

Universidade Federal do Piauí
Centro de Ciências da Natureza
Programa de Pós-Graduação em Ciência da Computação

União de Dados por Clusterização para Construção de Modelos de Predição de Evasão

Cledjan Torres da Costa

Teresina-PI, Junho de 2024

Cledjan Torres da Costa

União de Dados por Clusterização para Construção de Modelos de Predição de Evasão

Defesa de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da UFPI (área de concentração: Sistemas de Computação), como parte dos requisitos necessários para a obtenção do Título de Mestre em Ciência da Computação.

Universidade Federal do Piauí – UFPI

Centro de Ciências da Natureza

Programa de Pós-Graduação em Ciência da Computação

Orientador: André Macedo Santana

Teresina-PI

Junho de 2024

FICHA CATALOGRÁFICA
Universidade Federal do Piauí
Biblioteca Comunitária Jornalista Carlos Castello Branco
Divisão de Representação da Informação

C837u Costa, Cledjan Torres da.
União de dados por clusterização para construção de Modelos de Predição de Evasão / Cledjan Torres da Costa. -- 2024.
145 f.

Dissertação (Mestrado em Ciência da Computação)– Universidade Federal do Piauí, Centro de Ciências da Natureza, Programa de Pós-Graduação em Ciência da Computação, Teresina, 2024.
"Orientação: Prof. Dr. André Macedo Santana".

1. Mineração de dados educacionais. 2. Predição de evasão. 3. Clusterização. 4. Classificação. I. Santana, André Macedo. II. Título.

CDD - 006.32

Bibliotecário: Rigoberto Veloso de Carvalho – CRB-3/988

Cledjan Torres da Costa

União de Dados por Clusterização para Construção de Modelos de Predição de Evasão

Defesa de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da UFPI (área de concentração: Sistemas de Computação), como parte dos requisitos necessários para a obtenção do Título de Mestre em Ciência da Computação.

Trabalho aprovado. Teresina-PI, 08 de junho de 2024:

Documento assinado digitalmente
gov.br
ANDRE MACEDO SANTANA
Data: 22/07/2024 16:15:28-0300
Verifique em https://validar.iti.gov.br

André Macedo Santana
Orientador

Documento assinado digitalmente
gov.br
RICARDO DE ANDRADE LIRA RABELO
Data: 23/07/2024 09:26:09-0300
Verifique em https://validar.iti.gov.br

Ricardo de Andrade Lira Rabelo
Convidado 1

Documento assinado digitalmente
gov.br
VINICIUS PONTE MACHADO
Data: 24/07/2024 10:03:50-0300
Verifique em https://validar.iti.gov.br

Vinicius Ponte Machado
Convidado 2

Documento assinado digitalmente
gov.br
IALIS CAVALCANTE DE PAULA JUNIOR
Data: 24/07/2024 23:17:30-0300
Verifique em https://validar.iti.gov.br

Iális Cavalcante de Paula Júnior
Convidado 3

Teresina-PI
Junho de 2024

À Deus, cuja luz ilumina meu caminho e cujo amparo sustenta cada passo. Ao meu esposo, cujo amor, apoio, encorajamento e paciência têm sido minha inspiração constante. Ao meu filho, cuja presença é a fonte de minha força e coragem, moldando-me em um ser humano melhor a cada dia. Este trabalho é dedicado a vocês, com todo o meu amor.

Agradecimentos

Agradeço ao meu esposo, cujo apoio, compreensão e amor foram fundamentais para enfrentar os desafios e alcançar os objetivos deste trabalho. Agradeço também ao meu filho, por ser minha fonte de alegria, inspiração e motivação constante.

Ao Maurilio Lacerda Leonel Junior por sua inestimável parceria no desenvolvimento deste projeto, sempre disponível à ajudar trazendo boas idéias que enriqueceram o trabalho, e aos demais amigos da Superintendência de Tecnologia da Informação/UFPI que contribuíram significativamente para a compreensão dos dados utilizados e para a identificação dos melhores atributos, compartilhando suas vastas experiências no negócio.

Agradeço ao meu orientador, André Macedo Santana, por me conduzir nessa longa jornada de aprendizagem, pelos preciosos conselhos, pela paciência e pela assistência dedicada ao longo deste período, e aos demais professores do Programa de Pós-Graduação em Ciência da Computação/UFPI, cujos ensinamentos foram de inestimável valor para o desenvolvimento deste trabalho.

"Por vezes sentimos que aquilo que fazemos
não é senão uma gota de água no mar.
Mas o mar seria menor se lhe faltasse uma gota."
(Madre Teresa de Calcutá)

Resumo

O rápido avanço da tecnologia, sobretudo nas instituições de ensino, tem dado origem a vastos volumes de dados que, por meio do processo de Descoberta de Conhecimento em Bancos de Dados, têm o potencial de proporcionar consideráveis benefícios a instituições, estudantes, professores e colaboradores. Este estudo se concentra na aplicação desse processo para a previsão da evasão escolar, enfocando uma situação em que os conjuntos de dados podem não ser tão extensos quando considerados em um contexto de cursos individuais. No entanto, quando esses conjuntos são agrupados com outros que compartilham características semelhantes, podem aprimorar significativamente a capacidade preditiva. Para atingir esse objetivo, adotou-se uma abordagem de duas etapas em cascata. Na primeira etapa, uma abordagem não supervisionada de clusterização foi empregada, visando agrupar cursos com características afins. A segunda etapa, por sua vez, empregou uma técnica supervisionada de classificação para criar modelos de predição de evasão. Esses modelos foram criados tanto de forma individualizada para cada curso como a partir de dados de cursos agrupados, seguindo a estrutura definida pelo algoritmo de clusterização da etapa anterior. Essa abordagem permitiu uma comparação direta entre os modelos. Além disso, a temporalidade dos dados foi levada em consideração, com a divisão dos dados em períodos, do 1º ao 8º, sendo o último, na maioria dos cursos, o semestre de conclusão esperado. Os resultados obtidos destacam a eficácia da utilização de agrupamentos de cursos por meio da clusterização para otimizar a capacidade preditiva dos modelos de evasão escolar.

Palavras-chaves: mineração de dados educacionais. predição de evasão. clusterização. classificação.

Abstract

The rapid advancement of technology, especially in educational institutions, has given rise to vast volumes of data that, through the process of Knowledge Discovery in Databases, have the potential to provide significant benefits to institutions, students, teachers, and staff. This study focuses on the application of this process for predicting school dropout, with a focus on a situation where the data sets may not be as extensive when considered in the context of individual courses. However, when these sets are grouped with others sharing similar characteristics, they can significantly enhance predictive capacity. To achieve this goal, a two-step cascading approach was adopted. In the first step, an unsupervised clustering approach was used to group courses with similar characteristics. The second step, in turn, employed a supervised classification technique to create dropout prediction models. These models were created both individually for each course and based on data from grouped courses, following the structure defined by the clustering algorithm from the previous step. This approach allowed for a direct comparison between the models. Additionally, the temporal aspect of the data was taken into account, with data divided into periods, from the 1st to the 8th, with the latter, in most courses, representing the expected semester of completion. The results highlight the effectiveness of using course clusters through clustering to optimize the predictive capacity of school dropout models.

Keywords: educational data mining. dropout prediction. clustering. classification.

Lista de ilustrações

Figura 1 – Indicadores de trajetória dos ingressantes em 2012 dos cursos de Graduação (Fonte: Elaborada pela Deed/Inep com base em dados do Censo da Educação Superior). 1
Figura 2 – Processo proposto. 8
Figura 3 – Artigo publicado. 10
Figura 4 – O ciclo do processo de KDD (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). 14
Figura 5 – Representação de um modelo classificador (COSTA et al., 2012). 27
Figura 6 – Três maneiras diferentes de agrupar o mesmo conjunto de pontos. (TAN et al., 2019). 28
Figura 7 – Utilizando o algoritmo *K-Means* para encontrar três *clusters* em dados de amostra (TAN et al., 2019). 31
Figura 8 – Visualização de grupos com compacidades e separabilidades baixas e altas (SILVA; PERES; BOSCARIOLI, 2016). 36
Figura 9 – Hierarquia do aprendizado indutivo. 41
Figura 10 – Árvore de decisão para previsão de evasão utilizando o algoritmo J48 (BARRETO et al., 2019). 43
Figura 11 – Comparativo entre técnicas Supervisionada e Não-Supervisionada. . . . 46
Figura 12 – Coleta efetuada para as duas etapas do processo. 55
Figura 13 – Mapa de calor de correlação de *Pearson* para os atributos da etapa Não Supervisionada. 58
Figura 14 – Curva de Cotovelo. 59
Figura 15 – Coeficiente de Silhueta para Diferentes Valores de K. 60
Figura 16 – Gráfico de dispersão com o resultado do *K-Means* para $K = 3$. 61
Figura 17 – Gráfico de dispersão com o resultado do *K-Means* para $K = 5$. 62
Figura 18 – Gráfico de dispersão com o resultado do *K-Medoids* para $K = 3$. 62
Figura 19 – Gráfico de dispersão com o resultado do *DBSCAN* para *eps=0.2* e *minsamples=2*. 63
Figura 20 – Dendrograma Hierárquico. 63
Figura 21 – Atributos mais frequentes presentes nas listas dos 10 melhores atributos de cada modelo. 67
Figura 22 – Ranqueamento dos atributos por ganho de informação dos conjuntos dos grupos G1, G2 e G3 para cada período. 69
Figura 23 – Totais de exemplos de cada conjunto de dados com destaque para a classe “evadido”. 71
Figura 24 – Proporção da classe "evadido"para cada periodo e curso. 72

Figura 25 – Diagramação da técnica desenvolvida. 74
Figura 26 – Total de evadidos por período e curso comparados aos dos agrupamentos G1(a), G2(b) e G3(c). 75
Figura 27 – Resultados das métricas RC e MCC do 1º ao 8º período para *Random Forest*, *Gaussian Naive Bayes*, *Multi-layer Perceptron* e *K-Nearest Neighbors*. 79
Figura 28 – Comparação entre os algoritmos *Random Forest*, *Gaussian Naive Bayes*, *Multi-layer Perceptron* e *K-Nearest Neighbors*. 81
Figura 29 – Porcentagem de aplicabilidade e melhora significativa da solução proposta. 86

Lista de tabelas

Tabela 1 – Resumo dos Trabalhos Relacionados 8
Tabela 2 – Totais de dados presentes nos modelos de dados dos períodos. 9
Tabela 3 – Visão geral dos métodos de agrupamento (HAN; PEI; KAMBER, 2012). 29
Tabela 4 – Representação de uma matriz de confusão para classificadores binários. 38
Tabela 5 – Atributos coletados/criados para repositórios R1 e R2 durante a coleta. 54
Tabela 6 – Atributos cumulativos criados na etapa de coleta para R2. 55
Tabela 7 – Atributos utilizados na mineração de dados da etapa Não Supervisionada. 59
Tabela 8 – Agrupamentos formados pelo *K-Means* para K=3 e K=5 clusters. . . . 60
Tabela 9 – Atributos utilizados na mineração de dados da etapa supervisionada. . 66
Tabela 10 – Atributos menos significativos para os modelos dos grupos. 68
Tabela 11 – Percentual de aumento de dados proporcionado pelos agrupamentos em comparação aos cursos. 76
Tabela 12 – Teste Z - RC. 84
Tabela 13 – Teste Z - MCC. 85
Tabela 14 – Resultados das métricas RC e MCC para todos os modelos e algoritmos aplicados. 105
Tabela 15 – Ranqueamento dos atributos dos grupos em cada período. 121

Lista de abreviaturas e siglas

ADM	Administração
AUC-ROC	Área Sob a Curva da Característica de Operação do Receptor
AM	Aprendizado de Máquina
BIOM	Biomedicina
BIO	Ciências Biológicas
D1	Dataset de Trabalho 1
D2	Dataset de Trabalho 2
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
CON	Ciências Contábeis
COVID - 19	Coronavirus Disease
ECON	Ciências Econômicas
FIS	Fisioterapia
FN	Falsos Negativos
FP	Falsos Positivos
GaussianNB	Gaussian Naive Bayes
G1	Grupo 1
G2	Grupo 2
G3	Grupo 3
IA	Inteligência Artificial
IFES	Instituição Federal de Ensino Superior
KDD	Knowledge Discovery in Databases
KNeighborsClassifier	K-Nearest Neighbors
MCC	Matthews Correlation Coefficient

MD	Mineração de Dados
MDE	Mineração de Dados Educacionais
MAT	Matemática
MLP	Multi-Layer Perceptron
MOODLE	Modular Object-Oriented Dynamic Learning Environment
PCA	Principal Component Analysis
PED	Pedagogia
PES	Engenharia de Pesca
PSI	Psicologia
R1	Repositório 1
R2	Repositório 2
RC	Recall
SIGAA	Sistema Integrado de Gestão de Atividades Acadêmicas
TUR	Turismo
UFPI	Universidade Federal do Piauí
UFDPar	Universidade Federal do Delta do Parnaíba
VN	Verdadeiros Negativos
VP	Verdadeiros Positivos
WCSS	Within-Cluster Sum of Square

Sumário

1 INTRODUÇÃO 1
1.1 Objetivos 3
1.2 Trabalhos Relacionados 4
1.3 Visão Geral da Proposta 7
1.4 Organização do Trabalho 9

I REFERENCIAL TEÓRICO 11

2 REFERENCIAL TEÓRICO 13
2.1 Descoberta de Conhecimento em Base de Dados 13
2.1.1 Coleta 14
2.1.2 Pré-Processamento 16
2.1.2.1 Limpeza dos Dados 16
2.1.2.2 Redução de Dados 18
2.1.2.3 Balanceamento dos Dados 20
2.1.3 Transformação 24
2.1.3.1 Normalização 24
2.1.3.2 Discretização de atributos quantitativos 25
2.1.3.3 Transformação de atributos qualitativos em quantitativos 25
2.1.3.4 Atributos de tipos de dados complexos 26
2.1.3.5 Construção de atributos 26
2.1.4 Mineração de Dados 26
2.1.4.1 Predição 27
2.1.4.2 Agrupamento 28
2.1.4.3 Mineração de Relações 36
2.1.5 Análise e Interpretação dos Resultados 37
2.1.5.1 Métricas de Avaliação de Classificadores 37
2.2 Aprendizado de Máquina 40
2.2.1 Aprendizado Supervisionado 41
2.2.1.1 Paradigma Simbólico 42
2.2.1.2 Paradigma Estatístico 43
2.2.1.3 Paradigma Baseado em Exemplos 44
2.2.1.4 Paradigma Conexionista 44
2.2.2 Aprendizado Não-Supervisionado 45

II PROPOSTA E RESULTADOS 47

3 METODOLOGIA PROPOSTA 49
3.1 Ferramentas Utilizadas 49
3.2 O Processo Proposto 50
3.3 Coleta 50
3.4 Etapa 1 – Não Supervisionada 54
3.4.1 Pré-Processamento: 54
3.4.2 Transformação 57
3.4.3 Mineração de Dados 58
3.4.4 Resultados 60
3.5 Etapa 2 – Supervisionada 63
3.5.1 Pré-Processamento 64
3.5.1.1 Análise dos Atributos 67
3.5.2 Transformação 68
3.5.3 Mineração de Dados 70

4 RESULTADOS E DISCUSSÃO 75
4.1 Resultados 76
4.2 Avaliação dos Resultados 76
4.2.1 Avaliação dos Algoritmos 76
4.2.2 Avaliação dos Desempenhos dos Grupos em Relação aos Cursos 83

III PARTE FINAL 89

5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS 91
5.1 Considerações Finais 91
5.1.1 Desafios 93
5.1.2 Contribuições 93
5.1.3 Limitações 94
5.2 Trabalhos Futuros 94

REFERÊNCIAS 97

ANEXOS 103

ANEXO A – RESULTADOS DA ETAPA 2 105

ANEXO B – RANQUEAMENTO DOS ATRIBUTOS DOS GRUPOS 121

1 Introdução

Em 2021, quase 4 milhões de alunos ingressaram em cursos de graduação. Desse total, 87,5% em instituições privadas. No período compreendido entre 2011 e 2021, a rede privada cresceu 86,0% e a rede pública aumentou 0,3% no mesmo período. Com relação aos estudantes que concluíram cursos no mesmo nível acadêmico, tem-se mais de 1,3 milhão, sendo que apenas 16,5% são da rede pública. A variação percentual do número de concluintes em cursos de graduação foi maior na rede privada, com 38,8%, na pública esse crescimento foi de 0,4% no mesmo período (INEP, 2022). Esses números indicam que as instituições de ensino estão contribuindo com um número de profissionais abaixo das expectativas para a sociedade.

Segundo INEP (2022) 59% dos alunos que ingressaram em seus cursos de Graduação em 2012 evadiram até 2021 (39% até o final do 3º ano após ingresso), conforme pode ser observado na Figura 1. Na instituição estudada, Universidade Federal do Delta do Parnaíba (UFDPar), uma IFES recém criada (2019) através de desmembramento de um Campi da Universidade Federal do Piauí (UFPI), esses números chegam a ser bem mais altos, como em seus cursos de Engenharia de Pesca e Matemática que tem sua Taxa de Desistência Acumulada até 2021 igual a 75% e 79,5% respectivamente, no mesmo coorte (Ingressantes em 2012).

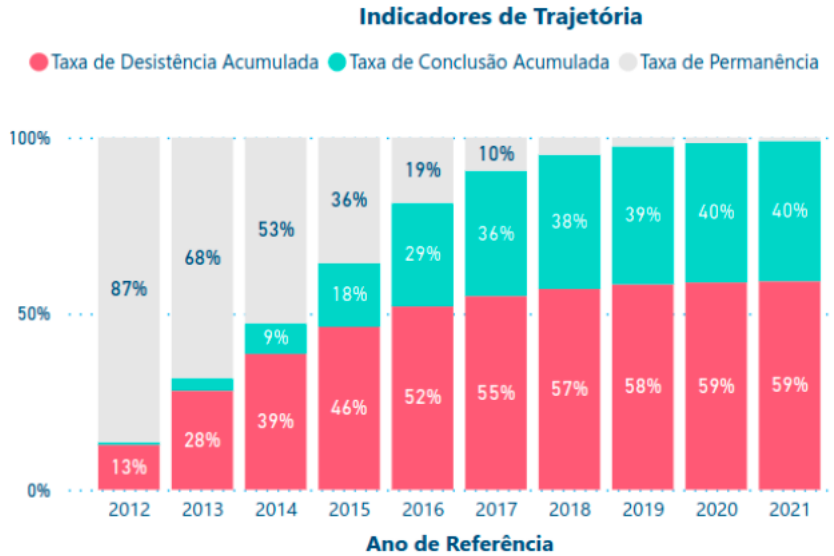


Figura 1 – Indicadores de trajetória dos ingressantes em 2012 dos cursos de Graduação (Fonte: Elaborada pela Deed/Inep com base em dados do Censo da Educação Superior).

O conceito atual de evasão possui aspectos que podem dificultar os diagnósticos e prognósticos para a elaboração de políticas públicas de combate a evasão (SILVA; MARIANO, 2021), de modo que para este estudo trataremos simplesmente como a saída

definitiva do estudante do curso de origem, sem concluí-lo (ANDIFES; ABRUEM et al., 1996), em nível de ensino de graduação de uma Instituição Federal de Ensino Superior (IFES). A evasão é um problema complexo que afeta a educação em vários níveis e gera prejuízos significativos, tanto sociais, econômicos, políticos, acadêmicos e financeiros. É fundamental desenvolver métodos de previsão de alunos em risco de evasão para tomar ações preventivas e reduzir seus impactos (MARTINHO; NUNES; MINUSSI, 2013).

À medida que a quantidade de dados educacionais disponíveis cresceu, tornou-se evidente que esses dados continham informações valiosas que poderiam ser exploradas para identificar padrões e tendências, surgindo a necessidade de se utilizar ferramentas específicas e técnicas de mineração de dados para explorar adequadamente esses dados, visando obter informações de grande importância e, com elas, ser possível criar ações estratégicas, trazendo benefícios tanto para a instituição como para seus alunos, profissionais relacionados ao contexto e sociedade como um todo.

Nesse contexto se destaca o *Knowledge Discovery in Databases* - KDD, também conhecido como Descoberta de Conhecimento em Bases de Dados. Segundo Fayyad, Piatetsky-Shapiro e Smyth (1996), é um processo não trivial de identificar padrões válidos, novos, potencialmente úteis e, em última análise, compreensíveis em dados. É abrangente, composto por várias etapas, como Coleta, Preparação dos Dados e Mineração de Dados, esta última a qual se concentra nas técnicas específicas de análise e descoberta de padrões, chamada de Mineração de Dados Educacionais (MDE), quando aplicada a dados no contexto educacional.

A MDE é uma área que explora estatística, aprendizado de máquina e algoritmos de mineração de dados aplicados a diferentes tipos de dados de ensino e permite descobrir novos conhecimentos, de modo a estabelecer bases para um processo de aprendizagem mais eficaz (ROMERO; VENTURA, 2010). Ela advém de uma das muitas aplicações dadas à Mineração de Dados, e se subdivide em várias linhas de pesquisa, dentre as quais a Predição, o Agrupamento e a Mineração de Relações (BAKER; ISOTANI; CARVALHO, 2011).

Segundo Manhães e Cruz (2019), existem diversos obstáculos que por vezes impedem melhores resultados dos modelos de predição, como a quantidade de dados disponíveis, em especial para instituições com informatização recente dos dados referentes à vida acadêmica dos discentes. A precisão da predição, taxa de acertos, entre outras coisas, depende da qualidade e da quantidade de dados de entrada. Esta, quando disponível em pouca quantidade, pode trazer uma série de problemas, como a dificuldade de identificar padrões significativos nos dados, *Overfitting* (superajuste), *Underfitting* (sub-ajuste) e inclusão de viés. Isso cria um desafio substancial para a aplicação eficaz da Mineração de Dados. Segundo Li et al. (2017), obter um alto desempenho de classificação em conjuntos de dados desequilibrados é uma tarefa desafiadora para os modelos de aprendizagem. Isso

ocorre porque, em conjuntos desequilibrados, onde uma das classes é substancialmente maior que as outras, a maioria dos algoritmos de aprendizado de máquina e mineração de dados tende a ser excessivamente influenciada pelas classes majoritárias, enquanto negligencia as classes minoritárias. Como consequência, esses algoritmos frequentemente apresentam um baixo desempenho de aprendizagem devido à sua lenta adaptação às classes menos representadas. Hu et al. (2019) destaca que modelos de *Deep Learning* treinados em pequenos conjuntos de dados mostram o baixo desempenho de versatilidade e generalização do conjunto de validação e teste. Portanto, esses modelos sofrem com o problema causado pelo *Overfitting*.

Existem várias estratégias que podem ajudar a reduzir os problemas causados pela baixa quantidade de dados, parte delas envolvendo a geração sintética de dados. Essa abordagem não apenas aumenta a quantidade e a diversidade dos dados, mas também é eficaz para mitigar o *Overfitting* nos modelos e aprimorar o desempenho de generalização (KRIZHEVSKY; SUTSKEVER; HINTON, 2012 apud LEI et al., 2019). Entretanto, a criação de instâncias artificiais inadequadas resulta na inserção de ruído e na diminuição do desempenho do classificador (CHAU; GARCÍA; CERVANTES, 2015).

1.1 Objetivos

A UFDPar, instituição alvo deste estudo, foi criada em 2019 a partir do desmembramento de um campi avançado da UFPI para tornar-se uma universidade autônoma. Os dados relativos aos cursos e estudantes só foram devidamente organizados e estruturados para fins de estudos de Mineração a partir do ano de 2012, após a implantação de um novo sistema acadêmico, restando um volume limitado de informações chave. Essa escassez de dados por curso é um fator dificultador no uso de ferramentas de aprendizagem de máquina para a predição de evasão, o que impõe um grande desafio, tornando a instituição um bom *case* para este estudo.

Ao contrário de instituições acadêmicas maiores e mais antigas, a UFDPar não estrutura seus cursos em departamentos, centros de ensino ou faculdades tradicionais. Ela oferece 12 cursos, a saber: Psicologia (PSI), Pedagogia (PED), Ciências Contábeis (CON), Ciências Econômicas (ECON), Administração (ADM), Turismo (TUR), Biomedicina (BIOM), Fisioterapia (FIS), Engenharia de Pesca (PES), Ciências Biológicas (BIO), Matemática (MAT) e Medicina. Todos os cursos foram incluídos neste estudo, com exceção do curso de Medicina, que foi implantado posteriormente aos demais. Esse curso concluiu sua primeira turma durante a pandemia de COVID-19. Os alunos tiveram suas colações de grau antecipadas, sem completar integralmente a carga horária exigida, para contribuir no combate à doença (conforme a Portaria 374/2020 do MEC). Além disso, o curso apresenta uma baixíssima taxa de evasão. Por conta destas peculiaridades, ele foi descartado do

estudo.

Diante desse cenário, a principal questão que esta pesquisa pretende abordar é: **"É possível desenvolver modelos de predição confiáveis a partir dos dados disponíveis no contexto do estudo?"**

O objetivo é desenvolver modelos confiáveis para a predição de evasão escolar na UFDPar, utilizando dados ampliados por meio da combinação de informações de vários cursos com características similares. Pretende-se demonstrar que esses modelos, quando treinados com dados agrupados, apresentam melhores resultados em termos de representatividade e capacidade de generalização do que modelos individualizados por curso, conforme abordado no trabalho de Viana, Santana e Rabêlo (2022). A quantidade dos dados disponível por curso é pequena, quando analisados de forma isolada, mas, ao serem agrupados, podem melhorar essa representatividade das classes. Em vez de adotar uma abordagem sintética aplicando técnicas como as realizadas no trabalho de Hu et al. (2019) e Li et al. (2017), busca-se a ampliação por meio da combinação de dados de cursos, identificando similaridades entre eles com base em suas características e de seus alunos. Esse processo é realizado por meio de clusterização: a partir dos grupos resultantes, novos modelos de predição são desenvolvidos ao combinar os conjuntos de dados dos cursos. Ao aumentar a quantidade de exemplos, o objetivo é melhorar a capacidade de generalização dos modelos de predição, pois os modelos têm a oportunidade de aprender padrões mais intrincados, permitindo uma representação mais precisa da relação entre as características dos dados e os resultados esperados.

1.2 Trabalhos Relacionados

A crescente disponibilidade de dados nas instituições de ensino tem levado à busca por abordagens inovadoras para enfrentar o desafio da evasão de alunos. Nesse contexto, as técnicas de mineração de dados têm emergido como um recurso valioso, oferecendo insights significativos que podem contribuir de maneira substancial para o entendimento e mitigação desse problema. Ao explorar e analisar padrões e relações ocultas nos dados acadêmicos, as técnicas de mineração de dados têm o potencial de identificar fatores de risco, padrões de comportamento e indicadores-chave que podem ser essenciais na tomada de decisões estratégicas e na implementação de intervenções preventivas.

Neste tópico é apresentado como a aplicação dessas técnicas tem sido abordada na literatura recente, destacando a sua importância como apoio fundamental no enfrentamento da evasão em instituições de ensino.

O estudo realizado por Martins, Bertuci e Peniani (2020) apresenta uma revisão de trabalhos primários publicados entre 2016 e 2020, que tratam da aplicação da mineração de dados como uma ferramenta de apoio para enfrentar a evasão escolar em instituições

de ensino superior no Brasil. O foco está na análise de técnicas de mineração utilizadas, nos atributos considerados mais relevantes, na avaliação da acurácia e na descrição das metodologias empregadas em cada um desses trabalhos. Em seu trabalho, foi identificado que a técnica de aprendizagem de máquina mais utilizada no cenário de evasão escolar é a árvore de decisão, totalizando 53,33% dos trabalhos selecionados, sendo que estes trabalhos têm média de acurácia acima dos 86%, demonstrando eficiência na extração de conhecimento. Com relação a metodologia utilizada, 38% dos trabalhos selecionados não fazem menção, entretanto, dos trabalhos que citam, aproximadamente 63% utilizam o KDD. Similarmente, na revisão de literatura realizado por Colpo et al. (2020), a qual selecionou apenas trabalhos de classificação para identificar alunos em risco de evasão ou verificar características com maior influência no intervalo de 2006 a 2019, também observou-se que na maioria dos trabalhos foi utilizado algoritmos de árvores de decisão. Segundo Han, Pei e Kamber (2012 apud COLPO et al., 2020), isto pode ser explicado pela facilidade de interpretação dos seus resultados. Na revisão de literatura de Santos, Saraiva e Oliveira (2021), além dos algoritmos de árvore de decisão figurarem como os mais utilizados tanto no cenário nacional quanto internacional, há o destaque para o *Random Forest* como o que apresenta melhor desempenho internacionalmente.

No trabalho de Manhães, CRUZ e ZIMBRÃO (2015) são aplicadas técnicas de mineração de dados educacionais visando avaliar o rendimento acadêmico dos alunos de graduação na UFRJ, possibilitando aos gestores identificar os estudantes que estão em situação de risco de evasão. Nele também foi levado em conta a semestralidade dos discentes para a predição do progresso acadêmico. Foram utilizados conjuntos de dados com um número reduzido de características, as quais descrevem dados acadêmicos nos períodos letivos abordados.

Da mesma forma, Santos, Martins e Plastino (2021) utiliza a semestralidade para previsão de evasão. Nele o objetivo é prever, ao final de cada semestre cursado, se o discente evadirá ou não, a partir apenas do seu desempenho nas disciplinas cursadas até o momento. O objetivo neste trabalho é verificar se é possível prever evasão somente com dados de desempenho dos alunos durante curso, e para isso foram utilizados algoritmos de classificação baseados em árvore. São criados 10 modelos preditivos, em que esse número representa o semestre ideal de conclusão para o curso de Ciência da Computação. O i-ésimo modelo, $1 \leq i \leq n$, é utilizado para prever se um aluno evadirá após transcorridos i semestres do seu ingresso.

Outro trabalho que também aplica a segmentação dos dados de acordo com os períodos, abrangendo desde o primeiro até o sexto, de forma a avaliar a evasão por semestre é Viana, Santana e Rabêlo (2022). Essa estratégia tem como finalidade a criação de um modelo de treinamento específico para cada um desses períodos, considerando assim o momento que o discente se encontra no curso. Cada modelo é desenvolvido incorporando

informações fixas de ingresso e informações acadêmicas acumulativas e variáveis, implicando que os dados acadêmicos evoluem a cada modelo.

No trabalho de Brito, Mello e Alves (2020), é realizado um estudo das características mais importantes para predição de evasão utilizando mais de 200 características relacionadas a dados demográficos municipais e 4 relacionados diretamente ao discente (cor/raça, gênero, ano de admissão e estado civil). Foram descartadas características que não estão presentes com o discente no momento do ingresso, tais como índice de rendimento escolar, por serem adquiridas no decorrer do curso, pois o objetivo é prever o risco de evasão de discentes ingressantes. Para tal, os dados foram divididos por área de conhecimento, ou seja, os dados dos cursos foram unidos por suas áreas, para então serem gerados modelos de classificação para cada grupo (área) com maior granularidade, verificando então as principais características para cada grupo. Também foi medida a acurácia do algoritmo aplicado, *Random Forest*, quanto à predição de cada grupo escolhido, obtendo resultados girando em torno de 70%.

O estudo de Saraiva et al. (2021) propõe o uso da mineração de dados e análise de agrupamentos para dividir uma base de dados do Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE) em diferentes perfis de estudantes, utilizando para tal o algoritmo de clusterização *K-Means*. Os estudantes são agrupados com base em suas características, objetivando detectar padrões. A análise dos grupos formados permite distinguir os perfis de estudantes que evadem e os que concluem o curso, identificando influências de variáveis como renda familiar, turno de estudo, sexo, faixa etária e grau de instrução da mãe no desempenho acadêmico. O estudo busca fornecer informações para orientar decisões pedagógicas e intervenções visando reduzir a evasão estudantil com base nos indicadores identificados em cada grupo.

Na tese de doutorado de Tamada et al. (2022), o objetivo é identificar riscos de evasão para cursos de Ensino a Distância, aplicando técnicas de aprendizado de máquina divididas em duas etapas em cascata: uma abordagem não supervisionada, utilizando o algoritmo de clusterização *K-Means* para agrupar os discentes, e outra supervisionada, utilizando atributos extraídos dos logs do Ambiente Virtual de Aprendizagem (AVA), dados socioeconômicos e um atributo transformado e descritivo indicando o grupo ao qual o exemplo pertence, obtido na abordagem anterior, nos modelos de classificação para predizer se o estudante irá evadir ou não. Os modelos propostos foram construídos em diferentes momentos dos cursos, coletando dados dos 10%, 25%, 50% e 75% de duração destes. Como resultado, conseguiu demonstrar que o peso da descrição com o grupo formado melhora o desempenho dos modelos, com uma métrica F1 superior a 80% nos diferentes classificadores testados, nos modelos referente aos 10% do curso.

Diferentemente dos demais trabalhos, este utiliza uma abordagem de aprendizagem de máquina não supervisionada, anterior a etapa supervisionada, objetivando o aumento

de dados dos cursos de forma não sintética. Ao identificar os cursos mais próximos entre si através de suas características e de seus discentes, por meio de técnicas de clusterização, novos conjuntos de dados podem ser criados a partir da união dos dados que compõem cada grupo/cluster identificado, objetivando criar modelos de predição específicos para cada um destes conjuntos, tornando-os mais robustos do que os modelos individuais (contexto do curso) na etapa supervisionada, de classificação. Quanto mais dados estão disponíveis, mais informações e padrões podem ser capturados pelo modelo, resultando em previsões mais precisas e confiáveis. Além disso, o aumento de dados frequentemente leva a uma melhor generalização do modelo, permitindo que ele lide com uma variedade mais ampla de situações e cenários, obtendo-se também uma maior representatividade da população ou do domínio no qual o modelo será aplicado.

Para tanto, foram utilizadas várias técinas de clusterização, dentre elas o *K-Means*, para identificação dos cursos que possuem maior proximidade entre seus dados, similar ao que foi feito por Saraiva et al. (2021), entretanto utilizando os resultados da clusterização para fazer a união dos dados dos cursos que compõem cada grupo/cluster identificado, criando modelos de dados mais robustos. Estes modelos seguiram a mesma abordagem de segmentação dos dados conforme o avanço do curso pelos discentes, similar as realizadas por Viana, Santana e Rabêlo (2022), Manhães e Cruz (2019), Tamada et al. (2022) e Santos, Martins e Plastino (2021), sendo criados modelos de predição por semestre, do 1º ao 8º, utilizando o algoritmo *Random Forest*. A Tabela 1 contém um resumo dos trabalhos relacionados.

1.3 Visão Geral da Proposta

Para demonstrar que modelos de predição com dados ampliados de forma não sintética, a partir da união de dados de cursos similares, são superiores a modelos individualizados por curso, este trabalho descreve um processo realizado em duas etapas. Na primeira etapa, os cursos foram agrupados através da clusterização de seus dados, utilizando algoritmos de aprendizado de máquina não supervisionado.

Na segunda etapa, foram formulados modelos de predição de evasão, tanto individualizados por curso quanto baseados em dados de cursos agrupados, conforme determinado pelos algoritmos de clusterização da etapa anterior. Esses agrupamentos foram utilizados para combinar os conjuntos de dados dos cursos, gerando assim modelos com maior volume de dados, mas ainda mantendo a similaridade entre os dados. O diagrama na Figura 2 apresenta as etapas do processo proposto.

A proposta utilizada é similar a aplicada por Tamada et al. (2022), entretanto, neste trabalho, as bases de dados das duas etapas são diferenciadas entre si em relação aos atributos e aos registros, embora haja sobreposição. E o resultado da clusterização

Tabela 1 – Resumo dos Trabalhos Relacionados

Autor	**Abordagens**	**Algoritmo**	**Métrica**	**Semestralidade**	**Agrupamento de cursos**
Manhães, CRUZ e ZIMBRÃO (2015)	Supervisionada	Total de 12, sendo o NaiveBayes com melhor desempenho	Acurácia	Sim	Não
Santos, Martins e Plastino (2021)	Supervisionada	CART	Acurácia, Precision, Recall e F-measure	Sim	Não
Viana, Santana e Rabêlo (2022)	Supervisionada	Random Forest, Decision Tree, Extra Trees, Multilayer Perception, Support Vector Machine, K-Nearest Neighbors e Gaussian Naive Bayes	Acurácia, Precisão, Recall, F1 score, Kappa, Área da curva ROC	Sim	Não
Brito, Mello e Alves (2020)	Supervisionada	Random Forest	Acurácia	Não	Sim, por área de conhecimento
Saraiva et al. (2021)	Não-Supervisionada	K-Means	-	Não	Não
Tamada et al. (2022)	Supervisionada e Não-Supervisionada	K-Means; Random Forest, Decision Tree,Gradiente boot tree, Regressão logistica, NaiveBayes e Support Vector Machine	Precisão, Recall e F1-Score	Sim	Não
Costa, Junior e Santana (2023)	Supervisionada e Não-Supervisionada	K-Means e Random Forest	Recall e MCC	Sim	Sim, por similaridade entre cursos

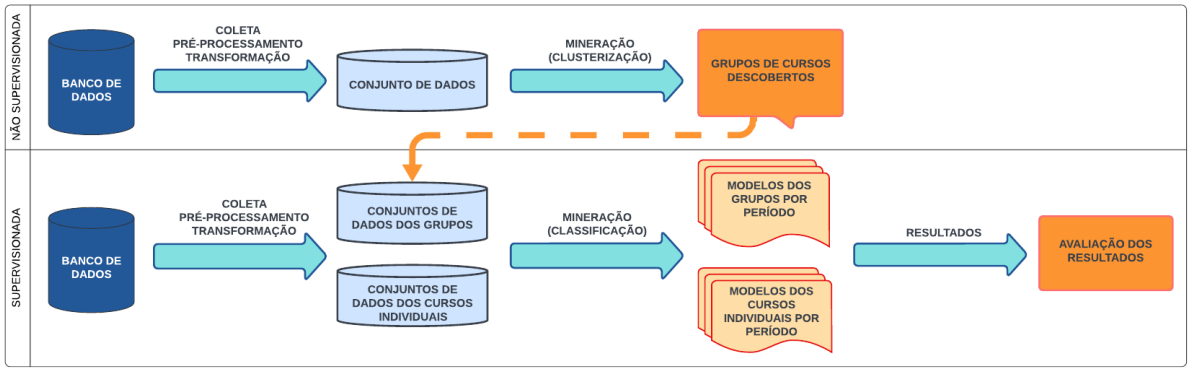


Figura 2 – Processo proposto.

é utilizado para identificar novos conjuntos de dados, gerando grupos de cursos os quais passarão pela etapa de classificação, diferentemente da tese, a qual utiliza o resultado da clusterização como um atributo na etapa de classificação.

Para os modelos de predição, foi utilizada a mesma arquitetura proposta por (VIANA; SANTANA; RABÊLO, 2022), em que há a separação dos dados por período em cada curso e grupos de cursos. No caso deste trabalho, do 1º ao 8º, em que os modelos

de 1º período são utilizados para predizer evasão de discentes que encontrem-se no 1º semestre do curso, os do 2º período, para discentes que encontrem-se no 2º semestre e assim sucessivamente. O modelo do 8º período é utilizado para prever os dados deste e dos seguintes. Cada modelo é constituído de características imutáveis (fixas), como raça e sexo, incluindo o *target* "evadido" e características acadêmicas cumulativas, conforme o período, refletindo a realidade do discente naquele momento da sua linha temporal no curso. É importante destacar que os dados foram organizados de maneira que os discentes deixam de constar nas bases de dados dos períodos à frente à sua evasão. Por exemplo, um discente que evade no 2º semestre, estará presente na base de dados do 1º e 2º períodos, pois houveram atividades acadêmicas, entretanto não constará mais nas bases do 3º ao 8º período. Dessa forma, a grande maioria dos discentes não evadidos (concluintes) estão presentes em todas as bases dos períodos, com exceção daqueles que concluem o curso em um prazo menor que 8 semestres, e os alunos evadidos estão presentes em todos os modelos do primeiro período, mas a quantidade diminui à medida que os períodos subsequentes avançam. Na tabela 2 pode-se observar os quantitativos de registros por período para cada classe, "evadido" e "não evadido", sem separacão dos cursos.

Tabela 2 – Totais de dados presentes nos modelos de dados dos períodos.

