelectDifficult,"PL",".csv")
    write.csv(mtxMetrics2,namemtxMetrics2)

    namebestSlice<-
paste0(path,'/Splits/bestSlice/','bestSlice',"D",DatasetCorrente,"Alg",algorithm,
                'Difficult',SelectDifficult,"_",stepMoment,".csv");
    write.csv(bestSlice,namebestSlice)

    setwd("c:/TRICluster-out"); getwd();


savevetDifficult(vetDown,SelectDifficult,vetDifficult,stepMoment,vetMSEAll,vetM
ASEAll,vetMetricsStepAll,
            mtxPredict,mtxvetDificultPL,vetDificultPL)

    allmtxMetrics2<-rbind(allmtxMetrics2,mtxMetrics2)

  }

  setwd("c:/TRICluster-out"); getwd();

  print(c("#@@@@@@@@@@@@@ next model",algorithm))

  print('xxxxxx')
  funcGraphic(vetDifficult)
  #SaveParameters(dataTemp,meandificultBest,vetDificult1PL)
  SaveParameters(dataTemp,meandificultBest,vetDificultPL)
  saveVetDown(vetDown,SelectDifficult,vetDifficult)
```

```
  }
    nameMtx3PL<-
paste0("C:/backup_Projeto/SMOTE/Splits/mtx3PL/",'Matrix',DatasetCorrente,"Met
ric",metric,".csv")
  write.csv(mtxDiffcultjoinAlgs,nameMtx3PL)
    #plotG1(vetDifficult,metric)
  }
}
```

```
## Code Run_Join


setwd("C:\\backup_Projeto\\Resultados\\OutputClusters\\50\\OutputClusters\\time
"); getwd();
#setwd("C:\\backup_Projeto\\Resultados\\OutputClusters\\100\\OutputClusters\\ti
me"); getwd();

TableTimes<-read.csv(file = 'MAtrix24TimesClusters%1.csv',header = TRUE, sep
= ",")
#TableTimes<-read.csv(file = 'MAtrix24TimesClusters%3.csv',header = TRUE, sep
= ",")

mtxTimes<-c();TableTimesTemp<-c();tCont<-0
initSlice<-1
endSlice<-38

for(sliceCol in 1:11){


  TableTimesTemp<-TableTimes[initSlice:endSlice,2]
  mtxTimes<-cbind(mtxTimes,TableTimesTemp)

  initSlice<-initSlice+38
  endSlice<-initSlice+38
  print(c(initSlice,endSlice))
  tCont<-tCont+1
  print(tCont)

}

dim(mtxTimes)[1]

dim(TableTimes)[1]/11

#colnames(dados) <- c("Coluna1", "Coluna2", "Coluna3")

nameTabGer50<-
paste0("C:\\backup_Projeto\\Resultados\\OutputClusters\\50\\OutputClusters\\tim
e\\TabTime50.csv")
write.csv(mtxTimes,nameTabGer50)


######################
```

```
#setwd("C:\\backup_Projeto\\Resultados\\OutputClusters\\50\\OutputClusters\\tim
e"); getwd();
setwd("C:\\backup_Projeto\\Resultados\\OutputClusters\\100\\OutputClusters\\tim
e"); getwd();

#TableTimes<-read.csv(file = 'MAtrix24TimesClusters%1.csv',header = TRUE, sep
= ",")
TableTimes<-read.csv(file = 'MAtrix24TimesClusters%3.csv',header = TRUE, sep
= ",")

mtxTimes<-c();TableTimesTemp<-c();tCont<-0
initSlice<-1
endSlice<-38

for(sliceCol in 1:11){

  TableTimesTemp<-TableTimes[initSlice:endSlice,2]
  mtxTimes<-cbind(mtxTimes,TableTimesTemp)

  initSlice<-initSlice+38
  endSlice<-initSlice+38
  print(c(initSlice,endSlice))
  tCont<-tCont+1
  print(tCont)

}

dim(mtxTimes)[1]

dim(TableTimes)[1]/11

nameTabGer100<-
paste0("C:\\backup_Projeto\\Resultados\\OutputClusters\\100\\OutputClusters\\ti
me\\TabTime100.csv"); getwd();
write.csv(mtxTimes,nameTabGer100)

#######################
vetMtx<-c(1,3,4,5,13,14,17,19,20,21,24)
for (nme in 1:11){


setwd("C:\\backup_Projeto\\Resultados\\OutputClusters\\50\\OutputClusters\\mtx
TargetClassificator"); getwd();
  nmeSet<-vetMtx[nme]
  nmeMtx<-paste0("mtxTargetClassificator_D",nmeSet,"%1.csv")

  TableMtxCurrent<-read.csv(file = nmeMtx,header = TRUE, sep = ",")
  ordMtx<-cbind(TableMtxCurrent[,41:43],TableMtxCurrent[,2:40])
  colnames(ordMtx)<-c("1PL","2PL","3PL",

"Alg1","Alg2","Alg3","Alg4","Alg5","Alg6","Alg7","Alg8","Alg9","Alg10",

"Alg11","Alg12","Alg13","Alg14","Alg15","Alg16","Alg17","Alg18","Alg19","Alg2
0",
```

```
"Alg21","Alg22","Alg23","Alg124","Alg25","Alg26","Alg27","Alg28","Alg29","Alg
30",

"Alg31","Alg32","Alg34","Alg35","Alg36","Alg37","Alg38","Alg39","Alg40")

  newnmeMtx<-paste0("mtxTargetClassificator_D",nmeSet,"new%1.csv")
  nameTabGer100New<-
paste0("C:\\backup_Projeto\\Resultados\\OutputClusters\\50\\OutputClusters\\mt
xTargetClassificator\\Rename\\",newnmeMtx)
  write.csv(ordMtx,nameTabGer100New)

  library(corrplot)
  setwd("C:\\backup_Projeto\\OutputClusters\\50\\Plots Exports"); getwd();

  nameGrafico<-paste0('Gráfico de Correlação do Dataset ',nme,"50_G1.tiff");
  g1<-as.data.frame(cbind(ordMtx[,1:12],ordMtx[,17],ordMtx[,39]))
  colnames(g1)<-
c("1PL","2PL","3PL","Alg1","Alg2","Alg3","Alg4","Alg5","Alg6","Alg7","Alg8","A
lg9","Alg14","Alg36")
  tiff(nameGrafico, units="in", width=10, height=10, res=300)
  matriz_cor <- cor(g1)
  gg1<-corrplot(matriz_cor, method = "color",col = COL2('RdYlBu', 10),type =
'lower')$corrPos -> p1
  text(p1$x, p1$y, round(p1$corr, 2))
  dev.off()
  ##################

