FUNDAÇÃO GETULIO VARGAS
ESCOLA DE ADMINISTRAÇÃO DE EMPRESAS DE SÃO PAULO

MAURILIO APARECIDO BENEVENTO

O USO DE ALGORITMOS DE APRENDIZAGEM DE MÁQUINA PARA PREVER O DESEMPENHO DO ALUNO

SÃO PAULO
2024

MAURILIO APARECIDO BENEVENTO

O USO DE ALGORITMOS DE APRENDIZAGEM DE MÁQUINA PARA PREVER O DESEMPENHO DO ALUNO

Tese apresentada à Escola de Administração de Empresas de São Paulo da Fundação Getulio Vargas, como requisito para obtenção do título de Doutor em Administração de Empresas.

Campo de Conhecimento: Administração e Análise de Tecnologia de Informação

Orientador: Prof. Dr. Fernando de Souza Meirelles

SÃO PAULO

2024

Dados Internacionais de Catalogação na Publicação (CIP)
Ficha catalográfica elaborada pelo Sistema de Bibliotecas/FGV

Benevento, Maurilio Aparecido.

O uso de algoritmos de aprendizagem de máquina para prever o desempenho do aluno / Maurilio Aparecido Benevento. - 2024.

118 f.

Orientador: Fernando de Souza Meirelles.

Tese (doutorado CDAE) – Fundação Getulio Vargas, Escola de Administração de Empresas de São Paulo.

1. Avaliação educacional. 2. Aprendizagem. 3. Inovações educacionais. 4. Tecnologia da informação. 5. Inteligência artificial. I. Meirelles, Fernando de Souza. II. Tese (doutorado CDAE) – Escola de Administração de Empresas de São Paulo. III. Fundação Getulio Vargas. IV. Título.

CDU 37.091.212.6:007.52

Ficha Catalográfica elaborada por: Raphael Figueiredo Xavier CRB SP-009987/O

MAURILIO APARECIDO BENEVENTO

O USO DE ALGORITMOS DE APRENDIZAGEM DE MÁQUINA PARA PREVER O DESEMPENHO DO ALUNO

Tese apresentada à Escola de Administração Empresas de São Paulo da Fundação Getulio Vargas, como requisito para obtenção do título de Doutor em Administração de Empresas.

Campo de conhecimento: Administração, Análise e Tecnologia da Informação.

Data de aprovação: 9/10/2024

Banca Examinadora:

_____
Prof. Dr. Fernando de Souza Meirelles
FGV EAESP (Orientador)

_____
Prof. Dr. André Luiz Silva Samartini
FGV EAESP

_____
Prof. Dr. Marcio Cardoso Machado
UNIP

_____
Prof. Dr. Marco Antônio Pinheiro da Silveira
USCS

Dedico este trabalho à minha esposa, Lilian Landi Benevento, por me apoiar e incentivar até a sua conclusão.

AGRADECIMENTOS

Agradeço ao Prof. Dr. Fernando de Souza Meirelles por ser meu orientador. Além disso, por ser sempre gentil e compartilhar seus *insights* valiosos.

Agradeço ao Prof. Dr. André Luiz Silva Samartini por ter aceitado o convite para participar como membro da banca avaliadora e por suas contribuições para a melhoria desta tese.

Agradeço ao Prof. Dr. Marcio Cardoso Machado e ao Prof. Dr. Marco Antônio Pinheiro da Silveira por participarem da banca avaliadora como membros externos, trazendo contribuições valiosas para este trabalho.

Agradeço à Fundação Getulio Vargas do Estado de São Paulo pela incrível oportunidade de estudar nessa escola de negócios.

Agradeço aos professores da FGV EAESP, em especial, da minha linha de pesquisa: AATI, pelo aprendizado e boa convivência.

Agradeço ao Prof. Dr. Marcos Ventura pelas dicas durante a construção da ideia e pela amizade ao longo do caminho.

Agradeço à Lilian Landi Benevento pela cumplicidade de décadas e por estar sempre ao meu lado acreditando que posso sempre fazer melhor.

Agradeço a Deus, Pai do meu Senhor e Salvador Jesus Cristo, cujo nome é o único que importa, por permitir toda essa jornada e por estar comigo em momentos turbulentos sendo o meu escudo e confidente.

"A aprendizagem de máquina é a arte e a ciência de ensinar as máquinas a aprenderem por si mesmas." (Geoffrey Hinton)

RESUMO

Esta tese investiga a aplicação de algoritmos de aprendizagem de máquina como ferramenta para aprimorar a administração escolar e personalizar o ensino, focando a previsão do desempenho acadêmico e a redução de reprovação e abandono escolar. A pesquisa utilizou dados do Censo Escolar 2023 e da PNAD 2023, que destacam preocupantes taxas de reprovação no ensino fundamental (4,2%) e elevadas taxas de evasão no ensino médio e superior, especialmente em cursos de Tecnologia da Informação (66,5%). A metodologia envolveu a análise de 900 registros de alunos ao longo de 8 meses, abrangendo 15 disciplinas. Foi estabelecida uma classificação de cada aluno como "Evasão (Não Apto e provável abandono)" e "Ativo (Apto com provável aprovação)", com base na maximização de métricas de seu desempenho, e perfil de sensibilidade e de especificidade. Essas categorias foram integradas a um modelo de gestão escolar que permite o monitoramento contínuo do progresso dos alunos, facilitando intervenções educacionais personalizadas. Essas intervenções foram fundamentadas nas teorias da Dependência do Aprendizado (TDA) e da Expectativa da Motivação (TEM). O modelo foi implementado em um Ambiente Virtual de Aprendizagem (AVA), promovendo uma administração escolar mais adaptativa e responsiva às necessidades individuais dos alunos. A tese demonstra que a combinação de um algoritmo de aprendizagem de máquina com abordagens pedagógicas estruturadas e uma administração moderna pode melhorar significativamente a capacidade de prever o desempenho acadêmico e reduzir a evasão escolar. Assim, esta pesquisa propõe uma estratégia para enfrentar os desafios educacionais atuais no Brasil, promovendo a personalização do ensino e aumentando a eficiência na gestão escolar.

Palavras-chave: Administração moderna, Personalização do ensino, Evasão escolar, Previsão de desempenho do aluno, Algoritmos de aprendizagem de máquina, Inteligência Artificial, Caminhos de aprendizagem, Tecnologia de Informação.

ABSTRACT

This thesis investigates the application of machine learning algorithms as a tool to improve school management and personalize education, with a focus on predicting academic performance and reducing school failure and dropout. The research used data from the 2023 School Census and the 2023 PNAD, which show worrying failure rates in elementary school (4.2%) and high dropout rates in secondary and higher education, especially in computer science courses (66.5%). The methodology involved the analysis of 900 student records over 8 months, covering 15 subjects. For each student, a classification was made into "Dropout (Not Fit and Likely to Drop Out)" and "Active (Fit and Likely to Pass)" based on the maximization of their performance metrics and sensitivity and specificity profiles. These categories were integrated into a school management model that allowed continuous monitoring of student progress and facilitated personalized educational interventions. These interventions were based on the theories of Learning Dependence (LD) and Motivational Expectancy (ME). The model was implemented in a Virtual Learning Environment (VLE) to promote school management that is more adaptive and responsive to students' individual needs. The thesis demonstrates that the combination of a machine learning algorithm with structured pedagogical approaches and modern administration can significantly improve school performance.

Keywords: Modern administration, Personalized learning, School dropout, Student performance prediction, Machine learning algorithms, Artificial Intelligence, Learning pathways, Information Technology.

LISTA DE FIGURAS

Figura 1 – *Etapas e Componentes do Sistema Proposto para AM* 30
Figura 2 – *Mineração Tradicional de Dados (Simplificado)* 33
Figura 3 – *Processo de Aprendizagem de Máquina (Simplificado)* 33
Figura 4 - *Modelo de Aprendizagem de Máquina para Prever o Desempenho do Aluno* 52
Figura 5 - *Estrutura da Metodologia de Pesquisa Aplicada nesta Tese* 55
Figura 6 – *Recorte do Resultado do Processo One-Hot Encoding no DataFrame (3)* 75
Figura 7 – *Separação para Treinamento e Teste* 76
Figura 8 – *Recorte dos Dados que Serão Utilizados para Treinamento (train_x)* 76
Figura 9 – *Recorte dos Dados do DataFrame train_y* 77
Figura 10 – *Recorte das Predições Realizadas com o Melhor Algoritmo* 81
Figura 11 – *Recorte da Predição no DataSet Original* 82
Figura 12 – *Teste Kappa para Scores* 85

LISTA DE TABELAS

Tabela 1 – *Exemplos para o Uso do Tipo Supervisionado* 34
Tabela 2 – *Exemplos de Aplicações Utilizando o Tipo Não Supervisionado* 35
Tabela 3 – *Exemplos de Aplicações Utilizando o Tipo semi-supervisionado* 36
Tabela 4 – *Aplicações Possíveis para o Tipo por Reforço* 37
Tabela 5 – *Aplicações Possíveis para o Aprendizado Profundo* 38
Tabela 6 – *Resumo dos Tipos de AM* 38
Tabela 7 – *Técnicas de Pesquisa Utilizadas nesta Tese* 56
Tabela 8 – *Fonte de Dados* 58
Tabela 9 – *Exemplo de Saída do DataFrame após Aplicação do One-Hot Encoding* 63
Tabela 10 – *Exemplo da Escolha do Melhor Algoritmo* 71
Tabela 11 – *Métricas de Acurácia* 78
Tabela 12 – *Os Cinco Melhores Resultados de Acurácia* 79
Tabela 13 – *Outros Scores para os Cinco Melhores Algoritmos* 79
Tabela 14 – *Registros de Alunos Utilizados para Treinamento e Teste e seus Pontos de Dados* 83
Tabela 15 – *Resultados das Hipóteses* 84
Tabela 16 – *Resultado do Teste de Kappa para cada Disciplina* 86
Tabela 18 - *Distribuição de Estudantes "Ativo" e "Evasão" por Perfil Comportamental* 88
Tabela 19 – *Contingência – Teste de Qui-Quadrado – Hipótese D* 89
Tabela 20 – *Distribuição de Estudantes "Ativo" e "Evasão" por Sistemas Representacionais* 89