TARGET	P1	P2	P3	P4	P5	P6	P7	P8
Evadidos	3298	2041	1560	1257	996	756	584	463
Não Evadidos	2507	2507	2507	2504	2504	2502	2498	2496

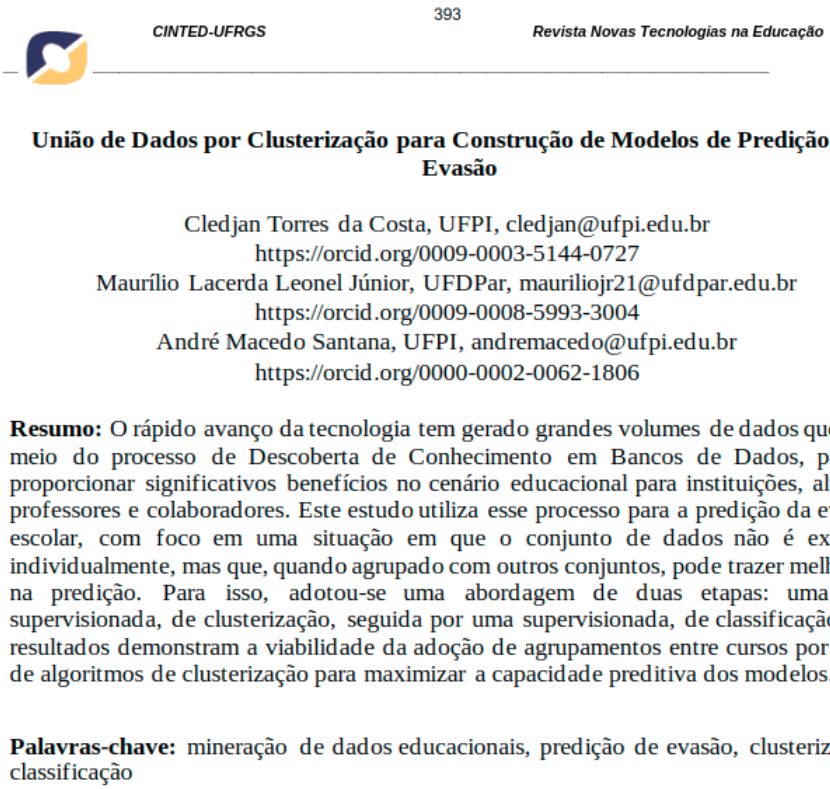
Uma vez obtidos os resultados da classificação dos modelos dos cursos de forma individual e dos grupos, foi realizada uma avaliação da diferença entre eles atravéz do Teste de Hipótese Z, com nível de significância de 5%.

Este trabalho deu origem ao artigo científico de Costa, Junior e Santana (2023) ilustrado na Figura 3 , em que foi empregada uma metodologia de duas etapas. Na etapa não supervisionada, utilizou-se o algoritmo *K-Means* para identificar cursos com maior similaridade, enquanto na etapa supervisionada, empregou-se o algoritmo *Random Forest* para gerar modelos de classificação dos cursos de forma isolada e dos grupos formados na etapa anterior. A comparação entre os modelos dos cursos e dos grupos mostrou maior robustez dos modelos dos grupos proporcionado pelo aumento de dados de forma não sintética. Os resultados demonstram a viabilidade da aplicação da solução proposta em aproximadamente 73,86% dos casos (65/88), considerando o RC, e em 88,64% (78/88), considerando o MCC.

1.4 Organização do Trabalho

Este trabalho está dividido em quatro partes. A primeira, destinado a Introdução, onde são apontados a contextualização, objetivos, trabalhos relacionados e visão geral da

proposta. A segunda realiza uma fundamentação teórica do KDD, suas fases e demais tecnologias utilizadas para melhor compreensão do trabalho. A terceira, composta por 2 capítulos nos quais é realizada a descrição da metodologia empregada, detalhando cada etapa do processo de KDD executado neste trabalho, das métricas empregadas, os resultados obtidos e análise dos resultados. E a última parte contém a conclusão e trabalhos futuros.



393

CINTED-UFRGS Revista Novas Tecnologias na Educação

União de Dados por Clusterização para Construção de Modelos de Predição de Evasão

Cledjan Torres da Costa, UFPI, cledjan@ufpi.edu.br
https://orcid.org/0009-0003-5144-0727
Maurílio Lacerda Leonel Júnior, UFDPar, mauriliojr21@ufdpar.edu.br
https://orcid.org/0009-0008-5993-3004
André Macedo Santana, UFPI, andremacedo@ufpi.edu.br
https://orcid.org/0000-0002-0062-1806

Resumo: O rápido avanço da tecnologia tem gerado grandes volumes de dados que, por meio do processo de Descoberta de Conhecimento em Bancos de Dados, podem proporcionar significativos benefícios no cenário educacional para instituições, alunos, professores e colaboradores. Este estudo utiliza esse processo para a predição da evasão escolar, com foco em uma situação em que o conjunto de dados não é extenso individualmente, mas que, quando agrupado com outros conjuntos, pode trazer melhorias na predição. Para isso, adotou-se uma abordagem de duas etapas: uma não supervisionada, de clusterização, seguida por uma supervisionada, de classificação. Os resultados demonstram a viabilidade da adoção de agrupamentos entre cursos por meio de algoritmos de clusterização para maximizar a capacidade preditiva dos modelos.

Palavras-chave: mineração de dados educacionais, predição de evasão, clusterização, classificação

Figura 3 – Artigo publicado.

Parte I

Referencial Teórico

2 Referencial Teórico

2.1 Descoberta de Conhecimento em Base de Dados

Sistemas de gerenciamento acadêmico desempenham um papel fundamental no registro e acompanhamento de informações sobre os estudantes, resultando na acumulação de grandes volumes de dados. Diante dessa quantidade significativa de informações, surge a necessidade premente de explorá-las de forma automatizada e eficiente. Nesse contexto, o processo de Descoberta de Conhecimento em Bancos de Dados assume uma importância significativa. O KDD é o processo pelo qual esses dados brutos são transformados em informações úteis e acionáveis, envolvendo etapas como pré-processamento, mineração de dados e interpretação de resultados. Isso possibilita que as instituições educacionais tomem decisões informadas, identifiquem tendências e melhorem o suporte aos estudantes de maneira personalizada. Em geral, os estudos que empregam mineração de dados para investigar a evasão escolar aderem às etapas do KDD (MEHTA; BUCH, 2016 apud TAMADA et al., 2022).

Fayyad, Piatetsky-Shapiro e Smyth (1996) desempenharam um papel pioneiro na área de KDD, fornecendo uma contribuição fundamental na definição e popularização do seu conceito como um processo completo, não trivial, interativo e iterativo, de identificar padrões válidos, novos, potencialmente úteis e, em última análise, compreensíveis em dados. Por iterativo e interativo, entende-se que cada fase pode ser repetida várias vezes, na sequencia tradicional ou de forma não linear, dependendo do conjunto de dados em questão e das escolhas feitas pelo analista (especialista no domínio) (SILVA; PERES; BOSCARIOLI, 2016).

Frequentemente, três atores estão presente no processo de KDD (BATISTA et al., 2003):

- Analista de dados: Responsável por conduzir a análise de dados, aplicar técnicas de mineração de dados e extrair conhecimento dos dados, embora não seja necessariamente especializado no domínio dos dados em questão.

- Especialista no domínio: Alguém com conhecimento profundo no campo de aplicação dos dados, que auxilia na interpretação dos resultados da mineração de dados e na definição dos objetivos da análise.

- Usúario: São as pessoas ou partes interessadas que utilizarão os resultados da análise de dados para tomar decisões ou realizar ações com base no conhecimento extraído.

É importante enfatizar que os três papéis distintos não precisam, obrigatoriamente, ser desempenhados por indivíduos diferentes. Com frequência, é observado que uma mesma pessoa assume simultaneamente dois papéis, como o de analista de dados e especialista de domínio, principalmente quando essa pessoa possui um profundo entendimento no campo de aplicação. No entanto, é necessária a presença dos três atores no processo de Descoberta de Conhecimento para assegurar a obtenção de resultados satisfatórios, uma vez que cada um deles contribui de forma única e complementar para o sucesso da análise de dados.

Fayyad, Piatetsky-Shapiro e Smyth (1996) define as seguintes etapas, ilustradas na Figura 4, do processo de KDD: coleta (seleção), pré-processamento, transformação, mineração de dados e análise dos resultados. É importante notar que a abordagem das etapas do processo de KDD pode variar entre diferentes autores, como as de (HAN; PEI; KAMBER, 2012), que trata a limpeza de dados como uma etapa única, e Tan et al. (2019) que trata a transformação como parte do pré-processamento. A diversidade de abordagens no KDD reflete a adaptabilidade da disciplina e a necessidade de flexibilidade para atender a diferentes cenários e desafios de mineração de dados, sendo fundamental que os profissionais optem pelas etapas e métodos que mais se ajustem às suas necessidades. Para este trabalho, considera-se a estruturação conforme Fayyad, Piatetsky-Shapiro e Smyth (1996), detalhadas a seguir.

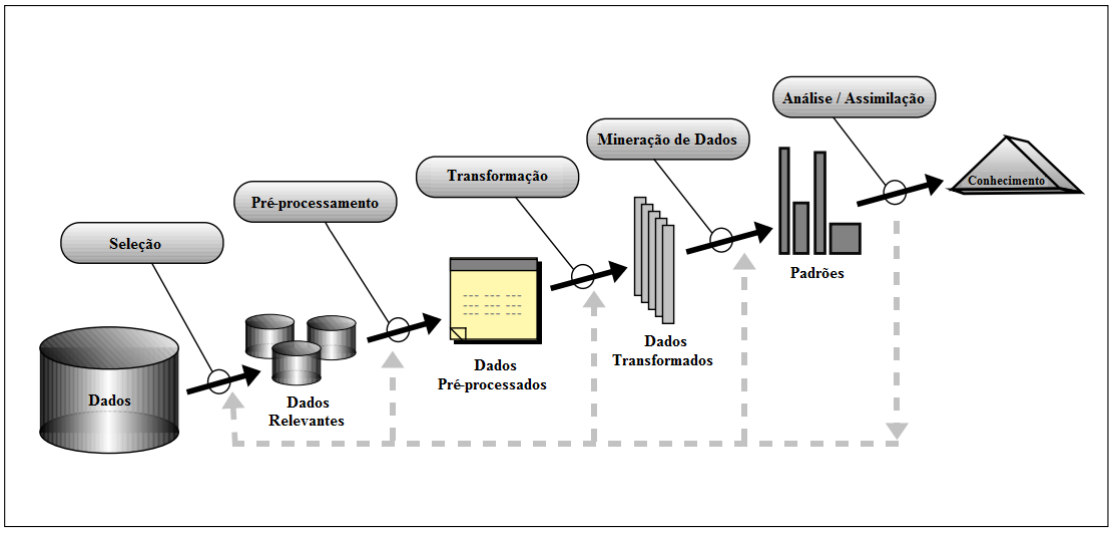


Figura 4 – O ciclo do processo de KDD (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

2.1.1 Coleta

A coleta de dados no processo de Descoberta de Conhecimento em Bancos de Dados é a fase inicial em que são reunidos e adquiridos dados brutos de diversas fontes, com o objetivo de criar um conjunto de dados abrangente. Essa etapa desempenha um papel fundamental para um resultado satisfatório na mineração de dados, uma vez que a

qualidade e a representatividade dos dados coletados impactam diretamente na qualidade das descobertas subsequentes.

Para garantir que os dados escolhidos sejam relevantes e adequados à tarefa de mineração de dados, a presença de um especialista de domínio é essencial. Esse especialista possui o conhecimento necessário para selecionar os atributos mais pertinentes e compreender quais informações são pertinentes para a análise, contribuindo significativamente para o sucesso do processo de KDD e para a extração de conhecimento a partir dos dados. Nesta etapa, é necessário que o analista de dados possua clareza sobre os objetivos definidos.

A maioria dos sistemas de gerenciamento de dados em uso atualmente foi desenvolvida em uma época anterior ao amadurecimento e à disseminação das técnicas de Engenharia de Software, resultando em sistemas proprietários com documentação insuficiente, tornando assim o processo de coleta de dados desafiador e dependente de especialistas de domínio (ARAUJO, 2014).

Existem diversos desafios que podem estar presentes nesta etapa. Alguns dos principais, levantados por Araujo (2014), são:

- Problemas Legais e Éticos: Há potenciais obstáculos de natureza legal ou ética que podem impedir a disponibilização de dados para fins de análise. Como exemplo, deve-se cumprir a Lei Geral de Proteção de Dados (LGPD), que regula as atividades de tratamento de dados pessoais, e garantir que a coleta de dados seja justa e não discriminatória, sendo necessária para evitar o reforço de preconceitos existentes ou a marginalização de grupos minoritários.

- Motivos Estratégicos: As organizações muitas vezes tem a necessidade de proteger informações sensíveis ou estratégicas, restringindo o acesso a parte dos dados ou estatísticas relacionadas a eles, muitas vezes para preservar a vantagem competitiva. Como exemplo, Chan e Stolfo (1998 apud BATISTA et al., 2003) descrevem uma análise voltada para a detecção de transações fraudulentas em cartões de crédito, na qual a distribuição das classes foi modificada. A proporção entre transações fraudulentas e não fraudulentas representa uma informação estratégica mantida em absoluto sigilo pelas empresas de cartão de crédito.

- Problemas de Conectividade: Existem diversos problemas de conectividade, como dados contidos em sistemas sem acesso via rede de computadores, por vezes sendo necessário criação de interfaces de comunicação, necessidade de manipular dados que estejam dispersos em vários locais e em diferentes formatos.

2.1.2 Pré-Processamento

Preparação ou pré-processamento de dados e uma etapa anterior à mineração que visa preparar os dados para uma análise eficiente e eficaz. Dada a diversidade de métodos de coleta e armazenamento de dados, esta etapa se destaca como uma das mais trabalhosas e demoradas no contexto geral do processo de descoberta do conhecimento (TAN et al., 2019).

As técnicas de processamento de dados, quando aplicadas antes da mineração, podem melhorar substancialmente a qualidade geral dos padrões extraídos e/ou o tempo necessário para a mineração real (HAN; PEI; KAMBER, 2012).

Comumente, os dados originários de diversas fontes, ou mesmo de uma única, apresentam vários tipos de falhas ou são organizados de forma inadequada para a aplicação de alguns algoritmos. Por conta disso, o processo de descoberta de conhecimento é dificultado, podendo levar ao fracasso da análise. Portanto, é fundamental aplicar procedimentos que pré-processem os dados, ou seja, os preparem, corrigindo inconsistências, agregando valor e possibilitando clegar a resultados de qualidade (SILVA; PERES; BOSCARIOLI, 2016). O pré-processamento compreende várias estratégias e técnicas diferentes que estão inter-relacionadas de formas complexas. Algumas das principais etapas envolvidas no pré-processamento são: limpeza dos dados, redução de dados e balanceamento dos dados, detalhados a seguir.

2.1.2.1 Limpeza dos Dados

Os dados do mundo real tendem a ser incompletos, ruidosos e inconsistentes. As tarefas de limpeza de dados têm como finalidade "limpar"os dados, abordando questões como o preenchimento de valores ausentes, a redução de ruídos nos dados, a detecção ou remoção de valores discrepantes, bem como a resolução de inconsistências. Quando os usuários percebem que os dados estão contaminados, é improvável que depositem confiança nos resultados de qualquer análise de dados realizada. Além disso, dados incorretos podem introduzir confusão no processo de análise de dados (HAN; PEI; KAMBER, 2012).

Tratamento de Valores Ausentes

Abaixo são elencados os principais métodos para tratamento de valores ausentes:

I) **Remoção da Tupla:** Essa prática costuma ser adotada quando falta um rótulo de classe (supondo que a tarefa de mineração envolva classificação). No entanto, esse método não é particularmente eficaz, a menos que a tupla contenha diversos atributos com valores ausentes. Sua eficácia é especialmente limitada quando a porcentagem de valores ausentes varia consideravelmente entre os atributos. Ao descartar a tupla,

deixamos de aproveitar os valores dos atributos remanescentes na mesma, os quais poderiam ter sido úteis para a tarefa em questão.

II) **Preenchimento manual dos valores:** Encontrar o valor correto para um atributo em um exemplo de dados pode ser viável, caso possa haver a repetição do processo de aquisição de dados ou a consulta a um especialista ne domínio. Entretanto torna-se impraticável quando há muitos valores ausentes, além disso o preenchimento pode introduzir viés nos dados se o especialista tiver apenas conhecimento parcial do contexto.

III) **Preencher Valores Ausentes com uma Constante Global:** Esse método funciona substituindo todos os valores de atributos faltantes por uma mesma constante, como um rótulo como "Desconhecido"ou - ∞ . Entretanto, é importante notar que o programa de mineração pode erroneamente interpretá-los como um conceito interessante, uma vez que todos compartilham o valor comum. Portanto, embora esse método seja simples, ele não é à prova de falhas.

IV) **Preencher Valores Ausentes com uma medida de tendência central:** Em distribuições de dados normais (simétricas), é apropriado usar a média, enquanto em distribuições de dados assimétricas, é mais aconselhável empregar a mediana.

V) **Utilizar a média ou mediana do atributo para todas as amostras pertencentes à mesma classe que a tupla fornecida:** Por exemplo, ao classificar clientes de acordo com o risco de crédito, é possível substituir o valor ausente pelo valor médio da renda para os clientes na mesma categoria de risco de crédito da tupla em questão. Se a distribuição de dados para uma classe específica estiver distorcida, a mediana é uma escolha mais adequada (HAN; PEI; KAMBER, 2012).

VI) **Utilizar o Valor mais Provável para Preencher a Lacuna no Dado Ausente:** Isso pode ser determinado por meio de métodos como regressão, abordagens baseadas em inferência com um formalismo bayesiano ou indução de árvore de decisão.

Suavização de Ruídos

O termo "ruído"se refere a qualquer tipo de interferência, variação ou distorção presente nos dados que não está diretamente relacionada à informação de interesse ou ao sinal que se deseja analisar. Em geral, o ruído pode ser causado por diversos fatores, incluindo erros de medição, imprecisões nos dados ou a presença de valores notavelmente discrepantes, conhecidos como outliers. Essas perturbações nos dados podem dificultar a identificação de padrões, tendências e informações relevantes durante a análise.

Algumas técnicas básicas de descrição estatística e métodos de visualização de dados, como boxplots e gráficos de dispersão, podem ser usados para identificar valores discrepantes, que pode representar ruído.

Segundo Silva, Peres e Boscarioli (2016), há duas abordagens para o tratamento de dados ruidosos:

I) **Inspeção e correção manual:**A análise exploratória proporciona ao analista a capacidade de identificar a presença de ruído nos dados, o que pode ser realizado, por exemplo, por meio da análise de gráficos de caixas ou pela observação da forma da distribuição de frequências. Quando há poucos valores ruidosos, é possível corrigi-los manualmente. No entanto, em casos em que esses valores ruidosos estão em grande quantidade, podem ser realizados procedimentos de suavização que envolvem a criação de intervalos de valores, orientados por medidas de posição, como a média ou a mediana, juntamente com medidas de dispersão.

II) **Identificação e limpeza automática:** Algoritmos estão disponíveis para mitigar ou eliminar ruídos em conjuntos de dados. Algoritmos projetados para abordar problemas de predição e agrupamento podem ser adaptados para identificar e isolar instâncias ruidosas em um conjunto de dados (um exemplo é o algoritmo DBSCAN) ou para mapear um valor ruidoso para uma classe ou intervalo de valores que o torne compatível com os valores esperados. Dependendo do tipo de dado em análise, técnicas de processamento de sinais podem ser empregadas para distinguir entre valores ruidosos e não ruidosos (SILVA; PERES; BOSCARIOLI, 2016).

2.1.2.2 Redução de Dados

A redução de dados geralmente ocorre em duas dimensões: por linhas, para diminuir a amostra de dados, e por colunas, para reduzir as variáveis de dados, ambas foram aplicadas neste estudo. Existem três métodos principais para a redução de variáveis em colunas. O primeiro consiste em utilizar o conhecimento do domínio para uma seleção direta das variáveis de interesse. O segundo envolve o uso de métodos estatísticos de seleção de características para escolher as variáveis relevantes para análise subsequente. O terceiro método adota técnicas de extração de características com o intuito de construir características úteis para a análise de dados (FAN et al., 2021). A seguir, são apresentadas as duas últimas.

Técnicas de Seleção de Atributos

Os métodos de seleção de atributos podem ser amplamente divididos em três categorias: *Filter*, *Wrapper* e *Embedded*.

- ***Filter*:** O processo geralmente envolve o cálculo de alguma medida de relevância, como coeficiente de correlação, entropia, qui-quadrado ou ganho de informação, para cada atributo em relação à variável alvo. Com base nessas métricas, os atributos são classificados ou selecionados com base em sua importância percebida. Os atributos que obtêm pontuações mais altas são considerados mais relevantes e são mantidos, enquanto os menos relevantes são descartados.

  O método *Filter* é rápido e eficiente, adequado para conjuntos de dados grandes e complexos, mas tem a limitação de não considerar interações entre atributos. Portanto, ele pode não ser apropriado para todos os tipos de problemas de seleção de atributos, especialmente quando as relações entre os atributos são cruciais para a tarefa de análise ou modelagem. Este método foi aplicado neste trabalho para identificação dos atributos mais relevantes.

- ***Wrapper*:** Utiliza um algoritmo de aprendizagem específico para avaliar a relevância dos atributos. O processo envolve a geração de diferentes subconjuntos de atributos a partir do conjunto de dados original e a criação de modelos de aprendizado de máquina usando cada subconjunto. Em seguida, esses modelos são avaliados com base em uma métrica de desempenho, como precisão ou *F1-score*, e a qualidade de cada subconjunto é medida. O objetivo é encontrar o conjunto de atributos que produz o melhor desempenho de acordo com a métrica escolhida.

  O método *Wrapper* é o mais computacionalmente intensivo e demorado do que o método *filter*, uma vez que requer a criação e avaliação de múltiplos modelos. No entanto, ele é mais robusto, pois leva em consideração as interações entre atributos e, portanto, tende a ser mais adequado quando as relações entre atributos são cruciais para a tarefa de análise ou modelagem.

- ***Embedded*:** Incorpora a seleção de atributos diretamente no processo de treinamento de um modelo de aprendizado de máquina. Ao contrário dos métodos *filter* e "wrapper", que realizam a seleção de atributos como uma etapa separada e independente, o método *Embedded* faz isso de forma integrada durante o treinamento do modelo.

  Nesse método, os algoritmos de aprendizado de máquina são projetados para considerar automaticamente a relevância dos atributos enquanto ajustam os parâmetros do modelo. Isso significa que o próprio modelo aprende quais atributos são mais importantes para a tarefa em questão e quais podem ser ignorados. Alguns exemplos incluem Árvores de Decisão, Florestas Aleatórias e Redes Neurais.

Técnicas de Extração de Características

São um conjunto de métodos e abordagens que têm como objetivo principal a redução da dimensionalidade dos dados. Essas técnicas são aplicadas para transformar

atributos existentes em um novo conjunto de características, de forma a preservar as informações mais relevantes e informativas enquanto descarta informações redundantes ou menos importantes. A extração de características é especialmente valiosa em situações em que o conjunto de dados é complexo e de alta dimensionalidade, o que pode levar a problemas de eficiência computacional e *overfitting* em modelos de aprendizado de máquina.

Ao contrário da seleção de atributos, que seleciona apenas recursos úteis a partir de variáveis existentes, a extração de características visa construir novos recursos com base em combinações lineares ou não lineares de variáveis existentes (FAN et al., 2021).

Uma das técnicas representativas de extração de características lineares mais conhecidas é a análise de componentes principais (*Principal Component Analysis* – PCA), a qual foi utilizada neste estudo. Ele realiza uma transformação linear dos dados para criar um novo conjunto de características não correlacionadas chamadas de componentes principais. O número de componentes principais é limitado a ser igual ou menor do que o número de atributos originais na base de dados. Essa transformação é projetada de tal forma que o primeiro componente principal capture a maior variabilidade dos dados, seguido pelo segundo componente principal que captura a segunda maior variabilidade, e assim por diante. Ela realiza um mapeamento linear dos dados em um espaço de dimensão menor, de tal forma que haja a maximização da variância dos dados nesse espaço (CASTRO; FERRARI, 2016).

2.1.2.3 Balanceamento dos Dados

O desbalanceamento de dados ocorre quando há uma distribuição desigual entre as classes ou categorias em um conjunto de dados. Em outras palavras, algumas classes têm um número significativamente maior de exemplos em comparação com outras, resultando em uma disparidade na representação das classes. Isso é especialmente comum em problemas de classificação, onde as classes podem estar desproporcionalmente distribuídas.

Muitos sistemas de aprendizado, ao presumirem o balanceamento das classes, falham em desenvolver classificadores capazes de prever com precisão a classe minoritária em dados desbalanceados, resultando frequentemente em alta precisão para a classe majoritária, mas baixa precisão para a classe minoritária, especialmente quando os custos associados à classificação incorreta da classe minoritária são significativamente maiores do que os da classe majoritária (ARAUJO, 2014).

Segundo Fujiwara et al. (2020), existem várias metodologias disponíveis para enfrentar o desafio dos dados desequilibrados, as quais podem ser agrupadas em seis abordagens distintas: detecção de anomalias, sensibilidade ao custo, baseada em regras, amostragem, aprendizagem com *Ensemble* e uma abordagem híbrida, que combina amostragem e aprendizagem em conjunto.

Abordagem Detecção de Anomalias

Essa abordagem trata o problema de dados desequilibrados como um problema de detecção de anomalias. Isso implica tratar a classe minoritária como uma classe "anômala" ou "anormal" em relação à classe majoritária.

Nesse contexto, envolve a aplicação de técnicas de detecção de anomalias para identificar instâncias da classe minoritária que são consideradas "anômalas" ou "raras" em comparação com a classe majoritária. Essas técnicas podem incluir métodos estatísticos, aprendizado de máquina não supervisionado ou baseado em regras.

Como exemplos de algoritmos bem conhecidos que aplicam essa abordagem, tem-se:

***One-Class SVM (Support Vector Machine)*:** Uma máquina de vetores de suporte de classe única é treinada em uma classe de dados (geralmente a classe majoritária) e, em seguida, avalia a proximidade das instâncias da classe minoritária em relação a essa fronteira de decisão. Instâncias que estão longe da fronteira são consideradas anomalias.

***Local Outlier Factor (LOF)*:** Este algoritmo calcula a densidade local de instâncias e compara com a densidade local das instâncias vizinhas. As instâncias que têm uma densidade local significativamente menor do que as de seus vizinhos são consideradas anomalias.

Abordagem Sensível ao custo

O principal conceito das abordagens sensíveis ao custo é atribuir diferentes custos (ou pesos) às diferentes classes durante o treinamento de um modelo de aprendizado de máquina. Essa abordagem visa dar mais ênfase e penalizar mais fortemente a classificação incorreta da classe minoritária em comparação com a classe majoritária.

Por exemplo, se um algoritmo rotula erroneamente uma pessoa saudável em um exame de saúde, o impacto desse diagnóstico incorreto não é tão significativo. No entanto, em contrapartida, se um paciente for erroneamente considerado saudável, isso pode resultar na perda da oportunidade de tratamento. Nesse cenário, o custo associado à classificação errônea do segundo caso é substancialmente mais alto do que no primeiro caso (FUJIWARA et al., 2020).

A Abordagem Sensível ao Custo pode ser aplicada a uma variedade de algoritmos de aprendizado de máquina, alguns exemplos são:

Árvores de Decisão: Algoritmos de árvore de decisão, como o C4.5, CART e Random Forest, geralmente suportam a especificação de pesos de classe. Pode ser atribuído pesos mais altos à classe minoritária durante o treinamento.

***Cost-sensitive Support Vector Machine* (C-SVM):** é uma variação do algoritmo SVM que lida especificamente com problemas de classificação desbalanceada,

permitindo que se atribua custos diferentes às classes durante o treinamento. O objetivo é dar mais ênfase ao desempenho da classe minoritária, penalizando mais fortemente os erros de classificação nessa classe.

Redes Neurais: Os pesos são aplicados às camadas de saída, de modo que erros de classificação na classe minoritária são penalizados de acordo com o peso atribuído a essa classe. Por exemplo, se o custo da classificação incorreta da classe minoritária for maior, pode-se atribuir pesos maiores aos neurônios de saída correspondentes a essa classe. Isso faz com que a rede neural seja mais sensível ao custo e focada em melhorar o desempenho da classe minoritária durante o treinamento.

***AdaBoost*:** Os pesos são atribuídos às instâncias de treinamento com base na dificuldade de classificação, e amostras mal classificadas têm seus pesos aumentados em cada iteração. Na abordagem Sensível ao Custo, pode ser atribuído pesos mais altos às instâncias da classe minoritária, incentivando o modelo *AdaBoost* a se concentrar em melhorar a classificação dessa classe. Isso significa que erros de classificação na classe minoritária são penalizados de forma mais acentuada, tornando o *AdaBoost* mais sensível ao custo e voltado para a melhoria do desempenho da classe minoritária. O algoritmo continuará ajustando os pesos das instâncias para enfatizar a correção da classe minoritária durante o processo de treinamento do conjunto de classificadores fracos.

Abordagem Baseada em Regras

É uma estratégia que envolve a criação de regras específicas para classificar instâncias de dados com base em critérios predeterminados. Essas regras são projetadas de forma a lidar com o desbalanceamento das classes, atribuindo mais atenção ou peso às instâncias da classe minoritária durante o processo de classificação.

Segundo Fujiwara et al. (2020), uma metodologia importante da abordagem baseada em regras é o uso de árvores de decisão, que requerem medidas como o ganho de informação para identificar regras de classificação. Liu et al. (2010) projetaram uma abordagem, para gerar regras que sejam estatisticamente significativas, que utiliza o teste exato de Fisher para podar galhos da árvore que não são estatisticamente significativos, utilizando a medida Proporção de Confiança de Classe (*Class Confidence Proportion* - CCP).

Abordagem de amostragem

Envolve a manipulação da distribuição das instâncias em um conjunto de dados desbalanceado por meio da criação de novas amostras (instancias sintéticas) ou pela redução das amostras da classe majoritária. O objetivo é equilibrar a proporção entre as classes majoritárias e minoritárias, melhorando assim o desempenho do modelo de aprendizado de máquina na detecção da classe minoritária. Duas técnicas comuns dentro da abordagem de amostragem são:

***Oversampling* (Sobreamostragem):** Nessa técnica, novas instâncias são geradas a partir das amostras da classe minoritária, aumentando a quantidade de exemplos dessa classe, ajudando a equilibrar a distribuição das classes. Métodos comuns de oversampling incluem a duplicação de instâncias existentes, a geração de instâncias sintéticas usando técnicas como *SMOTE* (*Synthetic Minority Over-sampling Technique*), *ADASYN* (*Adaptive Synthetic Sampling*), entre outros.

***Undersampling* (Subamostragem):** Na subamostragem, algumas instâncias da classe majoritária são removidas para equilibrar a distribuição entre as classes. Isso reduz a quantidade de dados da classe majoritária. Como esta técnica diminui o tamanho dos dados, menos tempo é necessário para o aprendizado, entretanto, descartar informações presentes nas instâncias da classe majoritária pode levar à perda de informações úteis da classe. Existem várias técnicas de subamostragem, como *Random Undersampling*, *Tomek Links* e *Neighborhood Cleaning*.

Abordagem de Aprendizagem com *Ensemble*

Envolve a combinação de múltiplos classificadores ou modelos de aprendizado de máquina para melhorar o desempenho na detecção da classe minoritária. O objetivo é aproveitar a diversidade de diferentes algoritmos para abordar o problema de desbalanceamento e melhorar a generalização do modelo.

A ideia central é combinar as previsões de vários modelos ou classificadores fracos em um único modelo mais robusto e preciso. Os modelos individuais podem ser diferentes em termos de algoritmos, parâmetros, estratégias de treinamento ou conjuntos de dados de treinamento. A combinação de múltiplos modelos permite que o conjunto seja mais capaz de detectar a classe minoritária, mesmo em cenários de desbalanceamento acentuado. Como exemplos, tem-se:

Bagging (Bootstrap Aggregating): Envolve o treinamento de múltiplos modelos com diferentes subconjuntos de dados de treinamento amostrados aleatoriamente com substituição. As previsões dos modelos são combinadas por votação. Pode ser útil para melhorar a estabilidade do modelo.

Boosting: É uma técnica em que os modelos são treinados iterativamente, com mais peso sendo atribuído às instâncias mal classificadas em cada iteração. Algoritmos de boosting como AdaBoost e Gradient Boosting são amplamente usados para lidar com o desbalanceamento, pois dão mais atenção à classe minoritária.

Abordagem Híbrida

É uma estratégia que combina múltiplas técnicas e abordagens para abordar o problema de classes desbalanceadas. Ela visa aproveitar os pontos fortes de diferentes

métodos para melhorar a eficácia da classificação em cenários de desbalanceamento. Essa abordagem é altamente flexível e permite a combinação de técnicas de amostragem, criação de regras, aprendizado em conjunto e outras, conforme necessário.

Contudo, nem sempre há uma bom funcionamento desses algoritmos híbridos, especialmente quando o número absoluto de exemplos minoritários é pequeno (FUJIWARA et al., 2020).

2.1.3 Transformação

A transformação de dados tem como objetivo converter os dados em formatos adequados para que seja possível a aplicação de algoritmos da fase de mineração de dados, uma vez que esses algoritmos não são capazes de operar com certos tipos de dados. A decisão de quais transformações serão necessárias depende do algoritmo de mineração escolhido. Para (BATISTA et al., 2003) as transformações mais comum são: normalização, discretização de atributos quantitativos, transformação de atributos qualitativos em quantitativos e atributos de tipos de dados complexos. Além desses, Han, Pei e Kamber (2012) cita a construção de atributos como uma das estratégias dessa etapa.

2.1.3.1 Normalização

A normalização de dados é um processo no qual os valores de diferentes atributos são ajustados para uma escala comum, de modo que possam ser comparados e analisados de maneira equitativa, visando, dessa forma, garantir que nenhum atributo tenha um impacto desproporcional em algoritmos de aprendizado de máquina devido a variações nas unidades ou faixas de valores.

Ela é especialmente importante quando os dados têm atributos com escalas muito diferentes, o que pode afetar negativamente algoritmos sensíveis à escala, como regressão linear e *K-Means*. As técnicas mais comuns de normalização são: *Min-Max Scaling*, *Z-Score Normalization (Standardization)* e Escalonamento Decimal.

Para este trabalho, foi utilizada *L2 Normalization*. Nesta técnica, também conhecida como *Norma Euclidiana*, cada amostra, representando cada linha da matriz de dados, é redimensionada independentemente das outras amostras de forma que a sua norma seja igual a um. A normalização das entradas para que tenham normas unitárias é uma operação comum em tarefas como classificação de texto ou agrupamento, entre outras (SCIKITLEARN, sem data de publicação).

Essa técnica é útil quando se há a necessidade de que todas as amostras tenham comprimento unitário (norma igual a 1), tornando-as comparáveis em termos de direção e magnitude, independentemente das escalas originais dos atributos. É frequentemente

usado em problemas de aprendizado de máquina que envolvem medidas de similaridade ou distância, como *clustering*, PCA e cálculos de similaridade entre vetores.

2.1.3.2 Discretização de atributos quantitativos

Refere-se ao processo de converter variáveis numéricas contínuas em variáveis categóricas ou discretas. Isso é feito dividindo o intervalo de valores possíveis em intervalos menores e atribuindo a cada valor numérico uma categoria que corresponde ao intervalo no qual ele se encontra.

Essa transformação é frequentemente usada quando se deseja lidar com dados quantitativos de uma forma mais simples e interpretável. A discretização pode ser aplicada a atributos como idades, salários, temperaturas, entre outros, que são originalmente representados por números contínuos. Por exemplo, pode-se discretizar a idade de uma pessoa em categorias como “jovem”, “adulto” e “idoso” com base em intervalos predefinidos.

É importante escolher os intervalos de discretização de forma cuidadosa, uma vez que isso pode afetar os resultados da análise de dados e a capacidade de generalização dos modelos.

2.1.3.3 Transformação de atributos qualitativos em quantitativos

Devido alguns algoritmos de aprendizado de máquina não conseguirem manipular atributos qualitativos, é necessário realizar uma conversão dessas variáveis categóricas, como cores, categorias de produtos ou estados civis, em representações numéricas.

Quando existem atributos qualitativos com uma ordem intrínseca, como “educação infantil”, “ensino fundamental” e “ensino médio”, é comum usar uma técnica chamada "Codificação Ordinal"para mapeá-los em valores numéricos que preservem essa ordem. Pode-se atribuir os valores 1, 2 e 3 para representar a ordem crescente do dos níveis de ensino.

Já quando não há essa relação de ordem, aplicar a codificação ordinal cria uma ordem nos valores que não existe. Então, uma abordagem mais recomendada é a *One-Hot encoding*, onde cada categoria do atributo se torna uma nova coluna binária que indica a presença ou ausência da categoria para cada amostra. Segundo (BATISTA et al., 2003), esse mapeamento pode ser feito sempre que a quantidade de categorias para o atributo não seja muito grande, tipicamente entre 2 e 5. Nos casos em que for igual a 2, é feito mapeamento de tal forma que um dos valores do atributo represente um valor baixo, tipicamente 0, e o outro alto, tipicamente 1.

2.1.3.4 Atributos de tipos de dados complexos

Atributos como data e hora não são facilmente manipulados por algoritmos de aprendizagem de máquina. Dados como estes, que não são qualitativos nem quantitativos, necessitam ser transformados em um outro tipo de dado em que possa ser reconhecido pelo algoritmo. No caso, por exemplo, de um atributo de data de nascimento, poderia ser transformado em um atributo de idade, convertendo-se dessa forma para o tipo inteiro.

2.1.3.5 Construção de atributos

Refere-se a construção e adição de novos atributos a partir de um determinado conjunto de atributos para ajudar no processo de mineração. Essa criação a partir de atributos que, inicialmente, podem não ser considerados com alta relevância, pode mostrar-se altamente representativa na descrição de um conceito em tarefas de mineração de dados.

De acordo com (BATISTA et al., 2003), o processo de construção de atributos pode ser categorizado em duas abordagens principais: a automática e a orientada pelo usuário. A abordagem de indução automática cria atributos automaticamente, com orientação do próprio método de construção, avaliando-os em relação aos dados para excluir ou incorporar, enquanto a abordagem de indução orientada pelo usuário utiliza conhecimento do usuário ou especialistas no domínio para guiar a composição de atributos, adaptando-os às necessidades e contexto do problema.

2.1.4 Mineração de Dados

Mineração de dados refere-se à utilização de técnicas implementadas por algoritmos computacionais para processar um conjunto de eventos reais como entrada e fornecer como saída um padrão de comportamento, que pode ser representado, por exemplo, como uma regra de associação, uma função de mapeamento ou a modelagem de um perfil (SILVA; PERES; BOSCARIOLI, 2016).

A escolha da técnica de mineração de dados depende das características exclusivas do problema em questão e de seus objetivos específicos. Por isso é de suma importância a compreensão do problema, entender os objetivos que se deseja alcançar, os tipos de dados disponíveis, as características do domínio e as metas de negócios associadas ao problema. A análise exploratória dos dados ajuda a identificar padrões, características importantes e relações entre variáveis, possibilitando determinar quais técnicas podem ser mais apropriadas.

Mineração de Dados Educacionais envolve a aplicação de técnicas de mineração de dados a conjuntos de dados educacionais, com o propósito de analisar essas informações a fim de abordar questões de pesquisa no campo educacional (BARNES et al., 2009). Uma

análise abrangente dos estudos de pesquisa em MDE foi conduzido por Baker, Isotani e Carvalho (2011), categorizando as principais subáreas de pesquisa como taxonomia, destacando-se predição, agrupamento e mineração de relações, detalhadas a seguir.

2.1.4.1 Predição

O objetivo é criar modelos que possam inferir características específicas dos dados, referidas como variáveis preditivas, por meio da análise e combinação de diversos elementos presentes nos dados, chamados de variáveis preditoras. Para viabilizar a identificação precisa de uma ou mais variáveis preditoras conhecidas antecipadamente, é necessário codificar manualmente parte dos dados. Existem três tipos de predição: classificação, regressão, e estimação de densidade. (BAKER; ISOTANI; CARVALHO, 2011).