  nameGrafico<-paste0('Gráfico de Correlação do Dataset ',nme,"50_G2.tiff");
  g2<-
as.data.frame(cbind(ordMtx[,1:3],ordMtx[,13:16],ordMtx[,18:23],ordMtx[,40:41]))
  colnames(g2)<-
c("1PL","2PL","3PL","Alg10","Alg11","Alg12","Alg13","Alg15","Alg16","Alg17","
Alg18","Alg19","Alg20","Alg37","Alg38")
  tiff(nameGrafico, units="in", width=10, height=10, res=300)
  matriz_cor <- cor(g2)
  gg2<-corrplot(matriz_cor, method = "color",col = COL2('RdYlBu', 10),type =
'lower')$corrPos -> p1
  text(p1$x, p1$y, round(p1$corr, 2))
  dev.off()
  ##################

  nameGrafico<-paste0('Gráfico de Correlação do Dataset ',nme,"50_G3.tiff");
  g3<-as.data.frame(cbind(ordMtx[,1:3],ordMtx[,24:38]))
  colnames(g3)<-
c("1PL","2PL","3PL","Alg21","Alg22","Alg23","Alg24","Alg25","Alg26","Alg27","
Alg28","Alg29","Alg30","Alg31","Alg32","Alg33","Alg34","Alg35")
  tiff(nameGrafico, units="in", width=10, height=10, res=300)
  matriz_cor <- cor(g3)
  gg3<-corrplot(matriz_cor, method = "color",col = COL2('RdYlBu', 10),type =
'lower')$corrPos -> p1
  text(p1$x, p1$y, round(p1$corr, 2))
  dev.off()
  ##################
```

```
  png(filename=paste0("Figura",nmeSet,".png"), height=50, width=50, unit="cm",
res=300)
  par(mfrow=c(3,1), mar=c(3,3,2,2), oma=c(3,3,2,2))

  matriz_cor <- cor(g1)
  gg1<-corrplot(matriz_cor, method = "color",col = COL2('RdYlBu', 10),type =
'lower')$corrPos -> p1
  text(p1$x, p1$y, round(p1$corr, 2))

  matriz_cor <- cor(g2)
  gg2<-corrplot(matriz_cor, method = "color",col = COL2('RdYlBu', 10),type =
'lower')$corrPos -> p1
  text(p1$x, p1$y, round(p1$corr, 2))

  matriz_cor <- cor(g3)
  gg3<-corrplot(matriz_cor, method = "color",col = COL2('RdYlBu', 10),type =
'lower')$corrPos -> p1
  text(p1$x, p1$y, round(p1$corr, 2))
  #mtext(side=1, text="Meu eixo X lindo", outer=T)
  #mtext(side=2, text="Meu eixo Y lindo", outer=T)
  dev.off()

}

########################
setwd("C:\\backup_Projeto\\Resultados\\OutputClusters\\100\\OutputClusters\\mt
xTargetClassificator"); getwd();

vetMtx<-c(1,3,4,5,13,14,17,19,20,21,24)
for (nme in 1:11){


setwd("C:\\backup_Projeto\\Resultados\\OutputClusters\\100\\OutputClusters\\mt
xTargetClassificator"); getwd();
  nmeSet<-vetMtx[nme]
  nmeMtx<-paste0("mtxTargetClassificator_D",nmeSet,"%3.csv")

  TableMtxCurrent<-read.csv(file = nmeMtx,header = TRUE, sep = ",")
  ordMtx<-cbind(TableMtxCurrent[,41:43],TableMtxCurrent[,2:40])
  colnames(ordMtx)<-c("1PL","2PL","3PL",

"Alg1","Alg2","Alg3","Alg4","Alg5","Alg6","Alg7","Alg8","Alg9","Alg10",

"Alg11","Alg12","Alg13","Alg14","Alg15","Alg16","Alg17","Alg18","Alg19","Alg2
0",

"Alg21","Alg22","Alg23","Alg124","Alg25","Alg26","Alg27","Alg28","Alg29","Alg
30",

"Alg31","Alg32","Alg34","Alg35","Alg36","Alg37","Alg38","Alg39","Alg40")

  newnmeMtx<-paste0("mtxTargetClassificator_D",nmeSet,"new%3.csv")
  nameTabGer100New<-
paste0("C:\\backup_Projeto\\Resultados\\OutputClusters\\100\\OutputClusters\\m
txTargetClassificator\\Rename\\",newnmeMtx)
```

```
  write.csv(ordMtx,nameTabGer100New)


  library(corrplot)
  setwd("C:\\backup_Projeto\\OutputClusters\\100\\Plots Exports"); getwd();

  nameGrafico<-paste0('Gráfico de Correlação do Dataset ',nme,"100_G1.tiff");
  g1<-as.data.frame(cbind(ordMtx[,1:12],ordMtx[,17],ordMtx[,39]))
  colnames(g1)<-
c("1PL","2PL","3PL","Alg1","Alg2","Alg3","Alg4","Alg5","Alg6","Alg7","Alg8","A
lg9","Alg14","Alg36")
  tiff(nameGrafico, units="in", width=10, height=10, res=300)
  matriz_cor <- cor(g1)
  gg1<-corrplot(matriz_cor, method = "color",col = COL2('RdYlBu', 10),type =
'lower')$corrPos -> p1
  text(p1$x, p1$y, round(p1$corr, 2))
  dev.off()
  ###################

  nameGrafico<-paste0('Gráfico de Correlação do Dataset ',nme,"100_G2.tiff");
  g2<-
as.data.frame(cbind(ordMtx[,1:3],ordMtx[,13:16],ordMtx[,18:23],ordMtx[,40:41]))
  colnames(g2)<-
c("1PL","2PL","3PL","Alg10","Alg11","Alg12","Alg13","Alg15","Alg16","Alg17","
Alg18","Alg19","Alg20","Alg37","Alg38")
  tiff(nameGrafico, units="in", width=10, height=10, res=300)
  matriz_cor <- cor(g2)
  gg2<-corrplot(matriz_cor, method = "color",col = COL2('RdYlBu', 10),type =
'lower')$corrPos -> p1
  text(p1$x, p1$y, round(p1$corr, 2))
  dev.off()
  ###################

  nameGrafico<-paste0('Gráfico de Correlação do Dataset ',nme,"100_G3.tiff");
  g3<-as.data.frame(cbind(ordMtx[,1:3],ordMtx[,24:38]))
  colnames(g3)<-
c("1PL","2PL","3PL","Alg21","Alg22","Alg23","Alg24","Alg25","Alg26","Alg27","
Alg28","Alg29","Alg30","Alg31","Alg32","Alg33","Alg34","Alg35")
  tiff(nameGrafico, units="in", width=10, height=10, res=300)
  matriz_cor <- cor(g3)
  gg3<-corrplot(matriz_cor, method = "color",col = COL2('RdYlBu', 10),type =
'lower')$corrPos -> p1
  text(p1$x, p1$y, round(p1$corr, 2))
  dev.off()
  ###################

  png(filename=paste0("Figura",nmeSet,".png"), height=50, width=50, unit="cm",
res=300)
  par(mfrow=c(3,1), mar=c(3,3,2,2), oma=c(3,3,2,2))

  matriz_cor <- cor(g1)
  gg1<-corrplot(matriz_cor, method = "color",col = COL2('RdYlBu', 10),type =
'lower')$corrPos -> p1
  text(p1$x, p1$y, round(p1$corr, 2))
```