LISTA DE ABREVIATURAS E SIGLAS

Abreviação	Significado	Descrição
AM	Aprendizagem de Máquina	Campo da IA que utiliza algoritmos para aprendizado e previsões.
API	Application Programming Interface	Definições e protocolos para construir e integrar *software*.
AUC-ROC	Area Under the Curve - Receiver Operating Characteristic	Métrica de avaliação usada para modelos de classificação, especialmente aqueles que lidam com classes desbalanceadas.
AVA	Ambiente Virtual de Aprendizagem	Plataforma para ensino e aprendizado *on-line*.
BNCC	Base Nacional Comum Curricular	Define as competências, habilidades e conteúdos essenciais que todos os estudantes da educação básica no Brasil devem aprender.
DTC	Decision Trees Classification	Algoritmo de classificação baseado em árvore de decisão.
DTR	Decision Trees Regression	Algoritmo de regressão baseado em árvore de decisão.
EAESP	Escola de Administração de Empresas de São Paulo	São Paulo Business School
EDM	Educational Data Mining	Mineração de dados voltada para a educação.
FGV	Fundação Getulio Vargas	Getulio Vargas Foundation
GBC	Gradient Boosting Classification	Algoritmo de classificação que combina múltiplos modelos fracos.
GBR	Gradient Boosting Regression	Algoritmo de regressão que cria um modelo preditivo a partir de modelos fracos.
IA	Inteligência Artificial	Campo da ciência da computação focado em criar sistemas inteligentes.
INEP	Instituto Nacional de Estudos e Pesquisas	Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira
KNC	Neural Network Models (Classification)	Algoritmo de classificação que identifica a classe de um dado ponto com base nas classes dos seus vizinhos mais próximos.
KNR	Neural Network Models (Regression)	Algoritmo de regressão que estima o valor de uma variável-alvo com base nos valores dos "K" vizinhos mais próximos.
KNP	Neural Network Perceptron	Tipo de rede neural usada para classificação.
KNR	Neural Network Regression	Algoritmo de regressão baseado em rede neural.
LDA	Linear Discriminant Analysis	Técnica de classificação que encontra a combinação linear de variáveis que melhor separa as classes.
LiR	Neural Network Models (S)	Método de regressão que modela relações lineares.

Abreviação	Significado	Descrição
LLM	Large Language Models	Usa técnicas avançadas de aprendizado de máquina, como redes neurais profundas.
LLR	Linear Logistic Regression	Algoritmo de regressão logística linear.
LRR	Logistic Ridge Regression	Regressão logística com penalização para prevenir *overfitting*.
MAE	Mean Absolute Error	Métrica de avaliação utilizada em problemas de regressão que calcula a média das diferenças absolutas entre os valores reais e as previsões feitas por um modelo.
MEC	Ministério da Educação	Órgão do governo federal responsável pela formulação e implementação de políticas públicas relacionadas à educação no Brasil.
ML	Machine Learning	Subcampo da IA que utiliza algoritmos para aprender padrões a partir de dados.
MSE	Mean Squared Error	Métrica de avaliação usada em problemas de regressão que calcula a média dos quadrados das diferenças entre os valores reais e as previsões feitas por um modelo.
NBB	Naïve Bayes Bernoulli	Algoritmo Naïve Bayes para variáveis binárias.
NBG	Naïve Bayes Gaussian	Algoritmo Naïve Bayes para variáveis contínuas.
NNC	Nearest Neighbors Classification	Algoritmo de classificação baseado nos vizinhos mais próximos.
NNR	Nearest Neighbors Regression	Algoritmo de regressão baseado em rede neural.
PCA	Principal Component Analysis	É uma técnica de redução de dimensionalidade que transforma os dados em um novo espaço de menor dimensão.
PDA	Previsão de Desempenho do Aluno	Método ou modelo utilizado para estimar o desempenho acadêmico dos alunos ao longo de um curso ou disciplina.
PNAD	Pesquisa Nacional por Amostra de Domicílios	Pesquisa realizada pelo Instituto Brasileiro de Geografia e Estatística (IBGE) que coleta informações socioeconômicas sobre a população brasileira.
RFC	Random Forest Classification	Algoritmo de classificação que utiliza múltiplas árvores de decisão.
RFR	Random Forest Regression	Algoritmo de regressão que utiliza múltiplas árvores de decisão.
SGD	Stochastic Gradient Descent	Método de otimização amplamente utilizado em algoritmos de aprendizado de máquina e redes neurais para minimizar funções de custo.
SGDC	Stochastic Gradient Descent Classification	Algoritmo de classificação que utiliza descida de gradiente estocástica.
SGDR	Stochastic Gradient Descent Regression	Algoritmo de regressão que utiliza descida de gradiente estocástica.
SVC	Support Vector Classification	Algoritmo de classificação baseado em máquinas de vetor de suporte.

Abreviação	Significado	Descrição
SVR	Support Vector Regression	Algoritmo de regressão baseado em máquinas de vetor de suporte.
TDA	Teoria da Dependência do Aprendizado	Teoria de reforço sobre a Qualidade e Equidade conforme a BNCC.
TEM	Teoria da Expectativa da Motivação	Teoria para instrumentalidade (Selo) orientada para a Motivação, Atitude e Metacognição integrada com o EDM.
TI	Tecnologia da Informação	Uso de sistemas e tecnologia para gerenciar informações.

SUMÁRIO

1 INTRODUÇÃO **16**
1.1 ADMINISTRAÇÃO EDUCACIONAL MODERNA 16
1.2 OBJETIVOS 18
1.2.1 Objetivo geral e pergunta de pesquisa **18**
1.2.2 Objetivos específicos **18**
1.3 JUSTIFICATIVAS – CONTRIBUIÇÕES TEÓRICAS E PRÁTICAS 19
1.3.1 Contribuição teórica **20**
1.3.2 Contribuição prática **21**
2 REVISÃO BIBLIOGRÁFICA **23**
2.1 ADMINISTRAÇÃO EDUCACIONAL 23
2.2 SOBRE A QUALIDADE DO ENSINO 24
2.3 PREVISÃO DO DESEMPENHO DO ALUNO (PDA) 26
2.3.1 A Previsão do Desempenho do Aluno (PDA) **28**
2.4 APRENDIZAGEM DE MÁQUINA (AM) 29
2.4.1 Mineração de dados *versus* Aprendizagem de Máquina (AM) **32**
2.4.2 Tipos de Aprendizagem de Máquina (AM) **33**
2.4.3 *Softwares* habilitadores **39**
2.4.4 Algoritmos de AM **39**
2.4.5 Iniciando na Aprendizagem de Máquina (AM) **40**
2.4.6 Algoritmos de classificação **40**
2.4.7 Algoritmos de regressão **41**
2.4.8 Variável: independente (x) e dependente (y) **41**
2.5 CONSOLIDAÇÃO DA REVISÃO BIBLIOGRÁFICA SOBRE AM 42
2.6 BASES TEÓRICAS 44
2.6.1 Teoria da Expectativa da Motivação (TEM) **44**
2.6.2 Teoria da Dependência do Aprendizado (TDA) **46**
2.7 HIPÓTESES DE PESQUISA 48
3 METODOLOGIA **51**
3.1 O MODELO DE APRENDIZADO DE MÁQUINA PARA PREVER O DESEMPENHO 51
3.1.1 Não Apto **52**

3.1.2 Apto **53**
3.1.3 Monitoramento **54**
3.2 METODOLOGIA DE PESQUISA 54
3.2.1 Abordagem de pesquisa **55**
3.2.2 Teorias **55**
3.2.3 Nível de pesquisa **55**
3.2.4 Tipo de pesquisa **56**
3.2.5 Técnica de pesquisa **56**
3.3 COLETA DE DADOS 57
3.4 DICIONÁRIO DE DADOS 59
3.5 VARIÁVEIS: INDEPENDENTES E DEPENDENTE 59
3.6 *TIMELINE* DAS ETAPAS DA AM 60
3.6.1 Algoritmos e bibliotecas **60**
3.6.2 DataSet e DataFrame **62**
3.6.3 Limpeza do DataFrame **63**
3.6.4 *One-hot encoding* e *dummies* **63**
3.6.5 Treinamento e teste (Separação do DataFrame) **64**
3.6.6 Métricas **66**
3.6.7 Predição para todo o DataSet **72**
3.6.8 Arquivo com Preditor **72**
3.7 ANÁLISES DOS RESULTADOS 73
3.7.1 Detalhamento do DataFrame **73**
3.7.2 Limpeza de colunas irrelevantes do DataFrame **74**
3.7.3 One-hot encoding **74**
3.7.4 Treinamento e teste **75**
3.7.5 Acurácia: performance ou medida de score **77**
3.7.6 Uso do melhor algoritmo **80**
3.7.7 Treinamento e teste com o melhor algoritmo **80**
3.7.8 Predição com todo o DataFrame **81**
3.7.9 Pontos de dados para a administração escolar **82**
3.8 ESTATÍSTICA 83
3.8.1. Hipóteses **84**