Classificação: A classificação é uma tarefa comum em Mineração de Dados, onde o modelo aprende a atribuir registros a classes conhecidas com base em exemplos previamente rotulados, permitindo que ele classifique novos registros (CAMILO; SILVA, 2009).

A Figura 5 ilustra o funcionamento de um modelo classificador que usa um conjunto de treinamento com amostras de dados em que a classe é conhecida para aprender a prever classes. Posteriormente, esse modelo é testado com um conjunto de testes que contém amostras com classes ocultas, e o objetivo é fazer previsões precisas com base no aprendizado anterior (COSTA et al., 2012).

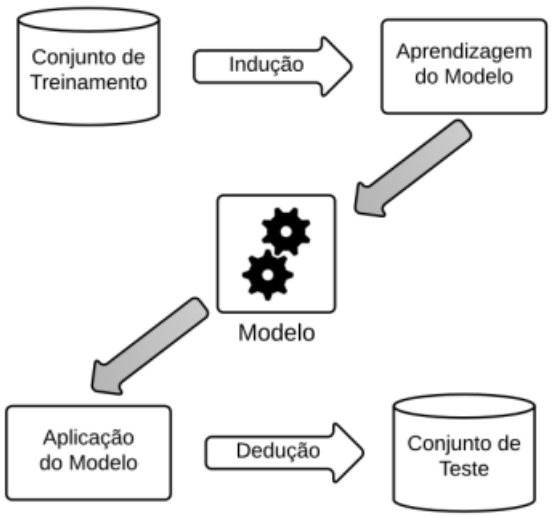


Figura 5 – Representação de um modelo classificador (COSTA et al., 2012).

Regressão: A regressão é semelhante à classificação, mas é usada quando o registro é representado por valores numéricos em vez de categorias. Isso permite estimar o valor de uma variável com base nos valores das outras.

Em EDM, os principais métodos de predição incluem classificação e regressão, em que a classificação é usada para variáveis preditivas binárias ou categóricas, a regressão

para variáveis preditivas contínuas, e ambas podem usar variáveis preditoras categóricas ou contínuas (COSTA et al., 2012).

Estimação de Densidade: A estimação de densidade é pouco comum na Mineração de Dados Educacionais devido à falta de independência estatística nos dados (BAKER; ISOTANI; CARVALHO, 2011).

2.1.4.2 Agrupamento

O agrupamento (*clustering*) envolve a subdivisão de uma população heterogênea em subgrupos mais coesos, sendo que esses grupos não são predefinidos e não existem exemplos previamente rotulados, como acontece na classificação. O agrupamento é frequentemente empregado como uma etapa preliminar para outras formas de mineração de dados (CAMARGO, 2002). Seu objetivo é identificar grupos de dados semelhantes. De acordo com Tamada et al. (2022), geralmente é empregada alguma forma de métrica de distância ou semelhança para identificar quais objetos compartilham características similares ou não.

Em muitas situações, a noção de cluster não possui uma definição clara. Para compreender a dificuldade em determinar o que constitui um cluster, (TAN et al., 2019) apresenta a Figura 6, que exibe 20 pontos e três diferentes abordagens para agrupá-los. Os símbolos utilizados indicam a pertinência a um cluster. As Figuras 6(b) e 6(d) dividem os dados em duas e seis partes, respectivamente. No entanto, a divisão aparente de cada um dos dois clusters maiores em três subclusters pode ser simplesmente uma ilusão humana. Além disso, não seria injustificado afirmar que os pontos formam quatro clusters, como mostrado na Figura 6(c). Essa figura demonstra que a definição de um cluster é imprecisa e que a melhor definição depende da natureza dos dados e dos resultados desejados.

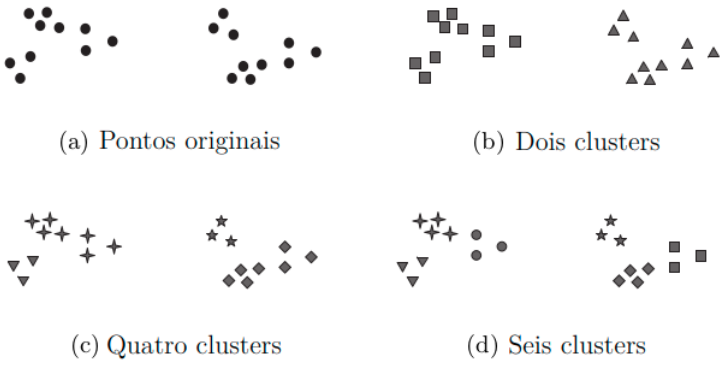


Figura 6 – Três maneiras diferentes de agrupar o mesmo conjunto de pontos. (TAN et al., 2019).

Conforme Murphy (2012), a natureza do agrupamento de dados, que não envolve um atributo alvo definido, juntamente com a falta de uma definição precisa e única do conceito de *cluster*, resulta em uma série de desafios, tanto na seleção dos algoritmos mais adequados

quanto na avaliação dos resultados obtidos. Portanto, em muitos aspectos, a análise de agrupamento possui uma natureza subjetiva, e seus resultados são significativamente influenciados pelo perfil do profissional responsável pela análise.

Existem muitos algoritmos de agrupamento na literatura, sendo difícil fornecer uma categorização precisa dos métodos de agrupamento, porque essas categorias podem se sobrepor, de modo que um método pode ter características de várias categorias. No entanto, é útil apresentar uma imagem relativamente organizada dos métodos de agrupamento, sendo, em geral, os principais métodos podendo ser classificados como: Métodos de Particionamento, Métodos Hierárquicos, Métodos Baseados em Densidade e Métodos Baseados em Grade (HAN; PEI; KAMBER, 2012). Um resumo das características de cada um é dado na Tabela 3.

Tabela 3 – Visão geral dos métodos de agrupamento (HAN; PEI; KAMBER, 2012).

Metódos	**Características Gerais**
Métodos de Particionamento	- Encontrar clusters mutuamente exclusivos de forma esférica; - Baseado em distância; - Pode usar a média ou o medoid (etc.) para representar o centro do cluster; - Eficaz para conjuntos de dados de pequeno a médio porte;
Métodos Hierárquicos	- O agrupamento é uma decomposição hierárquica (ou seja,múltiplos níveis); - Não pode corrigir fusões ou divisões errôneas; - Pode incorporar outras técnicas como microagrupamento ou considerar “vínculos” entre objetos;
Métodos Baseados em Densidade	- Pode encontrar clusters de formas arbitrariamente; - Clusters são regiões densas de objetos no espaço separadas por regiões de baixa densidade; - Densidade do cluster: Cada ponto deve ter um número mínimo de pontos dentro de sua “vizinhança”; - Pode filtrar outliers;
Métodos Baseados em Grade	- Usa uma estrutura de dados de grade de multirresolução; - Processamento rápido (geralmente independente do número de objetos de dados, mas dependente do tamanho da grade).

A forma mais simples e fundamental da análise de clusters é a partição, que agrupa objetos de um conjunto em vários grupos exclusivos. Geralmente, o número de grupos é especificado previamente. A formação dos clusters visa otimizar um critério de partição, como uma função de dissimilaridade baseada em distância, de modo que os objetos dentro de um cluster sejam "similares"entre si e "dissimilares"em relação a objetos de outros clusters, com base nos atributos do conjunto de dados (HAN; PEI; KAMBER, 2012). Alguns dos métodos de partição mais conhecidos e comumente usados são *K-Means* e *K-Medoids*. Outro método muito utilizado é o DBSCAN, mas este baseado em densidade, todos aplicados neste trabalho e abordados com mais detalhes a seguir.

K-Means

K-means é um dos algoritmos de agrupamento mais antigos e amplamente utilizados (TAN et al., 2019). O algoritmo utiliza cálculos de distância entre pontos para determinar a qual cluster os dados pertencerão, sendo necessário informar antecipadamente o número

de clusters (K). O algoritmo funciona encontrando centróides (pontos representativos) para os clusters e atribuindo cada objeto ao cluster cujo centróide está mais próximo. Seu algoritmo segue os seguintes passos:

I) Inicializar 'K' centróides aleatórios.;

II) Associar cada elemento da base ao centróide mais próximo;

III) Atualizar os centróides de cada grupo calculando a média dos dados.;

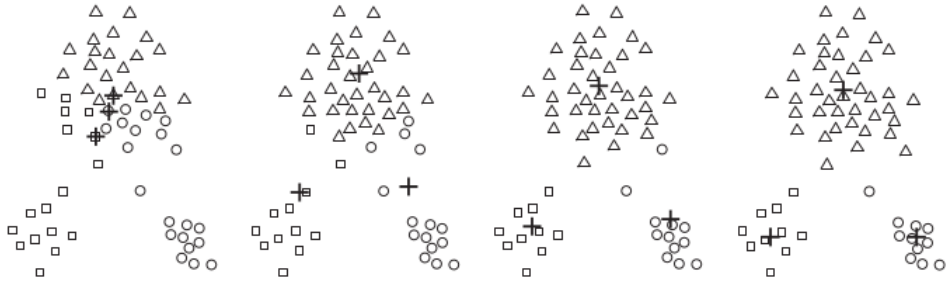
IV) Repetir os passos II e III até que os centróides não se alterem.

Os resultados produzidos pelo *K-Means* podem ser sensivelmente influenciados pelos valores iniciais selecionados para os centros dos clusters. Por conta disso, Uma abordagem eficaz para melhorar a seleção dos centróides no algoritmo *K-Means* é utilizar métodos de inicialização mais sofisticados do que a escolha puramente aleatória. Nesse contexto, o método *K-Means++* surgiu como uma alternativa promissora. Ele escolhe os centróides de maneira incremental, priorizando pontos que estão mais distantes dos centróides existentes. Isso ajuda a mitigar o impacto negativo de centróides iniciais ruins e contribui para clusters mais representativos e estáveis.

Tan et al. (2019) ilustrada na Figura 7 a operação do *K-Means*, que mostra como, a partir de três centróides iniciais, os clusters finais são encontrados em quatro passos de atribuição e atualização. Os centróides são indicados pelo símbolo "+"e todos os pontos pertencentes ao mesmo cluster têm o mesmo formato de marcador. No primeiro passo, os pontos são associados aos centróides iniciais, que estão todos localizados no grupo maior de pontos. Neste exemplo, a média dos pontos é usada para determinar o centróide. Após a atribuição dos pontos a um centróide, este é atualizado. Novamente, cada etapa é representada na figura com a posição inicial dos centróides e a atribuição dos pontos a eles. No segundo passo, os pontos são atribuídos aos centróides atualizados, e esses centróides são, por sua vez, atualizados novamente. Nos passos 2, 3 e 4, mostrados nas Figuras 7(b), (c) e (d), respectivamente, dois dos centróides deslocam-se em direção aos dois grupos menores de pontos na parte inferior das figuras. Quando o algoritmo K-means conclui, como na Figura 7(d), porque não ocorrem mais mudanças, os centróides identificaram os agrupamentos naturais dos pontos.

Para atribuir um ponto ao centroide mais próximo, precisamos de uma medida de proximidade que quantifique o conceito de "mais próximo"para os dados em questão. A Distância Euclidiana e a Distância *Manhattan* (*city-block*) são métodos de medição de distância comumente utilizados pelo *K-Means*.

Distância Euclidiana: é a métrica frequentemente empregada devido à sua capacidade de representar a distância física entre pontos em um espaço n-dimensional



(a) Interação 1. (b) Interação 2. (c) Interação 3. (d) Interação 4.

Figura 7 – Utilizando o algoritmo *K-Means* para encontrar três *clusters* em dados de amostra (TAN et al., 2019).

(SANTOS; MARTINS; PLASTINO, 2021). Sua fórmula soma os quadrados das diferenças entre as coordenadas em cada dimensão, tirando a raiz quadrada do resultado final. A mesma é apresentada a seguir:

$$\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.1}$$

Em que:

- n: Representa a dimensão do espaço em que os pontos estão localizados. Em um espaço bidimensional, n seria igual a 2, em um espaço tridimensional, n seria igual a 3, e assim por diante.
- x_i: Representa as coordenadas do ponto no espaço. O índice i varia de 1 a n para representar cada uma das n dimensões do espaço. Por exemplo, se tivermos um ponto em um espaço tridimensional, teríamos x_1 representando a coordenada ao longo do eixo X, x_2 representando a coordenada ao longo do eixo Y e x_3 representando a coordenada ao longo do eixo Z.
- y_i: Da mesma forma, y_i representa as coordenadas do segundo ponto no mesmo espaço dimensional, com o índice i variando de 1 a n.

Distância *Manhattan*: corresponde a soma das diferenças absolutas entre as coordenadas de dois pontos em um espaço n-dimensional. É chamada de *Manhattan* porque se assemelha à maneira como as ruas de *Manhattan*, em Nova York, estão dispostas em uma grade retangular. Em um espaço bidimensional (duas dimensões), isso se reduz a uma medida da distância em termos de "quarteirões"percorridos na grade, ao longo dos eixos x e y. Sua fórmula é apresentada a seguir:

$$\sum_{i=1}^{n}|x_i - y_i| \tag{2.2}$$

Em que n representa a dimensão do espaço, x_i e y_i são as coordenadas dos pontos nos eixos das dimensões.

Número ideal de *clusters* (K): Determinar o número ideal de clusters em um conjunto de dados é importante na análise de clusters porque influencia a granularidade adequada da análise, buscando um equilíbrio entre a compactação dos dados e a precisão. Isso envolve escolher entre diferentes níveis de detalhe na representação dos clusters (HAN; PEI; KAMBER, 2012). No caso da clusterização por particionamento, como no caso do *K-Means*, requer que o usuário determine previamente o número de clusters K. No entanto, de acordo com Kassambara (2017) não existe uma resposta definitiva para essa pergunta. A determinação do número ideal de clusters possui uma natureza subjetiva e depende, em grande parte, do método utilizado para medir as similaridades e dos parâmetros utilizados durante o processo de particionamento.

Existem várias abordagens para estimar esse número, alguns métodos populares e eficazes são o método de Elbow (cotovelo) e coeficiente de silhueta (silhouette), ambos utilizados neste trabalho.

***Elbow*:** A ideia central dos métodos de particionamento, como o algoritmo de agrupamento *K-Means*, é criar clusters para minimizar a variação total dentro deles, chamada de Soma dos Quadrados Dentro do Cluster (*Within-Cluster Sum of Square* - WCSS). Queremos que a WCSS total seja a menor possível, o que indica maior compacidade nos clusters. O método do Cotovelo avalia a WCSS total em relação ao número de clusters. O objetivo é escolher um número de clusters em que adicionar mais um cluster não melhore substancialmente a WCSS total (KASSAMBARA, 2017). O autor descreve os passos para identificar o número ótimo de clusters da seguinte forma:

I) Execute o algoritmo de agrupamento para diferentes valores de K. Por exemplo, variando K de 1 a 10 clusters.

II) Para cada valor de K, calcule a soma total dos quadrados dentro do cluster (WCSS).

III) Trace a curva do WCSS em relação ao número de clusters K.

IV) A localização de uma curva (cotovelo) no gráfico é geralmente considerada como um indicador do número apropriado de clusters.

Silhueta (*Silhouette*): Essa métrica avalia a qualidade dos clusters em uma análise de agrupamento. Ela mensura a semelhança dos objetos dentro do mesmo cluster em comparação com objetos de clusters distintos, ajudando a determinar quão bem os pontos estão agrupados em relação uns aos outros. O coeficiente é calculado para cada objeto no conjunto de dados e varia entre -1 e 1. Sua fórmula é:

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{2.3}$$

Em que:

- $S(i)$ é o coeficiente de silhueta para o ponto de dados i.
- $a(i)$ é a distância média entre o ponto i e todos os outros pontos no mesmo cluster (coesão)
- $b(i)$ é a menor distância média entre o ponto i e todos os pontos em qualquer cluster ao qual i não pertence (separação).

Um valor negativo é indesejável, pois corresponde a um caso em que a distância média para os pontos no cluster é maior do que a distância média mínima para os pontos em outro cluster. Queremos que o coeficiente de silhueta seja positivo, e que a distância média para os pontos no cluster esteja o mais próximo possível de 0, situação esta em que o coeficiente assume seu valor máximo de 1 (TAN et al., 2019).

Para calcular o coeficiente de silhueta médio de um cluster, soma-se os coeficientes de silhueta de todos os pontos no cluster e, em seguida, divide-se pelo número de pontos no cluster. Para obter uma medida geral da qualidade de um agrupamento, é possível calcular o coeficiente de silhueta médio de todos os pontos.

Para determinar a quantidade ideal de grupos (K) em um agrupamento usando o coeficiente de silhueta, deve-se executar o algoritmo de agrupamento para diferentes valores de K e calcular o coeficiente de silhueta médio para cada. O valor ideal de K é aquele que maximiza o coeficiente de silhueta médio.

K-Medoids

O *K-Medoids* é um algoritmo de agrupamento pertencente ao método de Particionamento. Ele é uma variação do *K-Means*, porém é mais robusto a ruído e valores discrepantes, pois o centro do grupo será necessariamente um objeto da base de dados, o que reduz a influência de ruídos e valores discrepantes na definição do centro (CASTRO; FERRARI, 2016).

A ideia fundamental por trás do *K-Medoids* é agrupar os dados de forma que os pontos em cada cluster sejam mais semelhantes entre si do que com os pontos de outros clusters. O algoritmo começa selecionando aleatoriamente K pontos do conjunto de dados como medoides iniciais. Em seguida, ele atribui cada ponto de dados ao medoide mais próximo, com base em uma medida de dissimilaridade, como a distância euclidiana ou a distância de *Manhattan*. Depois, ele recalcula os medoides de cada cluster, selecionando o ponto que minimiza a soma das dissimilaridades dos pontos do cluster em relação a

esse ponto. Esse processo de atribuição de pontos aos clusters e recálculo dos medoides é repetido até que não haja mais mudanças nos clusters ou até que um critério de parada seja atendido.

K-Medoids também possui algumas limitações, como a necessidade de definir o número de clusters K a priori e sua sensibilidade à escolha dos medoides iniciais. Em geral, o *K-Medoids* é uma técnica útil para agrupar dados quando a presença de outliers é uma preocupação e quando se deseja interpretabilidade nos medoides dos clusters.

DBSCAN

O *DBSCAN (Density-Based Spatial Clustering of Applications with Noise)* é um algoritmo de agrupamento baseado em densidade, utilizado para identificar agrupamentos em conjuntos de dados complexos e de alta dimensionalidade. Não requer a especificação do número de clusters a priori e é capaz de identificar clusters de diferentes formas e tamanhos.

A densidade de um objeto o pode ser medida pelo número de objetos próximos a ele. O DBSCAN identifica objetos com vizinhanças densas, conhecidos como objetos centrais, e os conecta com suas vizinhanças para criar regiões densas, que são tratadas como clusters. A vizinhança de um objeto é quantificada usando um parâmetro definido pelo usuário, $\epsilon > 0$, que especifica o raio de uma vizinhança centrada no objeto. A vizinhança ϵ de um objeto o é o espaço dentro de um raio ϵ centrado em o. A densidade dessa vizinhança é determinada pelo número de objetos dentro do raio ϵ. Para avaliar a densidade e identificar objetos centriais, o DBSCAN utiliza outro parâmetro do usuário, *MinPts*, que estabelece o limiar mínimo de objetos necessários para considerar uma vizinhança como densa. Um objeto é classificado como central se sua vizinhança ϵ contiver pelo menos *MinPts* objetos, sendo esses objetos centrais fundamentais para definir regiões densas (HAN; PEI; KAMBER, 2012).

É fundamental no DBSCAN a escolha adequada dos valores de *MinPts* e ϵ. Esses parâmetros determinam a sensibilidade do algoritmo para identificar regiões densas e influenciam diretamente a forma e o tamanho dos clusters identificados.

O DBSCAN é capaz de lidar com ruídos e outliers devido à sua capacidade de identificar regiões de densidade no conjunto de dados. Além disso, ele é capaz de identificar clusters de formas arbitrárias e não é sensível à ordem em que os dados são apresentados.

Agrupamento Hierárquico *(Hierarchical Clustering)*

Embora os métodos de particionamento possam agrupar objetos em conjuntos exclusivos, em certos casos, desejamos organizar nossos dados em uma hierarquia, dividindo-os em grupos em diferentes níveis. O agrupamento hierárquico faz isso ao formar uma

"árvore"de clusters, onde os objetos de dados são agrupados em diferentes níveis. Essa representação hierárquica é útil para resumir e visualizar os dados de forma mais organizada e compreensível (HAN; PEI; KAMBER, 2012). Existem duas abordagens principais para o agrupamento hierárquico:

- Aglomerativo: Essa abordagem organiza os objetos em uma hierarquia, seguindo uma estratégia de baixo para cima. No início, cada ponto de dados é considerado um cluster separado e, em cada passo, os clusters mais próximos são mesclados até que todos os pontos estejam em um único cluster.
- Divisivo: Os objetos são organizados em uma hierarquia usando uma estratégia de cima para baixo. Começa-se com todos os pontos em um único cluster e, em cada etapa, o cluster é dividido em clusters menores até que cada ponto de dados esteja em seu próprio cluster.

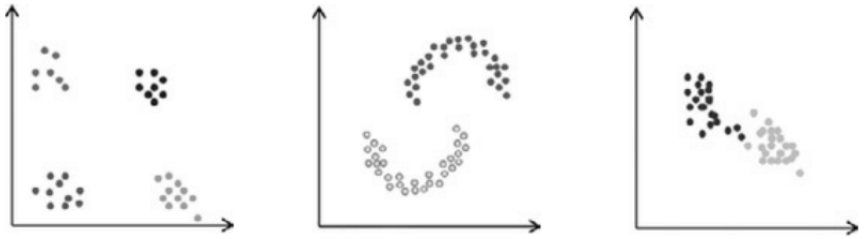
As técnicas de agrupamento hierárquico aglomerativo são amplamente utilizadas. Esse tipo de agrupamento é frequentemente representado graficamente por meio de dendrogramas, que exibem as relações entre clusters e subclusters, bem como a ordem em que os clusters foram mesclados (visão aglomerativa) ou divididos (visão divisiva) (TAN et al., 2019). Neste trabalho, utilizamos essa técnica aglomerativa bem como a representação gráfica por meio do denograma.

Durante o processo de agrupamento hierárquico, é necessário determinar a distância ou similaridade entre os pontos de dados. É a definição de proximidade entre clusters que diferencia as várias técnicas hierárquicas aglomerativas. O método empregado neste estudo, *Ward*, parte do pressuposto de que um cluster é representado pelo seu centróide. No entanto, ele mensura a proximidade entre dois clusters em função do aumento no SSE (do inglês *sum of the squared error*) que resulta da fusão desses dois clusters. Esse método busca minimizar a soma das distâncias ao quadrado dos pontos em relação aos seus centróides de cluster (TAN et al., 2019).

Avaliação do Agrupamento

Para avaliar o resultado da clusterização, dois critérios podem ser considerados: compacidade e separabilidade (HALKIDI; BATISTAKIS; VAZIRGIANNIS, 2001 apud SILVA; PERES; BOSCARIOLI, 2016). O primeiro critério, a compacidade, está diretamente ligada ao objetivo de encontrar grupos com o maior nível de similaridade intragrupos possível. A medida estatística de variância é uma maneira muito simples de medir a compacidade: quanto menor a variância dos exemplares dentro de um grupo, maior a compacidade. Já a separabilidade, objetiva encontrar grupos com a menor quantidade de similaridade entre eles. Portanto, os grupos devem estar longe uns dos outros. Uma

maneira simples de determinar a separabilidade é examinar as distâncias entre dois grupos (SILVA; PERES; BOSCARIOLI, 2016). Organizações de exemplares em grupos com alta e baixa compacidade e separabilidade são apresentados na Figura 8(a, b, c).



(a) Alta compacidade e alta separabilidade

(b) Baixa compacidade e alta separabilidade

(c) Alta compacidade e baixa separabilidade

Figura 8 – Visualização de grupos com compacidades e separabilidades baixas e altas (SILVA; PERES; BOSCARIOLI, 2016).

2.1.4.3 Mineração de Relações

Segundo Baker, Isotani e Carvalho (2011) na mineração de relações, o objetivo é descobrir relações entre variáveis em bancos de dados, podendo envolver a identificação das variáveis mais associadas a uma específica ou relações entre todas as variáveis presentes. Isso é realizado por meio de quatro tipos de mineração: regras de associação, correlações, sequências e causas.

a) Regras de Associação: Esse tipo de mineração busca identificar associações entre variáveis. É comumente usado para encontrar relações entre itens em conjuntos de dados, como em sistemas de recomendação. Por exemplo, a descoberta de que clientes que compram pão muitas vezes também compram manteiga é uma regra de associação.

b) Correlações: O objetivo é determinar se existem relações estatísticas entre as variáveis. Isso inclui examinar como as alterações em uma variável se relacionam com as alterações em outra variável. Por exemplo, pode-se investigar se existe uma relação entre as taxas de doenças respiratórias e a poluição do ar.

c) Sequências: O foco é descobrir padrões sequenciais nos dados. Análise de séries temporais e previsões são campos em que é amplamente usado. Por exemplo, pode-se buscar padrões sequenciais nos dados de vendas para descobrir como os eventos afetam o comportamento de compra dos clientes com o tempo.

d) Causas: A mineração de causas busca identificar relações de causa e efeito entre variáveis. Ela procura entender como uma variável afeta outra e quais fatores podem influenciar um evento. Por exemplo, pode-se investigar as razões pelas quais as taxas de desemprego aumentaram em uma região específica.

2.1.5 Análise e Interpretação dos Resultados

Concluída a etapa de mineração de dados, inicia-se a etapa de avaliação e interpretação dos resultados, conhecida também como pós-processamento, na qual os resultados são avaliados quanto à sua qualidade e relevância, o que envolve a aplicação de métricas de desempenho, validação cruzada e testes de hipóteses, entre outras técnicas. A etapa deve envolver todos os participantes do processo para que avaliem os resultados obtidos. Caso não o resultado não atenda as necessidades, o processo pode retornar as etapas anteriores e realizar ajustes até que os objetivos sejam alcançados.

O analista de dados avalia o desempenho dos algoritmos usando métricas. O especialista em domínio verifica a compatibilidade dos resultados com seu conhecimento do campo. O usuário toma a decisão final sobre a aplicabilidade dos resultados (BATISTA et al., 2003).

Neste trabalho, são utilizadas técnicas de clusterização e classificação na etapa de mineração de dados. Do ponto de vista de clusterização, a avaliação dos resultados foi detalhada no item 3.4.2.1. Quanto a classificação, as métricas de avaliação utilizadas foram *Recall* (RC) e Coeficiente de Correlação de *Matthews* (*Matthews Correlation Coefficient* - MCC), detalhadas a seguir.

2.1.5.1 Métricas de Avaliação de Classificadores

Existem várias métricas populares para avaliar a performance de classificadores. Uma das principais é a matriz de confusão, que permite analisar a quantidade de verdadeiros positivos (VP), verdadeiros negativos (VN), falsos positivos (FP) e falsos negativos (FN), conforme descrição abaixo:

- VP: O número de exemplos que foram corretamente classificados como positivos.
- VN: O número de exemplos que foram corretamente classificados como negativos.
- FP: O número de exemplos que foram incorretamente classificados como positivos (quando na verdade são negativos).
- FN: O número de exemplos que foram incorretamente classificados como negativos (quando na verdade são positivos).

Na figura 4 tem-se a representação de uma matriz de confusão (2x2) para classificadores binários. A partir desses valores, podemos calcular outras métricas como Acurácia, Precisão, *recall*, *F1 Score*, *Kappa* e MCC. Além disso, a AUC-ROC (Área sob a curva ROC) é outra métrica popular que avalia o desempenho de classificadores em várias configurações de limiar. Ela mede a capacidade do modelo de distinguir entre classes positivas e negativas.

Tabela 4 – Representação de uma matriz de confusão para classificadores binários.

TOTAL		REAL	
		Positivo	Negativo
PREDITO	Positivo	Verdadeiro Positivo	Falso Positivo
	Negativo	Falso Negativo	Verdadeiro Negativo

Acurácia

Representa a taxa de acerto geral do modelo, ou seja, a proporção de previsões corretas em relação ao total de previsões. Sua fórmula é apresentada na equação 2.4

$$\text{Acurácia} = \frac{VP + VN}{VP + FN + VN + FP} \tag{2.4}$$

Precisão

A precisão avalia a capacidade do modelo de fazer previsões corretas para a classe positiva, ou seja, mede a proporção de instâncias positivas identificadas corretamente em relação ao total de instâncias identificadas como positivas pelo modelo. Sua fórmula é apresentada na equação 2.5.

$$\text{Precisão} = \frac{VP}{VP + FP} \tag{2.5}$$

Recall

Recall mede a proporção de instâncias positivas que foram identificadas corretamente pelo modelo em relação ao total de instâncias positivas existentes nos dados reais. Sua fórmula é apresentada na equação 2.6.

$$\text{Recall} = \frac{VP}{VP + FN} \tag{2.6}$$

F1 - Score

Corresponde à média harmônica da precisão e do recall, a qual fornece um equilíbrio entre as duas métricas. Sua fórmula é apresentada na equação 2.7.

$$F1\text{-Score} = \frac{2 \cdot \text{Precisão} \cdot \text{Recall}}{\text{Precisão} + \text{Recall}} \tag{2.7}$$

Kappa

O índice Kappa é uma métrica de avaliação usada para medir a concordância entre as classificações de um modelo e as classificações reais em um problema de classificação. É especialmente útil quando se lida com problemas de classificação em que há mais de duas

classes ou quando a métrica de concordância deve levar em consideração a concordância que ocorreria ao acaso. Sua fórmula é apresentada na equação 2.8.

$$\kappa = \frac{P_o - P_e}{1 - P_e} \tag{2.8}$$

Onde:

- P_o (Proporção Observada de Concordância): corresponde a proporção de casos em que o modelo e as observações reais concordam, representada pela fórmula 2.9
- P_e (Proporção Esperada de Concordância ao Acaso): corresponde a proporção de concordância que seria esperada ao acaso, representada pela fórmula 2.10.

$$P_o = \frac{VP + VN}{\text{Total de Observações}} \tag{2.9}$$

$$P_e = \frac{(VP + FN) \cdot (VP + FP) + (VN + FP) \cdot (VN + FN)}{(\text{Total de Observações})^2} \tag{2.10}$$

MCC

o Coeficiente de Correlação *Matthews* é uma métrica estatística que avalia a qualidade das previsões em modelos de classificação, consideração os quatro quadrantes da matriz de confusão: VP, VN, FP e FN. Sua fórmula está representada pela equação 2.11. Essa métrica é especialmente relevante para lidar com conjuntos de dados desequilibrados, como os deste estudo.

$$\text{MCC} = \frac{VP \cdot VN - FP \cdot FN}{\sqrt{(VP + FP)(VP + FN)(VN + FP)(VN + FN)}} \tag{2.11}$$

Quando se trata de classes, o MCC é simétrico. Isso significa que trocar positivos e negativos produziria os mesmos resultados. O valor mínimo do MCC é -1, que indica que uma previsão é completamente errada, o classificador classifica todos os positivos como negativos e todos os negativos como positivos. O valor máximo é +1, o que indica uma classificação perfeita. Quando o valor é próximo de zero, indica que a previsão feita foi semelhante a um palpite aleatório (CHICCO; WARRENS; JURMAN, 2021).

Área Sob a Curva ROC (AUC-ROC)

Para compreender a AUC-ROC, é importante primeiro entender a curva ROC em si. Esta curva é uma representação gráfica do desempenho de um modelo de classificação em diferentes limiares de classificação. Ela é construída traçando a taxa de verdadeiros

positivos (TPR), que mede a capacidade do modelo de identificar corretamente exemplos da classe positiva, em relação à taxa de falsos positivos (FPR), que representa a proporção de exemplos da classe negativa erroneamente classificados como positivos. A curva ROC é, portanto, uma representação visual da capacidade do modelo de equilibrar a identificação de verdadeiros positivos sem aumentar excessivamente os falsos positivos.

A AUC-ROC, por sua vez, é uma métrica numérica que resume a curva ROC em um único valor. Ela varia entre 0 e 1, e um valor mais próximo de 1 indica um desempenho superior do modelo. Em outras palavras, quanto maior a AUC-ROC, melhor o modelo é em distinguir entre as classes. Um valor de 0,5 indica um desempenho equivalente ao acaso, enquanto um valor próximo a 1 representa uma capacidade de classificação excelente.

2.2 Aprendizado de Máquina

Aprendizado de Máquina (AM) é um ramo da Inteligência Artificial (IA) baseado na ideia de que sistemas podem aprender com dados, identificar padrões e tomar decisões com o mínimo de intervenção humana. Ele é capaz de identificar padrões em grandes conjuntos de dados e usá-los para fazer previsões ou tomar decisões sem ser explicitamente programado para realizar tarefas específicas. É amplamente utilizado em áreas como tradutores automáticos, reconhecimento de fala, reconhecimento facial, diagnóstico médico, análise financeira e muitas outras (LUDERMIR, 2021). Ainda conforme o autor, o sucesso das soluções de problemas complexos com técnicas de IA se baseia no fato de que esses problemas não possuem soluções algorítmicas conhecidas, mas contêm dados suficientes para treinar algoritmos de aprendizado de máquina, permitindo que as máquinas adquiram conhecimento intuitivo, que é difícil de expressar verbalmente ou formalmente, de maneira análoga à aprendizagem humana.

Mitchell (1997) define o aprendizado de máquina de forma ampla, abrangendo qualquer programa de computador que aprimore seu desempenho em determinada tarefa por meio da experiência. É um campo inerentemente multidisciplinar, baseando-se em resultados de inteligência artificial, probabilidade e estatística, teoria da complexidade computacional, teoria de controle, teoria da informação, filosofia, psicologia, neurobiologia e outros campos.

A indução é um método de inferência lógica que generaliza conclusões a partir de exemplos específicos, sendo um dos principais métodos utilizados para derivar conhecimento novo e prever eventos futuros, mas requer cuidado na escolha dos exemplos e na quantidade de dados disponíveis para evitar hipóteses de pouco valor. O aprendizado indutivo pode ser subdividido em dois tipos: supervisionado e não-supervisionado (MONARD; BARANAUSKAS, 2003), conforme pode ser observado na Figura 9.



Figura 9 – Hierarquia do aprendizado indutivo.

2.2.1 Aprendizado Supervisionado

No aprendizado supervisionado, o algoritmo é treinado para fazer previsões ou tomar decisões com base em dados de treinamento rotulados. Nesse contexto, "supervisionado" se refere ao fato de que o algoritmo é "guiado" por rótulos ou respostas conhecidas nos dados de treinamento. Classificação e regressão, abordados na sessão 2.1.4.1, são as principais tarefas associadas.

No trabalho de Batista et al. (2003), é realizada uma caracterização mais formal do Aprendizado Supervisionado, indicando que, neste, um sistema recebe um conjunto de exemplos rotulados, onde cada exemplo é uma combinação de características (representadas por um vetor $\vec{x_i}$) e um rótulo de classe y_i. O objetivo é criar um modelo, $y = f(\vec{x})$, que pode prever valores de y para novos exemplos não vistos. No entanto, devido à limitação do número de exemplos, o sistema não pode caracterizar completamente a função f. Em vez disso, ele cria uma hipótese h que se aproxima de f, ou seja, $h(\vec{x}) \approx f(\vec{x})$. Essa hipótese é uma estimativa da *função conceito* f.

Existem pontos importantes a serem consideradas no aprendizado supervisionado, dentre eles:

Sobreajuste e Sub-ajuste: Dois conceitos críticos no Aprendizado Supervisionado são o sobreajuste (*overfitting*) e o sub-ajuste (*underfitting*). O sobreajuste ocorre quando um modelo se ajusta em excesso aos dados de treinamento, capturando até mesmo o ruído nos dados. Isso resulta em uma má generalização para novos dados. Por outro lado, o sub-ajuste ocorre quando o modelo é muito simples para representar a complexidade dos dados de treinamento e, consequentemente, não consegue capturar os padrões essenciais. O equilíbrio entre esses extremos é fundamental para criar modelos robustos.

Validação Cruzada: Para avaliar o desempenho do modelo e evitar o sobreajuste,

a validação cruzada é uma técnica valiosa que foi utilizada neste trabalho. Ela envolve a divisão dos dados em subconjuntos de treinamento e validação em múltiplas iterações. Isso ajuda a estimar como o modelo se comportará em dados não vistos, fornecendo uma métrica mais robusta de desempenho.

Regularização: A regularização é uma técnica utilizada para controlar o sobreajuste. Ela introduz penalidades nos pesos dos atributos, incentivando o modelo a manter apenas os atributos mais relevantes. Métodos de regularização comuns incluem L1 (Lasso) e L2 (Ridge) para modelos de regressão linear e redes neurais.

Em seu trabalho (ARAUJO, 2014) elenca 4 paradigmas dentro do aprendizado supervisionado: simbólico, estatístico, baseado em exemplos e conexionista.

2.2.1.1 Paradigma Simbólico

Paradigma Simbólico enfatiza a construção de modelos que representam explicitamente o conhecimento em termos de conceitos, símbolos e relações lógicas. Isso torna os modelos simbólicos altamente interpretáveis e apropriados para tarefas em que a explicabilidade e a representação simbólica são fundamentais.

Dentre as representações simbólicas mais exploradas, destacam-se as árvores e as regras de decisão (GOMES, 2002). Dentro do contexto de evasão, os algoritmos mais amplamente utilizados são: *Decision Tree*, J48 e *Random Forest* (SANTOS; SARAIVA; OLIVEIRA, 2021).

***Decision Tree*:** ou árvore de decisão, é usada tanto para tarefas de classificação quanto de regressão. Ela é uma estrutura hierárquica composta por nós, incluindo o nó-raiz e os nós-folha, que representam os resultados finais. As conexões entre os nós são baseadas em regras "se-então". A árvore toma decisões com base nas características dos dados de entrada, seguindo a lógica das regras. A criação da árvore envolve a seleção dos nós em cada posição, geralmente utilizando medidas como ganho de informação e entropia para avaliar a organização e uniformidade dos dados.

J48: é um algoritmo específico de árvore de decisão desenvolvido por Ross Quinlan. No entanto, tanto ele quanto o decision tree, as implementações compartilham princípios comuns de construção de árvores de decisão, como a divisão de nós com base em critérios de ganho de informação, índice Gini ou erro médio quadrático. Na Figura 10 é apresentada uma árvore de decisão gerada pelo J48 no contexto de previsão de evasão como um dos resultados do trabalho de Barreto et al. (2019).

***Random Forest*:** ou Floresta Aleatória em português, é uma técnica de aprendizado de máquina que utiliza múltiplas Árvores de Decisão criadas de maneira aleatória para fazer previsões. Essas árvores são combinadas para produzir um resultado final. O Random Forest faz parte dos métodos ensemble, que reúnem vários modelos para obter

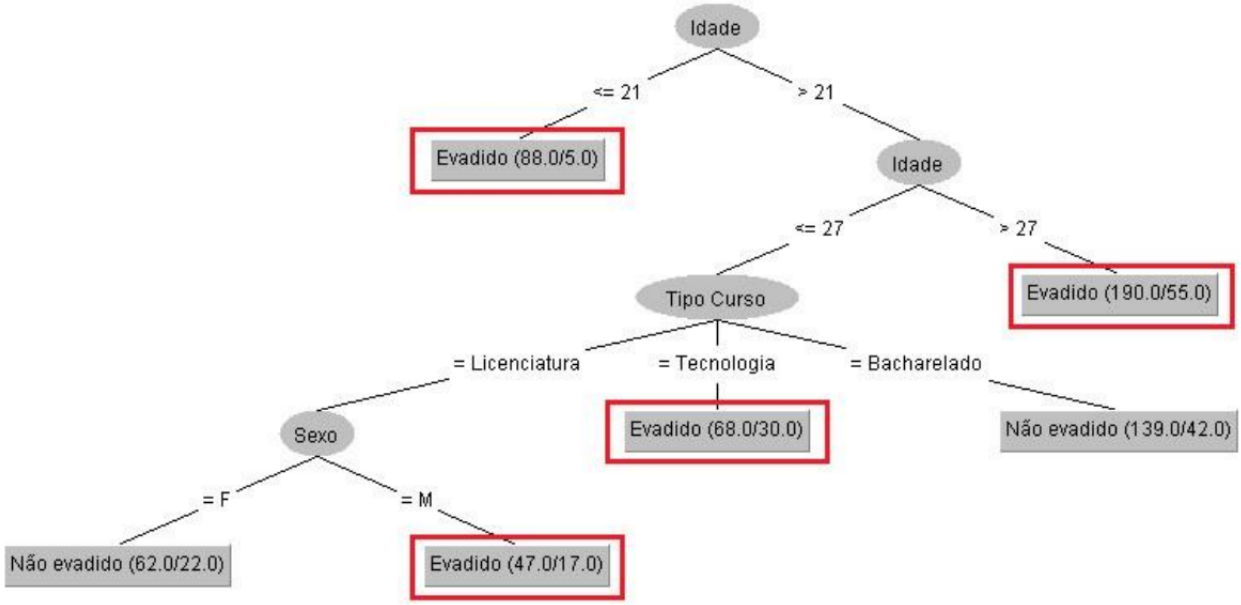


Figura 10 – Árvore de decisão para previsão de evasão utilizando o algoritmo J48 (BARRETO et al., 2019).

uma única previsão. Essa abordagem torna o algoritmo mais robusto, embora também levando a um maior custo computacional, geralmente resultando em melhor desempenho.