```
  matriz_cor <- cor(g2)
  gg2<-corrplot(matriz_cor, method = "color",col = COL2('RdYlBu', 10),type =
'lower')$corrPos -> p1
  text(p1$x, p1$y, round(p1$corr, 2))

  matriz_cor <- cor(g3)
  gg3<-corrplot(matriz_cor, method = "color",col = COL2('RdYlBu', 10),type =
'lower')$corrPos -> p1
  text(p1$x, p1$y, round(p1$corr, 2))
  #mtext(side=1, text="Meu eixo X lindo", outer=T)
  #mtext(side=2, text="Meu eixo Y lindo", outer=T)
  dev.off()
}
```

```
## Coe ComplexityByPL

rm(list=ls())

#### Functions
histGraphcs<-function(setDataset){
  #Criando os histogramas
  dataHist<-mtxpls;
  xmax<-max(mtxpls);xmin<-min(mtxpls)
  mPL<-1;

  if(setDataset==1){nameDataset<-"Dataset 1"}
  if(setDataset==2){nameDataset<-"Dataset 2"}
  if(setDataset==3){nameDataset<-"Dataset 3"}
  if(setDataset==4){nameDataset<-"Dataset 4"}
  if(setDataset==5){nameDataset<-"Dataset 5"}
  if(setDataset==6){nameDataset<-"Dataset 6"}
  if(setDataset==7){nameDataset<-"Dataset 7"}
  if(setDataset==8){nameDataset<-"Dataset 8"}
  if(setDataset==9){nameDataset<-"Dataset 9"}
  if(setDataset==10){nameDataset<-"Dataset 10"}
  if(setDataset==11){nameDataset<-"Dataset 11"}

  nameGrafico<-paste0('HistDataset',ListdataSet[stepdataset],".tiff")
  tituloGrafico<-paste0("Histograma do ",nameDataset)
  dataHist<-mtxpls
  #dataHist<-mtxtemposBP[,1]

  tiff(nameGrafico, units="in", width=5, height=5, res=300)
  if (setMode==1){
    ###### 50
    # compare difficulto with cluster - mtxTargetClassificator_D1%1.csv
    setwd("C:\\backup_Projeto\\Resultados\\OutputClusters\\50\\Plots\\BarrsHist");
getwd();

  }

  if (setMode==2){
    ###### 100
    # compare difficulto with cluster - mtxTargetClassificator_D1%1.csv
```

```
setwd("C:\\backup_Projeto\\Resultados\\OutputClusters\\100\\Plots\\BarrsHist");
getwd();
 }


 hist(mtxpls,
     main= tituloGrafico,
     xlab="Dificuldade",
     ylab = "Densidade",
     col="gray",
     nclass = 6,
     ylim=c(0,100),
     #ylim = NULL,
     xlim = c(xmin,xmax))
 #xlim=c(xmin,xmax))
 dev.off()
}
barrGraphcs<-function(setDataset){

 # Barrs
 # Grafico de setores - pizza
 sectors<-c()
 mtxplsOKF<-mtxpls
 colnames(mtxplsOKF)<-
as.character(c("MFC","FC","MDC","DC","MFE","FE","MDE","DE"))
 mi_tabla<-c("1PL","2PL","3PL")
 ListdataSet<-c(1,3,4,5,13,14,17,19,20,21,24);
 ListD7<-c(1,2,3,4,5,6,7,8,9,10,11);



 if(setDataset==1){nameDataset<-"Dataset 1"}
 if(setDataset==2){nameDataset<-"Dataset 2"}
 if(setDataset==3){nameDataset<-"Dataset 3"}
 if(setDataset==4){nameDataset<-"Dataset 4"}
 if(setDataset==5){nameDataset<-"Dataset 5"}
 if(setDataset==6){nameDataset<-"Dataset 6"}
 if(setDataset==7){nameDataset<-"Dataset 7"}
 if(setDataset==8){nameDataset<-"Dataset 8"}
 if(setDataset==9){nameDataset<-"Dataset 9"}
 if(setDataset==10){nameDataset<-"Dataset 10"}
 if(setDataset==11){nameDataset<-"Dataset 11"}

 mtxplsOKF<-mtxpls; posGraf<-1
 colnames(mtxplsOKF)<-
as.character(c("MFC","FC","MDC","DC","MFE","FE","MDE","DE"))
 nameGraficoD<-paste0('BarDataset',ListdataSet[stepdataset],".tiff")
 tituloGraficoD<-paste0("Gráfico de barras do ",nameDataset)

 tiff(nameGraficoD, units="in", width=5, height=5, res=300)
 # Corretas F
 if (setMode==1){
  ###### 50
  # compare difficulto with cluster - mtxTargetClassificator_D1%1.csv
```

```
   setwd("C:\\backup_Projeto\\Resultados\\OutputClusters\\50\\Plots\\BarrsHist");
getwd();

  }

  if (setMode==2){
   ###### 100
   # compare difficulto with cluster - mtxTargetClassificator_D1%1.csv

setwd("C:\\backup_Projeto\\Resultados\\OutputClusters\\100\\Plots\\BarrsHist");
getwd();
  }

  posGraf=ListD7[stepdataset];
  sectors<-as.matrix(mtxplsOKF)
  barplot(sectors,col = rainbow(3),
          #ylim = c(0,100),
          ylim = NULL,
          angle = 45,
          beside = TRUE,
          #ylim = c(0, 300),
          main = tituloGraficoD)
  #legend.text = mi_tabla)#rownames(mi_tabla),
  #args.legend = list(x = "left"))

  if(posGraf==1){
   #makePlot()
   legend("topright",
          legend=c("1PL","2PL","3PL"),
          fill = c("red","green","blue")
          #lty=1:2, cex=0.8, box.lty = 0
   )
  }


if((posGraf==2)|(posGraf==7)|(posGraf==8)|(posGraf==9)|(posGraf==10)|(posGraf
==11)){
   #makePlot()
   legend("topright",
          legend=c("1PL","2PL","3PL"),
          fill = c("red","green","blue")
          #lty=1:2, cex=0.8, box.lty = 0
   )
  }

  if((posGraf==3)|(posGraf==4)|(posGraf==5)|(posGraf==6)){
   #makePlot()
   legend("topright",
          legend=c("1PL","2PL","3PL"),
          fill = c("red","green","blue")
          #lty=1:2, cex=0.8, box.lty = 0
   )}

  dev.off()
}
```

```
boxplotGraphcs<-function(setDataset){
  # boxplot
  nameGrafico<-paste0('Boxplot de Dataset',ListdataSet[stepdataset],".tiff")
  tituloGrafico<-paste0("Boxplot dos niveis de b")
  dataHist<-mtxpls
  #dataHist<-mtxtemposBP[,1]

  if (setMode==1){
    ###### 50
    # compare difficulto with cluster - mtxTargetClassificator_D1%1.csv
    setwd("C:\\backup_Projeto\\Resultados\\OutputClusters\\50\\Plots\\BarrsHist");
getwd();