3.9 CAMINHOS DE APRENDIZAGEM 90
3.10 IA GENERATIVA INTEGRADA COM CAMINHOS DE APRENDIZAGEM 91
3.11 DISCUSSÕES 92
3.11.1 Análise dos resultados 92
3.11.2 Implicações práticas 93
3.11.3 Limitações do estudo 94
3.11.4 Perspectivas futuras 94
3.11.5 Reflexão final sobre o uso da IA na educação 95
4 CONCLUSÃO 96
REFERÊNCIAS 98
APÊNDICES 105
APÊNDICE A – *AMBIENTES E FERRAMENTAS PARA APRENDIZAGEM DE MÁQUINA* 105
APÊNDICE B – *EXEMPLO DE ALGORITMOS POR TIPOS E TÉCNICAS* 106
APÊNDICE C – *CONSOLIDAÇÃO DO REFERENCIAL: PREVISÃO DO DESEMPENHO E DE MÁQUINA E DA MINERAÇÃO DE DADOS* 107
APÊNDICE D – *ATRIBUTOS UTILIZADOS POR SHARIRI ET AL. (2015)* 109
APÊNDICE E – *DICIONÁRIO DE DADOS DAS COLUNAS DO DATASET* 110
APÊNDICE F – *TIMELINE DAS ETAPAS* 111
APÊNDICE G – *AGRUPAMENTO DE ALGORITMOS SELECIONADOS PARA ANÁLISE DE ACURÁCIA* 112
APÊNDICE H – *DESCRITIVO DOS ALGORITMOS SELECIONADOS PARA ESTUDO* 113
APÊNDICE I – *BIBLIOTECAS DO PYTHON UTILIZADAS NESTA TESE* 114
APÊNDICE J – *DATAFRAME (1) SEM TRANSFORMAÇÕES* 115
APÊNDICE K – *RESULTADO DA LIMPEZA DE COLUNAS IRRELEVANTES DO DATAFRAME (2)* 116
APÊNDICE L – *RESULTADO DO ONE-HOT ENCODING NO DATAFRAME (3)* 117
APÊNDICE M – *CAMINHOS DE APRENDIZADO COM BASE NO CANVAS INSTRUCTURE* 118

1 INTRODUÇÃO

1.1 Administração educacional moderna

Na administração educacional moderna, a capacidade de adaptação rápida às mudanças e a personalização dos serviços são essenciais para o sucesso das empresas (Kibby, 2022). No setor educacional, isso se aplica de maneira significativa, pois a administração escolar pode transformar a maneira como o ensino é oferecido. Ao adotar uma abordagem flexível e orientada para as necessidades específicas dos alunos, as instituições de ensino podem criar soluções educacionais personalizadas que não apenas melhoram o desempenho acadêmico, mas também aumentam a retenção de alunos (Libâneo, 2010).

A administração educacional, quando alinhada com os princípios de adaptabilidade e personalização, conforme Charania (2024), permite que escolas e universidades ajustem seus currículos, metodologias e recursos de acordo com as demandas individuais dos estudantes. Isso não apenas facilita uma aprendizagem focada no aluno, mas também assegura que a instituição permaneça em sintonia com as exigências do mercado e as expectativas dos alunos (Kibby, 2022).

A educação no Brasil enfrenta desafios significativos relacionados à reprovação e ao abandono escolar, tanto nos anos iniciais do ensino fundamental quanto no ensino superior. Dados do censo escolar divulgados pelo Ministério da Educação (MEC) e pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP, 2023), revelam que a taxa de reprovação na rede pública de ensino fundamental aumentou para 4,2%. Além disso, a Pesquisa Nacional por Amostra de Domicílios (PNAD, 2023) mostra que 9 milhões de alunos não conseguiram terminar o ensino médio, com maior prevalência entre homens.

No ensino superior, a situação é ainda mais preocupante, especialmente em cursos de Tecnologia da Informação (TI), onde a taxa de desistência chega a 66,5%. Esse problema é agravado pela falta de centralização e sincronização de dados educacionais, currículos desatualizados, dificuldades econômicas e baixa qualidade da educação básica. Essas questões resultam em altos índices de evasão escolar e desmotivação dos alunos.

É oportuno encontrar na administração educacional moderna a incorporação de práticas flexíveis e personalizadas, apresentando-se como uma solução para reduzir a reprovação e evasão. A centralização de dados e o uso de métodos preditivos permitem intervenções direcionadas e adequadas, promovendo o progresso acadêmico e a retenção dos alunos. Como observado por Kibby (2022), essa abordagem não apenas melhora a qualidade do ensino, mas também torna as

instituições mais competitivas em um cenário dinâmico, destacando-se pela capacidade de adaptação às necessidades individuais e pela personalização contínua do aprendizado.

Segundo Ofori et al. (2020), a previsão precoce do desempenho dos alunos é essencial para implementar medidas que melhorem os resultados de aprendizagem em tempo hábil. Essa prática pode orientar mudanças nas políticas acadêmicas e nas práticas pedagógicas, proporcionando uma base sólida para o aprimoramento contínuo do aprendizado dos estudantes.

No entanto, essa tarefa é desafiadora, especialmente pela ausência de um banco de dados centralizado, como destacado por Akuma e Abakpa (2021), e pela complexidade envolvida em lidar com grandes quantidades de dados educacionais. Pojon (2017) ressalta que a previsão do desempenho dos alunos é uma tarefa complexa devido ao vasto volume de dados que precisam ser analisados. A compreensão e a aplicação de ferramentas para integrar e processar esses dados tornaram-se, portanto, um foco importante nas pesquisas acadêmicas.

A pesquisa de Ahmad et al. (2022) aponta que a sincronização dos dados internos das escolas ainda é um desafio significativo, o que aumenta a dependência de sistemas externos de previsão. Esse contexto reforça a necessidade de avançar em soluções tecnológicas que permitam uma melhor centralização e utilização dos dados educacionais para prever e melhorar o desempenho acadêmico de maneira mais eficiente e precisa.

O uso de algoritmos de Aprendizagem de Máquina (AM) está se tornando uma prática cada vez mais comum na predição do desempenho acadêmico, possibilitando o processamento de grandes volumes de dados e a identificação de padrões e tendências que ajudam a personalizar o ensino e atender às necessidades individuais dos alunos. No entanto, essa tecnologia traz à tona preocupações importantes sobre a proteção de dados pessoais e o risco de discriminação ou desigualdade na avaliação do desempenho estudantil. A aplicação desses algoritmos deve ser realizada com cautela, garantindo que os princípios de equidade e qualidade defendidos pela Base Nacional Comum Curricular (BNCC, 2022) sejam respeitados.

Além disso, a pesquisa em Educational Data Mining (EDM) reforça a importância de utilizar dados educacionais de maneira responsável para promover intervenções pedagógicas. Segundo estudos psicológicos, como o de Ahmad et al. (2023), o desempenho dos alunos é fortemente influenciado por fatores emocionais, como frustração, ansiedade e expectativas familiares. Esses aspectos, muitas vezes negligenciados, são cruciais para o entendimento pleno do processo de aprendizado e podem ser mais bem explorados com uma abordagem que inclua dados neurocognitivos e emocionais.

Por exemplo, a aplicação de técnicas avançadas como a espectroscopia funcional de infravermelho próximo, conforme demonstrado por Lamb et al. (2022), pode fornecer *insights* em

tempo real sobre o engajamento e a compreensão dos alunos, permitindo a criação de modelos preditivos com alta precisão. Essa abordagem não apenas complementa as técnicas tradicionais baseadas em dados históricos, mas também se alinha aos objetivos da BNCC de promover uma educação inclusiva e personalizada, que valoriza tanto o desenvolvimento cognitivo quanto o emocional dos alunos.

Portanto, integrar dados psicométricos nesse contexto representa uma fronteira promissora para a pesquisa em predição de desempenho acadêmico, oferecendo oportunidades significativas para melhorar a qualidade do ensino e garantir que todos os alunos tenham as melhores condições para aprender.

1.2 Objetivos

1.2.1 Objetivo geral e pergunta de pesquisa

Desenvolver e validar um modelo preditivo utilizando algoritmos de AM em conjunto com métodos estatísticos para antecipar o desempenho acadêmico de alunos, com o intuito de identificar riscos de evasão e propor intervenções personalizadas que melhorem o desempenho escolar.

Além disso, integrar as técnicas preditivas em uma solução educacional que permita a personalização do ensino, a otimização dos recursos pedagógicos e a modernização da administração educacional, alinhando-se às práticas modernas de gestão, que valorizam a adaptabilidade, eficiência e centralização de dados para decisões estratégicas e orientadas às necessidades individuais dos alunos.

Resulta-se na pergunta de pesquisa: **Como o uso de algoritmos de Aprendizagem de Máquina pode prever o desempenho do aluno?**

1.2.2 Objetivos específicos

Os objetivos específicos foram formulados para criar uma base que conduza ao objetivo geral, permitindo a personalização do ensino e a modernização da administração educacional por meio de soluções tecnológicas atuais.

1.2.2.1 Desenvolver Modelo de Aprendizagem de Máquina

Desenvolver um modelo que agrupe estudantes em categorias como "Evasão – Não Apto" e "Ativo – Apto", baseando-se nos resultados dos algoritmos e considerando as diretrizes da BNCC e as teorias: Dependência do Aprendizado (TDA) e Expectativa da Motivação (TEM), com o objetivo de personalizar o ensino e otimizar os recursos educacionais. Nesta etapa foi necessário:

Definir a variável target de forma dicotômica para indicar "Evasão (Provável Evasão ou Reprovação)" ou "Ativo (Provável Aprovação)" e estruturar o DataSet (conjunto de dados), incluindo a seleção de variáveis preditoras, armazenamento, e preparação dos dados, como a criação de colunas *dummies* para facilitar a análise dos modelos de AM.

Escolher, testar e avaliar algoritmos de Aprendizagem de Máquina, utilizando diferentes bibliotecas, como Scikit-learn e TensorFlow. Realizar o treinamento e teste dos algoritmos com métricas adequadas (acurácia, recall, F1-score, AUC-ROC) para identificar o modelo de melhor performance.

1.2.2.2 Implementação e Personalização do Ensino

Integrar o modelo preditivo a uma plataforma educacional, permitindo intervenções pedagógicas personalizadas e automação de processos de gestão de risco. Focar na adaptação das práticas pedagógicas para melhor atender às necessidades dos alunos.