2.2.1.2 Paradigma Estatístico

É uma abordagem que se baseia em princípios estatísticos para modelar e fazer previsões a partir de dados rotulados. Esse paradigma envolve a aplicação de técnicas estatísticas e probabilísticas para entender a relação entre as características dos dados de entrada e as classes ou valores de saída que se deseja prever. Um exemplo de algoritmo deste paradigma que se destaca em termos de utilização dentro do contexto de evasão é o *Naive Bayes* (SANTOS; SARAIVA; OLIVEIRA, 2021), que é um método probabilístico baseado na Teoria Bayesiana de probabilidade.

***Gaussian Naive Bayes (GaussianNB)*:** é uma extensão do algoritmo *Naive Bayes*, que é conhecido por sua simplicidade e eficiência. A principal suposição por trás do Naive Bayes, incluindo o *Gaussian Naive Bayes*, é que todas as características são independentes entre si, dada a classe da instância. Isso significa que a presença de uma determinada característica em uma classe não influencia a presença de outra característica. No *Gaussian Naive Bayes*, assume-se que as características de cada classe seguem uma distribuição Gaussiana (normal). O algoritmo calcula a média e o desvio padrão das características para cada classe a partir dos dados de treinamento. Ao classificar uma nova instância, ele determina a probabilidade de pertencer a cada classe com base nas características da instância e nas distribuições de probabilidade. A classe atribuída é aquela com a maior probabilidade posterior, calculada usando o Teorema de *Bayes*.

2.2.1.3 Paradigma Baseado em Exemplos

Se concentra na representação e tomada de decisões com base em exemplos individuais do conjunto de dados. Nesse paradigma, o aprendizado ocorre por meio da comparação direta de instâncias de dados de treinamento para tomar decisões sobre novas instâncias.

Um exemplo representativo deste paradigma é o algoritmo *Nearest Neighbors* juntamente com sua variante conhecida como *K Nearest Neighbors* (KNeighborsClassifier), detalhadas a seguir (ARAUJO, 2014).

***K-Nearest Neighbors*:** ou K-Vizinhos Mais Próximos, nesse algoritmo, dado um ponto de dados não rotulado, ele calcula a distância desse ponto para todos os pontos de dados rotulados no conjunto de treinamento. Em seguida, identifica os K pontos mais próximos (vizinhos) e atribui ao ponto não rotulado a classe mais comum entre esses vizinhos. A escolha do valor de K afeta diretamente a fronteira de decisão do classificador, em que valores menores de K tendem a se ajustar muito aos dados de treinamento (alta variância), enquanto valores maiores de K produzem fronteiras de decisão mais suaves (baixa variância).

2.2.1.4 Paradigma Conexionista

Se baseia no funcionamento de redes neurais artificiais para modelar e resolver problemas complexos. Essa abordagem é fortemente inspirada pelo funcionamento do cérebro humano e consiste em várias camadas de unidades computacionais interconectadas, chamadas neurônios artificiais.

Dentro deste paradigma, os principais algoritmos são as Redes Neurais Artificiais, juntamente com suas variantes, *Multi-Layer Perceptron (MLPClassifier)*, *Radial Basis Function* e *Support Vector Machine* (ARAUJO, 2014).

***Multi-Layer Perceptron*:** é uma arquitetura de rede neural artificial composta por várias camadas de neurônios, incluindo uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Neurônios em camadas subsequentes estão conectados, e cada neurônio usa uma função de ativação para processar a entrada e gerar uma saída. O *Multi-Layer Perceptron* é capaz de aprender representações complexas dos dados e é comumente usado em tarefas de aprendizado supervisionado, como classificação e regressão. Seu treinamento envolve ajuste de pesos por algoritmos como a retropropagação.

Neste trabalho, foi utilizado um algoritmo de cada paradigma: *Random Forest*, *GaussianNB*, *K-Nearest Neighbors* e *Multi-Layer Perceptron*.

2.2.2 Aprendizado Não-Supervisionado

Já nas tarefas de aprendizado não-supervisionado, o algoritmo é treinado em um conjunto de dados que não possui rótulos, permitindo que o modelo encontre padrões e estruturas nos dados por conta própria. Segundo Murphy (2012), essas tarefas envolvem a identificação de informações relevantes nos dados, sem orientação externa, com foco na identificação de propriedades intrínsecas dos dados de entrada para construir representações úteis, com o propósito de auxiliar na tomada de decisões e na descoberta de conhecimento, permitindo que um algoritmo de AM aprenda a representar as entradas de acordo com um critério de qualidade específico. Dentre as tarefas mais importantes desse paradigma, está a clusterização. Ainda segundo o autor, as tarefas podem ser divididas em associação, sumarização e agrupamento.

A sumarização visa resumir os dados, condensando informações significativas de forma a manter o conteúdo mais importante, ao mesmo tempo em que reduz o volume de dados. A associação se refere a busca de padrões frequêntes, são um método baseado em regras para encontrar relacionamentos entre variáveis em um determinado conjunto de dados, conforme apresentado na sessão 2.1.4.3. E o agrupamento é uma técnica que envolve a divisão de um conjunto de dados em subconjuntos que são semelhantes entre si. Cada dado em um conjunto pertence ao cluster mais próximo, ou seja, aquele com o qual tem mais semelhança, sendo esta calculada usando medidas como a distância euclidiana e a distância de Manhattan, por exemplo, conforme detalhado em 2.1.4.2.

Com o intuito de estabelecer uma comparação entre as duas técnicas de Aprendizado de Máquina, Supervisionado e Não Supervisionado, discutidas neste capítulo, a Figura 11 apresenta um resumo em paralelo de ambas, possibilitando visualizar as principais diferenças entre essas abordagens.



Figura 11 – Comparativo entre técnicas Supervisionada e Não-Supervisionada.

Parte II

Proposta e Resultados

3 Metodologia Proposta

Este capítulo apresenta a caracterização e definição da metodologia empregada no processo de Descoberta do Conhecimento em Bases de Dados, informando ferramentas aplicadas, coleta de dados, pré-processamento, transformação e mineração de dados, além de uso de técnicas de Aprendizado de Máquina supervisionado e não supervisionado. Os resultados obtidos e a avaliação desses resultados são apresentados no capítulo seguinte, 4.

3.1 Ferramentas Utilizadas

Para o pré-processamento dos dados e a geração dos modelos preditivos, nossa escolha recaiu sobre o *Google Colaboratory*[1] (Colab), uma plataforma baseada em *Jupyter Notebook*[2] e alimentada pela linguagem de programação *Python*[3], bem como as bibliotecas *Pandas*[4], *Matplotlib*[5] e *Scikit-Learn*[6] (SKlearn).

O Google Colaboratory (Colab) oferece uma série de vantagens essenciais para esse projeto. Primeiramente, é uma plataforma em nuvem gratuita (com recursos limitados), o que significa que não é necessário a configuração e manutenção de infraestrutura local. Também permite que uma equipe trabalhe colaborativamente, independentemente da localização geográfica.

A linguagem Python, conhecida por sua simplicidade e vasta biblioteca de funções, facilita a manipulação de dados, a criação de modelos preditivos e a visualização de resultados, tornando todo o processo mais eficiente e adaptável às necessidades específicas do projeto, como a vasta quantidade de conjuntos de dados e modelos criados.

Com relação às bibliotecas, o Pandas oferece recursos de manipulação e análise de dados, garantindo que os dados estejam prontos para serem utilizados em nossos modelos de predição. A Matplotlib oferece diversos recursos para criação de gráficos em diversos formatos e estilos, sendo uma das bibliotecas mais populares do Python. O Scikit-Learn fornece uma ampla gama de algoritmos de Aprendizado de Máquina, simplificando a construção e avaliação de modelos de evasão escolar.

---

1 <https://colab.research.google.com/>
2 <https://jupyter.org/>
3 <https://www.python.org/>
4 <https://pandas.pydata.org/>
5 <https://matplotlib.org/>
6 <https://scikit-learn.org/>

3.2 O Processo Proposto

Neste estudo, adotou-se o esquema de Descoberta de Conhecimento em Bases de Dados proposto por Fayyad, Piatetsky-Shapiro e Smyth (1996), seguindo todas as etapas descritas na sessão 2.1, e adotado em diversos trabalhos, como em (ARAUJO, 2012), (VIANA; SANTANA; RABÊLO, 2022), (TAMADA et al., 2022). Os dados trabalhados são oriundos do Sistema Integrado de Gestão de Atividades Acadêmicas - SIGAA mantido pela UFPI. O SIGAA automatiza procedimentos acadêmicos em diversos níveis, incluindo graduação, pós-graduação, ensino técnico e ensino médio, gerenciamento de projetos de pesquisa, ações de extensão, monitoria, projetos de ensino, registro acadêmico de docentes, educação a distância (UFRN, 2020) e disponibiliza um ambiente virtual de aprendizado, similar ao do *Moodle*[7].

A técnica proposta se desdobra em duas etapas distintas. Na primeira, é adotada uma abordagem não supervisionada para identificar grupos de cursos que compartilham semelhanças em relação às suas características e de seus discentes. Isso é alcançado por meio de técnicas de clusterização aplicadas aos dados. Essa fase cria agrupamentos que representam cursos com características comuns.

Na segunda etapa, é utilizada uma abordagem supervisionada de classificação. Os resultados da etapa anterior são explorados para gerar conjuntos de dados robustos, formados pela combinação das informações de cursos pertencentes ao mesmo cluster. Com base nesses conjuntos de dados, foram desenvolvidos modelos de predição, tanto para os grupos quanto para cursos individuais. Isso possibilita não apenas analisar o desempenho de cursos em grupos, mas também avaliar como cada curso se destaca individualmente, possibilitando identificar se o aumento de dados proposto com a junção de dados de cursos é válida.

Para a comparação dos modelos agrupados e individuais, foi empregado o Teste de Hipótese Z, uma abordagem estatística para determinar se os resultados obtidos nos modelos de predição dos agrupamento de cursos é significativamente superior, inferior ou sem mudança relevante às previsões de cursos individuais.

3.3 Coleta

Conforme mencionado, os dados coletados são oriundos do SIGAA, o qual teve sua implantação iniciada em 2011, ano este em que a instituição mudou a forma de ingresso para os cursos da graduação regular, passando a adotar o Sistema de Seleção Unificada - SISU. Além disso, os dados migrados não possuem boa qualidade, tendo em vistas que o sistema legado possuia estrutura diferente do SIGAA, impossibilitando algumas migrações

[7] <https://moodle.org/>

diretas, sendo necessário conversões e consequentes perdas de informações. Como por exemplo o registro da nota de forma única (resultado final da disciplina) ao invés de por avaliações. Por conta disso, a qualidade dos dados para utilização nesse estudo é adequada apenas a partir de 2012, resultando um baixo volume de informações a serem analisadas.

A busca e seleção dos dados foram realizadas com apoio de analistas de domínio da Superintendência de Tecnologia da Informação da UFPI e da UFDPar, os quais foram fundamentais para identificar/construir os atributos considerados mais relevantes para o estudo em questão, tendo em vistas a grande quantidade de informações armazenada pelo sistema, assim como para a construção de funções SQL para já agrupar algumas informações transformando-as em atributos. Ela foi executada em maio de 2023 e se restringiu ao âmbito da graduação na modalidade presencial regular. Os dados coletados incluiram discentes que ingressaram entre o período de 2012.1 até 2023.1, abrangendo um intervalo de onze anos. A partir desta coleta, foram gerados dois repositórios de dados distintos, cada um para um propósito específico.

O primeiro repositório, R1, foi desenvolvido para suportar a Etapa 1 do processo, que se concentra na clusterização dos dados. Este repositório engloba informações de todos os discentes com status *ativo*, *concluído* (não evadidos) e *cancelado* (evadidos). Os dados contidos em R1 compreendem uma variedade de atributos, incluindo informações acadêmicas, características individuais dos discentes e detalhes dos cursos aos quais estão vinculado. O conjunto de dados coletado é composto por um total de 24 atributos e abrange 11.263 registros.

O segundo repositório, R2, é destinado para a Etapa 2 do processo, que envolve a predição. Neste repositório concentram-se os discentes com status *concluído* (não evadidos) e *cancelado* (evadidos), excluindo aqueles com status *ativo*, presentes no R1. O repositório R2 possui uma quantidade maior de atributos do que R1, com alguns sendo coincidentes, pois para R2 foi dado enfoque para utilização de caracteristicas acadêmicas cumulativas, que variam com o avanço cronológico do curso pelo discente, para viabilizar a criação dos modelos preditivos baseado em janelas semestrais, conforme aplicado no trabalho de Viana, Santana e Rabêlo (2022). Este conjunto de dados inicial totaliza 122 atributos e inclui um total de 5.979 registros.

Durante a coleta, foi realizada limpeza prévia e criação de atributos, diretamente via função SQL. Como limpeza, foi efetuado o tratamento do campo que contém a média final do discente em determinada disciplina, na qual foi feita a imputação de notas para os casos em que o campo encontrava-se nulo da seguinte forma: 7 para os registros em que a situação fosse de aprovação e 0 nos casos de reprovação. A ação foi necessária para que fosse possível a criação do atributo “média acumulada” (R2) do discente.

A criação de atributos nesta etapa geraram os seguintes atributos para o R1:

- *totalreprovacao*: Total de reprovações do discente no curso;
- *totaltrancamentocurso*: Total de trancamentos no curso do discente;
- *totaltrancamentocomponentes*: Total de trancamentos em turmas do discente.

Para R2, os atributos criados nessa etapa foram:

- *ano_referencia* e *periodo_referencia*: Ano-período de evasão ou de conclusão do curso, conforme seja o caso. Para os evadidos, existem muitos em que o ano-período de evasão registrado no sistema não coincide com a realidade, pois, muitas vezes, esses discentes já abandonaram os seus cursos a tempos, mas a instituição tardou em realizar os procedimentos de cancelamento via edital, por exemplo, acabando por ficar um registro de data bem posterior a data de real abandono. Por conta disso, o registro do sistema foi descartado e optamos por utilizar o ano-periodo de evasão como sendo um período posterior ao último em que o aluno cursou turmas. Para os não evadidos, utilizamos o último ano-periodo cursado;
- *cursounapandemia*: Identifica se o discente cursou turmas no período da pandemia (2020 a 2021);
- *quantidadeensino*: Quantidade de atividades de ensino em que participou. Cabe ressaltar que, além desses, os analistas de domínio pensaram em atributos que apresentassem os totais de participações dos discentes em atividades de monitoria, pesquisa e extensão. Entretanto, devido os resultados serem nulos, foram descartados;
- *qtde_periodo_cursado_cronologicamente*: Quantidade de semestres transcorridos desde o ingresso até o último registro de turmas no histórico;
- *temcredito_cc_ca*: Se possui crédito concedido ou crédito automático no histórico.

Além destes, ainda nesta fase, foram criados atributos acadêmicos cumulativos a cada semestre, conforme mencionado acima, cujas características evoluem à medida que o aluno avança no curso. São, em um total de 88, sendo oito atributos de cada tipo, dependendo do período acadêmico considerado, cobrindo do primeiro ao oitavo semestre. Esses atributos mantêm sua natureza constante ao longo de todos os semestres, mas seus valores variam à medida que os dados se acumulam com a progressão dos períodos. Por exemplo, existem 8 atributos de natureza "Total de aprovações", com terminações de 01 a 08, correspondente ao período referente, o qual indica o total de turmas aprovadas pelo discente até o final do semestre em questão. Para uma descrição mais sucinta, optamos por apresentá-los de forma genérica. No repositório real, o x presente na nomenclatura dos mesmos corresponde ao semestre referente. Esses atributos incluem:

- *trancamento_programa_x*: Quantidade de trancamentos no curso até o semestre x;
- *qtd_aprovacao_x*: Quantidade de turmas aprovadas no curso até o semestre x;
- *qtd_reprovacao_ft_nt_x*: Quantidade de turmas reprovadas por falta e nota no curso até o semestre x;
- *qtd_reprovacao_ft_x*: Quantidade de turmas reprovadas por falta no curso até o semestre x;
- *qtd_reprovacao_nt_x*: Quantidade de turmas reprovadas por nota no curso até o semestre x;
- *qtd_trancamento_turma_x*: Quantidade de trancamentos de turmas até o semestre x;
- *qtd_cancelamento_turma_x*: Quantidade de cancelamento de turmas até o semestre x;
- *media_acumulada_x*: Média do rendimento acadêmico até o semestre x;
- *ch_integralizada_x*: Carga horária integralizada até o semestre x;
- *rep_ob_semestre_x*: Quantidade de reprovações de disciplinas obrigatórias até o semestre x;
- *ap_ob_semestre_x*: Quantidade de aprovações em disciplinas obrigatórias cumpridas no prazo até o semestre x.

O atributo *evadido*, target para o R2, foi criado com base nos discentes que possuíam o status *concluído* e os aptos a colarem Grau (status *graduando*), classificados como *Não Evadido*. Já como *Evadido* foram classificados os discentes com status *cancelado* e os que não efetuaram matrícula nos 3 últimos semestres (ou mais) até a Coleta de dados, seguindo as normas de graduação para cancelamento por abandono da instituição, adicionando-se um período de margem. Cabe ressaltar que discentes com status *cancelado* por motivo de falecimento foram descartados.

Na Tabela 5 são apresentados os atributos coletados/criados dos dois repositórios, sendo marcados com "*" aqueles que são oriundos da criação prévia de atributos durante a fase. Já na Tabela 6, estão presentes os atributos cumulativos criados para o R2 de forma generalizada, apresentando apenas a natureza dos atributos criados para cada período.

Cada repositório foi então trabalhado de forma distinta, mesmo que similar em muitos pontos no que se referem aos atributos coincidentes, para se preparar os dados para as etapas de mineração de cada uma das etapas do processo, não supervisionada e supervisionada. Na Figura 12 tem-se as informações relativas a esta fase. A seguir, detalharemos as fases subsequentes dentro de cada etapa.

Tabela 5 – Atributos coletados/criados para repositórios R1 e R2 durante a coleta.

Rep.	**Atributo**	**Descrição**	**Tipo**
R1	centro	Centro ao qual o curso está associado (UFD-Par)	Categórico
R1	totalreprovacao*	Quantidade de reprovações no curso	Numérico
R1	totaltrancamentocomponentes*	Quantidade de trancamento em turmas	Numérico
R1	totaltrancamentocurso*	Quantidade de trancamento no curso	Numérico
R1, R2	area_cnpq	Áreas de conhecimento CNPq do curso	Categórico
R1, R2	curso	Curso do discente	Categórico
R1, R2	grau	Grau do curso (Bacharelado|Licenciatura)	Categórico
R1, R2	id_curriculo	Identificador do currículo do discente	Numérico
R1, R2	id_curso	Identificador do curso do discente	Numérico
R1, R2	turno	Turno do curso	Categórico
R1, R2	data_nascimento	Data de nascimento do discente	Data
R1, R2	quantidade_membros_familia	Quantidade de membros da família	Numérico
R1, R2	segundograuanoconclusao	Ano de conclusão do ensino médio	Numérico
R1, R2	sexo	Sexo do discente	Categórico
R1, R2	ano_ingresso	Ano de ingresso na instituição	Numérico
R1, R2	evadido*	Situação do discente (Target: Evadido|Não Evadido)	Booleano
R1, R2	id_discente	Identificador do discente	Numérico
R1, R2	ira	Índice de Rendimento Acadêmico	Numérico
R1, R2	nota_ciencias_humanas	Nota na prova de Ciências Humanas no Enem	Numérico
R1, R2	nota_ciencias_natureza	Nota na prova de Ciências da Natureza no Enem	Numérico
R1, R2	nota_linguagem	Nota na prova de Linguagem no Enem	Numérico
R1, R2	nota_matematica	Nota na prova de Matemática no Enem	Numérico
R1, R2	nota_redacao	Nota na prova de Redação no Enem	Numérico
R1, R2	periodo_ingresso	Período de ingresso na instituição	Numérico
R2	conclusao_escola_publica	Se o discente concluiu o ensino médio em escola pública	Booleano
R2	estadocivil	Estado civil	Categórico
R2	raca	Raça	Categórico
R2	uf_naturalidade	UF de nascimento	Categórico
R2	ano_referencia*	Ano de cancelamento ou conclusão do curso	Numérico
R2	cursounapandemia*	Se o discente cursou turmas no período da pandemia (2020 a 2021)	Booleano
R2	id_convocacao_processo_seletivo	Identificador da convocação do processo seletivo do discente	Numérico
R2	id_processo_seletivo	Identificador do processo seletivo ao qual o discente ingressou	Numérico
R2	opcao_inscricao	Opção de inscrição no processo seletivo (1|2)	Numérico
R2	periodo_referencia*	Período de cancelamento ou conclusão do curso	Numérico
R2	quantidadeensino*	Quantidade de atividades de ensino em que participou	Numérico
R2	qtde_periodo_cursado_cronologicamente*	Quantidade de semestres transcorridos desde o ingresso	Numérico
R2	temcredito_cc_ca*	Se possui crédito concedido ou crédito automático no histórico	Booleano
R2	tipoacaoafirmativa	Tipo de ação afirmativa do discente no ingresso	Categórico

3.4 Etapa 1 – Não Supervisionada

O objetivo desta etapa é identificar os cursos mais similares entre si, através de técnicas de agrupamento. Após a coleta, partiu-se para o pré-processamento dos dados, seguido da mineração e análise dos resultados.

3.4.1 Pré-Processamento:

No desenvolvimento deste projeto, as etapas de pré-processamento e transformação dos dados se revelaram processos iterativos e não estritamente sequenciais, envolvendo a necessidade de avançar e retroceder entre essas duas fases. Especificamente, no estágio de

Tabela 6 – Atributos cumulativos criados na etapa de coleta para R2.

Atributo	**Descrição**	**Tipo**
trancamento_programa_x	Quantidade de trancamentos no curso	Numérico
qtd_aprovacao_x	Quantidade de turmas aprovadas no curso	Numérico
qtd_reprovacao_ft_nt_x	Quantidade de turmas reprovadas por falta e nota no curso	Numérico
qtd_reprovacao_ft_x	Quantidade de turmas reprovadas por falta no curso	Numérico
qtd_reprovacao_nt_x	Quantidade de turmas reprovadas por nota no curso	Numérico
qtd_trancamento_turma_x	Quantidade de trancamentos de turmas	Numérico
qtd_cancelamento_turma_x	Quantidade de cancelamento de turmas	Numérico
media_acumulada_x	Média do rendimento acadêmico	Numérico
ch_integralizada_x	Carga horária integralizada	Numérico
rep_ob_semestre_x	Quantidade de reprovações de disciplinas obrigatórias	Numérico
ap_ob_semestre_x	Quantidade de aprovações em disciplinas obrigatórias	Numérico

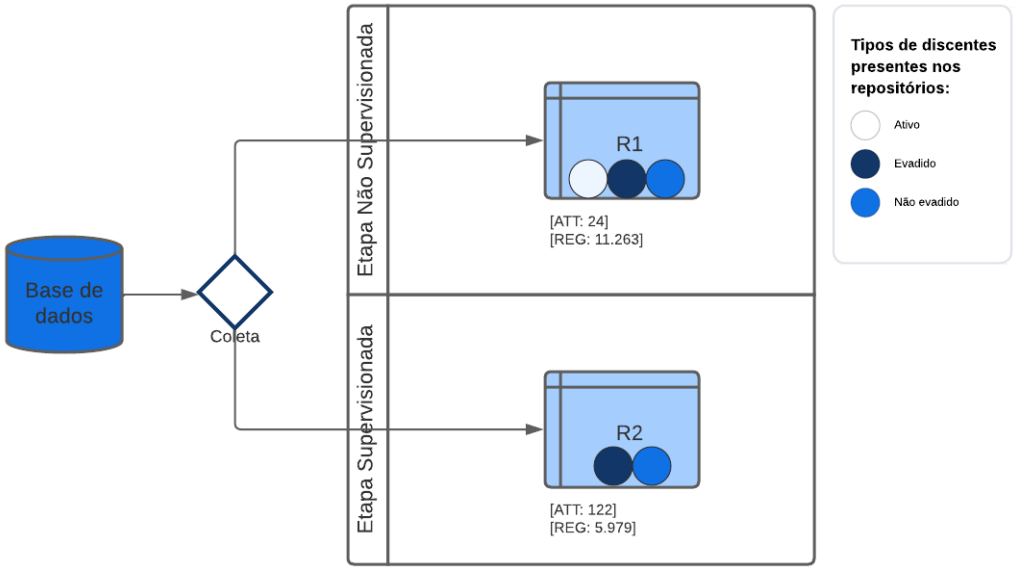


Figura 12 – Coleta efetuada para as duas etapas do processo.

pré-processamento, conduzimos a limpeza dos dados, juntamente com a seleção manual de atributos relevantes. Entretanto esta seleção só pode ser realizada, uma vez que avançamos para a fase de transformação e realizamos a criação de novos atributos e retrocedemos para o pré-processamento. Além disso, na fase de transformação, houve a necessidade de converter atributos categóricos em formatos numéricos e normalizar os dados. Dessa forma, essa abordagem iterativa e não linear permitiu uma preparação mais eficiente e refinada dos dados.

Limpeza dos Dados

Os dados coletados foram tratados de forma a corrigir ruídos e dados ausentes. O total de casos problemáticos se concentraram em grande parte nos registros referente ao ano de 2012, ano inicial de ingresso de alunos considerado na coleta dos dados. Os atributos tratados e as estratégias adotadas foram os seguintes:

- *data_nascimento*: Realizado o tratamento de dados ruidosos, com exclusão de um único registro, cuja data de nascimento encontrava-se igual a data de ingresso.

- *segundograuanoconclusao*: Inicialmente foi realizado o tratamento de dados ruidosos, realizando-se a atribuição de *0* (zero) para todos os casos em que o ano de conclusão do ensino médio era maior (posterior) que o ano de ingresso na instituição e também para os casos em que este ano fosse anterior a *1500*. Em seguida, para os registros nulos ou iguais a zero (0), foi realizada a imputação pelo ano de conclusão do ensino médio mais frequente dentre os discentes que possuiam o mesmo ano de nascimento do registro em questão e que o resultado não ultrapassasse o ano de ingresso. Casos residuais foram descartados.

- *quantidade_membros_familia*: Registros nulos foram imputados pela média da quantidade de membros da família de cada ano de ingresso.

- Notas no Enem (*nota_ciencias_humanas*, *nota_ciencias_natureza*, *nota_linguagem,* *nota_matematica* e *nota_redacao*): Imputados pela média de cada nota.

Criação de Atributos

Após a limpeza, os dados foram transformados de forma a se gerar caracteristicas representativas dos cursos, e não de cada discente, com o objetivo de generalizar todas as informações por curso e assim obter-se um *dataset* final composto de 11 registros, um para cada curso. Dessa forma, ao se aplicar o algoritmo de clusterização, obtenha-se grupos de cursos com características mais próximas. Os atributos construídos nesta etapa foram:

- Taxas de evasão: foram criados 12 atributos contendo as taxas de evasão por curso, um para cada ano de ingresso utilizado no estudo.

- Idade de ingresso: criação do atributo de idade de ingresso do discente a partir da data de nascimento e o ano de ingresso. Este atributo foi necessário para a criação, posterior, do atributo *mediaidadeingresso* citado logo mais.

- Intervalo em anos entre conclusão do ensino médio e ano de ingresso. Este atributo foi necessário para a criação, posterior, do atributo da média desse intervalo, *MediaInterv_EM_Ingresso*, citado logo mais.

- Médias: foram criados 11 atributos baseados em médias obtidas através do cálculo de informações dos discentes de um determinado curso. Por exemplo, a média da idade de ingresso de todos os alunos de Administração corresponde ao atributo *mediaidadeingresso* para o curso. Da mesma forma, foram calculados atributos de média do nº de membros da família, do intervalo em anos entre conclusão do ensino médio e ingresso, das notas do Enem, do índice de rendimento acadêmico, do nº de trancamentos em turmas bem como das reprovações.

- Totais: foram criados 2 atributos de totais, *SomaTotaReprovacoes* e *SomaTotalTrancamentos*, correspondento, respectivamente, ao total de reprovações no curso e total de trancamentos de curso.
- Percentual: 1 atributo correspondente ao percentual de mulheres no curso.

Para a construção dos atributos relativos às taxas de evasão, foram utilizados todos os registros do repositório R1. Para os demais, apenas dados relativos aos evadidos e não evadidos.

Seleção de Atributos

Durante a seleção de atributos, foram descartados os atributos identificadores, constantes, o target, assim como os que foram utilizados para gerar os atributos construídos especificados anteriormente. Além disso, foi aplicada a Matriz de Correlação de *Pearson* para medir a relação linear entre as variáveis. Para os resultados em que o coeficiente obtido entre dois atributos foi maior do que 0.95 foi realizada a análise e exclusão, quando adequado, de um deles (atributo).

A Figura 13 apresenta o mapa de calor da matriz. Conforme pode ser observado, foram detectados as seguintes correlações fortes entre pares de atributos:

- *mediaIdadeIngresso* e *MediaInterv_EM_Ingresso*: 0,97;
- *MediaNotaLinguagem* e *MediaNotaRedacao*: 0,97;
- *MediaReprovacoes* e *2013*: 0,96.

Optou-se então pela exclusão dos atributos *MediaInterv_EM_Ingresso* e *MediaNotaRedacao*. Quanto aos atributos *MediaReprovacoes* e *2013*, decidiu-se manter ambos, pois, apesar da forte relação linear, eles representam aspectos diferentes do fenômeno, podendo fornecer informações importantes para a análise. Ao final dessa etapa, obteve-se o *Dataset* de Trabalho 1 (D1) para a primeira etapa, com 27 atributos e 11 registros/tuplas (1 por curso). Os atributos finais são apresentados na tabela 7.

3.4.2 Transformação

Uma vez que o *K-Means*, um dos algoritmos escolhidos para ser utilizado na mineração de dados nesta etapa (não supervisionada), calcula distâncias entre pontos de dados, trabalhando portanto com dados do tipo numérico, foi realizada a transformação dos dados categóricos, adaptando-os para o algoritmo. Foi utilizada a técnica de codificação de rótulos, em que cada rótulo foi mapeado para um valor numérico único.

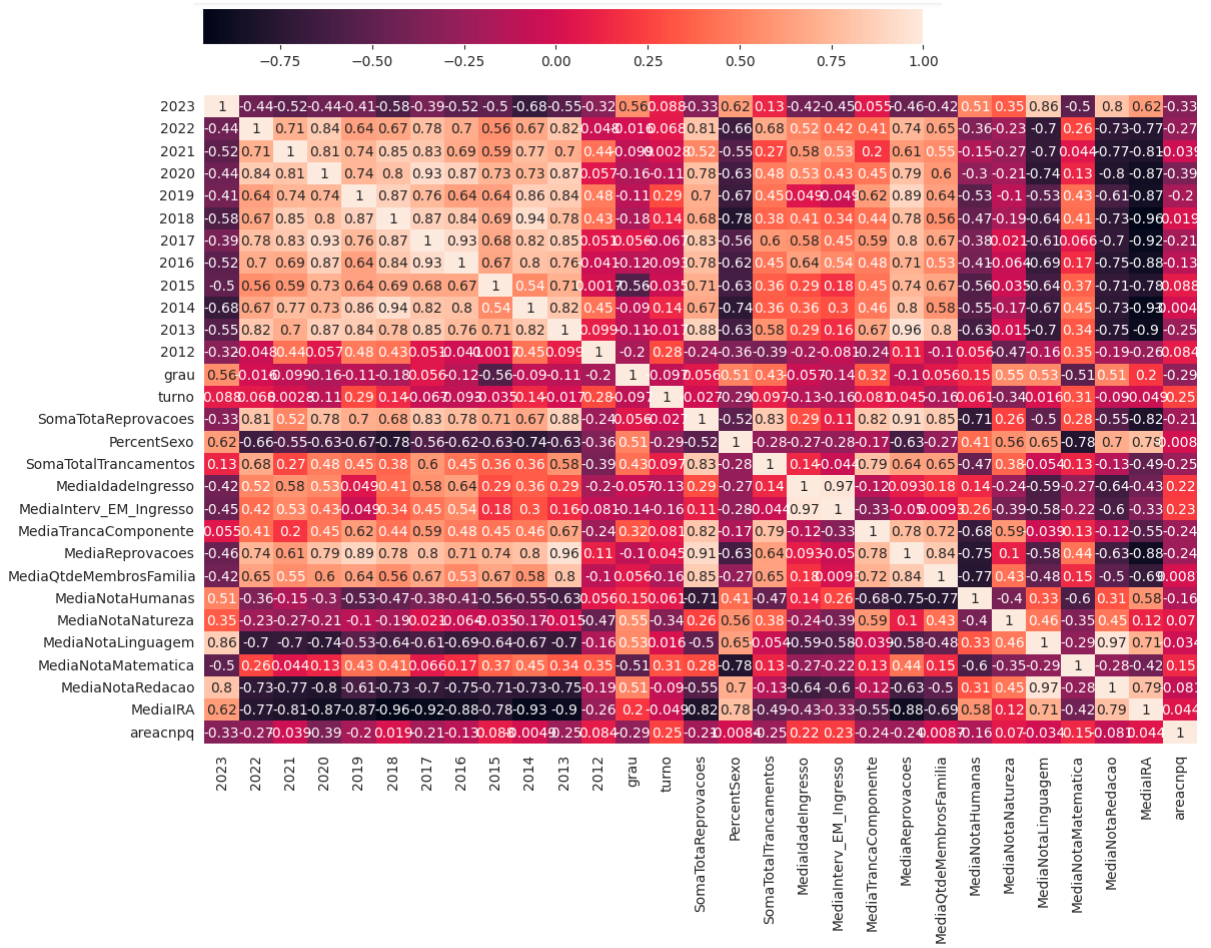


Figura 13 – Mapa de calor de correlação de *Pearson* para os atributos da etapa Não Supervisionada.

Além disso, o *K-Means* apresenta uma limitação ao não ser capaz de resolver problemas com escalas diferentes entre as variáveis de entrada. Isso significa que, para obter resultados confiáveis, é necessário que todos os dados estejam na mesma escala, caso contrário o *K-Means* pode dar peso desproporcional a variáveis que apresentam variâncias maiores, levando a uma influência desequilibrada na formação dos *clusters*. Por conta disso, foi realizada a normalização dos dados, utilizando *L2 Normalization*, conforme mencionado anteriormente, garantindo que todas as variáveis contribuam igualmente para a formação dos agrupamentos.

3.4.3 Mineração de Dados

Os algoritmos utilizados nesta fase foram *K-Means*, *K-Medoids*, *DBSCAN* e *Hierarchical Clustering* (ou Agrupamento Hierárquico). Para que seja possível a aplicação do *K-Means* e *K-Medoids*, é necessário passar como parâmetro o número de grupos desejado, *K*. Para determinar o valor ideal, para a base de dados em questão, foram aplicados os seguintes métodos de avaliação: o método de *Elbow* (conhecido também como método do

Tabela 7 – Atributos utilizados na mineração de dados da etapa Não Supervisionada.

Atributo	**Descrição**	**Tipo**
2012	Taxa de evasão do curso no ano de 2012	Numérico
2013	Taxa de evasão do curso no ano de 2013	Numérico
2014	Taxa de evasão do curso no ano de 2014	Numérico
2015	Taxa de evasão do curso no ano de 2015	Numérico
2016	Taxa de evasão do curso no ano de 2016	Numérico
2017	Taxa de evasão do curso no ano de 2017	Numérico
2018	Taxa de evasão do curso no ano de 2018	Numérico
2019	Taxa de evasão do curso no ano de 2019	Numérico
2020	Taxa de evasão do curso no ano de 2020	Numérico
2021	Taxa de evasão do curso no ano de 2021	Numérico
2022	Taxa de evasão do curso no ano de 2022	Numérico
2023	Taxa de evasão do curso no ano de 2023	Numérico
areacnpq	Áreas de conhecimento CNPq do curso	Categórico
grau	Grau do curso (Bacharelado|Licenciatura)	Categórico
turno	Turno do curso	Categórico
MediaIRA	Índice de Rendimento Acadêmico médio dos discentes do curso	Numérico
MediaIdadeIngresso	Média da idade de ingresso dos discentes do curso	Numérico
MediaNotaHumanas	Média das notas na prova de Ciências Humanas no Enem	Numérico
MediaNotaLinguagem	Média das notas na prova de Linguagem no Enem	Numérico
MediaNotaMatematica	Média das notas na prova de Matemática no Enem	Numérico
MediaNotaNatureza	Média das notas na prova de Ciências da Natureza no Enem	Numérico
MediaQtdeMembrosFamilia	Média da quantidade de membros da família dos discentes do curso	Numérico
MediaReprovacoes	Média da quantidade de reprovações em turmas	Numérico
MediaTrancaComponente	Média da quantidade de trancamento em turmas	Numérico
PercentSexo	Percentual de mulheres no curso	Numérico
SomaTotalTrancamentos	Total de trancamentos do curso	Numérico
SomaTotaReprovacoes	Total de reprovações do curso	Numérico

cotovelo) e o coeficiente de Silhueta.

O método de Elbow identifica o número ideal de *clusters* ao aumentar K até que o custo caia drasticamente e depois se estabilize, indicando que novos clusters não trazem benefícios significativos (KODINARIYA; MAKWANA et al., 2013). A Figura 14 exibe os resultados obtidos com a aplicação desta técnica. Embora a curva não forneça uma resposta definitiva, ela sugere de maneira sutil que o número ótimo de *clusters* seja 3, evidenciado pelo ponto em que o formato da curva assemelha-se a um "cotovelo" no gráfico.

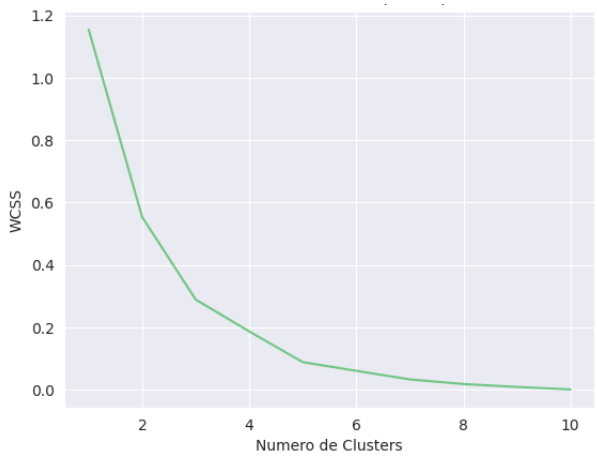


Figura 14 – Curva de Cotovelo.

Diante da sutileza do resultado obtido com a curva do cotovelo, o qual possibilita outra interpretação, como 5 *clusters*, optamos por aplicar o coeficiente de silhueta como

uma abordagem complementar, ajudando a avaliar a coesão interna dos clusters em relação à separação entre eles. Os resultados obtidos para os diferentes valores de K são estão apresentados na Figura 15.