  }

  if (setMode==2){
    ###### 100
    # compare difficulto with cluster - mtxTargetClassificator_D1%1.csv

setwd("C:\\backup_Projeto\\Resultados\\OutputClusters\\100\\Plots\\BarrsHist");
getwd();
  }

  if(setDataset==1){nameDataset<-"Dataset 1"}
  if(setDataset==2){nameDataset<-"Dataset 2"}
  if(setDataset==3){nameDataset<-"Dataset 3"}
  if(setDataset==4){nameDataset<-"Dataset 4"}
  if(setDataset==5){nameDataset<-"Dataset 5"}
  if(setDataset==6){nameDataset<-"Dataset 6"}
  if(setDataset==7){nameDataset<-"Dataset 7"}
  if(setDataset==8){nameDataset<-"Dataset 8"}
  if(setDataset==9){nameDataset<-"Dataset 9"}
  if(setDataset==10){nameDataset<-"Dataset 10"}
  if(setDataset==11){nameDataset<-"Dataset 11"}

  tiff(nameGrafico, units="in", width=20, height=20, res=300)
  colnames(mtxpls)<-c("MFC","FC","MDC","DC","MFE","FE","MDE","DE")
  TitleBP<-paste0("Boxplot - ",nameDataset)
  boxplot(mtxpls, main=TitleBP)


  dev.off()
}
#########################################################

ListdataSet<-c(1,3,4,5,13,14,17,19,20,21,24);
setMode<-2

if (setMode==1){
  ###### 50
  # compare difficulto with cluster - mtxTargetClassificator_D1%1.csv

setwd("C:\\backup_Projeto\\OutputClusters\\50\\OutputClusters\\mtxTargetClassi
ficator"); getwd();
```

```
}

if (setMode==2){
###### 100
# compare difficulto with cluster - mtxTargetClassificator_D1%1.csv
setwd("C:\\backup_Projeto\\Resultados\\OutputClusters\\100\\OutputClusters\\mt
xTargetClassificator"); getwd();

}

##########################################

# matrizes acumuladores


# dataset 1
#stepdataset<-1
for (stepdataset in 1:(length(ListdataSet))){

  mtxCM1<-c();
  b_questo1PL<-c();b_questo2PL<-c();b_questo3PL<-c();
  contMuitoFcilecorreto1<-c();contMuitoDificilecorreto1<-
c();contMuitoFcileerrada1<-c();contMuitoDificileerrada1<-c();
  contMuitoFcilecorreto2<-c();contMuitoDificilecorreto2<-
c();contMuitoFcileerrada2<-c();contMuitoDificileerrada2<-c();
  contMuitoFcilecorreto3<-c();contMuitoDificilecorreto3<-
c();contMuitoFcileerrada3<-c();contMuitoDificileerrada3<-c();
  vet1pl<-c();vet2pl<-c();vet3pl<-c();mtxpls<-c()


  contMuitoFcilecorreto1<-0;contFcilecorreto1<-0;contMuitoDificilecorreto1<-
0;contDificilecorreto1<-0;
  contMuitoFcileerrada1<-0;contFcileerrada1<-0;contMuitoDificileerrada1<-
0;contDificileerrada1<-0;
  # 2PL
  contMuitoFcilecorreto2<-0;contFcilecorreto2<-0;contMuitoDificilecorreto2<-
0;contDificilecorreto2<-0;
  contMuitoFcileerrada2<-0;contFcileerrada2<-0;contMuitoDificileerrada2<-
0;contDificileerrada2<-0;
  # 3PL
  contMuitoFcilecorreto3<-0;contFcilecorreto3<-0;contMuitoDificilecorreto3<-
0;contDificilecorreto3<-0;
  contMuitoFcileerrada3<-0;contFcileerrada3<-0;contMuitoDificileerrada3<-
0;contDificileerrada3<-0;

  if (setMode==1){
    ###### 50
    # compare difficulto with cluster - mtxTargetClassificator_D1%1.csv

setwd("C:\\backup_Projeto\\OutputClusters\\50\\OutputClusters\\mtxTargetClassi
ficator"); getwd();

  }

  if (setMode==2){
```

```
   ###### 50
   # compare difficulto with cluster - mtxTargetClassificator_D1%1.csv

setwd("C:\\backup_Projeto\\Resultados\\OutputClusters\\100\\OutputClusters\\mt
xTargetClassificator"); getwd();


  }


if (setMode==1){
  modelDXPLX<-
paste0("mtxTargetClassificator_D",ListdataSet[stepdataset],"%1.csv")

}

if (setMode==2){
  modelDXPLX<-
paste0("mtxTargetClassificator_D",ListdataSet[stepdataset],"%3.csv")

}


mtxCM<-read.csv(file = modelDXPLX,header = TRUE, sep = ",")
mtxCM1<-mtxCM[,2:43]
szmtxCM11PL<-dim(mtxCM1)[1]
wdow<-c();mtxplsall<-c()

for (stepb in 1:szmtxCM11PL){
  print(c("Dataset",ListdataSet[stepdataset]))
  for(stepc in 1:38){

    #print(Wdow<-mtxCM1[stepb])
  #1 PL
  if((mtxCM1[stepb,40] < -
2)&&((mtxCM1[stepb,1]==mtxCM1[stepb,stepc]))){b_questo1PL[stepb]<-
"MFC";contMuitoFcilecorreto1<-contMuitoFcilecorreto1+1;Wdow<-
"MFC";print(contMuitoFcilecorreto1)}
  if((mtxCM1[stepb,40] > -2)&&((mtxCM1[stepb,1] <
0)&&((mtxCM1[stepb,stepc]==(mtxCM1[stepb,stepc])))){b_questo1PL[stepb]<-
"FC";contFcilecorreto1<-contFcilecorreto1+1;Wdow<-"FC"}
  if((mtxCM1[stepb,40] >
2)&&(mtxCM1[stepb,1]==mtxCM1[stepb,stepc])){b_questo1PL[stepb]<-
"MDC";contMuitoDificilecorreto1<-contMuitoDificilecorreto1+1;Wdow<-"MDC"}
  if((mtxCM1[stepb,40] <= 2)&&((mtxCM1[stepb,1] >=
0)&&((mtxCM1[stepb,stepc]==(mtxCM1[stepb,stepc])))){b_questo1PL[stepb]<-
"DC";contDificilecorreto1<-contDificilecorreto1+1;Wdow<-"DC"}
  if((mtxCM1[stepb,40] < -
2)&&((mtxCM1[stepb,1]!=mtxCM1[stepb,stepc]))){b_questo1PL[stepb]<-
"MFE";contMuitoFcileerrada1<-contMuitoFcileerrada1+1;Wdow<-"MFE"}
  if((mtxCM1[stepb,40] > -2)&&((mtxCM1[stepb,1] <
0)&&((mtxCM1[stepb,stepc]!=(mtxCM1[stepb,stepc])))){b_questo1PL[stepb]<-
"FE";contFcileerrada1<-contFcileerrada1+1;Wdow<-"FE"}
```