1.3 Justificativas – contribuições teóricas e práticas

Em um ambiente educacional cada vez mais complexo e competitivo, a administração moderna precisa ir além das práticas tradicionais para enfrentar os desafios de reprovação e abandono escolar. A personalização do ensino surge como uma solução central para essas questões, permitindo que as instituições de ensino se adaptem rapidamente às necessidades dos alunos e às exigências do mercado.

Essa personalização é facilitada pela adoção de soluções tecnológicas, que, quando integradas a uma administração ágil e flexível, possibilitam a criação de estratégias educacionais direcionadas. Nesse contexto, o uso de algoritmos de AM não é o fim em si, mas uma ferramenta que auxilia a transformação de dados educacionais em *insights* acionáveis, permitindo que gestores e educadores intervenham de maneira proativa e personalizada.

Ao abordar a administração escolar com um enfoque moderno, esta tese busca preencher lacunas significativas no uso de dados e na implementação de práticas educacionais personalizadas. Por meio da integração de teorias e práticas inovadoras, o trabalho visa fornecer um modelo replicável que possa ser utilizado por administradores educacionais para melhorar a eficácia do ensino e a retenção de alunos, respondendo de maneira mais precisa e adaptada aos desafios educacionais contemporâneos.

1.3.1 Contribuição teórica

Esta tese oferece uma contribuição relevante para o campo da administração educacional ao explorar e aplicar algoritmos de AM na previsão do desempenho acadêmico, com ênfase na personalização do ensino e na gestão escolar. Diferentemente de muitos estudos que se concentram apenas em métricas como acurácia, precisão, *recall* e F1-Score, a pesquisa vai além, demonstrando a aplicabilidade prática desses algoritmos em contextos educacionais, proporcionando um ponto de partida para discussões teóricas.

Ao avaliar diferentes algoritmos de maneira detalhada, a tese não apenas identifica o modelo preditivo mais eficiente, mas também apresenta exemplos de como essas ferramentas podem ser integradas às práticas educacionais. Isso inclui soluções que permitem que instituições adaptem seus currículos e métodos de ensino com base nas necessidades individuais dos alunos, favorecendo a personalização e a eficiência. Dessa forma, a pesquisa contribui para o desenvolvimento de práticas educacionais mais adaptativas.

Além disso, a conexão entre as contribuições práticas e a TEM e a TDA oferece uma perspectiva teórica importante. A TEM foi utilizada para explicar como os algoritmos podem otimizar a identificação de fatores que influenciam a motivação dos alunos, facilitando ajustes no processo de aprendizado. A TDA embasa a adaptação do currículo e as intervenções pedagógicas baseadas no progresso e nas necessidades dos alunos ao longo do tempo. Essas teorias fundamentam a abordagem prática da tese e abrem espaço para futuras investigações que busquem integrar tecnologia educacional e motivação.

Dessa forma, a contribuição teórica desta tese não se limita à avaliação técnica dos algoritmos, mas também promove uma reflexão sobre como as ferramentas preditivas podem apoiar a personalização do ensino e a modernização da administração educacional. Ao associar essas teorias à prática, a pesquisa oferece uma base sólida para novos estudos e futuras aplicações, promovendo uma maior integração entre a Inteligência Artificial (IA) e a gestão educacional. Isso contribui para o entendimento de como o uso de algoritmos pode melhorar a experiência de

aprendizado, ao mesmo tempo que permite que instituições educacionais se adaptem de maneira mais eficiente às demandas de seus alunos.

Além disso, esta tese resultou na publicação de dois artigos científicos que compartilham os *insights* e resultados obtidos:

- "Prever e melhorar o desempenho dos alunos com o uso combinado de aprendizagem de máquina e GPT", 2023, Revista de Gestão e Avaliação Educacional (REGAE).
- "Prever o desempenho do aluno com a aprendizagem de máquina e melhorar com GPT", ISLA 7th Conference on Information Systems in Latin America, 2023 Proceedings.

1.3.2 Contribuição prática

Esta tese oferece uma contribuição prática relevante ao fornecer orientações claras para administradores escolares que buscam integrar algoritmos de AM em suas estratégias de gestão educacional. A aplicação dessas ferramentas possibilita a personalização do ensino e a criação de soluções educacionais que atendam com mais precisão às necessidades individuais dos alunos, promovendo uma administração moderna e orientada por dados.

Os resultados e descobertas da pesquisa podem ser diretamente aplicados no contexto educacional, auxiliando gestores na seleção e utilização dos algoritmos mais adequados para prever o desempenho acadêmico. Isso permite que as instituições não apenas obtenham *insights* detalhados sobre o progresso dos alunos, mas também implementem intervenções pedagógicas direcionadas, favorecendo uma experiência de aprendizado mais personalizada.

Além disso, ao integrar essas tecnologias no cotidiano da gestão escolar, as instituições de ensino podem otimizar recursos, melhorar a retenção de alunos e elevar a qualidade do ensino oferecido. As estratégias delineadas nesta tese permitem que as escolas se adaptem às exigências do mercado educacional, ao mesmo tempo que se alinham às expectativas de uma educação mais personalizada e centrada no aluno. Assim, a pesquisa fornece ferramentas práticas para que

gestores possam transformar suas abordagens administrativas, adotando práticas que favorecem tanto o desempenho acadêmico quanto a eficiência institucional.[1]

[1] Para validação prática, esta tese foi remodelada e submetida como projeto para o International Competition for Business-Driven Artificial Intelligence Projects, aprovada em 17/11/2023, cujo alcance envolveu mais de 800 pessoas de 65 países, totalizando 279 projetos, resultando em 20 projetos finalistas, sendo esta tese entre eles na Global Management Week 2023. Foi um momento de troca de ideias, com a participação de 14 palestrantes de 8 países e mais de 5.000 inscritos, debatendo alguns dos assuntos mais importantes do mundo dos negócios na atualidade.

2 REVISÃO BIBLIOGRÁFICA

2.1 Administração educacional

A importância da administração educacional como solução personalizada ganhou força a partir de 2005, com a EDM (Kabakchieva, 2012). A EDM é um campo de aplicações de mineração de dados, com foco em desenvolver métodos para explorar os tipos únicos de dados que vêm de ambientes educacionais e usar esses métodos para entender os alunos e o seu aprendizado (Al-Shabandar et al., 2017; Obsie & Adem, 2018; Tjandra et al., 2022). A mineração de dados de educação é feita em alta escala, pois a educação tornou-se notável para o crescimento pessoal e econômico de um indivíduo (Agrawal et al., 2017).

Charania (2024) afirma que, com o avanço das tecnologias educacionais, os processos de ensino têm se transformado significativamente. A inclusão dessas tecnologias, associada ao uso de *softwares* educacionais, tem gerado grandes volumes de dados, permitindo novas formas de personalização e avaliação.

Os grandes volumes de dados, conhecidos como EDM, são um campo emergente que utiliza técnicas de mineração de dados, como estatísticas e AM, para analisar dados educacionais e extrair informações valiosas. Tais informações ajudam a otimizar estratégias de ensino e a melhorar o desempenho dos alunos (Ozyurt et al., 2023).

A EDM é aplicada para desenvolver modelos de alunos que expressam o conhecimento atual, a motivação, a metacognição e as atitudes dos alunos. Esse campo também é uma ponte na análise de dados produzidos por qualquer sistema de informação relacionado à aprendizagem e educação. Esses dados podem incluir a interação de um aluno individual com o sistema de aprendizagem, dados administrativos escolares, dados demográficos, e dados sobre a situação cognitiva e emocional dos alunos (Ozyurt et al., 2023). A EDM oferece uma compreensão mais profunda sobre como os alunos aprendem, preveem o aprendizado e entendem o comportamento de aprendizagem real, proporcionando *insights* valiosos para a melhoria dos sistemas educacionais.

O desenvolvimento de um aluno inicia no primeiro dia de aula, independentemente se é no ensino médio ou na graduação, e não na sua formatura, trazendo maior complexidade às escolas em acompanhar o histórico dos alunos devido, por exemplo, às rupturas que ocorrem ao longo da formação. Pesquisadores como Shovon e Haque (2012, p. 146) afirmam que nos cursos de engenharia e ciência da computação o primeiro ano é um ponto de virada na trajetória educacional

do estudante". Em concordância, estão Obsie e Adem (2018, p. 39), quando afirmam que, por meio desse conhecimento prévio, "a universidade poderia prever com 85% de precisão quais alunos se formarão ou não". A escola, de modo geral, poderia concentrar seus esforços acadêmicos nos alunos em maior risco.

Kabakchieva (2012, p. 686) declara em seu trabalho que "as universidades são confrontadas com uma forte competição entre si, tentando atrair os alunos mais adequados que passaram com sucesso pelo processo seletivo universitário e envidando esforços para lidar com a retenção de alunos".

Segundo Buenaño-Fernández et al. (2019), acompanhar o desempenho do aluno é uma das aplicações mais populares na mineração de dados educacionais, que pode ser usada inclusive para melhorar esse desempenho. A alta precisão dessa tarefa é útil para identificar aqueles com baixo desempenho no início do processo de aprendizagem (Ofori et al., 2020).

As instituições de ensino esperam o resultado do processo como predicado para melhorar o desempenho acadêmico do aluno e o aprendizado (Keshtkar, 2018 como citado em Ofori et al., 2020). Como dito, a vasta quantidade de dados para se avaliar sempre foi um desafio para as escolas, mas isso já pode ser superado com ferramentas que foram desenvolvidas para lidar com essa complexidade. Evidente que, além de a escola ter os dados, é necessário entender como utilizá-los (Koutina & Kermanidis, 2011 como citado em Ofori et al., 2020).

2.2 Sobre a qualidade do ensino

O sucesso da escola, segundo Agrawal et al. (2017), está na qualidade do sistema de ensino, nas regras e regulamentos acadêmicos. Esse objetivo pode ser alcançado por meio da análise do desempenho acadêmico dos alunos e dos fatores que afetam o desempenho do aluno (Butt et al., 2023). Deve-se ter uma medida para saber se a educação de qualidade está chegando a todos os alunos ou não (Yanamandra & Praasd, 2022). O mesmo ocorre no processo educacional universitário peruano, por exemplo, cujo desafio é gerar estratégias que melhorem a qualidade do ensino para formar indivíduos com capacidades cognitivas, criativas e inovadoras (Vives et al., 2024).