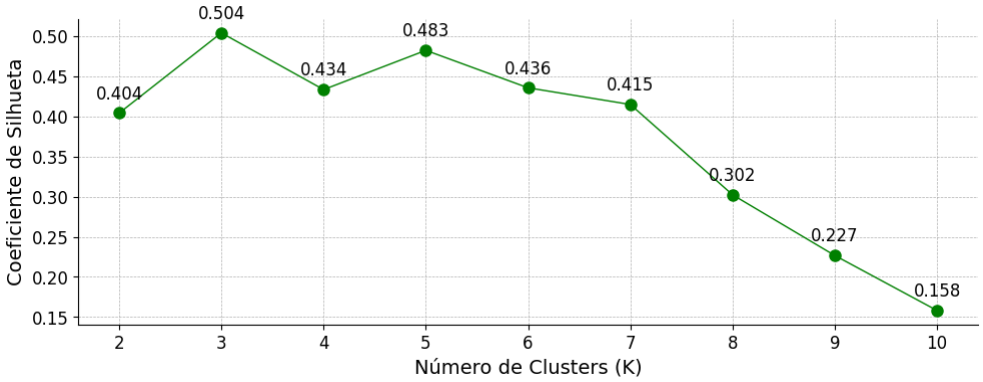


Figura 15 – Coeficiente de Silhueta para Diferentes Valores de K.

Avaliando os resultados para o coeficiente de silhueta, observamos que para 3 *clusters* obteve-se um valor mais alto. Entretanto, para 5, o valor não foi tão inferior. Os dados foram então submetidos ao *K-Means* para ambos valores de K, possibilitando a avaliação dos resultados obtidos.

3.4.4 Resultados

Os resultados obtidos com o processo de clusterização com o *K-Means*, aplicados para um número de 3 e 5 grupos, estão apresentados na Tabela 8.

Tabela 8 – Agrupamentos formados pelo *K-Means* para K=3 e K=5 clusters.

K=3	
Grupo 1 (G1)	Ciências Biológicas, Biomedicina, Engenharia de Pesca, Fisioterapia, Matemática e Turismo;
Grupo 2 (G2)	Pedagogia e Psicologia;
Grupo 3 (G3)	Administração, Ciências Contábeis e Ciências Econômicas.
K=5	
Grupo 1 (G1)	Ciências Biológicas, Engenharia de Pesca e Matemática;
Grupo 2 (G2)	Pedagogia e Psicologia;
Grupo 3 (G3)	Administração, Ciências Contábeis e Ciências Econômicas;
Grupo 4 (G4)	Biomedicina, Fisioterapia;
Grupo 5 (G5)	Turismo.

Em ambos resultados, os grupos G2 e G3 são os mesmos. Entretanto, o G1 para $K = 3$ é subdividido em 3 (G1, G4 e G5) para $K = 5$. Optamos, então, por descartar a opção com 5 *clusters*, permanecendo a de 3, pois um dos grupos resultantes, G5, possuía um único curso inviabilizando a solução de aumento de dados para o mesmo.

Foi realizada a aplicação da técnica PCA para reduzir a dimensionalidade dos dados mantendo 90% da representação original. Como resultado desse processo, foram gerados três componentes principais. Esses componentes principais são responsáveis por capturar a maior parte da variação presente nos dados, permitindo uma representação mais compacta. Para visualizar a proximidade dos pontos de dados, foram criados 2 gráficos de dispersão, um para $K = 3$, Figura 16, e outro para $K = 5$, Figura 17. As coordenadas dos pontos de dados são representadas pelos três componentes principais, o que possibilitou a geração de gráficos no espaço tridimensional. Os centroides de cada cluster foram identificados e marcados nos gráficos, assim como os pontos/cursos.

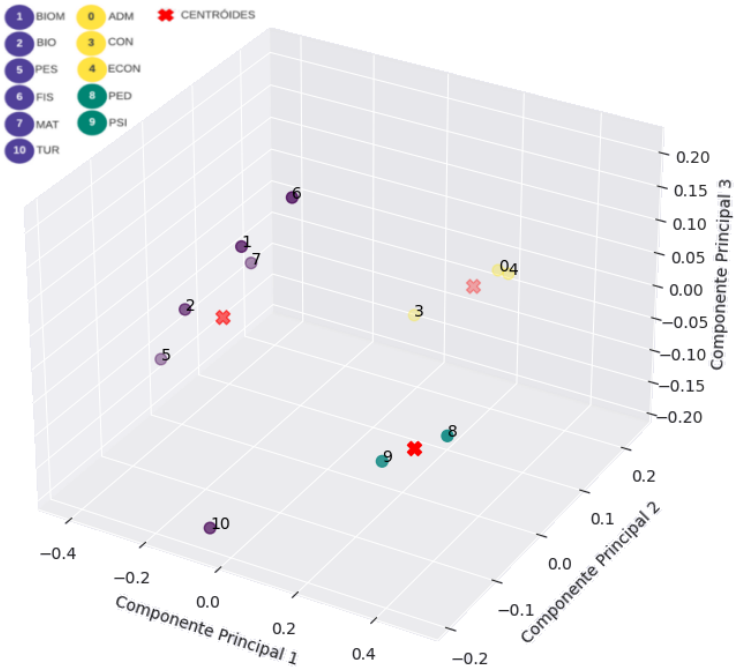


Figura 16 – Gráfico de dispersão com o resultado do *K-Means* para $K = 3$.

Para os demais algoritmos, foi realizada a redução da dimensionalidade dos dados por PCA para 2 componentes principais. Os resultados obtidos foram todos similares aos do *K-Means* para 3 *clusters*, com os mesmos agrupamentos de cursos. Na Figura 18 tem-se o resultado para o *K-Medoids*.

O DBSCAN dispensa a necessidade prévia de conhecer o número total de clusters. No entanto, a definição de dois parâmetros-chave, ϵ *(eps) = 0,2* e *MinPts (min_samples) = 2*, têm um grande impacto no desempenho e nos resultados do algoritmo. A aplicação do algoritmo aos dados resultou na formação de três grupos sem a presença de ruídos, refletindo a mesma composição dos algoritmos anteriores, conforme ilustrado na Figura 19.

Por fim, optou-se pela aplicação da técnica de Agrupamento Hierárquico. Esta abordagem também apresenta a vantagem de não exigir conhecimento prévio sobre a quantidade de clusters, permitindo que os dados revelem naturalmente sua estrutura hierárquica. Foi utilizada a técnica de agrupamento hierárquico aglomerativo *Ward*. A

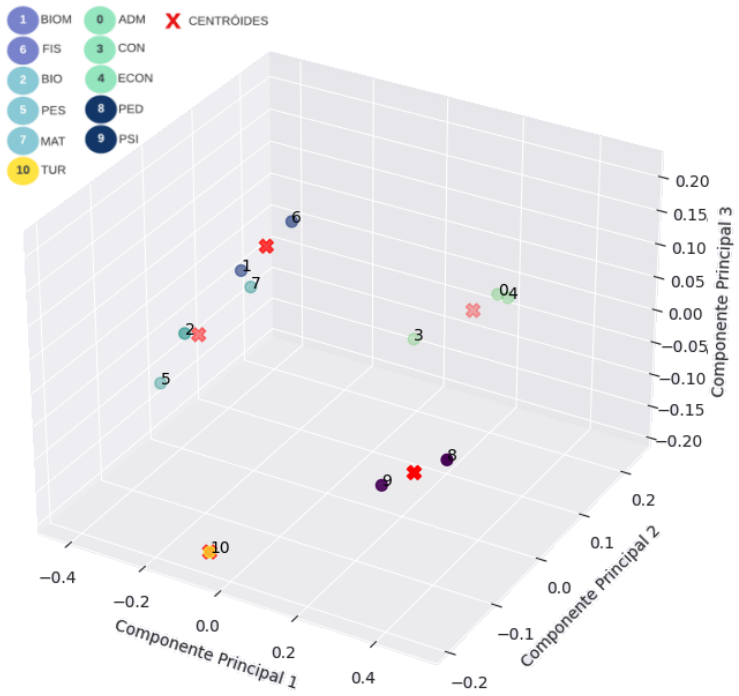


Figura 17 – Gráfico de dispersão com o resultado do *K-Means* para $K = 5$.

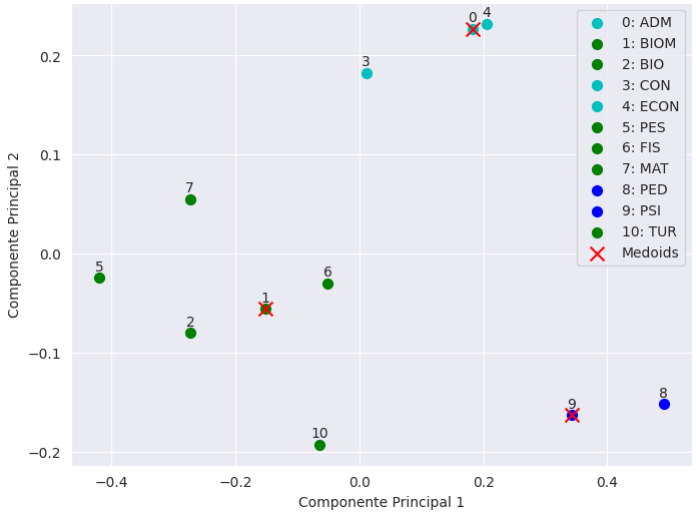


Figura 18 – Gráfico de dispersão com o resultado do *K-Medoids* para $K = 3$.

representação gráfica foi realizada por meio do dendrograma, podendo ser observada na Figura 20, o qual proporcionou uma visão mais aprofundada da formação dos clusters. No dendrograma, os pontos de dados são representados como folhas individuais e a altura na qual as linhas do dendrograma são unidas indica a distância ou dissimilaridade entre os clusters que estão sendo combinados.

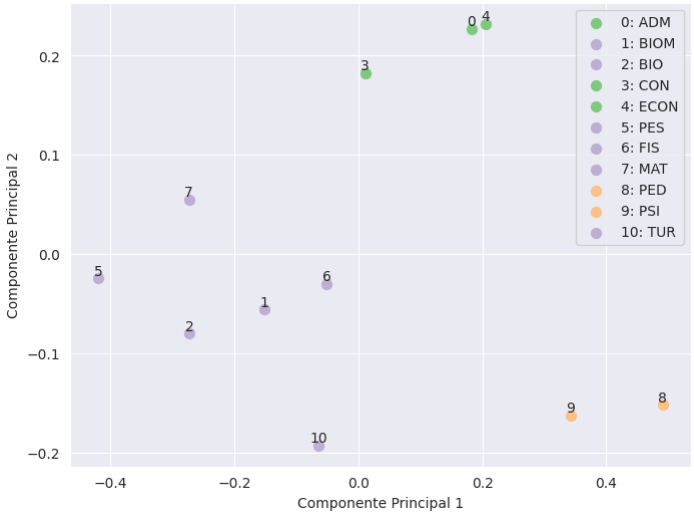


Figura 19 – Gráfico de dispersão com o resultado do *DBSCAN* para *eps=0.2* e *minsamples=2*.

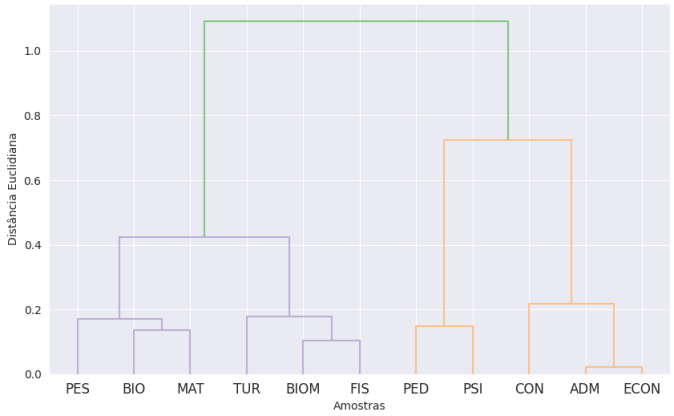


Figura 20 – Dendrograma Hierárquico.

3.5 Etapa 2 – Supervisionada

O objetivo desta etapa é criar modelos de predição para os grupos de cursos identificados na etapa anterior e os cursos de forma individual, utilizando a abordagem de separação dos dados por períodos, levando em conta a temporalidade das informações. Após realização da coleta e geração do repositório R2, seguiu-se para o pré-processamento dos dados, seguido da mineração e análise dos resultados.

3.5.1 Pré-Processamento

Similar a Etapa Não Supervisionada, o pré-processamento e a transformação dos dados também foram iterativos, necessitando de avançar e retroceder entre essas duas fases. No pré-processamento foi realizada limpeza e seleção de atributos manual, seguido pela transformação dos dados, com a criação de atributos e conversões de atributos categóricos em numéricos.

Limpeza

A Limpeza foi realizada da mesma forma que na Etapa 1 para os atributos coincidentes. Para os demais, as estratégias adotadas foram as seguintes:

- *conclusao_escola_publica*: Os registros nulos foram imputados conforme ação afirmativa referente. Se esta era "Ampla Concorrência", imputado como *False*, caso contrário, *True*. Os casos residuais (em que a ação afirmativa encontrava-se nula) imputados com *False*;
- *estadocivil*: Registros nulos imputados pelo mais frequente;
- *raca*: Registros nulos imputados pelo mais frequente;
- *uf_naturalidade*: Registros nulos imputados pelo mais frequente;
- *opcao_inscricao*: Registros nulos imputados pelo mais frequente;
- *tipoacaoafirmativa*: Registros nulos imputados com valor zero (0);
- *id_processo_seletivo* e *id_convocacao_processo_seletivo*: ambos atributos foram tratados de forma conjunta, pois quando um está nulo, necessariamente o outro também está. Eles foram necessários para criar um atributo de ordem de convocação, detalhado logo mais. Os casos nulos foram imputados com zero (0).

Criação de Atributos

A criação de atributos desempenhou um papel significativo no processo, ocorrendo em duas fases distintas: durante a coleta de dados e posteriormente, dentro da transformação. Durante a coleta, a criação de atributos envolveu a definição e extração de informações adicionais a partir das fontes de dados originais. Essa ação foi direcionada para enriquecer o conjunto de dados com informações relevantes para a análise, tornando-o mais abrangente e informativo, compilando grandes quantidades de registros em atributos representativos. Os atributos criados durante a coleta foram apresentados na sessão 3.3.

Já na transformação, a criação de atributos ocorreu como parte do refinamento dos dados coletados. Nesse contexto, a ênfase estava na engenharia de atributos, onde novas

características foram derivadas com base nas existentes, a fim de melhorar a capacidade de modelagem e a extração de conhecimento dos dados. São eles:

- *idade_ingresso*: Atributo de idade de ingresso do discente criado a partir da data de nascimento e do ano de ingresso, conforme realizado na Etapa 1;
- *intervalo_em_ingresso*: Intervalo em anos entre conclusão do ensino médio e ano de ingresso, conforme realizado na Etapa 1;
- *convocacao*: Relaciona-se à convocação de um discente dentro de um processo seletivo, indicando se a convocação ocorreu na primeira chamada, segunda, terceira, ou em chamadas subsequentes, conforme apropriado. Como o sistema armazena essa informação como um atributo de texto, no qual os usuários têm a liberdade de inserir informações de forma não padronizada, o que aumenta a probabilidade de erros e inconsistências, tomamos a decisão de eliminar o atributo original e criar um novo a partir dos atributos disponíveis, *id_processo_seletivo* e *id_convocacao_processo_seletivo*. Isso foi feito para garantir a integridade e a padronização dos dados, tornando a análise mais confiável;
- *tem_ensino*: Indica se o discente participou de atividades de ensino ou não. Criado a partir do atributo *quantidadeensino*.

Seleção de Atributos

Foi realizada ainda uma seleção manual de atributos na qual excluiu-se atributos identificadores e atributos que foram utilizados para a criação de outros. Além disso, assim como da Etapa 1, foi aplicada a Matriz de Correlação de *Pearson* para medir a relação linear entre as variáveis. Para os resultados em que o coeficiente obtido entre dois atributos foi maior que $|0.95|$, foi-se realizada a exclusão de um deles.

As fortes correlações identificadas foram para dois tipos de atributos de cada semestre, *qtd_aprovacao_x* e *ch_integralizada_x*, dos atributos acadêmicos cumulativos, totalizando 8 pares de correlações. Os valores para os 8 semestres ficaram muito próximos uns dos outros, sendo o do 1º semestre ($x = 1$) igual a 0.990694. Optou-se pela exclusão dos atributos *ch_integralizada_x*.

O resultado desta fase foi um Dataset de Trabalho 2 (D2) composto de 106 atributos e 5805 registros, dos quais 3298 da classe “evadido” e 2507 “não evadido”. Na Tabela 9 são listados todos os atributos resultantes desta fase e que serão utilizados na mineração de dados, com exceção dos atributos *curso* e *qtde_periodo_cursado_cronologicamente* que são utilizados como chaves nas divisões dos conjuntos dos cursos e períodos e excluídos posteriormente, conforme será detalhado mais a frente.

Tabela 9 – Atributos utilizados na mineração de dados da etapa supervisionada.

Atributos Fixos	**Descrição**	**Tipo**
area_cnpq	Áreas de conhecimento CNPq do curso	Categórico
conclusao_escola_publica	Se o discente concluiu o ensino médio em escola pública	Booleano
convocacao	Número da convocação do discente dentro de um processo seletivo	Numérico
curso	Curso do discente	Categórico
cursounapandemia	Se o discente cursou turmas no período da pandemia (2020 a 2021)	Booleano
estadocivil	Estado civil	Categórico
evadido	Situação do discente (Target: Evadido|Não Evadido)	Booleano
grau	Grau do curso (Bacharelado|Licenciatura)	Categórico
idade_ingresso	Idade em que o discente ingressou no curso	Numérico
intervalo_em_ingresso	Intervalo em anos entre conclusão do ensino médio e ano de ingress	Numérico
nota_ciencias_humanas	Nota na prova de Ciências Humanas no Enem	Numérico
nota_ciencias_natureza	Nota na prova de Ciências da Natureza no Enem	Numérico
nota_linguagem	Nota na prova de Linguagem no Enem	Numérico
nota_matematica	Nota na prova de Matemática no Enem	Numérico
nota_redacao	Nota na prova de Redação no Enem	Numérico
opcao_inscricao	Opção de inscrição no processo seletivo (1|2)	Numérico
periodo_ingresso	Período de ingresso na instituição	Numérico
qtde_periodo_cursado_cronologicamente	Quantidade de semestres transcorridos desde o ingresso	Numérico
quantidade_membros_familia	Quantidade de membros da família	Numérico
raca	Raça	Categórico
sexo	Sexo do discente	Categórico
tem_ensino	Se o discente participou de atividades de ensino	Booleano
temcredito_cc_ca	Se possui crédito concedido ou crédito automático no histórico	Booleano
tipoacaoafirmativa	Tipo de ação afirmativa do discente no ingresso	Categórico
turno	Turno do curso	Categórico
uf_naturalidade	UF de nascimento	Categórico
Atributos Cumulativos	**Descrição**	**Tipo**
ap_ob_semestre_x	Quantidade de aprovações em disciplinas obrigatórias no prazo	Numérico
media_acumulada_x	Média do rendimento acadêmico	Numérico
qtd_aprovacao_x	Quantidade de turmas aprovadas no curso	Numérico
qtd_cancelamento_turma_x	Quantidade de cancelamento de turmas	Numérico
qtd_reprovacao_ft_nt_x	Quantidade de turmas reprovadas por falta e nota no curso	Numérico
qtd_reprovacao_ft_x	Quantidade de turmas reprovadas por falta no curso	Numérico
qtd_reprovacao_nt_x	Quantidade de turmas reprovadas por nota no curso	Numérico
qtd_trancamento_turma_x	Quantidade de trancamentos de turmas	Numérico
rep_ob_semestre_x	Quantidade de reprovações de disciplinas obrigatórias	Numérico
trancamento_programa_x	Quantidade de trancamentos de curso	Numérico

3.5.1.1 Análise dos Atributos

Para compreender quais atributos contribuem mais significativamente para a predição dos resultados, foi realizada uma análise utilizando o ganho de informação. Para isso, aplicamos a função *mutual_info_classif* em cada um dos conjuntos de dados, obtendo um resultado único para cada um deles.

De forma geral, foram identificados os 10 melhores atributos para cada um dos 112 conjuntos de dados formados. A frequência com que cada um desses atributos aparece nos modelos está representada na Figura 21. Podemos observar que os atributos *ap_ob_semestre*, *rep_ob_semestre*, *qtd_aprovacao* e *qtd_reprovacao_ft_nt* estão presentes entre os 10 melhores atributos em quase 100% dos modelos. Além disso, os atributos referente às notas do Enem (*nota_aprovacao*, *nota_linguagem*, *nota_ciencias_humanas*, *nota_ciencias_natureza* e *nota_matematica*) se destacaram, sendo mais frequentes do que atributos como sexo, idade de ingresso, estado civil, raça, dentre outros.

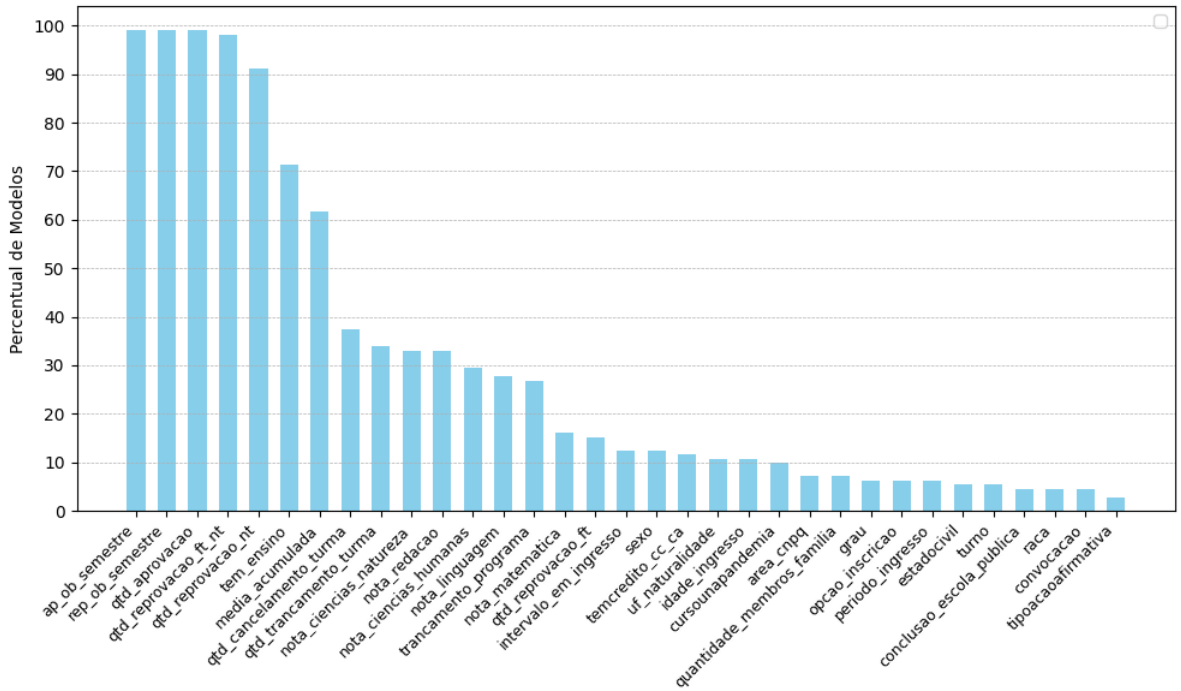


Figura 21 – Atributos mais frequentes presentes nas listas dos 10 melhores atributos de cada modelo.

De forma mais detalhada, os gráficos ilustrados na Figura 22 demonstram o posicionamento dos atributos para G1, G2 e G3 ao longo dos 8 períodos estudados, considerando o ganho de informação individual de cada atributo. Optou-se por limitar a exibição dos atributos que se posicionaram abaixo do 10º lugar nos gráficos, uma vez que, após essa posição, observa-se uma oscilação significativa que pode comprometer a interpretação dos dados. Em todos os três gráficos, o atributo que exerce a maior influência é o *qtd_aprovacao*, que se destaca por ocupar a primeira posição em todos os períodos de

G1 e G3. Em G2, este atributo alterna sua posição com *ap_ob_semestre*. Adicionalmente, os atributos *rep_ob_semestre*, *qtd_reprovacao_ft_nt* e *qtd_reprovacao_nt* variam entre a segunda e a quinta posição nos três gráficos.

Por outro lado, os atributos que possuem menor relevância nos modelos dos grupos, considerando como critério que a posição no ranqueamento por ganho de informação para todos os períodos seja superior a 15, podem ser observados na Tabela 10.

Tabela 10 – Atributos menos significativos para os modelos dos grupos.

				GRUPO1					
ATRIBUTO	**P1**	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**SOMA**
cursounapandemia	30	23	28	30	32	33	26	27	229
turno	21	33	33	31	27	19	31	32	227
quantidade_membros_familia	26	32	23	29	29	27	18	29	213
opcao_inscricao	29	29	19	27	31	25	25	23	208
periodo_ingresso	20	27	27	24	21	32	22	31	204
nota_matematica	16	26	29	32	24	26	24	26	203
convocacao	18	28	26	20	33	28	16	33	202
conclusao_escola_publica	25	24	30	21	25	20	30	16	191
raca	22	21	21	26	28	29	21	20	188
uf_naturalidade	23	31	17	18	19	16	28	19	171
				GRUPO2					
ATRIBUTO	**P1**	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**SOMA**
cursounapandemia	26	31	32	24	26	31	26	31	227
nota_matematica	25	28	27	23	27	27	30	27	214
temcredito_cc_ca	19	30	21	18	23	30	28	30	199
opcao_inscricao	23	24	29	19	24	25	27	20	191
quantidade_membros_familia	21	18	18	16	28	21	25	22	169
				GRUPO3					
ATRIBUTO	**P1**	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**SOMA**
area_cnpq	16	25	33	33	33	33	22	22	217
uf_naturalidade	28	19	29	25	27	27	21	31	207
raca	30	27	27	27	22	24	19	29	205
turno	24	26	32	18	24	31	16	23	194
quantidade_membros_familia	25	30	18	22	30	17	20	32	194

3.5.2 Transformação

Na transformação foi realizada a conversão dos dados categóricos para numérico, utilizando a técnica de codificação de rótulos, em que cada rótulo foi mapeado para um valor numérico único. Após a conversão, os dados se tornam mais acessíveis para uma ampla variedade de algoritmos de aprendizado de máquina.

D2 foi então subdividido em oito conjuntos de dados distintos, denominados P1, P2, P3, P4, P5, P6, P7 e P8, representando cada um dos oito períodos analisados. Cada conjunto de dados inclui todos os atributos estáticos (fixos), que não mudam ao longo do progresso acadêmico, via de regra, e os atributos cumulativos que são pertinentes ao semestre em que o estudante se encontra.

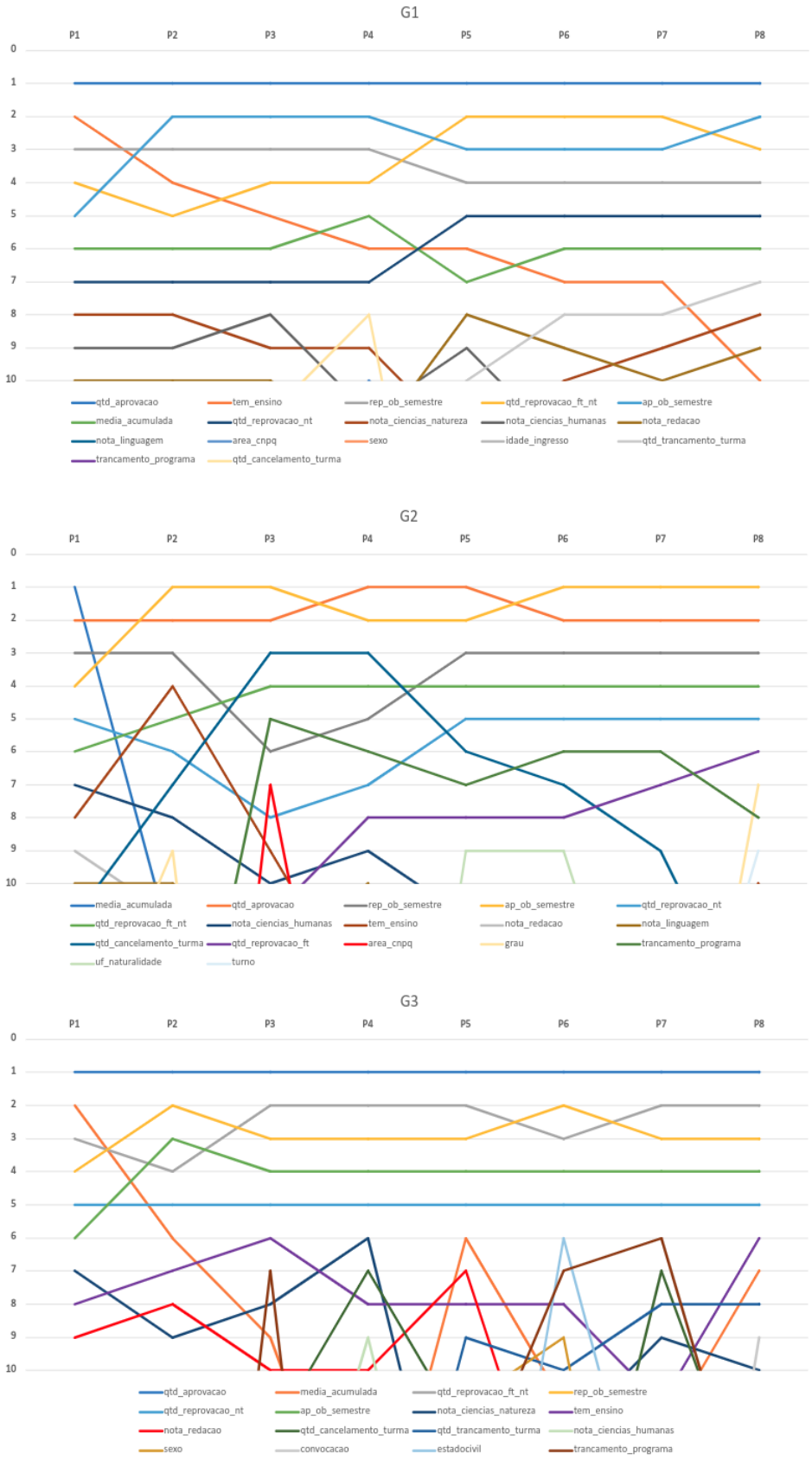


Figura 22 – Ranqueamento dos atributos por ganho de informação dos conjuntos dos grupos G1, G2 e G3 para cada período.

Além disso, os dados foram segmentados de forma que, a medida que os discentes deixam a instituição, evadindo ou concluindo o curso, eles são retirados dos conjuntos subsequentes. Dessa forma a maioria dos não evadidos (concluíram o curso) estão presentes em todos os conjuntos de dados, com exceção aqueles que concluíram antecipadamente, e os evadidos estão todos nos modelos do 1º período, mas o número cai com o avanço dos períodos e os discentes evadem.

Por fim, de cada conjunto de dados dos períodos, foram gerados os conjuntos de dados para cada curso. Além desses, utilizando o resultado dos agrupamentos formados na Etapa 1, foram criados conjuntos de dados para os grupos de cursos. Dessa forma obteve-se 112 conjuntos: 24 oriundos de 3 grupos x 8 períodos, que se referem a proposta deste trabalho, somados a 88, oriundos de 11 cursos x 8 períodos, que são os conjuntos a gerarem os modelos para comparação.

Para a separação dos dados entre períodos, foi utilizado o atributo *qtde_periodo_ cursado_ cronologicamente*, e para a separação entre cursos, foi utilizado o atributo *curso*, sendo ambos descartados após a separação. Dessa forma, tem-se 34 atributos em cada um dos conjuntos a serem utilizados na etapa de mineração de dados.

Na Figura 23 pode ser observado os totais de exemplos para cada modelo dos cursos e dos grupos formados conforme resultados dos algoritmos da etapa não supervisionada. Já na Figura 24 tem-se a proporção da classe "evadido" ($evadidos/total$) para cada curso/grupo e período, podendo-se observar que os cursos de Engenharia de Pesca, Matemática e Ciências Econômicas, são, nessa ordem, os cursos em que se tem mais evasão. Na contramão, o curso com menor evasão é o de Pedagogia.

3.5.3 Mineração de Dados

Cada um dos 112 conjuntos de dados foram então submetidos aos algoritmos *Random Forest*, *Gaussian Naive Bayes*, *K-Nearest Neighbors* e *Multi-layer Perceptron*, aplicando-se então um algoritmo para cada paradigma do aprendizado supervisionado.

Dada a diversidade de modelos disponíveis, optou-se por uma abordagem padronizada e genérica para configurar os algoritmos, com os valores dos hiperparâmetros escolhidos arbitrariamente devido à grande quantidade de modelos. Para o *Random Forest*, decidiu-se não realizar a poda das árvores de decisão, mantendo-as completas, e utilizar 10 árvores independentes ($n_estimators = 10$), correspondendo a 10% do valor padrão de 100, para reduzir o tempo de processamento. No caso do *MLP*, estabeleceu-se um número máximo de iterações, *max_iter=1000*. Para o *KNeighborsClassifier*, definiu-se *n_neighbors = 3*, o menor número ímpar após 1, para evitar empates na classificação.

Ao que se refere as métricas auferidas durante a mineração, no contexto da evasão escolar, é fundamental priorizar aquelas que se concentrem na precisão da predição de que

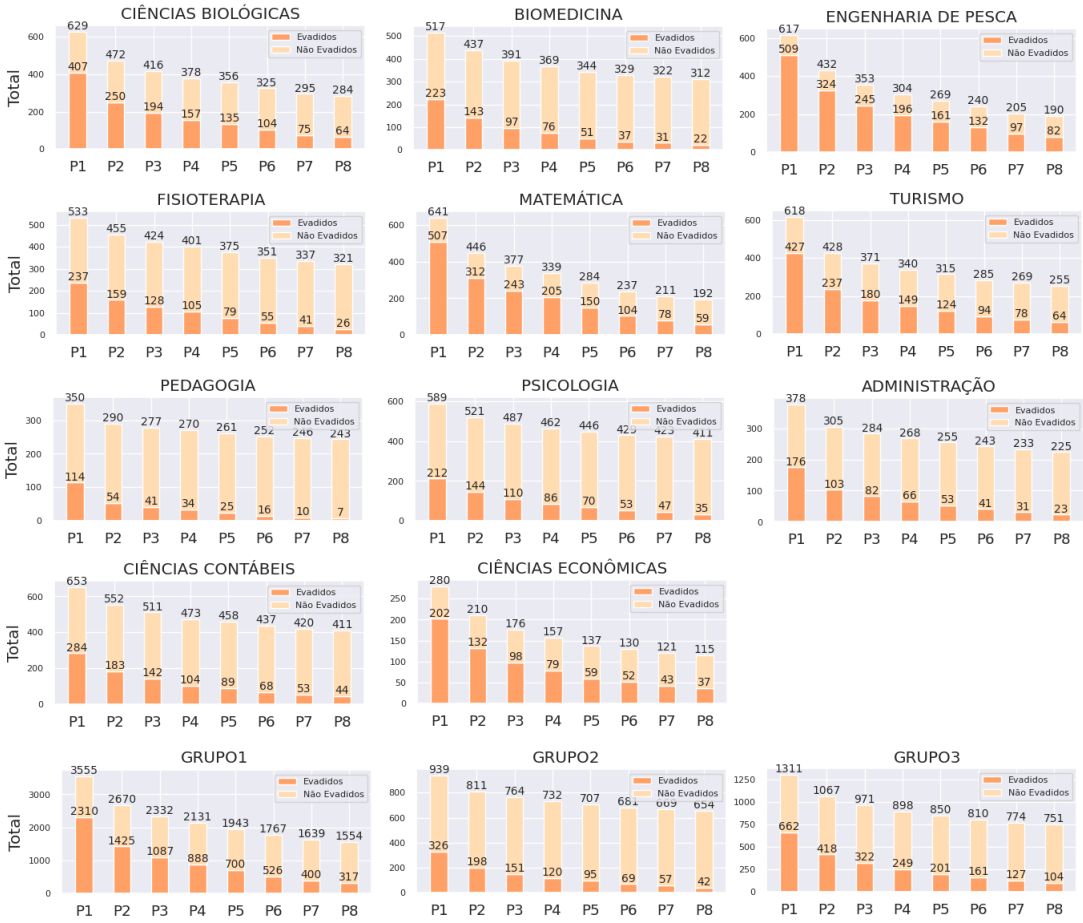


Figura 23 – Totais de exemplos de cada conjunto de dados com destaque para a classe "evadido".

um aluno não evadirá. Isso porque a identificação incorreta de alunos em risco de evasão pode ter implicações significativas, influenciando a implementação de políticas preventivas direcionadas à retenção desses alunos na instituição de ensino. Com essa perspectiva em mente, optamos por empregar duas métricas na avaliação dos modelos neste trabalho: o *Recall* e o Coeficiente de Correlação de *Matthews*. Ambas obtidas por meio de validação cruzada com 10 subdivisões (*K-Fold Cross-Validation*).

O RC é interessante para situações em que os Falsos Negativos são considerados mais prejudiciais que os Falsos Positivos. Especificamente no contexto deste trabalho, ele se concentra na capacidade do modelo de identificar corretamente os casos positivos, ou seja, os alunos que estão em risco de evasão. A predição de evasão é crítica para permitir a intervenção precoce e a implementação de estratégias de apoio a alunos em risco. Se o *Recall* for baixo, isso significa que o modelo está perdendo muitos casos de alunos que realmente precisam de ajuda, o que pode resultar em uma alta taxa de evasão não detectada. Portanto, uma medição precisa do *Recall* ajuda a garantir que o sistema de predição esteja identificando a maioria dos alunos em risco, permitindo que as instituições educacionais ajam proativamente para fornecer suporte e evitar a evasão.

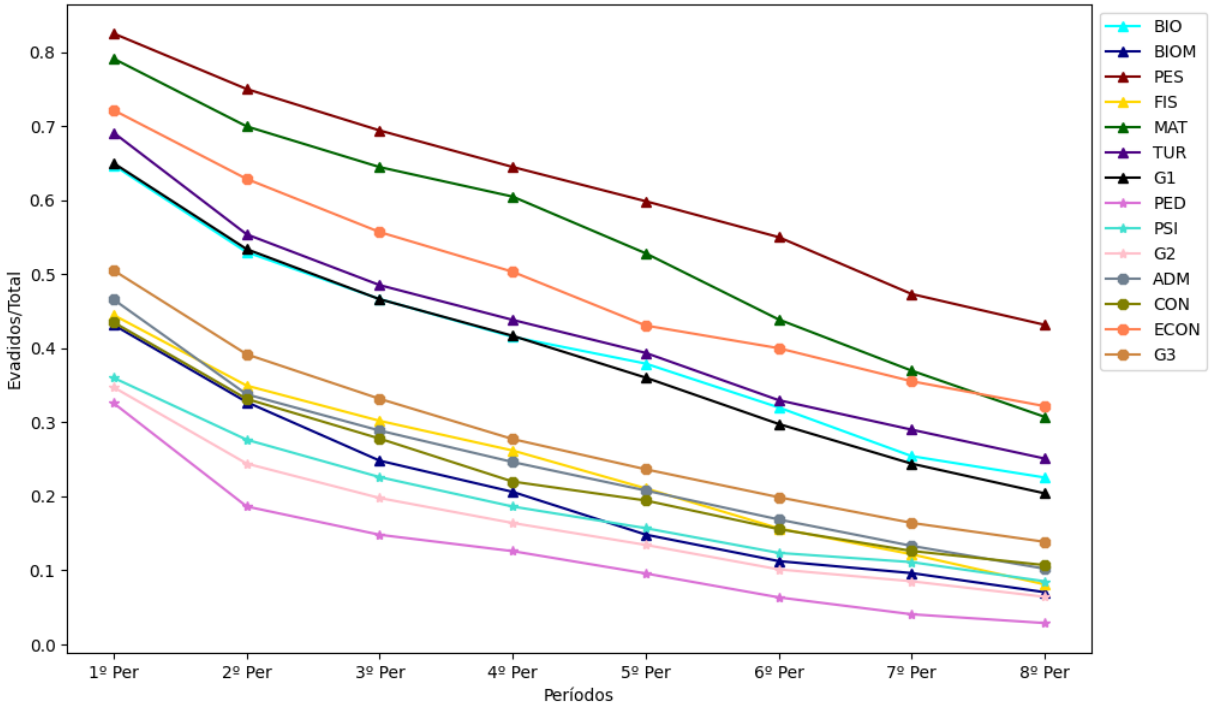


Figura 24 – Proporção da classe "evadido"para cada periodo e curso.