```
  if((mtxCM1[stepb,40] >
2)&&((mtxCM1[stepb,1]!=mtxCM1[stepb,stepc]))){b_questo1PL[stepb]<-
"MDE";contMuitoDificileerrada1<-contMuitoDificileerrada1+1;Wdow<-"MDE"}
  if((mtxCM1[stepb,40] <= 2)&&((mtxCM1[stepb,1] >=
0)&&((mtxCM1[stepb,stepc]!=(mtxCM1[stepb,stepc])))){b_questo1PL[stepb]<-
"DE";contDificileerrada1<-contDificileerrada1+1;Wdow<-"DE"}

  #2 PL
  if((mtxCM1[stepb,41] < -
2)&&((mtxCM1[stepb,2]==mtxCM1[stepb,stepc]))){b_questo2PL[stepb]<-
"MFC";contMuitoFcilecorreto2<-contMuitoFcilecorreto2+1;Wdow<-"MFC"}
  if((mtxCM1[stepb,41] > -2)&&((mtxCM1[stepb,2] <
0)&&(mtxCM1[stepb,stepc]==(mtxCM1[stepb,stepc])))){b_questo2PL[stepb]<-
"FC";contFcilecorreto2<-contFcilecorreto2+1;Wdow<-"FC"}
  if((mtxCM1[stepb,41] >
2)&&((mtxCM1[stepb,2]==mtxCM1[stepb,stepc]))){b_questo2PL[stepb]<-
"MDC";contMuitoDificilecorreto2<-contMuitoDificilecorreto2+1;Wdow<-"MDC"}
  if((mtxCM1[stepb,41] <= 2)&&((mtxCM1[stepb,2] >=
0)&&(mtxCM1[stepb,stepc]==(mtxCM1[stepb,stepc])))){b_questo2PL[stepb]<-
"DC";contDificilecorreto2<-contDificilecorreto2+1;Wdow<-"DC"}
  if((mtxCM1[stepb,41] < -
2)&&((mtxCM1[stepb,2]!=mtxCM1[stepb,stepc]))){b_questo2PL[stepb]<-
"MFE";contMuitoFcileerrada2<-contMuitoFcileerrada2+1;Wdow<-"MFE"}
  if((mtxCM1[stepb,41] > -2)&&((mtxCM1[stepb,2] <
0)&&(mtxCM1[stepb,stepc]!=(mtxCM1[stepb,stepc])))){b_questo2PL[stepb]<-
"FE";contFcileerrada2<-contFcileerrada2+1;Wdow<-"FE"}
  if((mtxCM1[stepb,41] >
2)&&((mtxCM1[stepb,2]!=mtxCM1[stepb,stepc]))){b_questo2PL[stepb]<-
"MDE";contMuitoDificileerrada2<-contMuitoDificileerrada2+1;Wdow<-"MDE"}
  if((mtxCM1[stepb,41] <= 2)&&((mtxCM1[stepb,2] >=
0)&&(mtxCM1[stepb,stepc]!=(mtxCM1[stepb,stepc])))){b_questao2PL[stepb]<-
"DE";contDificileerrada2<-contDificileerrada2+1;Wdow<-"DE"}

  #3 PL
  if((mtxCM1[stepb,42] < -
2)&&((mtxCM1[stepb,3]==mtxCM1[stepb,stepc]))){b_questo3PL[stepb]<-
"MFC";contMuitoFcilecorreto3<-contMuitoFcilecorreto3+1;Wdow<-"MFC"}
  if((mtxCM1[stepb,42] > -2)&&((mtxCM1[stepb,3] <
0)&&(mtxCM1[stepb,stepc]==(mtxCM1[stepb,stepc])))){b_questo3PL[stepb]<-
"FC";contFcilecorreto3<-contFcilecorreto3+1;Wdow<-"FC"}
  if((mtxCM1[stepb,42] >
2)&&((mtxCM1[stepb,3]==mtxCM1[stepb,stepc]))){b_questo3PL[stepb]<-
"MDC";contMuitoDificilecorreto3<-contMuitoDificilecorreto3+1;Wdow<-"MDC"}
  if((mtxCM1[stepb,42] <= 2)&&((mtxCM1[stepb,3] >=
0)&&(mtxCM1[stepb,stepc]==(mtxCM1[stepb,stepc])))){b_questo3PL[stepb]<-
"DC";contDificilecorreto3<-contDificilecorreto3+1;Wdow<-"DC"}
  if((mtxCM1[stepb,42] < -
2)&&((mtxCM1[stepb,3]!=mtxCM1[stepb,stepc]))){b_questo3PL[stepb]<-
"MFE";contMuitoFcileerrada3<-contMuitoFcileerrada3+1;Wdow<-"MFE"}
  if((mtxCM1[stepb,42] > -2)&&((mtxCM1[stepb,3] <
0)&&(mtxCM1[stepb,stepc]!=(mtxCM1[stepb,stepc])))){b_questo3PL[stepb]<-
"FE";contFcileerrada3<-contFcileerrada3+1;Wdow<-"FE"}
  if((mtxCM1[stepb,42] >
2)&&((mtxCM1[stepb,3]!=mtxCM1[stepb,stepc]))){b_questo3PL[stepb]<-
"MDE";contMuitoDificileerrada3<-contMuitoDificileerrada3+1;Wdow<-"MDE"}
```

```
  if((mtxCM1[stepb,42] <= 2)&&((mtxCM1[stepb,3] >=
0)&&(mtxCM1[stepb,stepc]!=(mtxCM1[stepb,stepc])))){b_questo3PL[stepb]<-
"DE";contDificileerrada3<-contDificileerrada3+1;Wdow<-"DE"}

  }

  print("#######################################")
  print(c(stepdataset,stepb,stepc))
  print("#######################################")

  vet1pl<-
c(contMuitoFcilecorreto1,contFcilecorreto1,contMuitoDificilecorreto1,contDificile
correto1,contMuitoFcileerrada1,contFcileerrada1,contMuitoDificileerrada1,contDi
ficileerrada1)
  vet2pl<-
c(contMuitoFcilecorreto2,contFcilecorreto2,contMuitoDificilecorreto2,contDificile
correto2,contMuitoFcileerrada2,contFcileerrada2,contMuitoDificileerrada2,contDi
ficileerrada2)
  vet3pl<-
c(contMuitoFcilecorreto3,contFcilecorreto3,contMuitoDificilecorreto3,contDificile
correto3,contMuitoFcileerrada3,contFcileerrada3,contMuitoDificileerrada3,contDi
ficileerrada3)

  mtxpls<-rbind(vet1pl,vet2pl,vet3pl)

  print(vet1pl)
  print(vet2pl)
  print(vet3pl)

  ###################################

  if (setMode==1){
    setwd("C:\\backup_Projeto\\Resultados\\OutputClusters\\50\\Plots\\OutPlots");
getwd();
  }

  if (setMode==2){

setwd("C:\\backup_Projeto\\Resultados\\OutputClusters\\100\\Plots\\OutPlots");
getwd();
  }

  colnames(mtxpls)<-c("MFC","FC","MDC","DC","MFE","FE","MDE","DE")
  mtxplsname<-paste0("Dataset",ListdataSet[stepdataset],"_mtxpls",".csv")
  write.csv(mtxpls, mtxplsname);

  histGraphcs(stepdataset)
  barrGraphcs(stepdataset)
  boxplotGraphcs(stepdataset)


  if (setMode==1){
    ###### 50
    # compare difficulto with cluster - mtxTargetClassificator_D1%1.csv
```