Além disso, essa análise pode ser efetivamente utilizada para recomendações aos tomadores de decisão da escola. Por outro lado, as escolas possuem muitos dados para a tomada de decisão, mas são pobres em informações, resultando em decisões não confiáveis (Akuma & Abakpa, 2021). Para Ahmad et al. (2021), a qualidade educacional é essencial para o

desenvolvimento do país e considera que os dados coletados dos alunos precisam ser mais bem utilizados para se alcançar o desempenho esperado. Para Ahmad et al. (2022), o sistema de ensino necessita de ferramentas para a previsão de desempenho completo e detalhado que possa garantir a interação e coordenação entre os fatores dos alunos que influenciem a qualidade da formação.

No Brasil, a BNCC, que estabelece as diretrizes educacionais para todas as etapas e modalidades da educação básica, dá orientações explicitas sobre a qualidade esperada do ensino.

De acordo com a BNCC (2022), a qualidade de ensino envolve uma série de elementos interconectados, que contribuem para o desenvolvimento integral dos estudantes e para o alcance dos objetivos da educação básica.

Entre os elementos que compõem a qualidade de ensino, segundo a BNCC (2022), destacam-se:

- o desenvolvimento das competências e habilidades previstas na matriz curricular, que incluem conhecimentos, habilidades e valores necessários para a formação plena dos estudantes;
- a promoção da equidade e da inclusão, garantindo que todos os estudantes tenham acesso às mesmas oportunidades e recursos educacionais, independentemente de suas características pessoais ou sociais;
- a valorização da diversidade e da pluralidade, reconhecendo a importância da convivência pacífica e respeitosa entre diferentes grupos sociais, culturais, étnicos, religiosos e de gênero;
- a utilização de metodologias ativas e contextualizadas, que estimulem a participação ativa dos estudantes na construção de seu próprio conhecimento e na resolução de problemas concretos;
- a formação integral dos estudantes, que inclui o desenvolvimento de competências socioemocionais, como a empatia, a colaboração, a responsabilidade e a capacidade de tomar decisões éticas e responsáveis;
- a articulação entre os diferentes componentes curriculares e áreas de conhecimento, buscando a integração e a transdisciplinaridade para a construção de conhecimentos mais amplos e significativos.

Em suma, a qualidade de ensino, segundo a BNCC (2022), envolve a construção de um projeto pedagógico centrado no desenvolvimento integral dos estudantes, valorizando sua diversidade e promovendo a equidade e a inclusão. Esse documento, para esta tese, é orientativo no aspecto de suporte sobre a qualidade do ensino, entretanto não é utilizado para se avaliar em

qual grau impacta a predição do desempenho do estudante. Porém, vale ressaltar que, se a coleta de dados dos alunos fosse nos níveis socioeconômico e socioemocional, a predição teria resultados absolutamente conectados com a BNCC.

Há relações diretas entre a BNCC com os objetivos específicos desta tese e que são tratadas em seções posteriores, porém citam-se aqui os elementos:

- o desenvolvimento das competências e habilidades previstas na matriz curricular com apoio de caminhos de aprendizagem, termo usado em nosso modelo;
- a promoção da equidade e da inclusão, por meio do resultado da previsão de alunos em risco, em alerta e no caminho, componente do nosso modelo;
- a utilização de metodologias ativas e contextualizadas, por meio da previsão e sugestão de caminhos de aprendizagem[2];
- a articulação entre os diferentes componentes curriculares e áreas de conhecimento por meio das teorias utilizadas e caminhos de aprendizagem.

2.3 Previsão do Desempenho do Aluno (PDA)

A capacidade de prever o desempenho dos alunos é muito importante para melhorar suas habilidades de ensino e tornou-se um conhecimento valioso que pode ser utilizado para diversos fins; por exemplo, um plano estratégico pode ser aplicado para o desenvolvimento de uma educação de qualidade (Ünal, 2020), com especial atenção para os primeiros dois semestres, que se caracterizam como pontos de virada para a jornada completa do aluno. Entende-se que durante esses primeiros semestres poderia ser iniciada a coleta de dados relevantes para compor a avaliação do desempenho acadêmico por meio de algoritmos, permitindo uma atenção cirúrgica às dificuldades do estudante. Segundo Kibby (2022), a personalização por meio de algoritmos permite identificar rapidamente as necessidades dos alunos, oferecendo caminhos de aprendizagem flexíveis e focados.

Em linha com Shovon e Haque (2012), as informações acadêmicas do aluno precisam ser bem aproveitadas e, em tempo oportuno, utilizadas para um acompanhamento mais efetivo.

> Sobre os fatores de avaliação dos alunos, como testes de classe, laboratório de tarefas de exames intermediários e finais, são estudados. Recomenda-se que todas estas informações

[2] Os *links* em azul são âncoras para as seções onde se aborda o assunto com mais profundidade.

> correlacionadas sejam transmitidas ao professor da turma antes da realização do exame final. Este estudo ajudará os professores a reduzirem a taxa de abandono para um nível significativo e melhorar o desempenho dos alunos. (Shovon & Haque, 2012, p. 146)

De fato, o desafio é homérico, tendo em vista a vasta quantidade de informações armazenadas durante a jornada do aluno, sem considerar dados que não são coletados e que seriam importantes para uma análise mais eficiente, tais como variáveis independentes sobre desemprego, renda *per capita* familiar, infraestrutura de moradia e recursos de TI disponíveis. Ainda sobre a qualidade, deve-se lembrar que o entrante em qualquer curso de formação, mesmo no mais fundamental, apresenta o seu desempenho escolar anterior, e este pode ser o *input* para a previsibilidade do desempenho do aluno durante o seu processo de aprendizado. Cabe ressaltar que não há como precisar se o histórico do aluno garante que seu desempenho foi bom, tendo em vista o desconhecimento da avaliação adotada pela escola e seus professores.

É um ponto de partida para correlacionar com a PDA, alinhado com Agrawal et al. (2017), que usaram o resultado de notas históricas para a previsão de notas do próximo semestre, inclusive com o propósito de minimizar a taxa de reprovação.

Para Ofori et al. (2020), a PDA ajuda a equipe acadêmica a rapidamente identificar grupos de alunos que precisam de assistência acadêmica adicional. A precisão com base em seus históricos acadêmicos é crucial para realizar efetivamente as intervenções pedagógicas necessárias para garantir a conclusão do curso no prazo e expectativa do aluno (Belachew & Gobena, 2017).

A corrida pela PDA tem crescido nas escolas, e em diversos níveis de formação. Porém, se espera ir além da base de dados básica, que geralmente são notas e frequências, podendo incluir-se os dados socioemocionais e socioeconômicos, horário de estudo e atributos familiares (acompanhamento dos pais, empregabilidade, condições financeiras, entre outros), mantendo-se o zelo pela privacidade e proteção de dados pessoais, principalmente para menores de idade.

De fato, os atributos socioemocionais são pouco explorados, segundo o extenso trabalho de revisão de Ahmad et al. (2022), mas são considerados grandes influenciadores do desempenho do aluno, em concordância com Kishor et al. (2022), que obtiveram dados de origem familiar e dados médicos como preditores importantes para a realização da pesquisa.

Em outra perspectiva de atributos, relaciona-se a instituição de ensino, que é representada pelo professor no papel ativo de influência no desempenho do aluno, pois fornece assistências diversas durante a interação com os estudantes. Além disso, há a relação com a metodologia de ensino, o envolvimento dos alunos na sala de aula (presencial ou remota) e a percepção do aluno em relação à instituição.

2.3.1 A Previsão do Desempenho do Aluno (PDA)

O objetivo da PDA é encontrar maneiras de propor melhorias durante a jornada do aluno. Entretanto, não é uma tarefa trivial para as escolas melhorarem o desempenho dos alunos (Shovon & Haque, 2012). Além disso, segundo Karthikeyan e Kavipriya (2017), devido ao incrível crescimento da tecnologia, como as mídias sociais, incluindo, mais recentemente, as plataformas de IA generativas, isso pode dissuadir os alunos de sua própria trilha, e essa é uma das razões para que os alunos tenham um desempenho ruim nas atividades acadêmicas que leve à desistência do curso. Porém, não se sabe, até o momento, se o uso das novas plataformas de IA poderia ajudar efetivamente os alunos em melhores desempenhos.

Esse contexto da distração e abandono abriu uma nova reflexão de contraponto para se pensar no aluno – o principal ator e motivo desta tese –, e não apenas na predição do seu desempenho, cuja contribuição poderá influenciar escolas a investigarem como tornar a jornada de aprendizado mais atraente, minimizando as distrações, longe de acabar com elas.

Na visão de Alamri e Alharbi (2021) e Karthikeyan e Kavipriya (2017), "prever o desempenho do aluno possibilita alertá-lo sobre a sua jornada e oferece uma chance de melhorá-la em tempo de formação". Há aqui uma oportunidade de criar um equilíbrio na competição pela atenção do aluno, pois a informação da PDA antecipadamente pode ser uma ferramenta de *feedback* futuro, já que se trata da predição de desempenho.

A PDA, conforme Al-Shabandar et al. (2017), é um campo emergente de pesquisa que visa aprimorar a oferta de educação por meio da aplicação de várias abordagens estatísticas e de AM. A escola é a 5ª indústria que mais investe em TI no Brasil (Cortex, 2022), e tem essa preocupação e anseia por ferramentas que apoiem essa iniciativa, haja vista a crescente quantidade de dados gerados nesse ecossistema educacional, assim como investimentos educacionais igualmente registrados em Portugal por Ünal (2020). Reorientar o desempenho acadêmico dos alunos é crucial, especialmente para instituições a partir do ensino técnico e superior (Obsie & Adem, 2018).