Quanto ao MCC, é uma métrica de avaliação de desempenho amplamente usada em problemas de classificação, especialmente em classificação binária. Seu grande benefício é poder explicar a qualidade do modelo que lida com conjuntos de dados desequilibrados, que é o que acontece nos modelos deste trabalho, acentuando-se com a evolução dos períodos. Segundo Chicco e Jurman (2020), a acurácia e o *F1-Score*, embora amplamente utilizadas, podem induzir a resultados enganosos em conjuntos de dados desequilibrados, pois não consideram adequadamente a proporção entre elementos positivos e negativos. O MCC é sugerido como uma alternativa eficaz devido às suas propriedades matemáticas que lidam com o desequilíbrio dos dados e sua consistência, mesmo quando as classes são trocadas. O MCC é simples e intuitivo, exigindo que o classificador faça predições corretas tanto para a maioria dos casos negativos quanto para a maioria dos casos positivos, independentemente de suas proporções no conjunto de dados global.

Finalizada a fase de mineração de dados da etapa supervisionada, partiu-se então para avaliação dos resultados. A Figura 25 fornece uma representação visual de todo o processo realizado, abrangendo ambas as etapas, não supervisionada e supervisionada, e destacando a interconexão entre elas. De forma resumida, na primeira etapa, não supervisionada, foi realizado o processo de KDD, transformando a base selecionada em um conjunto reduzido com características dos cursos. Aplicando-se a mineração de dados por meio da clusterização, foram identificados grupos de cursos. Com essas informações, partiu-se para a segunda etapa, supervisionada, onde tambpem foi aplicado o KDD e foram gerados modelos de predição para cada curso e período, além de modelos para cada grupo, identificado na etapa anterior, e período. A partir dos resultados de *Recall* e *MCC*,

foi realizado um teste estatístico para determinar o melhor resultado.

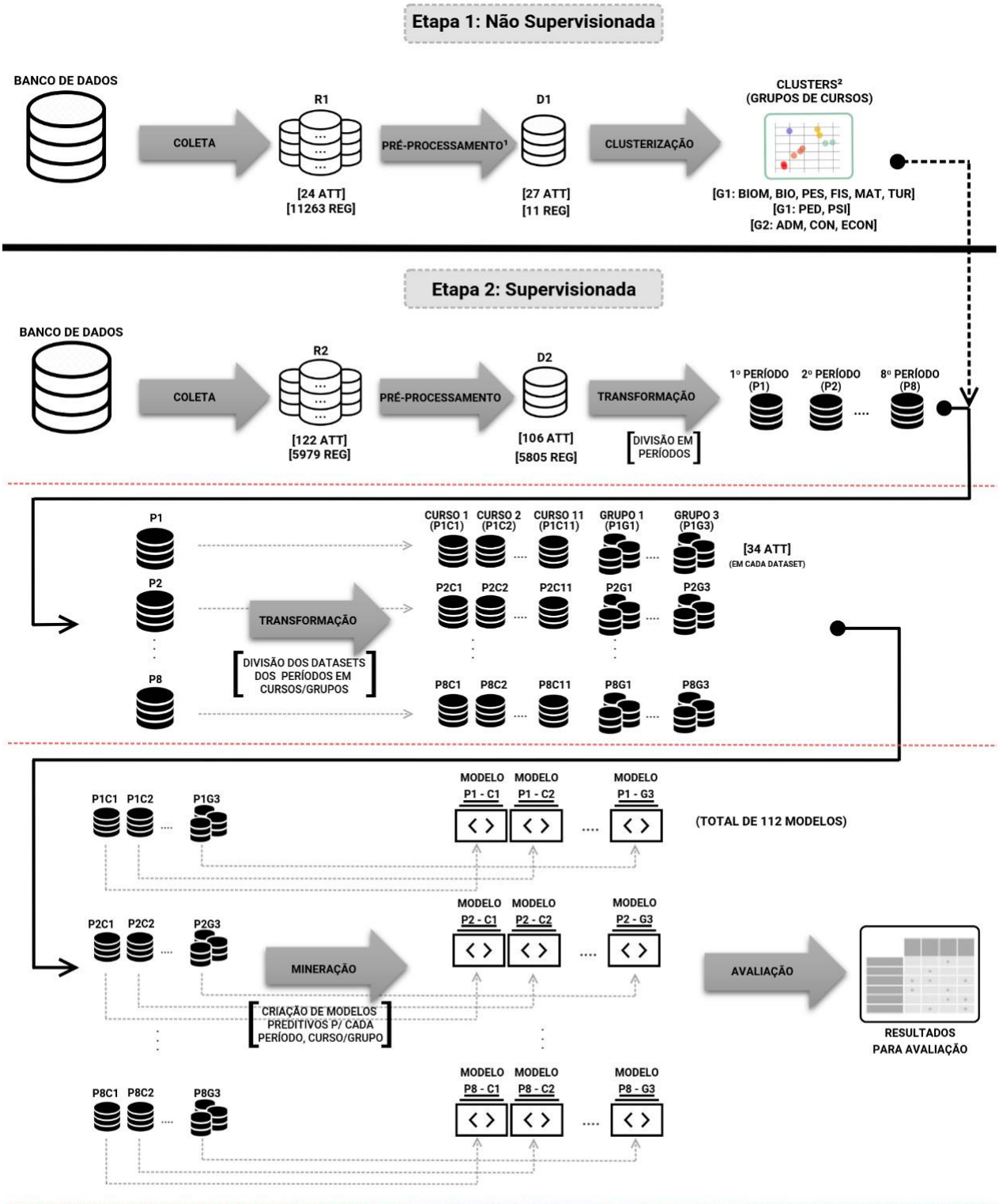


Figura 25 – Diagramação da técnica desenvolvida.

4 Resultados e Discussão

Com os resultados da Etapa 1, é importante realizarmos uma comparação dos quantitativos de amostras presentes nos conjuntos de dados formados por cada grupo, e os conjuntos de dados de cada curso individualmente para compreender como a formação de grupos influencia a disponibilidade de dados para cada modelo de predição criado.

Para isso, na Figura 26, é realizada uma comparação desses quantitativos relacionadas à classe minoritária (evadidos). A figura fornece uma representação visual dos grupos, identificados como G1, G2 e G3, juntamente com seus cursos correlacionados. Os valores numéricos são destacados sobre as linhas que representam cada grupo e cada curso que contém a maior quantidade de amostras dessa classe. Essa representação gráfica permite uma análise direta do ganho de dados da classe minoritária ao longo dos diferentes períodos proporcionado pelos agrupamentos realizados.

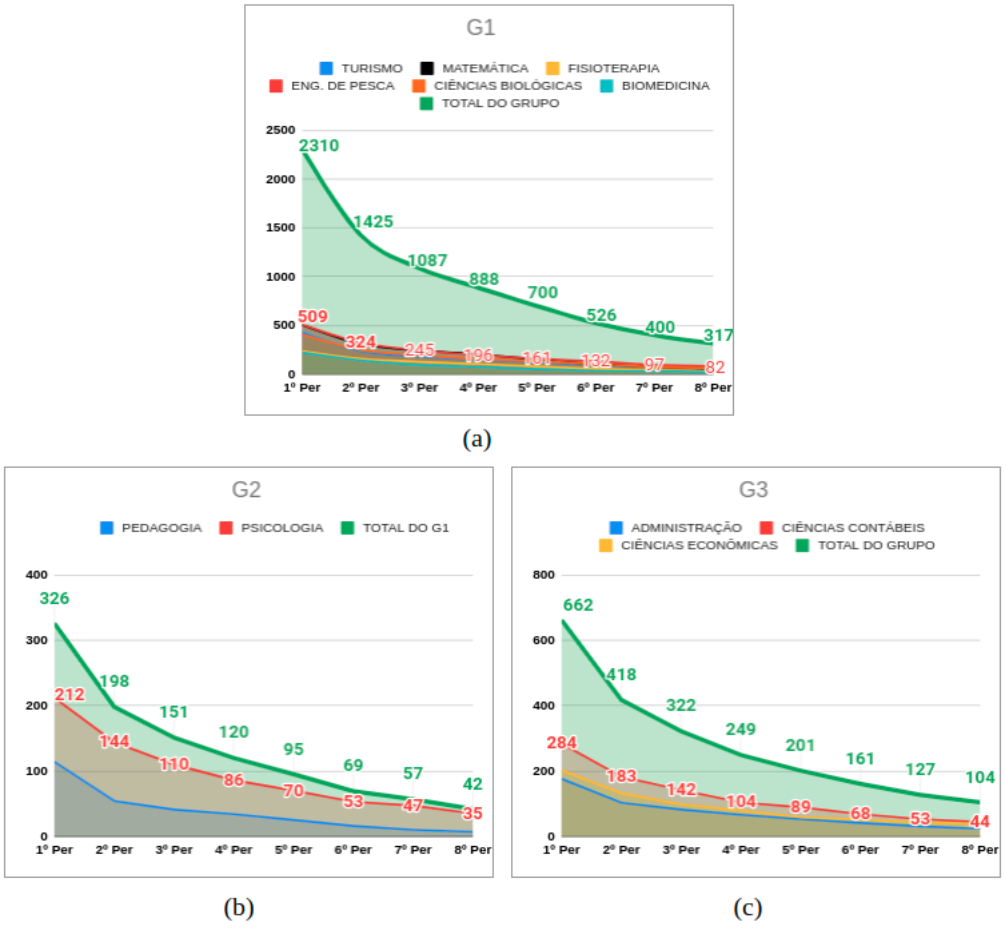


Figura 26 – Total de evadidos por período e curso comparados aos dos agrupamentos G1(a), G2(b) e G3(c).

Na Tabela 11, é possível obter uma visão abrangente da proporção do total de dados do Grupo (não apenas à classe minoritária) em relação à quantidade de dados do

curso que compõe esse grupo, expressa em valores percentuais. Em cursos nos quais o benefício promovido pelo agrupamento foi mais modesto, observou-se um aumento de pelo menos 55% como no caso de PSI, e 82% em CON. No entanto, a maioria dos cursos apresentou ganhos substanciais, com valores que ultrapassam 400%, com destaque para PES, onde o ganho atingiu 717,89%.

Tabela 11 – Percentual de aumento de dados proporcionado pelos agrupamentos em comparação aos cursos.

PER	BIO	BIOM	PES	FIS	MAT	TUR	PED	PSI	ADM	CON	ECON
1º	465,18%	587,62%	476,18%	566,98%	454,60%	475,24%	168,29%	59,42%	246,83%	100,77%	368,21%
2º	465,68%	510,98%	518,06%	486,81%	498,65%	523,83%	179,66%	55,66%	249,84%	93,30%	408,10%
3º	460,58%	496,42%	560,62%	450,00%	518,57%	528,57%	175,81%	56,88%	241,90%	90,02%	451,70%
4º	463,76%	477,51%	600,99%	431,42%	528,61%	526,76%	171,11%	58,44%	235,07%	89,85%	471,97%
5º	445,79%	464,83%	622,30%	418,13%	584,15%	516,83%	170,88%	58,52%	233,33%	85,59%	520,44%
6º	443,69%	437,08%	636,25%	403,42%	645,57%	520,00%	170,24%	58,74%	233,33%	85,35%	523,08%
7º	455,59%	409,01%	699,51%	386,35%	676,78%	509,29%	171,95%	58,16%	232,19%	84,29%	539,67%
8º	447,18%	398,08%	717,89%	384,11%	709,38%	509,41%	169,14%	59,12%	233,78%	82,73%	553,04%

4.1 Resultados

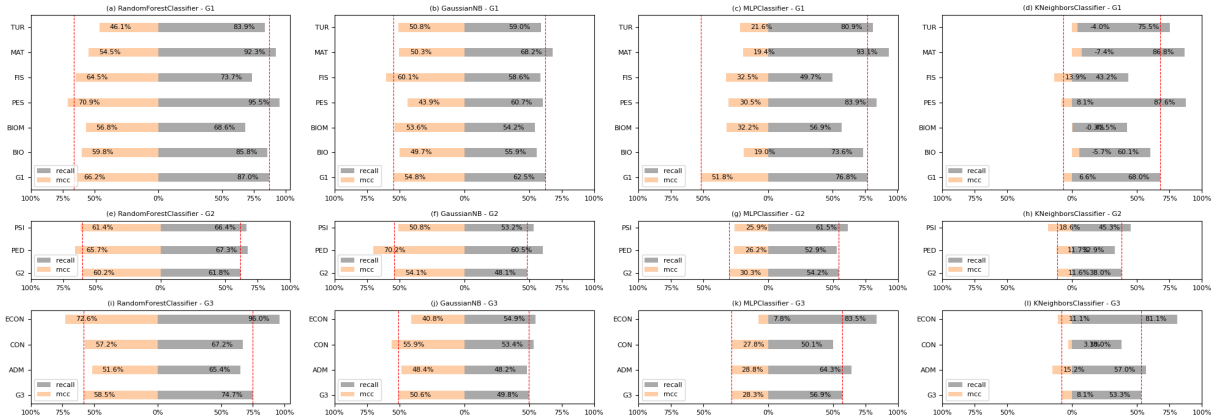
São apresentados na Figura 27 os resultados das métricas de *Recall* e Coeficiente de Correlação *Matthews* para os algoritmos *Random Forest*, *Gaussian Naive Bayes*, *Multilayer Perceptron* e *K-Nearest Neighbors* durante a Etapa 2 (Supervisionada), abrangendo os modelos desenvolvidos nos períodos de 1 a 8. Cada gráfico na figura exibe os desempenhos dos cursos individuais e dos grupos compostos por eles. Linhas tracejadas vermelhas foram adicionadas para marcar os resultados dos grupos G1, G2 e G3. Isso permite uma comparação visual entre os modelos, para cada uma das métricas, identificando aqueles em que a estratégia de aumento de dados por meio de união de dados de cursos aparentam gerar resultados superiores ou inferiores em comparação com a abordagem que considera os cursos de forma isolada.

Na Figura 28, apresentamos esses mesmos resultados em uma outra obordagem. Essa figura permite a análise comparativa dos algoritmos, facilitando a identificação do algoritmo com melhor desempenho para cada modelo. Nela, é realizada a comparação dos resultados dos modelos de cada curso e grupo ao longo dos 8 períodos para os 4 algoritmos avaliados, tanto para RC (lado esquerdo) como para MCC (lado direito).

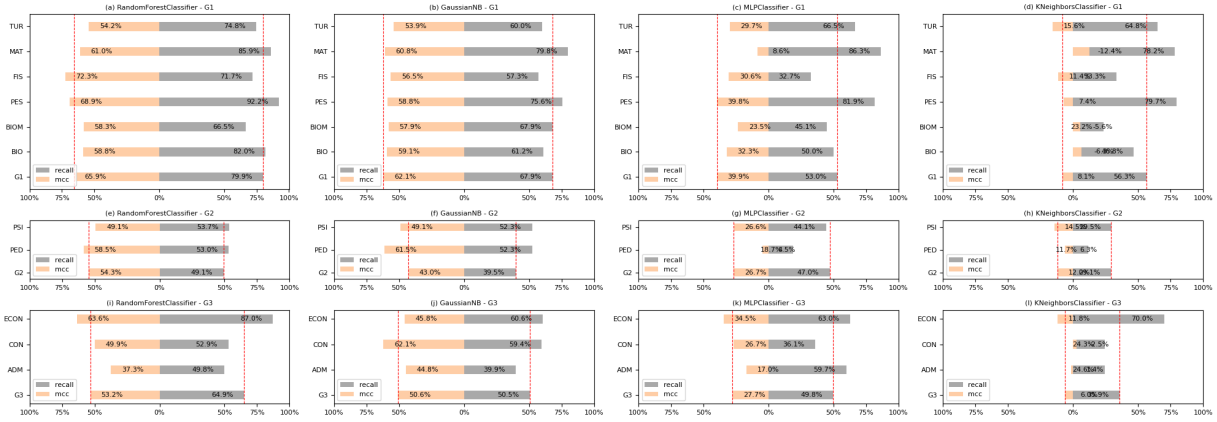
4.2 Avaliação dos Resultados

4.2.1 Avaliação dos Algoritmos

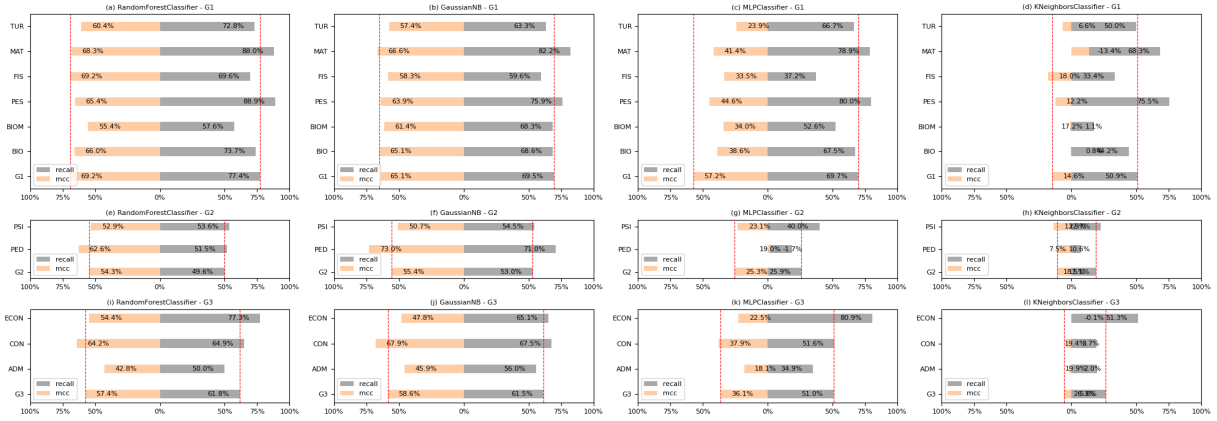
Avaliando os resultados apresentados na Figura 28, é perceptível que o algoritmo utilizado do paradigma baseado em exemplos, *KNeighborsClassifier*, demonstra o pior



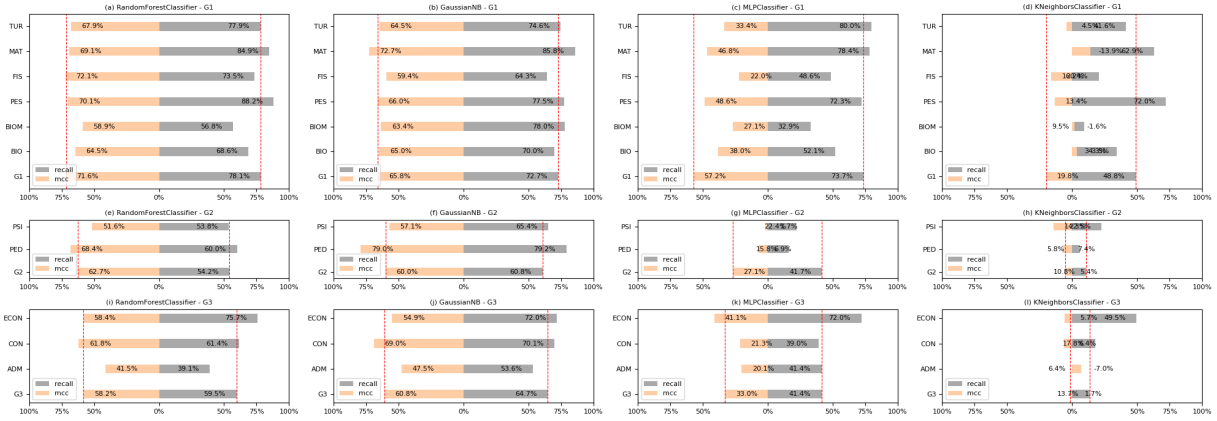
1º Período



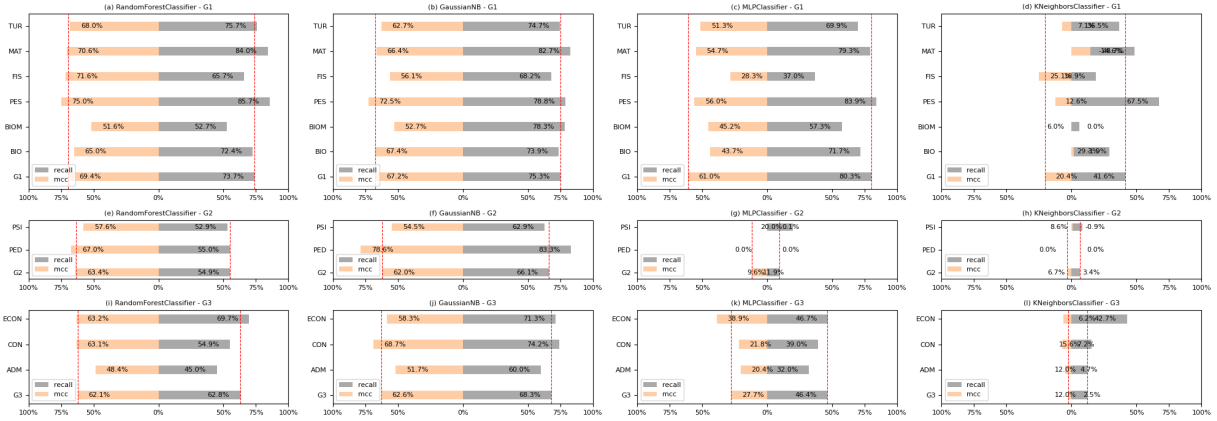
2º Período



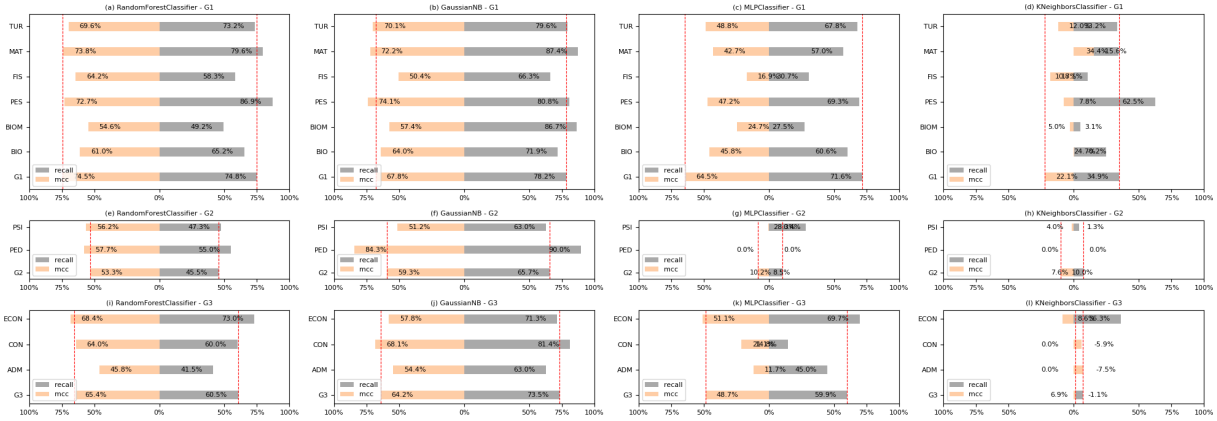
3º Período



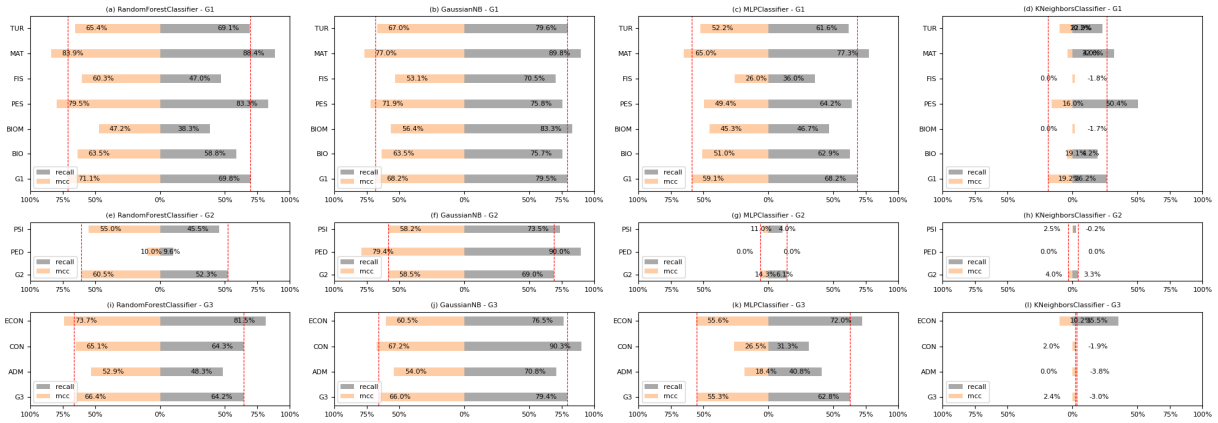
4º Período



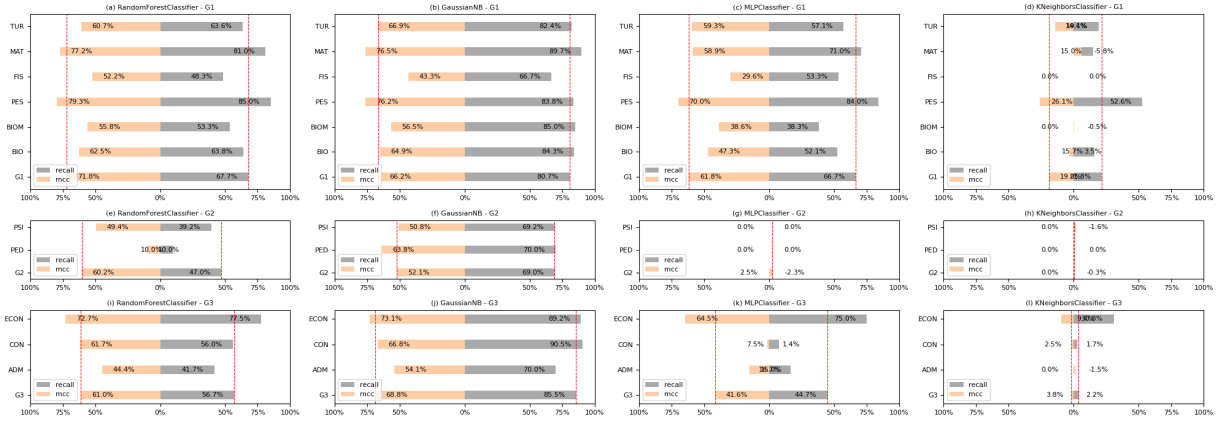
5º Período



6º Período



7º Período

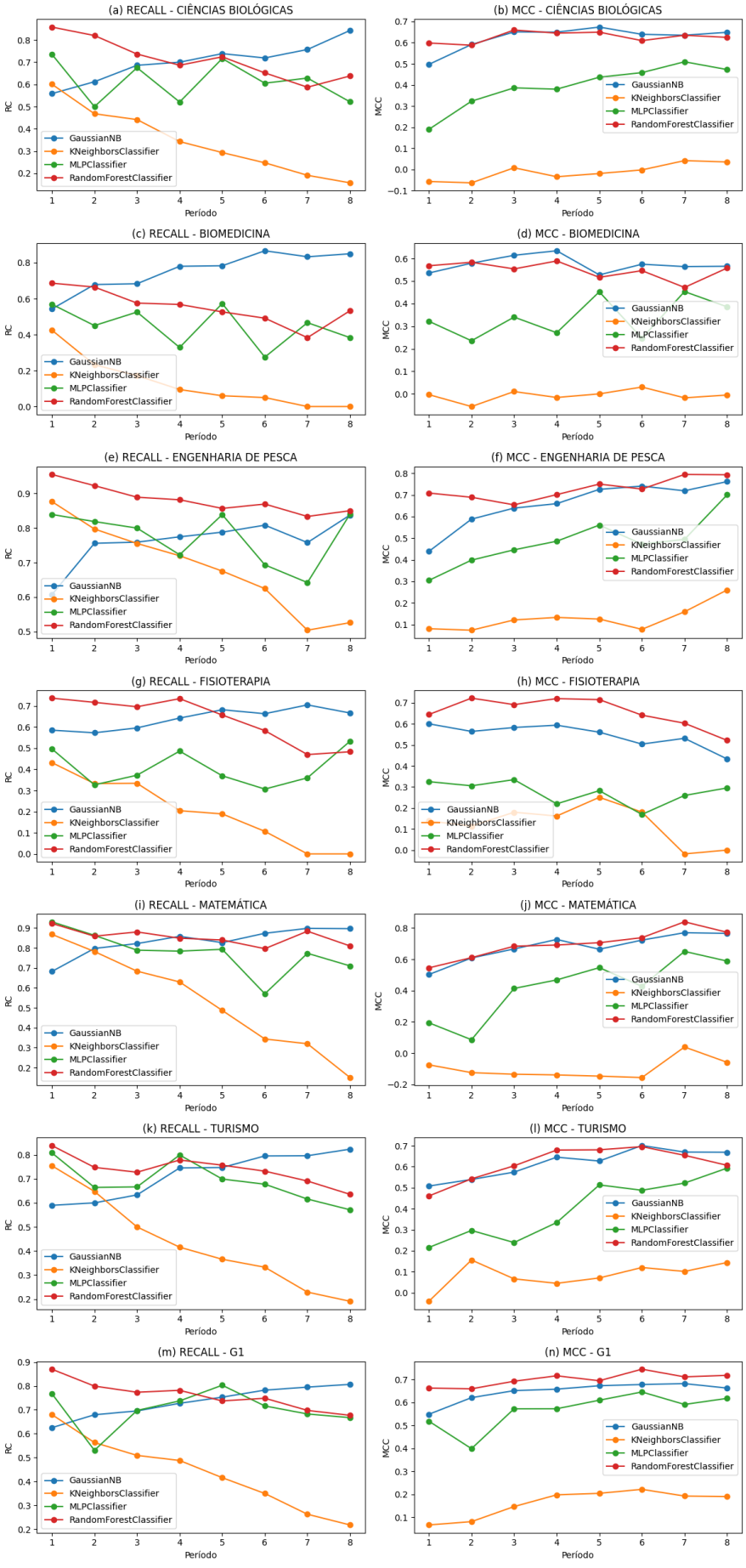


8º Período

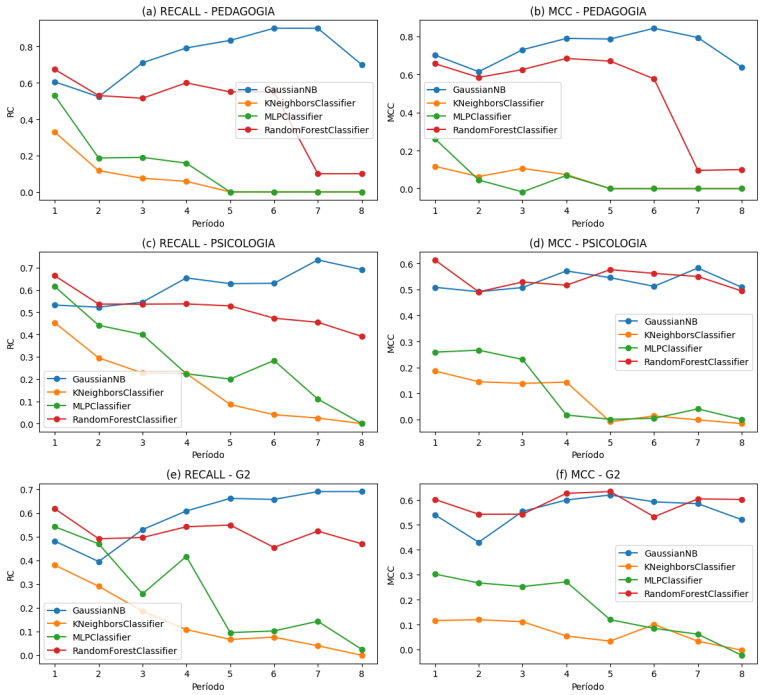
Figura 27 – Resultados das métricas RC e MCC do 1º ao 8º período para *Random Forest*, *Gaussian Naive Bayes*, *Multi-layer Perceptron* e *K-Nearest Neighbors*.

desempenho em comparação com os outros algoritmos. Os valores obtidos pelo *KNeighborsClassifier* são consistentemente baixos em praticamente todos os modelos, sugerindo que este paradigma pode não ser adequado para o tipo de problema abordado. Uma possível justificativa para esse desempenho é o fato de que o algoritmo requer um parâmetro, K, que determina a quantidade de vizinhos mais próximos a serem considerados durante a classificação. É importante ajustar o valor de K de acordo com a natureza dos dados e com as características específicas do problema em questão. No entanto, devido à grande variedade de modelos avaliados, o valor foi padronizado de forma arbitrária para todos os casos. Essa padronização pode ter contribuído para o desempenho inferior do algoritmo em comparação com os outros avaliados.

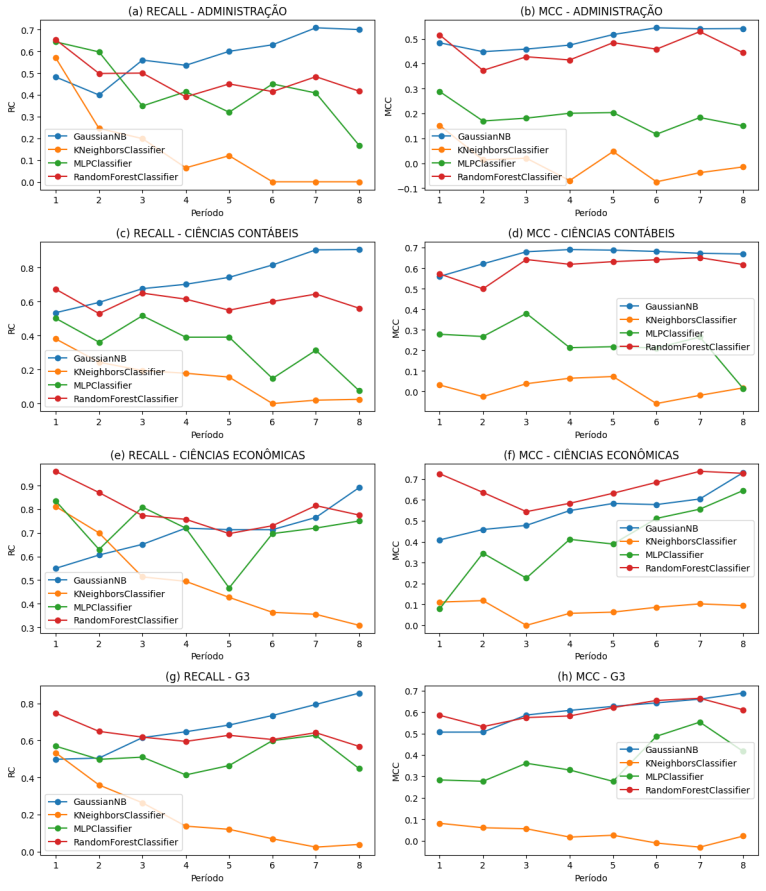
Quanto ao MLP, em linhas gerais, seu desempenho também não foi satisfatório, e em alguns casos, mostrou-se tão pouco eficaz quanto o *KNeighborsClassifier*, como evidenciado nos modelos do G2 e nos cursos associados a ele (por exemplo, PED do 5º ao 8º período). Os melhores resultados para RC, superando os demais algoritmos, foram



G1 e cursos que o compõem.



G2 e cursos que o compõem.



G3 e cursos que o compõem.

Figura 28 – Comparação entre os algoritmos *Random Forest*, *Gaussian Naive Bayes*, *Multilayer Perceptron* e *K-Nearest Neighbors*.

observados no 5º período de G1 (80,29%), no 4º período de TUR (79,95%), no 3º período de ECON (80,89%) e no 1º período de ADM (64,35%).

Os melhores desempenhos foram alcançados com *Random Forest* e *Gaussian Naive Bayes*, apresentando resultados mais similares em termos de MCC e alternando entre si em relação ao *Recall*. De modo geral, observamos que o *Random Forest* se destacou como o melhor algoritmo para todos os modelos de 1º período, em ambas as métricas, com exceção apenas do MCC para TUR, o curso mais distinto comparado aos demais, e PED, curso com menor índice de evasão. Contudo, ao analisarmos o *Recall* para este algoritmo, percebemos uma tendência de declínio, conforme avançamos nos períodos, e aclive, para o *Gaussian Naive Bayes*, apresentando este último melhores resultados para os últimos períodos.

Essa tendência é observada em praticamente todos os cursos e grupos, com a exceção de PES, em que, apesar do declínio no desempenho do *Random Forest* e do crescimento no *Gaussian Naive Bayes*, o *Random Forest* continua sendo superior em todos os períodos. Os resultados mais expressivos foram registrados para o *Random Forest* no 1º período de ECON, com 96,02%, e em PES, com 95,48%. Por outro lado, o *Gaussian Naive Bayes* alcançou os melhores resultados nos 7º e 8º períodos de CON, com 90,33% e 90,5%, respectivamente.

Destaca-se o curso de PED do 6º ao 8º período, onde a diferença nos resultados entre *Gaussian Naive Bayes* e *Random Forest* é bastante significativa, com o primeiro demonstrando superioridade em ambas as métricas consideradas. Essa discrepância também é observada, em menor escala, nos 6º, 7º e 8º períodos de BIOM, nos 7º e 8º períodos de CON, nos 7º e 8º períodos de PSI, no 8º período do G2 e no 8º período de ADM, todos para RC.

Ao analisarmos esses dados à luz da Figura 24, percebemos que, para esses cursos e períodos, a proporção da classe "Evadido" é consideravelmente menor. Isso sugere que o *Gaussian Naive Bayes* obteve os melhores resultados nos modelos com conjuntos de dados mais desbalanceados. Essa observação ressalta a capacidade do *Gaussian Naive Bayes* de lidar com desequilíbrios nos dados e produzir resultados mais precisos em cenários onde a classe minoritária é menos representativa.

Dos 112 modelos gerados, em 65 deles (58,03%) o *Gaussian Naive Bayes* tem RC superior ao *Random Forest*, ficando este último com 47 (41,96%). Já olhando para MCC, os resultados se invertem, ficando 51 deles (45,54%) melhores com o *Gaussian Naive Bayes* e 61 (54.46%) com o *Random Forest*.

4.2.2 Avaliação dos Desempenhos dos Grupos em Relação aos Cursos

Analisando a Figura 27, podemos perceber que os resultados dos grupos (G1, G2 e G3) ao longo dos períodos, aparentam ter melhor desempenho do que os modelos individuais em grande parte dos modelos. Entretanto, para realizar uma avaliação estatística da diferença entre os resultados de um modelo de um curso e do grupo ao qual esse curso pertence, no mesmo período e algoritmo, foi empregado o Teste de Hipótese Z com um nível de significância de 5%. Os resultados dessa avaliação, aplicados às métricas RC e MCC, são apresentados nas Tabelas 12 e 13 respectivamente. Nessas tabelas, os valores "0", "+1" e "-1" representam os seguintes cenários:

- Valor "0": Indica que não há diferença significativa entre os resultados do modelo do curso e do modelo agrupado;
- Valor "+1": Indica que o modelo agrupado é significativamente melhor do que o modelo do curso para a métrica em questão;
- Valor "-1": Indica que o modelo agrupado é significativamente pior do que o modelo do curso com relação à métrica em questão.

Essa abordagem estatística ajuda a quantificar a diferença entre os modelos de cursos individuais e os modelos agrupados, permitindo uma avaliação objetiva das melhorias ou desafios proporcionados pelos agrupamentos no contexto proposto.