```
setwd("C:\\backup_Projeto\\OutputClusters\\50\\OutputClusters\\mtxTargetClassi
ficator"); getwd();

 }

 if (setMode==2){
  ###### 100
  # compare difficulto with cluster - mtxTargetClassificator_D1%1.csv

setwd("C:\\backup_Projeto\\Resultados\\OutputClusters\\100\\OutputClusters\\mt
xTargetClassificator"); getwd();
 }

 vet1pl<-c();vet2pl<-c();vet3pl<-c();mtxpls<-c()
}
mtxplsall<-rbind(mtxplsall,mtxpls)
}

print(mtxplsall)
```

```
# Code Statistics

rm(list=ls())

library(rstatix)
library(dunn.test)
require(PMCMR)
library(report,insight)

#library(report)

#setMetric<-1 #1 to 3
#SetDataset<-1 #c(1,3,4,5,13,14,17,19,20,21,24)
#SetAlg<-1
#c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,
31,32,33,34,35,36,37,38)
#setDif<-1 #1to3

vetPositon<-c();mtxposition<-c();vetKKsignificance<-c()
DatasetCorrenteVet<-c(1,3,4,5,13,14,17,19,20,21,24)
mtxCompP<-c(); contPvalues<-0;contexecs<-0

for(isetMetric in 1:3){
 for (iSetDataset in 1:11){
  for (iSetAlg in 1:38){
   for (isetDif in 1:3){

    setMetric<-isetMetric
    SetDataset<-DatasetCorrenteVet[iSetDataset]
    SetAlg<-iSetAlg
    setDif<-isetDif

path1<-paste0("E:\\backup_Projeto\\SMOTE\\Métrica
",setMetric,"\\vetMetricsStepAll")
```

```
path2<-paste0("E:\\backup_Projeto\\SMOTE\\Métrica ",setMetric,"\\vetDifficult")
setTarget1<-
paste0("vetMetricsStepAllD",SetDataset,"Alg",SetAlg,"Difficult",setDif,"_500.csv
")
setTarget2<-
paste0("vetDifficult",SetDataset,"D",SetDataset,"Alg",SetAlg,"All3Difficult500.cs
v")

print(setTarget1)
print(setTarget2)

# Load files
setwd(path1); getwd();
mtx500set1<-read.csv(setTarget1,header = TRUE, sep = ",")
setwd(path2); getwd();
mtx500b1<-read.csv(setTarget2,header = TRUE, sep = ",")

# Set data
dim(mtx500set1)
mtx500set<-mtx500set1[1:10,2:501]
dim(mtx500set)
mode(mtx500set)
mtx500set<-as.data.frame(mtx500set)

dim(mtx500b1)
mtx500b<-mtx500b1[1:3,2:501]
dim(mtx500b)
mode(mtx500b)
mtx500b<-as.data.frame(mtx500b)

mtxJoinData<-c()
mtxJoinData<-mtx500b

for(lineMxt in 1:10){
  lineMxt<-lineMxt+3
  mtxJoinData[lineMxt,]<-mtx500set[lineMxt,]
}

dim(mtxJoinData)
mode(mtxJoinData)
#mtxJoinData1<-as.numeric(mtxJoinData)
#mode(mtxJoinData1)

rownames(mtxJoinData)<-
as.character(c("1PL","2PL","3PL","PRE","SEN","SPE","RAND","JAC","FM","DU
NN","SIL","DB","PBM"))

KK_output<-kruskal.test(mtxJoinData)
#zz<-pairwise.wilcox.test(mtxJoinData,p.adjust.method = "bonferroni")
vetTemp<-c(setMetric,SetDataset,SetAlg,setDif,"p-value: ",KK_output$p.value)
contexecs<-contexecs+1
if(KK_output$p.value<=0.05){
  vetKKsignificance<-rbind(vetKKsignificance,vetTemp)
  contPvalues<-contPvalues+1;
}}}}}
```

```
PropPvalues<-0
PropPvalues<-(contPvalues/contexecs)

namemtxmtxCompP<-
paste0("C:\\backup_Projeto\\Resultados\\StatisisticsmtxCompP.csv")
write.csv(mtxCompP,namemtxmtxCompP)

namemtxMetrics<-
paste0("C:\\backup_Projeto\\Resultados\\StatisisticsMetrics.csv")
write.csv(mtxposition,namemtxMetrics)

namemtxSig<-paste0("C:\\backup_Projeto\\Resultados\\StatisisticSig.csv")
write.csv(vetKKsignificance,namemtxSig)