As técnicas de mineração e AM na PDA, com base nas suas notas, durante o período de formação, são ferramentas úteis para prever desempenhos bons e ruins e apoiar as metodologias de ensino e aprendizagem, identificando métodos instrucionais que se adequam aos alunos com informações de bases variadas (Belachew & Gobena, 2017; Muliono et al., 2020; Ofori et al., 2020; Soni et al., 2018). Entretanto, cabe ressaltar que a mineração de dados é um dos passos para a geração de um algoritmo de aprendizagem e teste para a predição (Akuma & Abakpa, 2021; Muliono et al., 2020).

2.4 Aprendizagem de Máquina (AM)

A AM é uma área da IA que automatiza a análise de dados, permitindo que algoritmos aprendam por meio da experiência ao executar tarefas sem programação explícita, portanto a AM é um tipo de IA (Meirelles, 2024).

A IA e a AM são altamente valorizados no mercado global. Como exemplo, a IA, de acordo com o World Economic Forum (2018), estimou que, até 2025, as funções redundantes diminuam de 15,4% para 9% da força de trabalho, enquanto as profissões emergentes devem crescer de 7,8% para 13,5%. Isso significa que, com base em tais estimativas, até 2030, a mudança na divisão do trabalho entre humanos e máquinas poderá substituir cerca de 85 milhões de empregos, enquanto surgem 97 milhões de novos papéis adaptados a essa nova divisão do trabalho entre humanos, máquinas e algoritmos.

A AM popularizou-se e ganhou adeptos que procuram a sua utilização nos mais diversos problemas. Na interpretação de Jantakun et al. (2022), os campos de AM, IA e ciência da computação (ou de dados) estão interligados e são frequentemente citados em artigos e outros meios de divulgação. Porém, cada área é distinta, com suas particularidades de aplicações. Na IA, são diversas possibilidades de aplicações, tais como na mobilidade com carros autônomos, no mercado financeiro, na educação, na área da saúde e, independentemente do setor, tem-se a AM (Ünal, 2020). O consenso é que AM é um ramo da IA, em que os algoritmos aprendem padrões comparando milhares de pontos de dados.

A AM é o campo de estudo mais procurado quando se trata de estudos sobre o desempenho do aluno. A aplicabilidade de AM prevê o desempenho do aluno usando o conhecimento aprendido durante o processamento (jornada do aluno). Essa aprendizagem requer grandes conjuntos de dados com definições, informações e categorização; tais entradas são regras que serão calculadas, gerando saídas sobre a eficiência do desempenho do aluno.

O trabalho de Tjandra et al. (2022) identificou que prever o desempenho do aluno deve iniciar com categorizações que orientem o trabalho de classificação e sugere cinco tarefas principais:

1. Identificação;
2. Classificação do aluno;
3. Modelagem e melhoria do aluno;
4. Sistema de recomendação;
5. Alerta precoce e indexação de sobrevivência.

A taxa de categorização é muitas vezes um recurso decisivo que um classificador deve utilizar com tempo e memória razoáveis para treinamento e aplicação do que foi especificado. Abstrair tal aplicação de algoritmo de AM não é tarefa simples e muitas vezes requer algum tipo de diagrama. Como exemplo, apresentam-se na Figura 1 as cinco etapas de Kishor et al. (2022), cuja metodologia indica passos envolvidos para um processo de AM.

No **Passo 1** da Figura 1, estão as fontes de entrada para a coleta de dados.

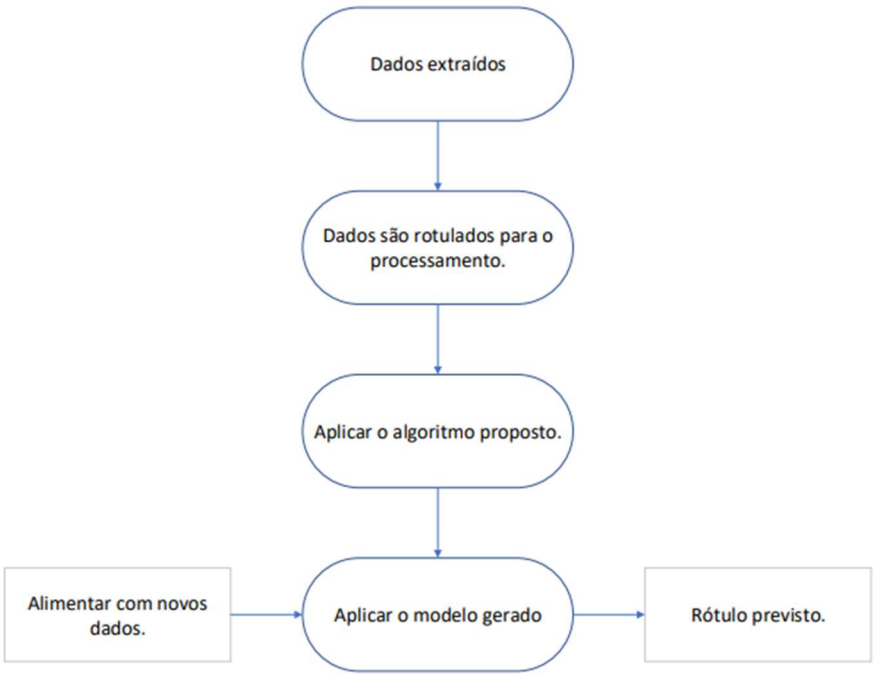
No **Passo 2**, os dados são processados para obter o conjunto de dados normalizados e *a posteriori* rotulá-los.

O **Passo 3** envolve o armazenamento dos resultados do passo 2 e o conjunto de dados treinados para o algoritmo de AM.

No **Passo 4**, o algoritmo de AM está pronto para projetar um algoritmo como saída que será utilizado com novas entradas.

Em alimentar com novos dados, o algoritmo de AM desenvolveu um algoritmo treinado ou um classificador que recebe novos dados e os rotula com predições.

Figura 1 – *Etapas e Componentes do Sistema Proposto para AM*



Fonte: Autor, adaptado de Kishor et al. (2022).

Um pouco diferente de Kishor et al. (2022), pode-se também descrever usando o ciclo de vida da AM de JavaTpoint (2021). O ciclo de vida da AM é um processo cíclico para construir um

projeto eficiente de AM. O principal objetivo do ciclo de vida é encontrar uma solução para o problema ou projeto.

O ciclo de vida da AM envolve sete etapas principais, que são dadas abaixo e que podem ter variações conforme a interpretação do pesquisador:

1. Coleta de dados;
2. Preparação dos dados;
3. Disputa de dados;
4. Analisar dados;
5. Treinar o modelo;
6. Testar o modelo;
7. Implantação.

No processo de ciclo de vida completo, para resolver um problema, cria-se um sistema de AM chamado "modelo", e esse modelo é criado fornecendo "treinamento". Mas, para treinar um modelo, precisa-se de dados, portanto o ciclo de vida começa pela sua coleta (Muliono et al., 2020; Ünal, 2020).

1. Coleta de dados

Nesta etapa, identificam-se as diferentes fontes de dados, como arquivos, banco de dados, internet ou dispositivos móveis. A quantidade e a qualidade dos dados coletados determinarão a eficiência da saída.

2. Preparação dos dados

Nesta etapa, primeiro, juntam-se todos os dados e, em seguida, executa-se a ordenação para correlações, tendências gerais e *outliers*.

3. Disputa de dados

A disputa de dados é o processo de limpeza e conversão de dados brutos em um formato utilizável.

4. Análise dos dados

O objetivo desta etapa é construir um modelo de AM para analisar os dados usando várias técnicas analíticas e revisar o resultado.

5. Treinamento do modelo

Usa-se os conjuntos de dados para treinar o modelo com vários algoritmos de AM. O treinamento é necessário para que ele possa entender os vários padrões, regras e recursos.

6. Teste do modelo

Nesta etapa, verifica-se a precisão do modelo fornecendo um conjunto de dados de teste para ele, geralmente, 30% do montante de registros.

7. Implantação

A última etapa do ciclo de vida da AM é a implantação, onde o modelo é utilizado no mundo real; também pode ser considerado como a elaboração do relatório final de um projeto.

2.4.1 Mineração de dados *versus* Aprendizagem de Máquina (AM)

A mineração de dados, segundo Zhang e Katsoulakis (2023), envolve a exploração e análise de grandes conjuntos de dados para descobrir padrões e relacionamentos úteis que possam ajudar as empresas a tomar decisões mais relevantes. É um processo exploratório que utiliza técnicas estatísticas e de visualização de dados para identificar tendências e padrões ocultos nos dados.

Por outro lado, a AM é uma abordagem computacional que permite que os sistemas "aprendam" a partir dos dados, sem serem explicitamente programados. Em vez disso, os algoritmos de AM usam algoritmos estatísticos para identificar padrões nos dados de treinamento e, em seguida, aplicam esses algoritmos para fazer previsões ou tomar decisões.

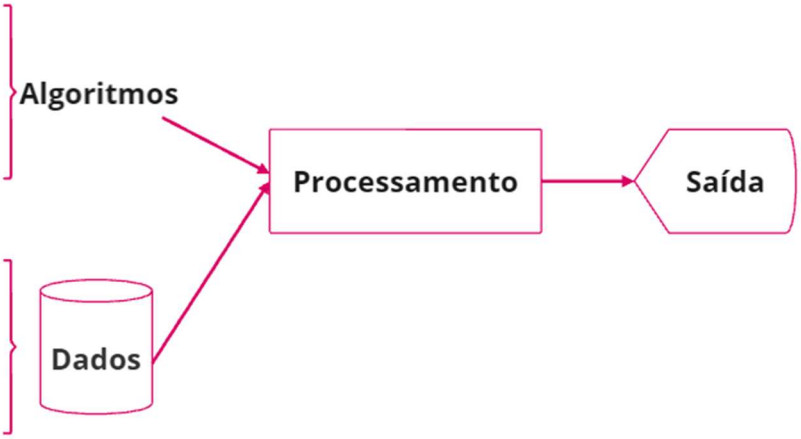
Em resumo, a principal diferença entre a mineração de dados e a AM é que a primeira é uma técnica exploratória que ajuda a identificar padrões nos dados, enquanto a segunda é uma técnica de modelagem preditiva que usa esses padrões para fazer previsões ou tomar decisões.