Ao examinarmos os dados expostos na Tabela 12 referente à métrica *Recall*, fica evidente que a aplicação da solução proposta por este estudo é viável para a maioria dos casos em todos os algoritmos avaliados. Destacam-se os cursos BIO e FIS, nos quais a viabilidade atinge 100% para todos os modelos em todos os algoritmos. Detalhadamente:

- *Random Forest*: considerando a perspectiva dos cursos, observa-se uma viabilidade de 100% para BIO, BIOM, FIS, TUR, PSI, ADM e CON. De maneira geral, a proposta pode ser aplicada em aproximadamente 73,86% dos casos, obtendo uma melhora significativa em 40,91% aproximadamente.
- *Gaussian Naive Bayes*: considerando a perspectiva dos cursos, constatamos uma viabilidade de 100% para BIO, FIS, TUR e ADM. Em termos gerais, a proposta pode ser aplicada em cerca de 68,18% dos casos, resultando em uma melhoria significativa de aproximadamente 21,59%.
- *Multi-layer Perceptron*: ao considerarmos os cursos, a viabilidade é de 100% para BIO, BIOM, FIS, PED e CON. De forma geral, a proposta pode ser aplicada em aproximadamente 72,73% dos casos, obtendo uma melhora significativa de cerca de 50%.

Tabela 12 – Teste Z - RC.

RandomForestClassifier											
PER	BIO	BIOM	PES	FIS	MAT	TUR	PED	PSI	ADM	CON	ECON
1º	0	**+1**	-1	**+1**	-1	**+1**	0	0	**+1**	**+1**	-1
2º	0	**+1**	-1	**+1**	-1	**+1**	0	0	**+1**	**+1**	-1
3º	0	**+1**	-1	**+1**	-1	0	0	0	**+1**	0	-1
4º	**+1**	**+1**	-1	**+1**	-1	0	0	0	**+1**	0	-1
5º	0	**+1**	-1	**+1**	-1	0	0	0	**+1**	**+1**	0
6º	**+1**	**+1**	-1	**+1**	0	0	-1	0	**+1**	0	-1
7º	**+1**	**+1**	-1	**+1**	-1	0	**+1**	**+1**	**+1**	0	-1
8º	0	**+1**	-1	**+1**	-1	0	**+1**	**+1**	**+1**	0	-1
GaussianNB											
PER	BIO	BIOM	PES	FIS	MAT	TUR	PED	PSI	ADM	CON	ECON
1º	**+1**	**+1**	0	0	-1	0	-1	0	0	0	0
2º	**+1**	0	-1	**+1**	-1	**+1**	-1	-1	**+1**	-1	-1
3º	0	0	-1	**+1**	-1	**+1**	-1	0	0	-1	0
4º	0	-1	0	**+1**	-1	0	-1	0	**+1**	-1	0
5º	0	0	0	**+1**	-1	0	-1	0	**+1**	-1	0
6º	**+1**	-1	0	**+1**	-1	0	-1	0	**+1**	-1	0
7º	0	0	0	**+1**	-1	0	-1	0	**+1**	-1	0
8º	0	0	0	**+1**	-1	0	0	0	**+1**	-1	0
MLPClassifier											
PER	BIO	BIOM	PES	FIS	MAT	TUR	PED	PSI	ADM	CON	ECON
1º	0	**+1**	-1	**+1**	-1	-1	0	-1	-1	**+1**	-1
2º	0	**+1**	-1	**+1**	-1	-1	**+1**	0	-1	**+1**	-1
3º	0	**+1**	-1	**+1**	-1	0	**+1**	-1	**+1**	0	-1
4º	**+1**	**+1**	0	**+1**	0	-1	**+1**	**+1**	0	0	-1
5º	**+1**	**+1**	0	**+1**	0	**+1**	**+1**	-1	**+1**	**+1**	0
6º	**+1**	**+1**	0	**+1**	**+1**	0	**+1**	-1	**+1**	**+1**	-1
7º	0	**+1**	0	**+1**	-1	**+1**	**+1**	0	**+1**	**+1**	-1
8º	**+1**	**+1**	-1	**+1**	0	**+1**	**+1**	**+1**	**+1**	**+1**	-1
KNeighborsClassifier											
PER	BIO	BIOM	PES	FIS	MAT	TUR	PED	PSI	ADM	CON	ECON
1º	**+1**	**+1**	-1	**+1**	-1	-1	0	-1	0	**+1**	-1
2º	**+1**	**+1**	-1	**+1**	-1	-1	**+1**	0	**+1**	**+1**	-1
3º	**+1**	**+1**	-1	**+1**	-1	0	**+1**	0	**+1**	**+1**	-1
4º	**+1**	**+1**	-1	**+1**	-1	**+1**	**+1**	-1	**+1**	-1	-1
5º	**+1**	**+1**	-1	**+1**	-1	0	**+1**	0	0	0	-1
6º	**+1**	**+1**	-1	**+1**	0	0	**+1**	**+1**	**+1**	**+1**	-1
7º	**+1**	**+1**	-1	**+1**	0	0	**+1**	0	**+1**	0	-1
8º	**+1**	**+1**	-1	**+1**	**+1**	0	0	0	**+1**	0	-1

- *K-Nearest Neighbors*: considerando os cursos, a viabilidade atinge 100% para BIO, BIOM, FIS, PED e ADM. De maneira geral, a proposta pode ser aplicada em aproximadamente 70,45% dos casos, proporcionando uma melhoria significativa de cerca de 48,86%.

Quando avaliamos a aplicabilidade da solução proposta pela métrica MCC, Tabela 13, observamos resultados ainda mais promissores. Destacam-se os cursos BIO e BIOM, nos quais a viabilidade é de 100% para todos os modelos em todos os algoritmos. Detalhadamente:

Tabela 13 – Teste Z - MCC.

RandomForestClassifier											
PER	BIO	BIOM	PES	FIS	MAT	TUR	PED	PSI	ADM	CON	ECON
1º	**+1**	**+1**	-1	0	**+1**	**+1**	0	0	**+1**	0	-1
2º	**+1**	**+1**	0	-1	**+1**	**+1**	0	0	**+1**	0	-1
3º	0	**+1**	0	0	0	**+1**	-1	0	**+1**	-1	0
4º	**+1**	**+1**	0	0	0	0	0	**+1**	**+1**	0	0
5º	0	**+1**	0	0	0	0	0	0	**+1**	0	0
6º	**+1**	**+1**	0	**+1**	0	0	0	0	**+1**	0	0
7º	**+1**	**+1**	-1	**+1**	-1	0	**+1**	0	**+1**	0	0
8º	**+1**	**+1**	-1	**+1**	0	**+1**	**+1**	**+1**	**+1**	0	-1
GaussianNB											
PER	BIO	BIOM	PES	FIS	MAT	TUR	PED	PSI	ADM	CON	ECON
1º	**+1**	0	**+1**	-1	**+1**	0	-1	0	0	-1	**+1**
2º	0	0	0	**+1**	0	**+1**	-1	-1	0	-1	0
3º	0	0	0	**+1**	0	**+1**	-1	0	**+1**	-1	**+1**
4º	0	0	0	**+1**	-1	0	-1	0	**+1**	-1	0
5º	0	**+1**	0	**+1**	0	0	-1	**+1**	**+1**	-1	0
6º	0	**+1**	-1	**+1**	0	0	-1	**+1**	**+1**	0	0
7º	0	**+1**	0	**+1**	-1	0	-1	0	**+1**	0	0
8º	0	**+1**	-1	**+1**	-1	0	-1	0	**+1**	0	0
MLPClassifier											
PER	BIO	BIOM	PES	FIS	MAT	TUR	PED	PSI	ADM	CON	ECON
1º	**+1**	**+1**	**+1**	**+1**	**+1**	**+1**	0	0	0	0	**+1**
2º	**+1**	**+1**	0	**+1**	**+1**	**+1**	**+1**	0	**+1**	0	-1
3º	**+1**	**+1**	**+1**	**+1**	**+1**	**+1**	**+1**	0	**+1**	0	**+1**
4º	**+1**	**+1**	**+1**	**+1**	**+1**	**+1**	**+1**	**+1**	**+1**	**+1**	-1
5º	**+1**	**+1**	0	**+1**	**+1**	**+1**	**+1**	**+1**	**+1**	**+1**	-1
6º	**+1**	**+1**	**+1**	**+1**	**+1**	**+1**	**+1**	**+1**	**+1**	**+1**	0
7º	**+1**	**+1**	**+1**	**+1**	0	**+1**	**+1**	0	**+1**	**+1**	0
8º	**+1**	**+1**	-1	**+1**	0	0	0	0	**+1**	**+1**	-1
KNeighborsClassifier											
PER	BIO	BIOM	PES	FIS	MAT	TUR	PED	PSI	ADM	CON	ECON
1º	**+1**	**+1**	0	-1	**+1**	**+1**	0	-1	-1	**+1**	0
2º	**+1**	**+1**	0	-1	**+1**	-1	**+1**	0	**+1**	**+1**	-1
3º	**+1**	**+1**	0	0	**+1**	**+1**	0	0	**+1**	0	**+1**
4º	**+1**	**+1**	**+1**	0	**+1**	**+1**	0	-1	0	-1	-1
5º	**+1**	**+1**	**+1**	-1	**+1**	**+1**	**+1**	**+1**	0	-1	-1
6º	**+1**	**+1**	**+1**	0	**+1**	**+1**	**+1**	**+1**	0	0	-1
7º	**+1**	**+1**	0	**+1**	**+1**	**+1**	**+1**	**+1**	0	0	0
8º	**+1**	**+1**	-1	**+1**	**+1**	0	0	0	**+1**	0	-1

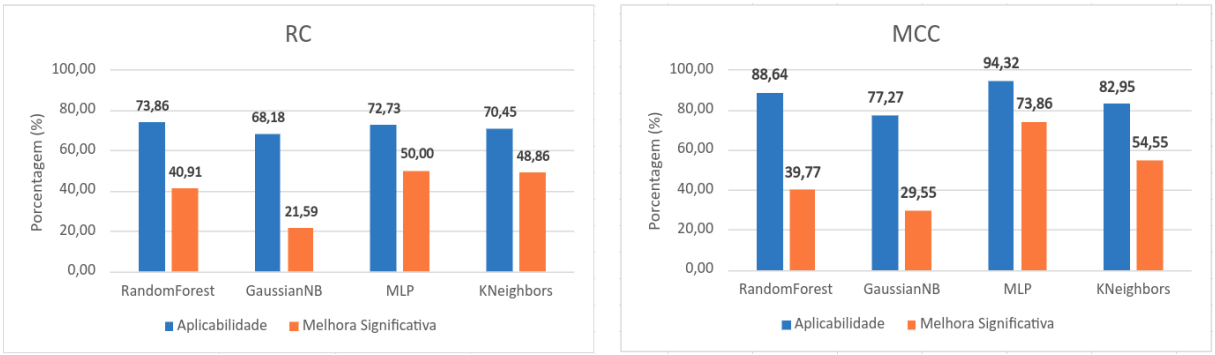
- *Random Forest*: ao considerarmos a perspectiva dos cursos, observamos uma viabilidade de 100% para BIO, BIOM, TUR, PSI e ADM. De forma geral, a proposta pode ser aplicada em cerca de 88,64% dos casos, proporcionando uma melhoria significativa de aproximadamente 39,77%.

- *Gaussian Naive Bayes*: ao considerarmos os cursos, observamos uma viabilidade de 100% para BIO, BIOM, TUR, ADM e ECON. De forma geral, a proposta pode ser aplicada em aproximadamente 77,27% dos casos, resultando em uma melhoria significativa de aproximadamente 29,55%.

- *Multi-layer Perceptron*: considerando os cursos, a viabilidade é de 100% para praticamente todos os cursos (9 dos 11), a exceção de PES e ECON. De forma geral, a proposta pode ser aplicada em aproximadamente 94,32% dos casos, obtendo uma melhoria significativa de cerca de 73,86%.

- *K-Nearest Neighbors*: ao considerarmos os cursos, a viabilidade atinge 100% para BIO, BIOM, MAT e PED. Em termos gerais, a proposta pode ser aplicada em aproximadamente 82,95% dos casos, proporcionando uma melhoria significativa de cerca de 54,55%.

De forma sintética, os percentuais de aplicabilidade e melhora significativa para RC e MCC nos 4 algoritmos são apresentados na Figura 29. Em termos de RC, o *Random Forest* se destaca com a maior aplicabilidade (73,86%) e o *Multi-layer Perceptron* com a maior melhora significativa (50%). Já para MCC, o *Multi-layer Perceptron* tem os maiores percentuais tanto para aplicabilidade (94,32%) quanto para melhora significativa (73,86%).



(a) RC (b) MCC

Figura 29 – Porcentagem de aplicabilidade e melhora significativa da solução proposta.

Nos casos em que as métricas de desempenho não exibem diferenças significativas, consideramos a aplicação da técnica proposta como positiva, uma vez que dados em maior quantidade podem, de fato, ser ativos mais valiosos para análises preditivas (FORTUNY; MARTENS; PROVOST, 2013). Esses modelos proporcionam uma representação mais abrangente da complexidade do problema, capturando padrões que podem passar despercebidos em amostras menores. Além disso, são mais robustos e possuem uma melhor capacidade de generalização. Isso significa que, quando aplicados a novos dados ou a situações diversas, eles tendem a manter seu desempenho, enquanto modelos com dados limitados podem ser mais suscetíveis a overfitting, isto é, ajustar-se excessivamente aos dados de treinamento e não serem tão eficazes na predição de novas instâncias.

Outra observação importante se refere aos casos em que a estratégia adotada se apresentou pior do que os modelos individualizados dos cursos. Avaliando a Tabela 12, em que a métrica utilizada foi mais crítica, é visível que os piores resultados se concentram

nos cursos de PES, MAT e ECON na maioria dos cenários. Esses três cursos se destacam por apresentar uma proporção significativamente maior de estudantes evadidos em relação ao número total de alunos, como ilustrado na Figura 24. Ou seja, quando os dados desses cursos são agrupados com os de outros cursos, a proporção de evadidos no conjunto de dados diminui. Acreditamos que, com isso, acarreta uma dificuldade adicional para o algoritmo de classificação, uma vez que a distinção entre alunos evadidos e não evadidos se torna menos clara em comparação aos modelos individuais dos cursos. Portanto, essa abordagem de unir os dados desses cursos específicos resulta em um desempenho inferior, sublinhando a importância de manter a singularidade de cada um ao desenvolver os modelos de predição.

Portanto, respondendo à questão da pesquisa, a aplicação da técnica demonstra a viabilidade de desenvolver modelos de predição confiáveis com os dados disponíveis no contexto do estudo, comprovando melhora no desempenho em termos de métricas em diversos cenários. Além disso, contribui para a robustez dos modelos de aprendizado de máquina, resultando em uma maior capacidade de lidar com a complexidade dos problemas e de realizar predições mais precisas.

Em resumo, a estratégia de agrupamento oferece, na maioria dos casos, a oportunidade de superar limitações associadas à escassez de dados e melhora a capacidade de generalização e predição dos modelos de aprendizado de máquina, resultando em benefícios significativos para os cursos considerados. Nosso método proposto mostrou eficácia significativa em prever a evasão em uma ampla gama de cursos, com melhorias particularmente notáveis em cursos como BIO e FIS. Essencialmente, a importância do tamanho das amostras na aprendizagem de máquina fica clara ao observarmos esse progresso. O aumento na quantidade de dados desempenha um papel fundamental na capacidade do modelo de prever com precisão.

Parte III

Parte Final

5 Considerações Finais e Trabalhos Futuros

5.1 Considerações Finais

Este trabalho apresenta um processo abrangente que aborda desafios comuns em instituições de ensino superior jovens, que geralmente enfrentam limitações na disponibilidade de dados para estudos de Mineração de Dados Educacionais. O objetivo principal desse trabalho foi lidar com essa limitação de dados de forma eficaz.

Para atingir esse objetivo, o processo incluiu duas etapas-chave. A primeira consistiu na clusterização dos cursos ofertados pela instituição de ensino. Essa clusterização visou identificar cursos com alta similaridade, permitindo a formação de grupos de cursos que compartilham características semelhantes.

A segunda etapa envolveu a aplicação de técnicas de classificação para prever a evasão de alunos, com a separação dos dados por períodos e cursos. Também foi possível a criação de conjuntos de dados específicos a partir dos grupos de cursos identificados na etapa anterior, de clusterização. Esses conjuntos de dados, mais completos e representativos, serviram como base para a construção de modelos de predição de evasão de alunos. A estratégia de agrupamento, Etapa 1, viabilizou a criação de modelos mais robustos e representativos na etapa de classificação, Etapa 2, uma vez que agregou dados de diversos cursos em um único grupo, sem a necessidade de gerar dados sintéticos. Isso, por sua vez, ampliou a disponibilidade de dados para análises posteriores.

Durante a fase não supervisionada da análise, os resultados decorrentes da aplicação de diversas técnicas, como os algoritmos *K-Means* e o *DBSCAN*, demonstraram consistência entre si. Essa consistência reflete uma clareza na estrutura dos dados, o que simplificou o processo de criação dos modelos supervisionados, permitindo uma transição mais fluida e direta para a fase seguinte da análise.

Com os modelos criados, foi realizada a comparação estatística do desempenho dos modelos de predição entre os cursos individualmente e os respectivos grupos num mesmo período, através do uso do Teste de Hipótese Z, possibilitando uma análise mais objetiva dos resultados.

Os resultados obtidos revelam a viabilidade da aplicação da solução proposta em uma parcela significativa dos casos, com índices de sucesso que variam entre os paradigmas do aprendizado supervisionado avaliados. No melhor cenário, com o *Random Forest*, a taxa de aplicabilidade atingiu aproximadamente 73,86%, enquanto no pior caso, com o *GaussianNB*, alcançou 68,18%, quando consideramos o RC. Já em termos de MCC, a solução obteve 94,32% de aplicabilidade no melhor caso, com o MLP, e 77,27% no pior

caso, com o (*GaussianNB*).

Essa consistência na aplicação da solução reflete o potencial e a eficácia do método desenvolvido. Mesmo nos algoritmos em que as métricas de desempenho foram desfavoráveis, observou-se uma melhora significativa com a inclusão do agrupamento. Por exemplo, o *MLP* apresentou uma taxa de viabilidade da solução de 94,32% com uma melhora significativa de 73,86% dos casos na métrica MCC. Esse destaque evidencia o impacto positivo da combinação de clusterização e classificação na construção de modelos de predição de evasão mais robustos e eficazes, contribuindo para uma melhor compreensão e gestão dos desafios relacionados à retenção de alunos em cursos de ensino superior.

Avaliando especificamente os resultados das métricas dos três grupos, o G1 obteve os melhores resultados, sendo o que apresenta a maior proporção de estudantes evadidos em relação ao número total de alunos entre os três grupos (Figura 24), seguido pelo G3, com uma proporção intermediária, e, por último, o G2, que registrou os menores desempenhos e a menor proporção de evasão.

Em relação aos cursos, aqueles com uma proporção significativamente maior de evasão, como PES, MAT e ECON, apresentaram os piores resultados no que se refere ao agrupamento. No entanto, quando tratados de forma individual, esses cursos demonstraram um desempenho superior. Isso pode ser atribuído, possivelmente, à dificuldade adicional enfrentada pelo algoritmo de classificação. A distinção entre alunos evadidos e não evadidos torna-se menos clara em comparação com os modelos individuais dos cursos, devido à diminuição da proporção de evasão no conjunto de dados.

O estudo dos atributos revelou que o mais significativo para os 3 grupos foi o *qtd_aprovacao*, destacando-se por ocupar a primeira posição no ranqueamento de ganho de informação em todos os períodos de G1 e G3. Em G2, este atributo alterna a 1ª posição com *ap_ob_semestre*. Adicionalmente, os atributos *rep_ob_semestre*, *qtd_reprovacao_ft_nt* e *qtd_reprovacao_nt* variam entre a segunda e a quinta posição nos três gráficos (Figura 22). Esses resultados demonstraram a importância da abordagem segmentada por períodos em que os atributos acadêmicos demonstraram maior relevância do que atributos com características pessoais.

Esses resultados têm implicações significativas, pois para o problema abordado, a capacidade aprimorada de identificar casos de evasão potencial e a redução dos erros associados têm um impacto positivo na gestão de políticas públicas preventivas em instituições de ensino superior. Ao possibilitar uma intervenção mais precisa e eficaz, a solução proposta pode contribuir para a retenção de alunos e o sucesso acadêmico, beneficiando tanto as instituições de ensino como os próprios estudantes.

5.1.1 Desafios

Logo abaixo, foram elencados os obstáculos encontrados durante a pesquisa, oferecendo uma visão crítica e reflexiva sobre as limitações e dificuldades enfrentadas ao longo do estudo.

- Baixa quantidade de dados por curso e desbalanceamento de classes: Enfrentamos o desafio da escassez de dados, especialmente na classe positiva (evadido) por conta da metodologia adotada de separação por períodos, resultando em conjuntos desbalanceados, o que impacta a robustez dos modelos.

- Grande quantidade de modelos gerados: A ampla variedade de cursos e períodos demandou a geração de um grande número de modelos. Este desafio destaca a complexidade da aplicação em um ambiente educacional diversificado e ressalta a importância de estratégias eficientes de gerenciamento de modelos.

- Segmentação dos dados por tempo: Foi um aspecto de grande relevância na etapa supervisionada, mas não adotada na não-supervisionada.

- Inclusão de dados do período da pandemia de COVID-19: A utilização de dados coletados durante a pandemia de COVID-19 trouxe um elemento adicional aos desafios, sem que fosse realizado um estudo aprofundado sobre os impactos dessa situação nos dados.

5.1.2 Contribuições

A significativa contribuição deste estudo está na criação e detalhamento de uma metodologia voltada para o aumento dos dados de forma não sintética no contexto do curso, podendo ser aplicada por diversas instituições que enfrentam a limitação de dados. Essa abordagem possibilita a predição contínua do risco de evasão ao longo do curso, combinando técnicas não supervisionadas e supervisionadas, resultando na criação de modelos mais robustos.

Além disso, o trabalho resultou na publicação de um artigo em periódico científico, contribuindo para a disseminação dos resultados e para a comunidade acadêmica interessada em temas relacionados à detecção de risco de evasão:

- COSTA, Cledjan Torres da; JUNIOR, Maurilio Lacerda Leonel; SANTANA, Andre Macedo. **União de Dados por Clusterização para Construção de Modelos de Predição de Evasão**. Revista Novas Tecnologias na Educação, v. 21, n. 2, p. 393–402, dez. 2023. Disponível em: <https://seer.ufrgs.br/index.php/renote/article/view/137761>.

5.1.3 Limitações

Este estudo deve ser compreendido considerando as seguintes limitações:

- O pré-processamento dos dados, na etapa supervisionada, foi realizado de forma global, considerando o conjunto completo dos dados, sem uma abordagem individualizada por curso. Essa abordagem pode limitar a sensibilidade do modelo a características específicas de determinados cursos.

- A grande quantidade de modelos dificulta análises individualizadas, assim como parametrizações adequadas dos algoritmos para cada um. Alguns modelos dos grupos tiveram resultados insatisfatórios, requerendo uma análise mais detalhada dos dados.

- Os dados relacionados ao período da pandemia de COVID-19 podem exercer uma influência significativa nos resultados analisados. Embora uma investigação detalhada dos seus efeitos não tenha sido realizada, por estar além do escopo deste estudo, foi criado um atributo para identificação dos dados relacionados ao período (*cursounapandemia*). Porém o mesmo se mostrou pouco significante para os modelos dos grupos G1 e G2 (Tabela 10), dentre outros. É importante reconhecer o potencial impacto negativo que esses dados podem apresentar.

- A etapa não supervisionada considerou os dados como um conjunto único, sem segmentação por períodos, como foi realizada na etapa supervisionada.

5.2 Trabalhos Futuros

Bons resultados foram observados ao longo deste trabalho, no entanto, há espaço para aprimoramentos das métricas obtidas. É possível testar outros algoritmos, mais aprimorados, baseados em árvores e baseados em probabilidade, uma vez que o *Random Forest* e o *GaussianNB* obtiveram os melhores desempenhos. Assim como ser realizado também um estudo mais detalhado dos modelos, principalmente aqueles com resultados insatisfatórios, e refiná-los com ajustes específicos nos parâmetros e balanceamento das classes, podendo assim contribuir significativamente para melhorar sua capacidade de generalização e desempenho preditivo.

Também pode-se realizar análises comparativas, excluindo os dados da pandemia de COVID-19, a fim de compreender melhor os efeitos desse período nos resultados obtidos. Por exemplo, seria interessante examinar como a pandemia afetou a evasão de estudantes com alto desempenho acadêmico, entre outros aspectos relevantes. Essa abordagem permitirá uma compreensão mais completa do cenário e das possíveis influências da pandemia nos dados analisados, possibilitando ajustá-los e melhorar os resultados.

Além disso, é possível aplicar, na etapa não supervisionada, uma segmentação temporal semelhante à realizada na etapa supervisionada, resultando em oito grupos distintos. Esses resultados segmentados podem ser posteriormente empregados no processo de classificação.

Referências

ANDIFES, A.; ABRUEM, A. et al. Diplomação, retenção e evasão nos cursos de graduação em instituições de ensino superior públicas: resumo do relatório apresentado a andifes, abruem e sesu/mec pela comissão especial. *Avaliação: Revista da Avaliação da Educação Superior*, v. 1, n. 2, 1996. Citado na página 2.

ARAUJO, F. d. *Descoberta de conhecimento em base de dados para o aprendizado da regulação médica/odontológica em operadora de plano de saúde*. Dissertação (Mestrado) — Programa de Pós-Graduação em Ciência da Computação, Universidade Federal do Piauí, 2014. Citado 4 vezes nas páginas 15, 20, 42 e 44.

ARAUJO, L. C. *Configuração*: uma perspectiva de Arquitetura da Informação da Escola de Brasília. Dissertação (Mestrado) — Universidade de Brasília, Brasília, mar. 2012. Citado na página 50.

BAKER, R.; ISOTANI, S.; CARVALHO, A. Mineraçao de dados educacionais: Oportunidades para o brasil. *Revista Brasileira de informática na educação*, v. 19, n. 02, p. 03, 2011. Citado 4 vezes nas páginas 2, 27, 28 e 36.

BARNES, T. et al. Educational data mining 2009: 2nd international conference on educational data mining, proceedings. In: . Cordoba, Spain: [s.n.], 2009. Citado na página 26.

BARRETO, D. L. et al. Evasão no ensino superior: investigação das causas via mineração de dados. *Revista Eletrônica Debates em Educação Científica e Tecnológica*, v. 3, n. 2, p. 3–21, 2019. Citado 3 vezes nas páginas 13, 42 e 43.

BATISTA, G. E. d. A. P. et al. *Pré-processamento de dados em aprendizado de máquina supervisionado*. Tese (Doutorado), 2003. Citado 7 vezes nas páginas 13, 15, 24, 25, 26, 37 e 41.

BRITO, B. C. P. de; MELLO, R. F. L. de; ALVES, G. Identificação de atributos relevantes na evasão no ensino superior público brasileiro. In: SBC. *Anais do XXXI Simpósio Brasileiro de Informática na Educação*. [S.l.], 2020. p. 1032–1041. Citado na página 6.

CAMARGO, S. S. *Mineração de regras de associação no problema da cesta de compras aplicada ao comércio varejista de confecção*. Dissertação (Mestrado) — Instituto de Informática - Universidade Federal do Rio Grande do Sul, 2002. Citado na página 28.

CAMILO, C. O.; SILVA, J. C. d. Mineração de dados: Conceitos, tarefas, métodos e ferramentas. *Universidade Federal de Goiás (UFC)*, v. 1, n. 1, p. 1–29, 2009. Citado na página 27.

CASTRO, L. N. de; FERRARI, D. G. *Introdução à Mineração de Dados: Conceitos básicos, algoritmos e aplicações*. São Paulo: Saraiva, 2016. ISBN 978-85-472-0099-2. Citado 2 vezes nas páginas 20 e 33.

CHAN, P. K.; STOLFO, S. J. The effects of training class distributions on performance using cost models. *Draft*, 1998. Citado na página 15.

CHAU, A. L.; GARCÍA, F.; CERVANTES, J. Classification on imbalanced data sets, taking advantage of errors to improve performance. In: *International Conference on Intelligent Computing*. [s.n.], 2015. Disponível em: <https://api.semanticscholar.org/CorpusID:1443072>. Citado na página 3.

CHICCO, D.; JURMAN, G. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC Genomics*, v. 21, 2020. Disponível em: <https://api.semanticscholar.org/CorpusID:209528322>. Citado na página 72.

CHICCO, D.; WARRENS, M. J.; JURMAN, G. The matthews correlation coefficient (mcc) is more informative than cohen's kappa and brier score in binary classification assessment. *IEEE Access*, v. 9, p. 78368–78381, 2021. Disponível em: <https://api.semanticscholar.org/CorpusID:235308708>. Citado na página 39.

COLPO, M. P. et al. Mineraçao de dados educacionais na previsao de evasao: uma rsl sob a perspectiva do congresso brasileiro de informática na educaçao. In: SBC. *Anais do XXXI Simpósio Brasileiro de Informática na Educaçao*. [S.l.], 2020. p. 1102–1111. Citado na página 5.

COSTA, C. T. d.; JUNIOR, M. L. L.; SANTANA, A. M. União de dados por clusterização para construção de modelos de predição de evasão. *Revista Novas Tecnologias na Educação*, v. 21, n. 2, p. 393–402, dez. 2023. Disponível em: <https://seer.ufrgs.br/index.php/renote/article/view/137761>. Citado na página 9.

COSTA, E. et al. Mineração de dados educacionais: conceitos, técnicas, ferramentas e aplicações. *Jornada de Atualização em Informática na Educação*, v. 1, n. 1, p. 1–29, 2012. Citado 3 vezes nas páginas 13, 27 e 28.

FAN, C. et al. A review on data preprocessing techniques toward efficient and reliable knowledge discovery from building operational data. *Frontiers in Energy Research*, v. 9, 2021. ISSN 2296-598X. Disponível em: <https://www.frontiersin.org/articles/10.3389/fenrg.2021.652801>. Citado 2 vezes nas páginas 18 e 20.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. *AI magazine*, v. 17, n. 3, p. 37–37, 1996. Citado 4 vezes nas páginas 13, 2, 14 e 50.

FORTUNY, E. Junqué de; MARTENS, D.; PROVOST, F. Predictive modeling with big data: Is bigger really better? *Big Data*, v. 1, n. 4, p. 215–226, 2013. PMID: 27447254. Disponível em: <https://doi.org/10.1089/big.2013.0037>. Citado na página 86.

FUJIWARA, K. et al. Over- and under-sampling approach for extremely imbalanced and small minority data problem in health record analysis. *Frontiers in Public Health*, v. 8, 2020. ISSN 2296-2565. Disponível em: <https://www.frontiersin.org/articles/10.3389/fpubh.2020.00178>. Citado 4 vezes nas páginas 20, 21, 22 e 24.

GOMES, A. K. *Análise do conhecimento extraído de classificadores simbólicos utilizando medidas de avaliação e de interessabilidade*. Tese (Doutorado) — Universidade de São Paulo, 2002. Citado na página 42.

HALKIDI, M.; BATISTAKIS, Y.; VAZIRGIANNIS, M. On clustering validation techniques. *Journal of intelligent information systems*, Springer, v. 17, p. 107–145, 2001. Citado na página 35.

HAN, J.; PEI, J.; KAMBER, M. *Data Mining: Concepts and Techniques*. 3nd. ed. Amsterdam, The Netherlands: Elsevier, 2012. 744 p. Citado 10 vezes nas páginas 15, 5, 14, 16, 17, 24, 29, 32, 34 e 35.

HU, B. et al. A preliminary study on data augmentation of deep learning for image classification. *Proceedings of the 11th Asia-Pacific Symposium on Internetware*, 2019. Disponível em: <https://api.semanticscholar.org/CorpusID:195750783>. Citado 2 vezes nas páginas 3 e 4.

INEP. *Indicadores de fluxo da educação superior. 2021*. 2022. Acesso em 15/06/2023. Disponível em: <https://www.gov.br/inep/pt-br/acesso-a-informacao/dados-abertos/indicadores-educacionais/indicadores-de-fluxo-da-educacao-superior>. Citado na página 1.

KASSAMBARA, A. *Practical Guide to Cluster Analysis in R: Unsupervised Machine Learning*. [S.l.]: Sthda, 2017. Citado na página 32.

KODINARIYA, T. M.; MAKWANA, P. R. et al. Review on determining number of cluster in k-means clustering. *International Journal*, v. 1, n. 6, p. 90–95, 2013. Citado na página 59.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, v. 60, p. 84 – 90, 2012. Disponível em: <https://api.semanticscholar.org/CorpusID:195908774>. Citado na página 3.

LEI, C. et al. A preliminary study on data augmentation of deep learning for image classification. In: *Proceedings of the 11th Asia-Pacific Symposium on Internetware*. [S.l.: s.n.], 2019. p. 1–6. Citado na página 3.

LI, D.-C. et al. Detecting representative data and generating synthetic samples to improve learning accuracy with imbalanced data sets. *PLoS ONE*, v. 12, 2017. Citado 2 vezes nas páginas 2 e 4.

LIU, W. et al. A robust decision tree algorithm for imbalanced data sets. In: _____. *Proceedings of the 2010 SIAM International Conference on Data Mining (SDM)*. [s.n.], 2010. p. 766–777. Disponível em: <https://epubs.siam.org/doi/abs/10.1137/1.9781611972801.67>. Citado na página 22.

LUDERMIR, T. B. Inteligência artificial e aprendizado de máquina: estado atual e tendências. *Estudos Avançados*, SciELO Brasil, v. 35, p. 85–94, 2021. Citado na página 40.

MANHÃES, L. M. B.; CRUZ, S. Predição do desempenho acadêmico de alunos da graduação utilizando mineração de dados. *XIX Simpósio de Pesquisa Operacional e Logística da Marinha. Rio de Janeiro, RJ, Brasil*, v. 6, 2019. Citado 2 vezes nas páginas 2 e 7.

MANHÃES, L. M. B.; CRUZ, S.; ZIMBRÃO, G. Predição do desempenho acadêmico de graduandos utilizando mineração de dados educacionais. *Doutorado em Engenharia de Sistemas e Computação Instituição de Ensino: Universidade Federal do Rio de Janeiro, Rio de Janeiro. Biblioteca Depositária: BIBLIOTECA DO CT*, 2015. Citado na página 5.

MARTINHO, V. R. D. C.; NUNES, C.; MINUSSI, C. R. An intelligent system for prediction of school dropout risk group in higher education classroom based on artificial neural networks. In: IEEE. *2013 IEEE 25th International Conference on Tools with Artificial Intelligence*. [S.l.], 2013. p. 159–166. Citado na página 2.

MARTINS, R. R.; BERTUCI, M. H.; PENIANI, L. P. Mineração de dados no combate à evasão escolar em instituições de ensino superior brasileiras. *Revista Interface Tecnológica*, v. 17, n. 2, p. 103–115, 2020. Citado na página 4.

MEHTA, A. A.; BUCH, N. J. Depth and breadth of educational data mining: Researchers' point of view. In: IEEE. *2016 10th International Conference on Intelligent Systems and Control (ISCO)*. [S.l.], 2016. Citado na página 13.

MITCHELL, T. M. *Machine Learning*. S.I.: McGraw-Hill, 1997. Citado na página 40.

MONARD, M. C.; BARANAUSKAS, J. A. *Conceitos sobre Aprendizado de Máquina*. 1st. ed. [S.l.]: Rezende, 2003. v. 1. Citado na página 40.

MURPHY, K. P. *Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina*. [S.l.]: Campus, 2012. ISBN 978-8535237019. Citado 2 vezes nas páginas 28 e 45.

ROMERO, C.; VENTURA, S. Educational data mining: a review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (applications and reviews)*, Ieee, v. 40, n. 6, p. 601–618, 2010. Citado na página 2.

SANTOS, C. H. D.; MARTINS, S. de L.; PLASTINO, A. É possível prever evasão com base apenas no desempenho acadêmico? In: SBC. *Anais do XXXII Simpósio Brasileiro de Informática na Educação*. [S.l.], 2021. p. 792–802. Citado 3 vezes nas páginas 5, 7 e 31.

SANTOS, V. H. B. dos; SARAIVA, D. V.; OLIVEIRA, C. T. de. Uma análise de trabalhos de mineração de dados educacionais no contexto da evasão escolar. In: SBC. *Anais do XXXII Simpósio Brasileiro de Informática na Educação*. [S.l.], 2021. p. 1196–1210. Citado 3 vezes nas páginas 5, 42 e 43.

SARAIVA, D. V. et al. Análise de agrupamentos para caracterização de indicadores de evasão. In: SBC. *Anais do XXIX Workshop sobre Educação em Computação*. [S.l.], 2021. p. 238–247. Citado 2 vezes nas páginas 6 e 7.

SCIKITLEARN. *scikit-learn - Normalizer*. sem data de publicação. Acessado em 06-08-2023. Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.Normalizer.html>. Citado na página 24.

SILVA, L. A.; PERES, S. M.; BOSCARIOLI, C. *Introdução à Mineração de Dados - Com Aplicações em R*. 1. ed. [S.l.]: GEN LTC, 2016. 296 p. ISBN 9788535284461. Citado 6 vezes nas páginas 13, 16, 18, 26, 35 e 36.

SILVA, L. B. E.; MARIANO, A. S. A definição de evasão e suas implicações (limites) para as políticas de educação superior. *Educação em Revista*, SciELO Brasil, v. 37, 2021. Citado na página 1.

TAMADA, M. M. et al. Predição de evasão de cursos técnicos em ead através de técnicas de aprendizado de máquina em duas etapas. Universidade Federal do Amazonas, 2022. Citado 5 vezes nas páginas 6, 7, 13, 28 e 50.

TAN, P.-N. et al. *Introduction to Data Mining*. 2nd. ed. [S.l.]: Pearson, 2019. Citado 9 vezes nas páginas 13, 14, 16, 28, 29, 30, 31, 33 e 35.

UFRN. *Página de Suporte do UFRN Info*. 2020. Disponível em: <https://docs.info.ufrn.br/doku.php?id=suporte>. Citado na página 50.

VIANA, F. S.; SANTANA, A. M.; RABÊLO, R. d. A. L. Avaliaçao de classificadores para prediçao de evasao no ensino superior utilizando janela semestral. In: SBC. *Anais do XXXIII Simpósio Brasileiro de Informática na Educação*. [S.l.], 2022. p. 908–919. Citado 6 vezes nas páginas 4, 5, 7, 8, 50 e 51.

Anexos

ANEXO A – Resultados da Etapa 2

Tabela 14 – Resultados das métricas RC e MCC para todos os modelos e algoritmos aplicados.