#######################
```

ANEXO A – Trabalho acadêmico publicado: **Teoria de Resposta ao Item e o Paradoxo de Moravec na obtenção de algoritmos ótimos**.

MOSTRA POLI/UPE 2021

Teoria de Resposta ao Item e o Paradoxo de Moravec na Obtenção de Algoritmos Ótimos

Paulo Fernando Leite-Filho, Centro de Informática - Universidade Federal de Pernambuco (paulo.leite@upe.br),
Silvio de Barros Melo, Centro de Informática - Universidade Federal de Pernambuco (sbm@cin.ufpe.br),
Rita Cassia-Moura, PPGES - Universidade de Pernambuco (cassia.moura@upe.br)

Introdução: O Paradoxo de Moravec aborda a complexidade na resolução de questões muito difíceis para os seres humanos e que são facilmente resolvidas pela Inteligência Artificial (IA), a qual tem dificuldade em resolver questões simples e que inclusive são resolvidas com facilidade por crianças, como a ação de andar e outras tarefas naturais aos seres humanos (SHAW, 2020). Identificar quando um algoritmo é ruim ou bom consiste no número de acertos em uma tarefa como a classificação de grupos, porém não identifica se o algoritmo acerta se as questões são fáceis ou difíceis (CHEN & AHN, 2020; DUDA et al., 2001). A Teoria da Resposta ao Item (TRI) possibilita diferenciar se questões são fáceis ou difíceis (MEC, 2021), e tem sido utilizada no Exame Nacional do Ensino Médio (Enem). **Objetivo**: Identificar algoritmos ótimos em questões difíceis ou fáceis. **Metodologia**: Foi empregado o Método da Máxima Verossimilhança do classificador de *Naive Bayes* para calcular a habilidade (θ) de resolver questões e o parâmetro da Dificuldade (b) do Modelo de 2 Parâmetros Logísticos (2PL) da TRI para classificar individualmente as questões consideradas fáceis ou não (MARTÍNEZ-PLUMED *et al.*, 2019) em 12 *datasets* disponíveis no *site* da *UCI*. Foi avaliado o desempenho de algoritmos do tipo não-supervisionados de *cluster* dos tipos *Fanny, Pam, Clara e Sota*, com os métodos de dissimilaridades de *Euclidean, Manhattan, Square Euclidean e Jaccard*. *Foi* calculado pelo Erro Quadrático Médio (do inglês *Mean Squared Error, MSE*) das classificações para determinar o melhor algoritmo dentre os testados. **Resultados:** Pelo parâmetro da Dificuldade (b) da TRI foram determinadas individualmente as questões fáceis ou não em cada um dos 12 *datasets*. O algoritmo de *k.means.HartiganWong* obteve o melhor *MSE* (0,998972) no *dataset Blood Transfusion* pelo cálculo dos *MSEs* das classificações. Nos 12 *datasets* pelo cálculo da Dificuldade (b) do Modelo 2PL da TRI, das classificações que foram acertadas pelos algoritmos, 45,86% foram difíceis e 22,46%, fáceis; e naquelas em que houve erro de classificação, 24,47% foram difíceis e 7,22%, fáceis. **Conclusão:** A possibilidade de identificar o grau de dificuldade de cada questão proporciona a seleção de quais algoritmos são ótimos para tarefas complexas.

Palavras-chave: *Paradoxo de Moravec; Machine Learning; Modelagem; Teoria da Resposta ao Item.*

Referências:

CHEN, Z.; AHN, H. Item Response Theory Based Ensemble in Machine Learning. **International Journal of Automation and Computing**, v. 17, n. 5, p. 621–636, 2020.

DUDA, R.; HART, P.; STORK, D. **Pattern Classification.** John Wiley & Sons Inc. West Sussex, 2001.

MARTÍNEZ-PLUMED, F.; PRUDÊNCIO, R.B.C.; MARTÍNEZ-USÓ, A.; HERNÁNDEZ-ORALLO, J. Item response theory in AI: Analysing machine learning classifiers at the instance level. **Artificial Intelligence**, v. 271, p. 18–42, 2019.

ANEXO B – Trabalho acadêmico publicado: **Métricas de rendimentos estatísticos em uma visão ampla dos dados**.

MÉTRICAS DE RENDIMENTOS ESTATÍSTICOS EM UMA VISÃO AMPLA DOS DADOS

Paulo Fernando Leite-Filho, Sílvio De Barros Melo, Rita Cassia-Moura.
CAMPUS SANTO AMARO – PRONTO SOCORRO CARDIOLÓGICO DE PERNAMBUCO – UNIVERSIDADE DE PERNAMBUCO
CENTRO DE INFORMÁTICA – UNIVERSIDADE FEDERAL DE PERNAMBUCO
CAMPUS BENFICA – PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE SISTEMAS, ESCOLA POLITÉCNICA DE PERNAMBUCO – UNIVERSIDADE DE PERNAMBUCO

Introdução: As métricas de rendimentos estatísticos são formas de avaliar os resultados em pesquisas científicas. Os algoritmos computacionais utilizam estas abordagens para identificar as melhores soluções para problemas distintos. Porém são obtidos diferentes resultados aplicando-se métricas distintas. Para isto as métricas podem ser ajustadas aos conjuntos de dados e aos resultados obtidos cientificamente. Objetivo: Avaliar as correlações das métricas de rendimento estatístico em sua completude das informações dos dados. Metodologia: Foram calculadas as métricas de *Accuracy, Precision, Specificity e Sensitivity* de cinco conjuntos de dados abertos em saúde, disponíveis no *site* de bases da *UCI*: *Echocardiogram, Heart-statlog, Parkinson, Blood Transfusion, Diabetes*. Foram calculadas as métricas pelos seguintes algoritmos de agrupamento: *Fanny, Pam, Clara, Sota; e nas seguintes medidas de dissimilaridades: Euclidean, Manhattan, Square Euclidean, Jaccard*. Foram correlacionadas estatisticamente as métricas por Spearman para avaliar as codominâncias. Resultados: Com o resultado das métricas para cada algoritmo, foi calculado o teste de *Shapiro-Wilk*, no qual os p-valores refutaram a normalidade com valor >= 5%. Com o cálculo de *Spearman* para cada métrica, 83% das correlações positivas corresponderam a 30,85% muito fracas, e destas foram obtidos os percentuais significativos de 10% para cada métrica de: *Precision x Accuracy; Accuracy x Specificity; Sensitivity x Specificity; Specificity x ROC;* e no mínimo de 5,83% para as demais correlações. Conclusão: A correlação positiva de *Sensitivity x Specificity* foi confirmada, e em destaque *Precision x Accuracy* e *Specificity x ROC*, ambas 10% com prevalência dominante de correlações positivas para todas as métricas.

Palavra-chave: Métricas, Rendimentos Estatísticos, Classificação, *Cluster*.

ANEXO C – Trabalho acadêmico publicado: **Tomada de decisões baseada no parâmetro *b* da Teoria da Resposta ao Item**.

RMP
Revista dos Mestrados Profissionais
UFPE / CCSA – MGP
https://periodicos.ufpe.br/revistas/rmp
ISSN – 2317 - 0115
V. 11 - n. 2 (2022)
Recife – PE

Tomada de decisões baseada no parâmetro *b* da Teoria da Resposta ao Item nas classificações de *ensemble*

Paulo Fernando Leite-Filho – Universidade Federal de Pernambuco
https://www.orcid.org/0000-0003-1345-8259 - pflf@cin.ufpe.br
Silvio Barros Melo – Universidade Federal de Pernambuco
https://www.orcid.org/0000-0001-8600-6427 - sbm@cin.ufpe.br

Resumo – A tomada de decisões baseada no auxílio de sistemas informatizados é uma tarefa complexa e alvo de muitos ajustes para evitar erros de classificações. Avaliar as classificações pelos acertos dos algoritmos e associar estes acertos com o parâmetro da dificuldade (*b*) da TRI em concordância com a estatística de Kappa proporciona uma maior segurança ao usuário de sistemas informatizados. Nesta avaliação entre os grupos de *undersample* e *oversample* foi possível selecionar grupos específicos de algoritmos de cluster baseados no intervalo de confiança de 5% e intensidade de concordância maiores que 0,9968 identificando concordância perfeita (0,48%) nos grupos de *SMOTE*.

Palavras-chave: Tomada de decisão, Teoria da Resposta ao Item, Classificação *Ensemble*

Decision making based on Item Response Theory b parameter in ensemble rankings