2.4.1.1 Processo de mineração

Tradicionalmente, a mineração de dados tem um fluxo de extrair dados precisos e confiáveis, transformá-los e programar cargas para mantê-los atualizados para posteriormente divulgar aos diversos interessados. No entanto, a alta precisão não é o único fator crítico para o sucesso. Além disso, Alamri e Alharbi (2021) destacam que é importante confiar no algoritmo, onde confiança significa que as partes interessadas podem entender as razões por trás dos dados. Esse processo se repete todos os dias, realizando uma foto das informações históricas.

Em AM, o fluxo pode iniciar da mesma forma, ou mesmo aproveitar os dados transformados, entretanto o fluxo é contínuo pelo fato de ser um aprendizado, assim o seu resultado gera um algoritmo. Esse algoritmo pode ser o resultado ou não, desse modo se inicia novamente o ciclo de aprendizagem. A Figura 2 descreve o modelo tradicional de mineração de dados.

Figura 2 – *Mineração Tradicional de Dados (Simplificado)*

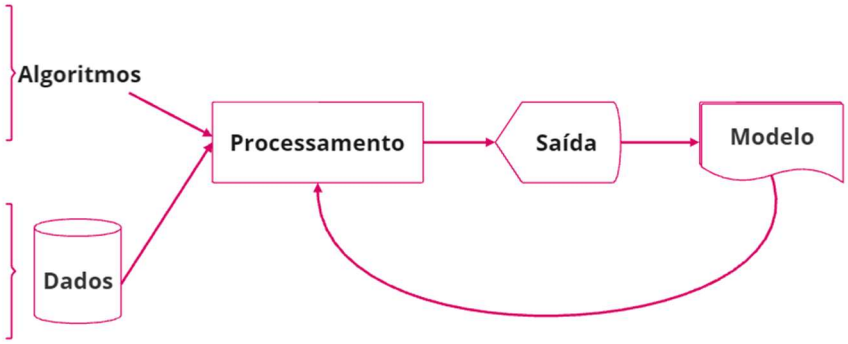


Fonte: Elaboração própria.

2.4.1.2 Processo de AM

Como dito anteriormente, o modelo de AM é ligeiramente semelhante e com atividades que fazem o papel da retroalimentação do processo. Na Figura 3, pode-se ver as diferenças entre ambas.

Figura 3 – *Processo de Aprendizagem de Máquina (Simplificado)*



Fonte: Elaboração própria.

2.4.2 Tipos de Aprendizagem de Máquina (AM)

Para Karthikeyan e Kavipriya (2017, p. 936), no aprendizado Supervisionado, "um modelo é construído antecipadamente para a análise e, em seguida, aplica o algoritmo aos dados para estimar os parâmetros do modelo". Esse tipo de aprendizado tem como base a regressão básica e

a classificação, onde uma entrada é sugerida, por exemplo, a imagem de um automóvel. Muito semelhante a Sevetlidis et al. (2024), que interpretam o aprendizado Supervisionado como treinar um modelo para prever o rótulo correto da classe para cada amostra em um conjunto de dados de treinamento.

As diferenças de fabricante (atributos ou rótulos), modelo, ano, cor e *design* variam, porém o algoritmo aprende o básico, tais como as quatro rodas, pneus, para-brisa, retrovisores, entre outros itens da imagem. A aprendizagem Supervisionada tem impulsionado diversos projetos em organizações como Amazon, Mercado Livre, Walmart, entre outras do setor varejista com operações logísticas gigantes. O reconhecimento de produtos por câmeras os leva para o lugar certo, sem depender de intervenção humana.

Não há um único tipo de AM que seja mais utilizado, pois diferentes tipos de aprendizado são adequados para diferentes tipos de problemas e dados. Assim, seguem possíveis aplicações para o tipo mais comum, o Supervisionado, na Tabela 1.

Tabela 1 – *Exemplos para o Uso do Tipo Supervisionado*

Tipo	Aplicação
Classificação	Categorização de dados em classes predefinidas, como detecção de *spam* em *e-mails*, classificação de imagens, diagnóstico médico, entre outros.
Regressão	Estimação de valores contínuos, como prever preços de imóveis, prever a demanda por um produto, prever a quantidade de vendas, entre outros.
Clusterização	Agrupamento de dados em *clusters* com base em suas características, como agrupamento de clientes em diferentes segmentos, agrupamento de notícias em diferentes tópicos, entre outros.
Análise de Sentimento	Classificação de texto em positivo, negativo ou neutro, como análise de comentários de clientes, análise de sentimentos em redes sociais, entre outros.
Reconhecimento de Padrões	Identificação de padrões em dados, como reconhecimento de voz, reconhecimento de escrita manual, detecção de fraude em cartões de crédito, entre outros.

Nota. Os números contínuos são com casas decimais e discretos sem casas decimais.
Fonte: Consolidado de pesquisas.

A aprendizagem Não Supervisionada é menos eficiente, cara e demorada, por não haver uma entrada (rótulos) para reconhecimento do padrão inicial, portanto o algoritmo precisa aprender a combinar os dados para chegar a alguma informação. Aplicações que sugerem perfis de pessoas que combinam, por exemplo, utilizam esse tipo de algoritmo, porém sem intervenção no resultado. Abdi (2016) interpreta o aprendizado Não Supervisionado “quando não se tem nenhuma variável alvo ou resultado para prever ou estimar”. Ele é usado para agrupar a população em diferentes grupos, que é amplamente utilizado para segmentar clientes para intervenção

específica. Na interpretação de Bonab et al. (2024), existem dois pontos de dados: um pequeno conjunto de dados rotulados e um conjunto mais extenso de dados não rotulados.

Esse tipo de aprendizado pode ser comparado com a forma de aprender de uma criança, ou seja, pela observação e imitação das ações de seus pais. As abordagens desse tipo podem ser combinadas para atenderem novas aplicações; alguns exemplos estão na Tabela 2.

Tabela 2 – *Exemplos de Aplicações Utilizando o Tipo Não Supervisionado*

Aplicações	Descrição	Exemplos práticos
Clustering (Agrupamento)	Divisão de dados em grupos (*clusters*) com base em similaridades ou dissimilaridades entre eles.	Segmentação de clientes, detecção de anomalias, identificação de padrões em dados não estruturados.
Redução de Dimensionalidade	Redução do número de variáveis em um conjunto de dados, mantendo a maior parte da informação relevante.	Visualização de dados em 2D ou 3D, compressão de imagens, pré-processamento de dados para algoritmos de AM.
Associação	Identificação de itens frequentemente associados em um conjunto de dados.	Recomendação de produtos, análise de cesta de compras, identificação de relações entre proteínas em biologia.
Detecção de Anomalias	Identificação de pontos de dados que são significativamente diferentes do restante do conjunto de dados.	Detecção de fraudes, diagnóstico médico, detecção de falhas em sistemas mecânicos.
Aprendizado por Competição	Uso de redes neurais para aprender a representação de dados em uma tarefa de competição interna.	Aprendizado de representações de imagens, reconhecimento de fala, tradução automática.
Aprendizado por Associação	Aprendizado de associações entre diferentes elementos de dados, sem a necessidade de um rótulo de saída.	Modelagem de tópicos, análise de sentimentos, sumarização de texto.

Nota. As combinações em diferentes abordagens são os principais diferenciais deste tipo Não Supervisionado. Fonte: Elaboração própria, com base em consolidado de pesquisas.

O algoritmo de aprendizado **Semi-supervisionado** usa os dados rotulados para aprender e fazer previsões, assim como no aprendizado Supervisionado. Em seguida, ele usa essas previsões para atribuir rótulos aos dados não rotulados, que são usados para ajustar o algoritmo. Esse processo é repetido várias vezes até que o algoritmo seja treinado com uma precisão satisfatória.

A principal vantagem do aprendizado semi-supervisionado é que ele pode ser aplicado a conjuntos de dados muito grandes, exemplos na Tabela 3, onde rotular todos os dados seria muito caro e demorado. Além disso, a incorporação de dados não rotulados pode melhorar a precisão do algoritmo, uma vez que esses dados podem fornecer informações adicionais sobre a distribuição

dos dados e ajudar a evitar o *overfitting*[3]. No entanto, o aprendizado semi-supervisionado pode ser desafiador, uma vez que é necessário que os dados rotulados sejam representativos do conjunto de dados como um todo e que a técnica de rotulagem dos dados não rotulados seja precisa e confiável.

Tabela 3 – *Exemplos de Aplicações Utilizando o Tipo semi-supervisionado*

Aplicações	Descrição
Processamento de Linguagem Natural	Classificação de texto, tradução automática, resumo de texto e reconhecimento de entidades nomeadas.
Visão Computacional	Detecção de objetos, segmentação de imagem, reconhecimento facial e reconhecimento de voz.
Análise de Redes Sociais	Identificação de comunidades, detecção de influenciadores e previsão de tendências.
Aprendizado por Reforço	Jogos, robótica e navegação.
Reconhecimento de Fala	Transcrição de áudio, identificação de alto-falantes e síntese de fala.

Nota. Podem existir outras combinações para o uso desse tipo semi-supervisionado. Fonte: Elaboração própria, com base em consolidado de pesquisas.

No tipo aprendizado por reforço, na visão de Abdi (2016, p. 2), "a máquina é treinada para tomar decisões específicas". A máquina é exposta a um ambiente onde se treina continuamente por tentativa e erro. Essa máquina aprende com a experiência passada e tenta capturar o melhor conhecimento possível para tomar decisões de negócios. Na abordagem de Yang e Xiao (2024), é possível que o aprendizado por reforço profundo aprenda e otimize automaticamente as estratégias de controle do sistema, fornecendo novas ideias e métodos para resolver problemas de controle de otimização em processos.