Algoritmo	**CursoGrupo**	**Período**	**RC**	**MCC**
RandomForestClassifier	PSI	1	66.41	61.37
RandomForestClassifier	PSI	2	53.67	49.09
RandomForestClassifier	PSI	3	53.64	52.88
RandomForestClassifier	PSI	4	53.75	51.64
RandomForestClassifier	PSI	5	52.86	57.63
RandomForestClassifier	PSI	6	47.33	56.18
RandomForestClassifier	PSI	7	45.50	54.99
RandomForestClassifier	PSI	8	39.17	49.43
RandomForestClassifier	PED	1	67.35	65.69
RandomForestClassifier	PED	2	53.00	58.45
RandomForestClassifier	PED	3	51.50	62.57
RandomForestClassifier	PED	4	60.00	68.41
RandomForestClassifier	PED	5	55.00	67.02
RandomForestClassifier	PED	6	55.00	57.69
RandomForestClassifier	PED	7	10.00	9.58
RandomForestClassifier	PED	8	10.00	10.00
RandomForestClassifier	ADM	1	65.36	51.58
RandomForestClassifier	ADM	2	49.82	37.31
RandomForestClassifier	ADM	3	50.00	42.79
RandomForestClassifier	ADM	4	39.05	41.51
RandomForestClassifier	ADM	5	45.00	48.45
RandomForestClassifier	ADM	6	41.50	45.84
RandomForestClassifier	ADM	7	48.33	52.89
RandomForestClassifier	ADM	8	41.67	44.38
RandomForestClassifier	CON	1	67.23	57.16
RandomForestClassifier	CON	2	52.89	49.90
RandomForestClassifier	CON	3	64.90	64.16
RandomForestClassifier	CON	4	61.36	61.83
RandomForestClassifier	CON	5	54.86	63.13
RandomForestClassifier	CON	6	60.00	64.05
RandomForestClassifier	CON	7	64.33	65.07

Algoritmo	CursoGrupo	Período	RC	MCC
RandomForestClassifier	CON	8	56.00	61.71
RandomForestClassifier	ECON	1	96.02	72.56
RandomForestClassifier	ECON	2	87.03	63.58
RandomForestClassifier	ECON	3	77.33	54.38
RandomForestClassifier	ECON	4	75.71	58.37
RandomForestClassifier	ECON	5	69.67	63.19
RandomForestClassifier	ECON	6	73.00	68.42
RandomForestClassifier	ECON	7	81.50	73.69
RandomForestClassifier	ECON	8	77.50	72.74
RandomForestClassifier	BIOM	1	68.64	56.76
RandomForestClassifier	BIOM	2	66.48	58.30
RandomForestClassifier	BIOM	3	57.56	55.37
RandomForestClassifier	BIOM	4	56.79	58.88
RandomForestClassifier	BIOM	5	52.67	51.64
RandomForestClassifier	BIOM	6	49.17	54.60
RandomForestClassifier	BIOM	7	38.33	47.17
RandomForestClassifier	BIOM	8	53.33	55.75
RandomForestClassifier	BIO	1	85.80	59.84
RandomForestClassifier	BIO	2	82.00	58.81
RandomForestClassifier	BIO	3	73.66	66.00
RandomForestClassifier	BIO	4	68.63	64.53
RandomForestClassifier	BIO	5	72.42	64.98
RandomForestClassifier	BIO	6	65.18	60.98
RandomForestClassifier	BIO	7	58.75	63.52
RandomForestClassifier	BIO	8	63.81	62.51
RandomForestClassifier	PES	1	95.48	70.85
RandomForestClassifier	PES	2	92.25	68.90
RandomForestClassifier	PES	3	88.90	65.39
RandomForestClassifier	PES	4	88.16	70.12
RandomForestClassifier	PES	5	85.66	75.00
RandomForestClassifier	PES	6	86.92	72.70
RandomForestClassifier	PES	7	83.33	79.45
RandomForestClassifier	PES	8	85.00	79.31
RandomForestClassifier	FIS	1	73.70	64.49
RandomForestClassifier	FIS	2	71.71	72.28
RandomForestClassifier	FIS	3	69.62	69.19
RandomForestClassifier	FIS	4	73.45	72.07

Algoritmo	CursoGrupo	Período	RC	MCC
RandomForestClassifier	FIS	5	65.71	71.58
RandomForestClassifier	FIS	6	58.33	64.19
RandomForestClassifier	FIS	7	47.00	60.34
RandomForestClassifier	FIS	8	48.33	52.19
RandomForestClassifier	MAT	1	92.31	54.54
RandomForestClassifier	MAT	2	85.89	61.04
RandomForestClassifier	MAT	3	88.03	68.35
RandomForestClassifier	MAT	4	84.88	69.11
RandomForestClassifier	MAT	5	84.00	70.60
RandomForestClassifier	MAT	6	79.64	73.80
RandomForestClassifier	MAT	7	88.39	83.93
RandomForestClassifier	MAT	8	81.00	77.24
RandomForestClassifier	TUR	1	83.90	46.08
RandomForestClassifier	TUR	2	74.80	54.24
RandomForestClassifier	TUR	3	72.78	60.43
RandomForestClassifier	TUR	4	77.86	67.93
RandomForestClassifier	TUR	5	75.71	68.05
RandomForestClassifier	TUR	6	73.22	69.56
RandomForestClassifier	TUR	7	69.11	65.43
RandomForestClassifier	TUR	8	63.57	60.67
RandomForestClassifier	GRUPO1	1	87.01	66.22
RandomForestClassifier	GRUPO1	2	79.93	65.91
RandomForestClassifier	GRUPO1	3	77.36	69.21
RandomForestClassifier	GRUPO1	4	78.15	71.57
RandomForestClassifier	GRUPO1	5	73.71	69.42
RandomForestClassifier	GRUPO1	6	74.84	74.45
RandomForestClassifier	GRUPO1	7	69.75	71.11
RandomForestClassifier	GRUPO1	8	67.68	71.78
RandomForestClassifier	GRUPO2	1	61.76	60.23
RandomForestClassifier	GRUPO2	2	49.13	54.29
RandomForestClassifier	GRUPO2	3	49.62	54.31
RandomForestClassifier	GRUPO2	4	54.17	62.68
RandomForestClassifier	GRUPO2	5	54.89	63.38
RandomForestClassifier	GRUPO2	6	45.48	53.27
RandomForestClassifier	GRUPO2	7	52.33	60.45
RandomForestClassifier	GRUPO2	8	47.00	60.23
RandomForestClassifier	GRUPO3	1	74.74	58.48

Algoritmo	CursoGrupo	Período	RC	MCC
RandomForestClassifier	GRUPO3	2	64.87	53.18
RandomForestClassifier	GRUPO3	3	61.78	57.41
RandomForestClassifier	GRUPO3	4	59.48	58.17
RandomForestClassifier	GRUPO3	5	62.76	62.07
RandomForestClassifier	GRUPO3	6	60.48	65.38
RandomForestClassifier	GRUPO3	7	64.17	66.40
RandomForestClassifier	GRUPO3	8	56.73	61.02
GaussianNB	PSI	1	53.23	50.84
GaussianNB	PSI	2	52.29	49.14
GaussianNB	PSI	3	54.55	50.75
GaussianNB	PSI	4	65.42	57.13
GaussianNB	PSI	5	62.86	54.54
GaussianNB	PSI	6	63.00	51.22
GaussianNB	PSI	7	73.50	58.25
GaussianNB	PSI	8	69.17	50.85
GaussianNB	PED	1	60.53	70.18
GaussianNB	PED	2	52.33	61.46
GaussianNB	PED	3	71.00	72.99
GaussianNB	PED	4	79.17	78.95
GaussianNB	PED	5	83.33	78.63
GaussianNB	PED	6	90.00	84.28
GaussianNB	PED	7	90.00	79.37
GaussianNB	PED	8	70.00	63.83
GaussianNB	ADM	1	48.20	48.41
GaussianNB	ADM	2	39.91	44.84
GaussianNB	ADM	3	55.97	45.87
GaussianNB	ADM	4	53.57	47.48
GaussianNB	ADM	5	60.00	51.71
GaussianNB	ADM	6	63.00	54.44
GaussianNB	ADM	7	70.83	54.03
GaussianNB	ADM	8	70.00	54.14
GaussianNB	CON	1	53.39	55.93
GaussianNB	CON	2	59.42	62.07
GaussianNB	CON	3	67.52	67.93
GaussianNB	CON	4	70.09	69.01
GaussianNB	CON	5	74.17	68.72
GaussianNB	CON	6	81.43	68.09

Algoritmo	CursoGrupo	Período	RC	MCC
GaussianNB	CON	7	90.33	67.20
GaussianNB	CON	8	90.50	66.84
GaussianNB	ECON	1	54.95	40.82
GaussianNB	ECON	2	60.60	45.82
GaussianNB	ECON	3	65.11	47.79
GaussianNB	ECON	4	71.96	54.88
GaussianNB	ECON	5	71.33	58.30
GaussianNB	ECON	6	71.33	57.76
GaussianNB	ECON	7	76.50	60.45
GaussianNB	ECON	8	89.17	73.09
GaussianNB	BIOM	1	54.25	53.56
GaussianNB	BIOM	2	67.86	57.90
GaussianNB	BIOM	3	68.33	61.40
GaussianNB	BIOM	4	78.04	63.40
GaussianNB	BIOM	5	78.33	52.73
GaussianNB	BIOM	6	86.67	57.44
GaussianNB	BIOM	7	83.33	56.40
GaussianNB	BIOM	8	85.00	56.55
GaussianNB	BIO	1	55.85	49.73
GaussianNB	BIO	2	61.20	59.14
GaussianNB	BIO	3	68.61	65.09
GaussianNB	BIO	4	70.00	64.99
GaussianNB	BIO	5	73.85	67.39
GaussianNB	BIO	6	71.91	63.96
GaussianNB	BIO	7	75.71	63.53
GaussianNB	BIO	8	84.29	64.92
GaussianNB	PES	1	60.70	43.88
GaussianNB	PES	2	75.58	58.80
GaussianNB	PES	3	75.90	63.90
GaussianNB	PES	4	77.45	65.96
GaussianNB	PES	5	78.79	72.55
GaussianNB	PES	6	80.82	74.07
GaussianNB	PES	7	75.78	71.91
GaussianNB	PES	8	83.75	76.16
GaussianNB	FIS	1	58.55	60.06
GaussianNB	FIS	2	57.29	56.45
GaussianNB	FIS	3	59.62	58.31

Algoritmo	CursoGrupo	Período	RC	MCC
GaussianNB	FIS	4	64.27	59.41
GaussianNB	FIS	5	68.21	56.08
GaussianNB	FIS	6	66.33	50.43
GaussianNB	FIS	7	70.50	53.15
GaussianNB	FIS	8	66.67	43.34
GaussianNB	MAT	1	68.19	50.27
GaussianNB	MAT	2	79.76	60.84
GaussianNB	MAT	3	82.20	66.61
GaussianNB	MAT	4	85.83	72.66
GaussianNB	MAT	5	82.67	66.42
GaussianNB	MAT	6	87.36	72.24
GaussianNB	MAT	7	89.82	76.98
GaussianNB	MAT	8	89.67	76.53
GaussianNB	TUR	1	59.00	50.82
GaussianNB	TUR	2	60.02	53.94
GaussianNB	TUR	3	63.33	57.43
GaussianNB	TUR	4	74.57	64.54
GaussianNB	TUR	5	74.74	62.72
GaussianNB	TUR	6	79.56	70.10
GaussianNB	TUR	7	79.64	66.97
GaussianNB	TUR	8	82.38	66.88
GaussianNB	GRUPO1	1	62.51	54.83
GaussianNB	GRUPO1	2	67.92	62.06
GaussianNB	GRUPO1	3	69.53	65.11
GaussianNB	GRUPO1	4	72.72	65.77
GaussianNB	GRUPO1	5	75.29	67.24
GaussianNB	GRUPO1	6	78.24	67.79
GaussianNB	GRUPO1	7	79.50	68.17
GaussianNB	GRUPO1	8	80.66	66.23
GaussianNB	GRUPO2	1	48.11	54.07
GaussianNB	GRUPO2	2	39.53	43.04
GaussianNB	GRUPO2	3	52.96	55.43
GaussianNB	GRUPO2	4	60.83	60.01
GaussianNB	GRUPO2	5	66.11	62.02
GaussianNB	GRUPO2	6	65.71	59.26
GaussianNB	GRUPO2	7	69.00	58.51
GaussianNB	GRUPO2	8	69.00	52.12

Algoritmo	CursoGrupo	Período	RC	MCC
GaussianNB	GRUPO3	1	49.80	50.62
GaussianNB	GRUPO3	2	50.47	50.65
GaussianNB	GRUPO3	3	61.49	58.60
GaussianNB	GRUPO3	4	64.68	60.76
GaussianNB	GRUPO3	5	68.29	62.62
GaussianNB	GRUPO3	6	73.46	64.25
GaussianNB	GRUPO3	7	79.42	66.02
GaussianNB	GRUPO3	8	85.55	68.82
KNeighborsClassifier	PSI	1	45.28	18.63
KNeighborsClassifier	PSI	2	29.52	14.50
KNeighborsClassifier	PSI	3	22.73	13.87
KNeighborsClassifier	PSI	4	22.50	14.34
KNeighborsClassifier	PSI	5	8.57	-0.89
KNeighborsClassifier	PSI	6	4.00	1.33
KNeighborsClassifier	PSI	7	2.50	-0.17
KNeighborsClassifier	PSI	8	0.00	-1.60
KNeighborsClassifier	PED	1	32.88	11.73
KNeighborsClassifier	PED	2	11.67	6.34
KNeighborsClassifier	PED	3	7.50	10.63
KNeighborsClassifier	PED	4	5.83	7.38
KNeighborsClassifier	PED	5	0.00	0.00
KNeighborsClassifier	PED	6	0.00	0.00
KNeighborsClassifier	PED	7	0.00	0.00
KNeighborsClassifier	PED	8	0.00	0.00
KNeighborsClassifier	ADM	1	57.03	15.20
KNeighborsClassifier	ADM	2	24.55	1.36
KNeighborsClassifier	ADM	3	19.86	2.02
KNeighborsClassifier	ADM	4	6.43	-6.97
KNeighborsClassifier	ADM	5	12.00	4.70
KNeighborsClassifier	ADM	6	0.00	-7.47
KNeighborsClassifier	ADM	7	0.00	-3.80
KNeighborsClassifier	ADM	8	0.00	-1.49
KNeighborsClassifier	CON	1	38.02	3.10
KNeighborsClassifier	CON	2	24.30	-2.55
KNeighborsClassifier	CON	3	19.38	3.73
KNeighborsClassifier	CON	4	17.82	6.40
KNeighborsClassifier	CON	5	15.56	7.23

Algoritmo	CursoGrupo	Período	RC	MCC
KNeighborsClassifier	CON	6	0.00	-5.93
KNeighborsClassifier	CON	7	2.00	-1.93
KNeighborsClassifier	CON	8	2.50	1.71
KNeighborsClassifier	ECON	1	81.14	11.05
KNeighborsClassifier	ECON	2	69.95	11.75
KNeighborsClassifier	ECON	3	51.33	-0.09
KNeighborsClassifier	ECON	4	49.46	5.67
KNeighborsClassifier	ECON	5	42.67	6.25
KNeighborsClassifier	ECON	6	36.33	8.56
KNeighborsClassifier	ECON	7	35.50	10.20
KNeighborsClassifier	ECON	8	30.83	9.36
KNeighborsClassifier	BIOM	1	42.53	-0.30
KNeighborsClassifier	BIOM	2	23.24	-5.57
KNeighborsClassifier	BIOM	3	17.22	1.07
KNeighborsClassifier	BIOM	4	9.46	-1.56
KNeighborsClassifier	BIOM	5	6.00	0.04
KNeighborsClassifier	BIOM	6	5.00	3.06
KNeighborsClassifier	BIOM	7	0.00	-1.71
KNeighborsClassifier	BIOM	8	0.00	-0.48
KNeighborsClassifier	BIO	1	60.11	-5.70
KNeighborsClassifier	BIO	2	46.80	-6.35
KNeighborsClassifier	BIO	3	44.18	0.79
KNeighborsClassifier	BIO	4	34.29	-3.45
KNeighborsClassifier	BIO	5	29.34	-1.94
KNeighborsClassifier	BIO	6	24.73	-0.22
KNeighborsClassifier	BIO	7	19.11	4.17
KNeighborsClassifier	BIO	8	15.71	3.55
KNeighborsClassifier	PES	1	87.65	8.14
KNeighborsClassifier	PES	2	79.71	7.44
KNeighborsClassifier	PES	3	75.53	12.20
KNeighborsClassifier	PES	4	71.97	13.36
KNeighborsClassifier	PES	5	67.54	12.61
KNeighborsClassifier	PES	6	62.47	7.85
KNeighborsClassifier	PES	7	50.44	15.97
KNeighborsClassifier	PES	8	52.64	26.05
KNeighborsClassifier	FIS	1	43.21	13.85
KNeighborsClassifier	FIS	2	33.29	11.41

Algoritmo	CursoGrupo	Período	RC	MCC
KNeighborsClassifier	FIS	3	33.40	18.00
KNeighborsClassifier	FIS	4	20.45	16.19
KNeighborsClassifier	FIS	5	18.93	25.09
KNeighborsClassifier	FIS	6	10.67	18.08
KNeighborsClassifier	FIS	7	0.00	-1.83
KNeighborsClassifier	FIS	8	0.00	0.00
KNeighborsClassifier	MAT	1	86.82	-7.41
KNeighborsClassifier	MAT	2	78.21	-12.40
KNeighborsClassifier	MAT	3	68.35	-13.42
KNeighborsClassifier	MAT	4	62.95	-13.87
KNeighborsClassifier	MAT	5	48.67	-14.63
KNeighborsClassifier	MAT	6	34.36	-15.57
KNeighborsClassifier	MAT	7	31.96	3.97
KNeighborsClassifier	MAT	8	15.00	-5.80
KNeighborsClassifier	TUR	1	75.50	-4.04
KNeighborsClassifier	TUR	2	64.78	15.60
KNeighborsClassifier	TUR	3	50.00	6.63
KNeighborsClassifier	TUR	4	41.57	4.51
KNeighborsClassifier	TUR	5	36.54	7.06
KNeighborsClassifier	TUR	6	33.22	12.03
KNeighborsClassifier	TUR	7	22.86	10.17
KNeighborsClassifier	TUR	8	19.05	14.42
KNeighborsClassifier	GRUPO1	1	68.01	6.63
KNeighborsClassifier	GRUPO1	2	56.29	8.08
KNeighborsClassifier	GRUPO1	3	50.89	14.64
KNeighborsClassifier	GRUPO1	4	48.77	19.75
KNeighborsClassifier	GRUPO1	5	41.57	20.40
KNeighborsClassifier	GRUPO1	6	34.94	22.14
KNeighborsClassifier	GRUPO1	7	26.25	19.22
KNeighborsClassifier	GRUPO1	8	21.76	19.02
KNeighborsClassifier	GRUPO2	1	38.00	11.60
KNeighborsClassifier	GRUPO2	2	29.08	11.96
KNeighborsClassifier	GRUPO2	3	18.54	11.12
KNeighborsClassifier	GRUPO2	4	10.83	5.39
KNeighborsClassifier	GRUPO2	5	6.67	3.35
KNeighborsClassifier	GRUPO2	6	7.62	10.02
KNeighborsClassifier	GRUPO2	7	4.00	3.31

Algoritmo	CursoGrupo	Período	RC	MCC
KNeighborsClassifier	GRUPO2	8	0.00	-0.32
KNeighborsClassifier	GRUPO3	1	53.31	8.12
KNeighborsClassifier	GRUPO3	2	35.93	6.05
KNeighborsClassifier	GRUPO3	3	26.32	5.57
KNeighborsClassifier	GRUPO3	4	13.70	1.69
KNeighborsClassifier	GRUPO3	5	11.98	2.55
KNeighborsClassifier	GRUPO3	6	6.88	-1.09
KNeighborsClassifier	GRUPO3	7	2.37	-2.99
KNeighborsClassifier	GRUPO3	8	3.82	2.16
GradientBoostingClassifier	PSI	1	53.59	53.89
GradientBoostingClassifier	PSI	2	43.24	41.91
GradientBoostingClassifier	PSI	3	40.91	50.25
GradientBoostingClassifier	PSI	4	41.39	52.60
GradientBoostingClassifier	PSI	5	48.57	57.26
GradientBoostingClassifier	PSI	6	38.00	50.62
GradientBoostingClassifier	PSI	7	41.00	51.57
GradientBoostingClassifier	PSI	8	40.83	47.12
GradientBoostingClassifier	PED	1	54.39	60.77
GradientBoostingClassifier	PED	2	39.33	51.32
GradientBoostingClassifier	PED	3	37.00	46.11
GradientBoostingClassifier	PED	4	65.83	74.53
GradientBoostingClassifier	PED	5	55.00	54.07
GradientBoostingClassifier	PED	6	50.00	46.95
GradientBoostingClassifier	PED	7	40.00	33.43
GradientBoostingClassifier	PED	8	40.00	40.00
GradientBoostingClassifier	ADM	1	58.43	48.30
GradientBoostingClassifier	ADM	2	39.82	40.37
GradientBoostingClassifier	ADM	3	45.97	47.34
GradientBoostingClassifier	ADM	4	35.48	37.77
GradientBoostingClassifier	ADM	5	41.00	41.73
GradientBoostingClassifier	ADM	6	34.50	38.18
GradientBoostingClassifier	ADM	7	35.00	46.34
GradientBoostingClassifier	ADM	8	41.67	40.54
GradientBoostingClassifier	CON	1	58.36	53.75
GradientBoostingClassifier	CON	2	52.46	49.08
GradientBoostingClassifier	CON	3	57.00	57.95
GradientBoostingClassifier	CON	4	51.00	57.55

Algoritmo	CursoGrupo	Período	RC	MCC
GradientBoostingClassifier	CON	5	57.08	65.73
GradientBoostingClassifier	CON	6	64.52	71.29
GradientBoostingClassifier	CON	7	58.67	64.15
GradientBoostingClassifier	CON	8	43.00	49.64
GradientBoostingClassifier	ECON	1	98.00	72.79
GradientBoostingClassifier	ECON	2	96.21	68.52
GradientBoostingClassifier	ECON	3	93.89	63.96
GradientBoostingClassifier	ECON	4	74.82	46.42
GradientBoostingClassifier	ECON	5	74.33	69.51
GradientBoostingClassifier	ECON	6	67.33	66.29
GradientBoostingClassifier	ECON	7	72.00	72.29
GradientBoostingClassifier	ECON	8	62.50	61.18
GradientBoostingClassifier	BIOM	1	63.06	51.36
GradientBoostingClassifier	BIOM	2	53.19	46.50
GradientBoostingClassifier	BIOM	3	46.67	50.71
GradientBoostingClassifier	BIOM	4	52.68	54.97
GradientBoostingClassifier	BIOM	5	41.33	41.70
GradientBoostingClassifier	BIOM	6	35.00	38.82
GradientBoostingClassifier	BIOM	7	47.50	53.29
GradientBoostingClassifier	BIOM	8	48.33	53.40
GradientBoostingClassifier	BIO	1	96.55	68.26
GradientBoostingClassifier	BIO	2	88.40	63.11
GradientBoostingClassifier	BIO	3	75.79	62.82
GradientBoostingClassifier	BIO	4	71.83	65.04
GradientBoostingClassifier	BIO	5	64.23	63.66
GradientBoostingClassifier	BIO	6	61.09	62.79
GradientBoostingClassifier	BIO	7	64.46	65.50
GradientBoostingClassifier	BIO	8	60.71	62.01
GradientBoostingClassifier	PES	1	96.66	72.16
GradientBoostingClassifier	PES	2	93.20	66.54
GradientBoostingClassifier	PES	3	92.62	66.56
GradientBoostingClassifier	PES	4	89.68	62.84
GradientBoostingClassifier	PES	5	91.87	77.85
GradientBoostingClassifier	PES	6	87.69	74.34
GradientBoostingClassifier	PES	7	83.00	76.61
GradientBoostingClassifier	PES	8	85.00	77.41
GradientBoostingClassifier	FIS	1	77.50	68.22

Algoritmo	CursoGrupo	Período	RC	MCC
GradientBoostingClassifier	FIS	2	62.25	58.22
GradientBoostingClassifier	FIS	3	65.77	66.36
GradientBoostingClassifier	FIS	4	62.18	64.62
GradientBoostingClassifier	FIS	5	61.79	66.60
GradientBoostingClassifier	FIS	6	60.33	67.60
GradientBoostingClassifier	FIS	7	44.00	54.59
GradientBoostingClassifier	FIS	8	46.67	47.78
GradientBoostingClassifier	MAT	1	98.21	42.25
GradientBoostingClassifier	MAT	2	90.69	58.13
GradientBoostingClassifier	MAT	3	90.88	64.66
GradientBoostingClassifier	MAT	4	85.36	64.59
GradientBoostingClassifier	MAT	5	82.67	72.18
GradientBoostingClassifier	MAT	6	80.55	76.11
GradientBoostingClassifier	MAT	7	83.57	81.31
GradientBoostingClassifier	MAT	8	79.67	80.54
GradientBoostingClassifier	TUR	1	96.95	42.11
GradientBoostingClassifier	TUR	2	67.63	47.53
GradientBoostingClassifier	TUR	3	61.11	53.58
GradientBoostingClassifier	TUR	4	72.67	56.46
GradientBoostingClassifier	TUR	5	73.08	65.74
GradientBoostingClassifier	TUR	6	73.33	72.76
GradientBoostingClassifier	TUR	7	64.64	61.59
GradientBoostingClassifier	TUR	8	54.52	52.13
GradientBoostingClassifier	GRUPO1	1	96.93	56.99
GradientBoostingClassifier	GRUPO1	2	75.85	61.81
GradientBoostingClassifier	GRUPO1	3	71.37	62.82
GradientBoostingClassifier	GRUPO1	4	68.56	64.83
GradientBoostingClassifier	GRUPO1	5	69.00	68.26
GradientBoostingClassifier	GRUPO1	6	66.82	68.45
GradientBoostingClassifier	GRUPO1	7	62.25	66.74
GradientBoostingClassifier	GRUPO1	8	61.39	67.86
GradientBoostingClassifier	GRUPO2	1	47.55	53.61
GradientBoostingClassifier	GRUPO2	2	40.03	45.98
GradientBoostingClassifier	GRUPO2	3	39.75	51.59
GradientBoostingClassifier	GRUPO2	4	51.67	63.45
GradientBoostingClassifier	GRUPO2	5	53.67	63.60
GradientBoostingClassifier	GRUPO2	6	40.71	52.34

Algoritmo	CursoGrupo	Período	RC	MCC
GradientBoostingClassifier	GRUPO2	7	49.33	59.76
GradientBoostingClassifier	GRUPO2	8	32.50	42.23
GradientBoostingClassifier	GRUPO3	1	76.38	54.71
GradientBoostingClassifier	GRUPO3	2	59.11	50.34
GradientBoostingClassifier	GRUPO3	3	56.85	56.08
GradientBoostingClassifier	GRUPO3	4	55.82	60.40
GradientBoostingClassifier	GRUPO3	5	57.33	63.89
GradientBoostingClassifier	GRUPO3	6	57.94	64.04
GradientBoostingClassifier	GRUPO3	7	52.50	61.84
GradientBoostingClassifier	GRUPO3	8	51.91	57.15
MLPClassifier	PSI	1	61.52	25.89
MLPClassifier	PSI	2	44.14	26.64
MLPClassifier	PSI	3	40.00	23.08
MLPClassifier	PSI	4	22.36	1.70
MLPClassifier	PSI	5	20.00	0.05
MLPClassifier	PSI	6	28.33	0.42
MLPClassifier	PSI	7	11.00	4.05
MLPClassifier	PSI	8	0.00	0.00
MLPClassifier	PED	1	52.88	26.23
MLPClassifier	PED	2	18.67	4.54
MLPClassifier	PED	3	19.00	-1.73
MLPClassifier	PED	4	15.83	6.93
MLPClassifier	PED	5	0.00	0.00
MLPClassifier	PED	6	0.00	0.00
MLPClassifier	PED	7	0.00	0.00
MLPClassifier	PED	8	0.00	0.00
MLPClassifier	ADM	1	64.35	28.83
MLPClassifier	ADM	2	59.73	16.97
MLPClassifier	ADM	3	34.86	18.12
MLPClassifier	ADM	4	41.43	20.08
MLPClassifier	ADM	5	32.00	20.36
MLPClassifier	ADM	6	45.00	11.65
MLPClassifier	ADM	7	40.83	18.35
MLPClassifier	ADM	8	16.67	15.00
MLPClassifier	CON	1	50.12	27.79
MLPClassifier	CON	2	36.08	26.71
MLPClassifier	CON	3	51.62	37.94

Algoritmo	CursoGrupo	Período	RC	MCC
MLPClassifier	CON	4	39.00	21.27
MLPClassifier	CON	5	39.03	21.75
MLPClassifier	CON	6	14.76	21.11
MLPClassifier	CON	7	31.33	26.46
MLPClassifier	CON	8	7.50	1.41
MLPClassifier	ECON	1	83.52	7.77
MLPClassifier	ECON	2	62.97	34.47
MLPClassifier	ECON	3	80.89	22.53
MLPClassifier	ECON	4	71.96	41.11
MLPClassifier	ECON	5	46.67	38.86
MLPClassifier	ECON	6	69.67	51.13
MLPClassifier	ECON	7	72.00	55.57
MLPClassifier	ECON	8	75.00	64.50
MLPClassifier	BIOM	1	56.92	32.18
MLPClassifier	BIOM	2	45.05	23.55
MLPClassifier	BIOM	3	52.56	34.03
MLPClassifier	BIOM	4	32.86	27.11
MLPClassifier	BIOM	5	57.33	45.23
MLPClassifier	BIOM	6	27.50	24.66
MLPClassifier	BIOM	7	46.67	45.33
MLPClassifier	BIOM	8	38.33	38.58
MLPClassifier	BIO	1	73.60	18.97
MLPClassifier	BIO	2	50.00	32.31
MLPClassifier	BIO	3	67.53	38.62
MLPClassifier	BIO	4	52.08	38.00
MLPClassifier	BIO	5	71.65	43.66
MLPClassifier	BIO	6	60.55	45.83
MLPClassifier	BIO	7	62.86	50.96
MLPClassifier	BIO	8	52.14	47.28
MLPClassifier	PES	1	83.91	30.48
MLPClassifier	PES	2	81.86	39.84
MLPClassifier	PES	3	79.97	44.61
MLPClassifier	PES	4	72.26	48.58
MLPClassifier	PES	5	83.86	56.05
MLPClassifier	PES	6	69.34	47.22
MLPClassifier	PES	7	64.22	49.44
MLPClassifier	PES	8	84.03	70.03

Algoritmo	CursoGrupo	Período	RC	MCC
MLPClassifier	FIS	1	49.71	32.54
MLPClassifier	FIS	2	32.67	30.58
MLPClassifier	FIS	3	37.24	33.50
MLPClassifier	FIS	4	48.64	21.99
MLPClassifier	FIS	5	36.96	28.27
MLPClassifier	FIS	6	30.67	16.94
MLPClassifier	FIS	7	36.00	25.98
MLPClassifier	FIS	8	53.33	29.56
MLPClassifier	MAT	1	93.10	19.44
MLPClassifier	MAT	2	86.32	8.62
MLPClassifier	MAT	3	78.88	41.37
MLPClassifier	MAT	4	78.40	46.82
MLPClassifier	MAT	5	79.33	54.66
MLPClassifier	MAT	6	57.00	42.74
MLPClassifier	MAT	7	77.32	65.03
MLPClassifier	MAT	8	71.00	58.85
MLPClassifier	TUR	1	80.86	21.58
MLPClassifier	TUR	2	66.45	29.66
MLPClassifier	TUR	3	66.67	23.92
MLPClassifier	TUR	4	79.95	33.44
MLPClassifier	TUR	5	69.94	51.34
MLPClassifier	TUR	6	67.78	48.80
MLPClassifier	TUR	7	61.61	52.22
MLPClassifier	TUR	8	57.14	59.33
MLPClassifier	GRUPO1	1	76.80	51.81
MLPClassifier	GRUPO1	2	53.03	39.85
MLPClassifier	GRUPO1	3	69.72	57.21
MLPClassifier	GRUPO1	4	73.73	57.23
MLPClassifier	GRUPO1	5	80.29	60.95
MLPClassifier	GRUPO1	6	71.61	64.53
MLPClassifier	GRUPO1	7	68.25	59.10
MLPClassifier	GRUPO1	8	66.71	61.75
MLPClassifier	GRUPO2	1	54.18	30.27
MLPClassifier	GRUPO2	2	47.00	26.70
MLPClassifier	GRUPO2	3	25.87	25.26
MLPClassifier	GRUPO2	4	41.67	27.13
MLPClassifier	GRUPO2	5	9.56	11.93

Algoritmo	**CursoGrupo**	**Período**	**RC**	**MCC**
MLPClassifier	GRUPO2	6	10.24	8.47
MLPClassifier	GRUPO2	7	14.33	6.13
MLPClassifier	GRUPO2	8	2.50	-2.33
MLPClassifier	GRUPO3	1	56.91	28.34
MLPClassifier	GRUPO3	2	49.77	27.71
MLPClassifier	GRUPO3	3	50.97	36.11
MLPClassifier	GRUPO3	4	41.38	32.97
MLPClassifier	GRUPO3	5	46.43	27.68
MLPClassifier	GRUPO3	6	59.89	48.69
MLPClassifier	GRUPO3	7	62.76	55.29
MLPClassifier	GRUPO3	8	44.73	41.64

ANEXO B – Ranqueamento dos Atributos dos Grupos

Tabela 15 – Ranqueamento dos atributos dos grupos em cada período.

CursoGrupo	**Atributo**	**P1**	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	$\sum$
GRUPO0	qtd_aprovacao	1	1	1	1	1	1	1	1	8
GRUPO0	tem_ensino	2	4	7	6	6	7	7	10	47
GRUPO0	rep_ob_semestre	3	3	4	3	4	4	4	4	28
GRUPO0	qtd_reprovacao_ft_nt	4	5	2	4	2	2	2	3	26
GRUPO0	ap_ob_semestre	5	2	3	2	3	3	3	2	22
GRUPO0	media_acumulada	6	6	6	5	7	6	6	6	48
GRUPO0	qtd_reprovacao_nt	7	7	5	7	5	5	5	5	48
GRUPO0	nota_ciencias_natureza	8	8	9	9	12	10	9	8	73
GRUPO0	nota_ciencias_humanas	9	9	12	11	9	12	12	12	82
GRUPO0	nota_redacao	10	10	10	12	8	9	10	9	78
GRUPO0	nota_linguagem	11	11	19	22	20	22	19	13	136
GRUPO0	temcredito_cc_ca	12	16	20	17	17	24	20	24	144
GRUPO0	area_cnpq	13	17	33	10	13	21	33	17	139
GRUPO0	sexo	14	12	29	19	11	15	29	30	162
GRUPO0	idade_ingresso	15	15	14	16	15	11	14	14	112
GRUPO0	nota_matematica	16	26	24	32	24	26	24	26	203
GRUPO0	intervalo_em_ingresso	17	30	23	33	18	14	23	21	172
GRUPO0	convocacao	18	28	16	20	33	28	16	33	202
GRUPO0	qtd_reprovacao_ft	19	22	15	23	16	13	15	28	156
GRUPO0	periodo_ingresso	20	27	22	24	21	32	22	31	204
GRUPO0	turno	21	33	31	31	27	19	31	32	227
GRUPO0	raca	22	21	21	26	28	29	21	20	188
GRUPO0	uf_naturalidade	23	31	28	18	19	16	28	19	171
GRUPO0	estadocivil	24	20	13	14	30	30	13	22	184
GRUPO0	conclusao_escola_publica	25	24	30	21	25	20	30	16	191
GRUPO0	quantidade_membros_familia	26	32	18	29	29	27	18	29	213
GRUPO0	qtd_trancamento_turma	27	14	8	15	10	8	8	7	113
GRUPO0	trancamento_programa	28	25	11	13	14	17	11	11	141
GRUPO0	opcao_inscricao	29	29	25	27	31	25	25	23	208
GRUPO0	cursounapandemia	30	23	26	30	32	33	26	27	229

CursoGrupo	Atributo	P1	P2	P3	P4	P5	P6	P7	P8	∑
GRUPO0	tipoacaoafirmativa	31	19	17	28	22	31	17	15	188
GRUPO0	qtd_cancelamento_turma	32	18	27	8	23	18	27	25	162
GRUPO0	grau	33	13	32	25	26	23	32	18	183
GRUPO1	media_acumulada	1	12	20	26	32	11	20	16	134
GRUPO1	qtd_aprovacao	2	2	2	1	1	2	2	2	14
GRUPO1	rep_ob_semestre	3	3	3	5	3	3	3	3	29
GRUPO1	ap_ob_semestre	4	1	1	2	2	1	1	1	13
GRUPO1	qtd_reprovacao_nt	5	6	5	7	5	5	5	5	46
GRUPO1	qtd_reprovacao_ft_nt	6	5	4	4	4	4	4	4	35
GRUPO1	nota_ciencias_humanas	7	8	14	9	11	15	14	17	91
GRUPO1	tem_ensino	8	4	17	14	19	12	17	10	93
GRUPO1	nota_redacao	9	11	31	17	30	28	31	28	179
GRUPO1	nota_linguagem	10	10	16	10	25	19	16	18	120
GRUPO1	qtd_cancelamento_turma	11	7	9	3	6	7	9	15	61
GRUPO1	intervalo_em_ingresso	12	23	8	33	21	13	8	19	157
GRUPO1	qtd_reprovacao_ft	13	13	7	8	8	8	7	6	74
GRUPO1	nota_ciencias_natureza	14	14	29	25	22	26	29	26	173
GRUPO1	area_cnpq	15	33	11	22	15	33	11	33	169
GRUPO1	grau	16	9	23	15	13	20	23	7	136
GRUPO1	convocacao	17	21	33	11	12	24	33	25	162
GRUPO1	tipoacaoafirmativa	18	22	32	29	10	18	32	29	184
GRUPO1	temcredito_cc_ca	19	30	28	18	23	30	28	30	199
GRUPO1	periodo_ingresso	20	16	10	31	18	17	10	13	139
GRUPO1	quantidade_membros_familia	21	18	25	16	28	21	25	22	169
GRUPO1	idade_ingresso	22	27	15	13	33	10	15	11	153
GRUPO1	opcao_inscricao	23	24	27	19	24	25	27	20	191
GRUPO1	conclusao_escola_publica	24	29	21	21	31	29	21	12	191
GRUPO1	nota_matematica	25	28	30	23	27	27	30	27	214
GRUPO1	cursounapandemia	26	31	26	24	26	31	26	31	227
GRUPO1	trancamento_programa	27	19	6	6	7	6	6	8	84
GRUPO1	raca	28	15	13	20	16	22	13	21	150
GRUPO1	estadocivil	29	25	22	27	14	23	22	24	194
GRUPO1	uf_naturalidade	30	17	18	28	9	9	18	23	165
GRUPO1	sexo	31	26	24	30	29	14	24	14	188
GRUPO1	turno	32	20	19	32	17	16	19	9	168
GRUPO1	qtd_trancamento_turma	33	32	12	12	20	32	12	32	186
GRUPO2	qtd_aprovacao	1	1	1	1	1	1	1	1	8

CursoGrupo	**Atributo**	**P1**	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	$\sum$
GRUPO2	media_acumulada	2	6	12	17	6	11	12	7	70
GRUPO2	qtd_reprovacao_ft_nt	3	4	2	2	2	3	2	2	20
GRUPO2	rep_ob_semestre	4	2	3	3	3	2	3	3	23
GRUPO2	qtd_reprovacao_nt	5	5	5	5	5	5	5	5	40
GRUPO2	ap_ob_semestre	6	3	4	4	4	4	4	4	33
GRUPO2	nota_ciencias_natureza	7	9	9	6	18	12	9	10	79
GRUPO2	tem_ensino	8	7	11	8	8	8	11	6	62
GRUPO2	nota_redacao	9	8	15	10	7	16	15	11	86
GRUPO2	temcredito_cc_ca	10	20	31	30	19	29	31	19	181
GRUPO2	nota_matematica	11	10	23	12	25	20	23	14	130
GRUPO2	qtd_cancelamento_turma	12	33	7	7	12	19	7	15	117
GRUPO2	cursounapandemia	13	31	32	31	26	18	32	33	214
GRUPO2	nota_linguagem	14	21	18	28	32	26	18	17	176
GRUPO2	opcao_inscricao	15	29	13	19	31	13	13	28	170
GRUPO2	area_cnpq	16	25	22	33	33	33	22	22	217
GRUPO2	qtd_reprovacao_ft	17	13	10	11	10	15	10	20	110
GRUPO2	conclusao_escola_publica	18	16	14	15	20	22	14	25	151
GRUPO2	qtd_trancamento_turma	19	15	8	20	9	10	8	8	113
GRUPO2	intervalo_em_ingresso	20	18	30	16	13	28	30	13	164
GRUPO2	nota_ciencias_humanas	21	14	28	9	21	25	28	27	164
GRUPO2	sexo	22	11	24	14	11	9	24	12	114
GRUPO2	grau	23	17	26	24	28	32	26	21	184
GRUPO2	turno	24	26	16	18	24	31	16	23	194
GRUPO2	quantidade_membros_familia	25	30	20	22	30	17	20	32	194
GRUPO2	periodo_ingresso	26	12	25	23	17	30	25	24	188
GRUPO2	convocacao	27	24	27	21	29	23	27	9	176
GRUPO2	uf_naturalidade	28	19	21	25	27	27	21	31	207
GRUPO2	estadocivil	29	28	17	13	23	6	17	18	162
GRUPO2	raca	30	27	19	27	22	24	19	29	205
GRUPO2	tipoacaoafirmativa	31	22	29	29	15	21	29	30	202
GRUPO2	trancamento_programa	32	32	6	32	14	7	6	16	146
GRUPO2	idade_ingresso	33	23	33	26	16	14	33	26	188