Abstract: Decision-making based on the aid of computerized systems is a complex task and the target of many adjustments to avoid classification errors. Evaluating the classifications by the hits of the algorithms and associating these hits with the Difficulty parameter (b) of the IRT in agreement with the Kappa statistics provides greater security to the user of computerized systems. In this evaluation between the undersample and oversample groups, it was possible to select specific groups of cluster algorithms based on the 5% confidence interval and agreement intensity greater than 0.9968, identifying Perfect agreement (0.48%) in the SMOTE groups.

Keywords: Decision-making, Item Response Theory, Ensemble, Classification

Data da Submissão: 13/08/2022 **Data de aceitação:** 11/12/2022

Este artigo está licenciado sob forma de uma licença Creative Commons Atribuição-Não Comercial-Sem Derivações 4.0 Internacional (CC BY-NC-ND 4.0).
https://creativecommons.org/licenses/by-nc-nd/4.0/

https://doi.org/10.51359/2317-0115.2022.256869

ANEXO D – Trabalho acadêmico publicado: ***Significance of Item Response Theory in Cluster Evaluation Metrics***.

Significance of Item Response Theory in Cluster Evaluation Metrics

Paulo Fernando Leite-Filho
Computer Center (CIn)
Universidade Federal de Pernambuco (UFPE)
Recife, Brazil
pflf@cin.ufpe.br

Silvio de Barros Melo
Computer Center (CIn)
Universidade Federal de Pernambuco (UFPE)
Recife, Brazil
sbm@cin.ufpe.br

***Abstract*—Artificial intelligence (AI) advances in different systems as a decision-making factor. Certain metrics can be used to gauge the success or failure of AI ranking. Item Response Theory (IRT) is widely known for being able to clearly identify the difficulty level of questions on educational tests. The importance of the metrics and the statistical correspondence between the classifications, considering the level of difficulty to be inferred by the TRI make up the objectives of this work, providing an assessment of how difficulties impact metrics. The ratings of 34 unsupervised algorithms were compared against the IRT difficulty parameter b on 11 datasets. Kruskal Wallis statistics were applied to identify the relationship between b values and algorithm rankings. The results indicate that 20.59% of the samples are statistically equivalent to those of variable b. This percentage is proportional to 157,078 of the total data statistically analyzed with a significance of α <= 0.05.**

***Keywords*—Item Response Theory, Classification, Metrics, Machine Learning**

2023 XLIX Latin American Computer Conference (CLEI) | 979-8-3503-1887-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/CLEI60451.2023.10346171

1 Introdução

A inteligência Artificial (IA) contribui constantemente nos avanços das aplicações da área da computação nos diversos contextos sociais e profissionais. Considerar uma aplicação inteligente consiste na aprovação do modelo computacional no teste de Alan Turing. O teste de Turing propõe comparar um sistema artificial com um humano, e cabe ao sistema artificial convencer o ser humano que pode ser comparado a uma pessoa em suas respostas e comportamento[1].

A IA proporciona soluções importantes em muitas áreas de aplicação, gerando um fator de grande impacto nas soluções de tomada de decisão. Para identificar os principais fatores relevantes para a tomada de decisão e a IA menos complexa para os seres humanos é uma tarefa difícil, e muitas vezes um desafio para a computação [2].

O uso de métricas de avaliação é uma forma de avaliar se a IA acerta ou erra uma classificação, regressão, agrupamento ou outros objetivos a serem alcançados pela IA. A aplicação das métricas se preocupa cada vez mais em medir a distância e a semelhança entre diferentes grupos, especialmente na classificação [3].

As métricas podem ser categorizadas como modelos avaliativos dos classificadores, porém a avaliação não identifica a diferença entre acertar classificações fáceis ou difíceis. A Teoria da Resposta ao Item (TRI) tem grande contribuição para identificar estas atribuições de dificuldades, auxiliando muitas vezes as classificações pela IA. A TRI complementa a tarefa de auxiliar sistemas baseados em IA nas tomadas de decisões [2]. O refino na seleção de questões fáceis ou difíceis proporciona identificar juntamente com as métricas quais algoritmos tem melhor desempenho em ambientes de decisões difíceis.

Pode-se atribuir parâmetros associados aos classificadores que acertem corretamente instâncias difíceis de classificar, melhorando a eficiência da Aprendizagem de Máquina supervisionada ou não supervisionada. O método de treinamento não supervisionado exige uma maior atenção, pois a decisão está totalmente direcionada aos modelos obtidos sem a intervenção humana, e atribuir o poder de decisão a este método exige que o sistema seja o mais preciso possível [4].

Há diversos trabalhos que utilizam as avaliações das métricas aplicando em classificações educacionais dos estudantes como [5] e [6]. Os objetivos deste trabalho são a análise das métricas de avaliação e suas correspondências estatísticas nas classificações dos algoritmos e as correspondências com as dificuldades calculadas pela TRI.

Este trabalho está organizado apresentando trabalhos científicos no Estado da arte no Capítulo 2, onde serão abordadas: as Aplicações da TRI, o conceito, os Modelos Logísticos, os Gráficos da TRI, a aplicação da TRI na Aprendizagem de Máquina e as métricas tradicionalmente utilizadas. No Capítulo 3 será descrito o método utilizado contendo os algoritmos e *datasets* avaliados. No Capítulo 4 serão apresentados e discutidos os resultados obtidos, seguidos da conclusão no Capítulo 5.

2 Estado da arte

Os algoritmos de IA estão cada dia mais eficientes em seus objetivos, obtendo bons resultados em diferentes tipos de *datasets* [7]. Este trabalho apresenta um método mais eficiente de identificar quais algoritmos são ótimos em determinadas situação, mediante a identificação das dificuldades das classificações dos itens dos *datasets*. A análise dos parâmetros do método da TRI tem sido considerada como uma solução para selecionar registros considerados difíceis de classificar, possibilitando uma forma de avaliar complexidades dos dados a serem triados por algoritmos de *Aprendizagem de Máquina*.

Nesta seção serão apresentados os tópicos principais citados na literatura e os avanços em cada tópico. Serão apresentadas comparações entre os algoritmos e estudantes: i) cada questão das avaliações corresponde a um item do *dataset*; ii) os estudantes serão os algoritmos e iii) as formas de avaliações das correções das provas serão as métricas de avaliação.

979-8-3503-1887-6/23/$31.00 ©2023 IEEE

Authorized licensed use limited to: UNIVERSIDADE DE PERNAMBUO. Downloaded on December 26,2023 at 23:37:48 UTC from IEEE Xplore. Restrictions apply.

ANEXO E – Prêmio de melhor trabalho.

Certificado

Escola Politécnica da Universidade de Pernambuco

Certificamos para os devidos fins que o trabalho intitulado, **Teoria de Reposta ao Item e o Paradoxo de Moravec na Obtenção de Algoritmos Ótimos**, de autoria de **Paulo Fernando Leite Filho, Rita de Cassia Moura do Nascimento e Silvio de Barros Melo**, apresentado na **Mostra de Extensão, Inovação e Pesquisa POLI/UPE 2021**, foi premiado como melhor trabalho na categoria **Pós Mestrado/Doutorado** do curso de Engenharia de Computação - Sistemas, durante o **INGENIA**, evento integrante da Semana Universitária da UPE, realizado no dia 07 de outubro de 2021 na Escola Politécnica de Pernambuco.

Recife, 16 de março de 2022.

Prof. Alberto Casado Lordsleem Jr., Dr.
Coordenador Setorial de Extensão e Cultura
Escola Politécnica da Universidade de Pernambuco

Prof.Hiran Ferreira de Lira, Mestre
Gerente da Divisão de Extensão
Escola Politécnica da Universidade de Pernambuco

Prof.ª Roberta Andrade de A. Fagundes, Dra
Gerente da Divisão de Cultura
Escola Politécnica da Universidade de Pernambuco