Uma outra maneira seria o algoritmo receber recompensas por uma determinada ação, que foi submetida ao usuário. O aprendizado por reforço registra a relação da ação e recompensa, aprendendo e decidindo novas ações, por meio desse conceito de recompensa cumulativa.

O uso desse aprendizado por reforço pode ser observado após assistir a um vídeo no YouTube ou Instagram, onde são recomendados títulos semelhantes, porém, se não terminar de assistir à recomendação, o algoritmo entenderá que deverá fazer outra abordagem no reforço. A sua aplicação segue em IA, robótica, jogos digitais, com outros exemplos na Tabela 4.

[3] *Overfitting* é um erro de modelagem em estatística que ocorre quando uma função está muito alinhada a um conjunto limitado de pontos de dados. Como resultado, o modelo é muito sensível aos dados de treinamento e não é capaz de generalizar o conhecimento para dados novos. Isso significa que o modelo pode ser muito preciso para prever dados de treinamento, mas não pode prever dados externos com precisão.

Tabela 4 – *Aplicações Possíveis para o Tipo por Reforço*

Aplicações	Descrição	Exemplos de práticos
Jogo	Aprendizado por meio de interações com jogos, em que o ator recebe recompensas ou penalidades por suas ações.	Jogos de tabuleiro, jogos de vídeo e jogos de cassino.
Robótica	Treinamento de robôs para realizar tarefas complexas, com base em recompensas ou penalidades fornecidas pelo ambiente.	Manipulação de objetos, navegação autônoma e montagem de peças.
Controle de Processos	Controle de processos em tempo real, ajustando as ações do ator com base em *feedbacks* de desempenho.	Controle de tráfego, controle de robôs em linhas de produção e controle de temperatura em ambientes industriais.
Simulação	Treinamento de atores em ambientes simulados, permitindo que eles aprendam em um ambiente seguro antes de serem implantados em ambientes reais.	Simulação de ambientes urbanos, simulação de processos industriais e simulação de tráfego.
Saúde	Treinamento de atores para tomar decisões em saúde com base em *feedbacks* de desempenho.	Diagnóstico médico, tratamento de doenças e otimização de recursos hospitalares.
Educação	Treinamento de atores para fornecer ensino personalizado e adaptativo, com base em *feedbacks* de desempenho dos alunos.	Ensino de línguas, ensino de matemática e ensino de habilidades práticas.

Nota. Essas combinações são semelhantes ao tipo Não Supervisionado, com possíveis diferentes abordagens. Fonte: Elaboração própria, com base em consolidado de pesquisas.

O tipo mais recente, e vastamente usado pelo algoritmo GPT, é o aprendizado profundo. O aprendizado profundo (ou *deep learning*) é um subcampo da AM que se baseia em Redes Neurais Artificiais (RNA) profundas para aprender a partir de dados (Jawed et al., 2024). As redes neurais são estruturas matemáticas inspiradas no cérebro humano que permitem que as máquinas processem informações e façam previsões.

No aprendizado profundo, essas redes neurais são compostas por muitas camadas ocultas (daí o nome "profundo"), permitindo que o algoritmo capture representações complexas e hierárquicas dos dados. Cada camada da rede neural extrai características ou padrões específicos dos dados de entrada e passa essa informação para a próxima camada, até que a rede produza uma saída final.

As redes neurais profundas são treinadas por meio de um processo chamado de *backpropagation*[4], no qual o algoritmo é alimentado com exemplos rotulados e ajusta os pesos das conexões entre as camadas para minimizar o erro de previsão. Com o tempo, a rede neural aprende

[4] O *backpropagation* é um algoritmo de otimização utilizado no treinamento de redes neurais artificiais profundas.

a extrair automaticamente características relevantes dos dados e a fazer previsões precisas, mesmo em dados nunca vistos antes.

O aprendizado profundo é usado em uma ampla variedade de aplicações de AM, incluindo visão computacional, processamento de linguagem natural, reconhecimento de fala e muitas outras. Essa técnica tem sido especialmente boa em tarefas que envolvem grandes quantidades de dados não estruturados, como imagens, vídeos e áudio.

São diversas aplicações que se beneficiam do aprendizado profundo; exemplos podem ser vistos na Tabela 5.

Tabela 5 – *Aplicações Possíveis para o Aprendizado Profundo*

Aplicações	Exemplos de aplicações
Visão computacional	Reconhecimento facial, detecção de objetos em imagens e vídeos, segmentação de imagens, classificação de imagens.
Processamento de linguagem natural	Tradução automática, sumarização de texto, análise de sentimento, reconhecimento de fala, geração de texto.
Análise de dados	Previsão de séries temporais, classificação de dados, agrupamento de dados, análise de redes sociais.
Robótica	Controle de robôs autônomos, navegação de robôs, detecção de objetos em tempo real.
Jogos	Agentes inteligentes em jogos, simulação de ambientes virtuais.
Saúde	Diagnóstico médico, análise de imagens médicas, descoberta de novas drogas.
Automotivo	Carros autônomos, sistemas de assistência ao motorista.
Finanças	Detecção de fraudes, previsão de investimentos, análise de risco.

Nota. O aprendizado profundo é muitas vezes utilizado em conjunto com outros tipos de AM para se chegar a um algoritmo eficiente. Na verdade, é comum combinar diferentes técnicas de AM para abordar diferentes aspectos de um problema e obter um algoritmo mais preciso. Fonte: Elaboração própria, com base em consolidado de pesquisas.

Esses são os tipos de AM envolvidos no treinamento em um conjunto de dados rotulados ou não rotulados para encontrar padrões ou estrutura de aprendizado. Na Tabela 6, apresenta-se um resumo dos tipos de AM citados nos parágrafos anteriores.

Tabela 6 – *Resumo dos Tipos de AM*

Tipos	Descrição
Supervisionado	Todos os dados são rotulados, e o algoritmo aprende a prever a saída a partir dos dados de entrada.
Não Supervisionado	Todos os dados não estão rotulados, e o algoritmo aprende a estrutura inerente a partir dos dados introduzidos.
semi-supervisionado	Combina elementos do aprendizado Supervisionado e Não Supervisionado. Em vez de ter um conjunto completo de dados rotulados (Supervisionado) ou não rotulados (Não

Tipos	Descrição
	Supervisionado), o aprendizado semi-supervisionado usa um conjunto de dados com uma pequena quantidade de dados rotulados e uma grande quantidade de dados não rotulados.
Aprendizado por Reforço	Reforçar a aprendizagem é uma área de AM preocupada com a forma como o agente de *software* deve tomar medidas num ambiente, de modo a maximizar alguma noção de recompensa.
Aprendizado Profundo	Uma subárea que utiliza RNA profundas para aprender a partir de dados. As redes neurais profundas são capazes de aprender automaticamente a extrair características complexas dos dados e produzir previsões precisas em tarefas de aprendizado Supervisionado e Não Supervisionado

Nota. Esta tese foi desenvolvida no tipo Supervisionado. Fonte: Elaboração própria, com base em consolidado de pesquisas.

2.4.3 *Softwares* habilitadores

Pode-se dizer que os recursos necessários para habilitar a AM são *softwares* especialistas com bibliotecas criadas especialmente para essa finalidade. Há ferramentas (*softwares*) proprietárias pagas e/ou de acesso livre que foram criadas especialmente para esse fim de AM. Incluem-se aqui não apenas *softwares*, mas também ambientes computacionais capazes de processar grande massas de dados minerados de diversos meios.

Para Tjandra et al. (2022), o uso de *software* tem como base o objetivo do pesquisador, podendo, inclusive, ser categorizado por manipulação de dados, análise de algoritmo e visualização de dados. Faz muito sentido que se entenda bem a aplicação do uso de *software* antes de se aventurar no aprendizado ou investimento antecipado, pois há muitas opções, e algumas são restritas a certas funcionalidades.

As *big techs* mais conhecidas lançaram seus produtos para atender a demanda por AM e, no Apêndice A, citam-se algumas delas. Uma observação relevante sobre adotar uma ferramenta é compreender o seu funcionamento e abrangência em relação aos algoritmos que se queira utilizar.

2.4.4 Algoritmos de AM

Para Kumar et al. (2017 como citado em Ofori et al., 2020), alguns algoritmos incluem árvores de decisão, rede neural, classificador bayesiano-vizinho mais próximo, máquina vetorial de apoio, floresta aleatória, regressão logística, análise discriminante linear, regressão múltipla e mapa auto-organizado.

Os algoritmos de AM são uma parte fundamental do campo de IA (Palkar et al., 2024). Eles são projetados para aprender padrões e estruturas de dados, permitindo que possam fazer previsões e tomar decisões com base em novas informações. Esses algoritmos podem ser divididos em tipos e técnicas de aprendizagem; o Apêndice B lista um resumo contendo os principais algoritmos.

2.4.5 Iniciando na Aprendizagem de Máquina (AM)

A AM está em constante crescimento e têm se tornado cada vez mais acessível a um público mais amplo. Entretanto, antes de mergulhar nas técnicas mais avançadas, é importante ter um entendimento dos conceitos básicos, como classificação e regressão.

A classificação e a regressão são duas tarefas importantes na AM. Como afirma Mitchell (1977, p. 2), "Classificação é a tarefa de prever um rótulo de classe discreto, como, por exemplo, se um e-mail é spam ou não. Regressão é a tarefa de prever um valor contínuo, como o preço de uma casa".

Há muitos algoritmos diferentes que podem ser usados para classificação e regressão. Murphy (2012, p. 553) escreve: "Alguns dos algoritmos mais comuns incluem árvores de decisão, máquinas de vetores de suporte e redes neurais".

A escolha do algoritmo depende do problema específico que está sendo resolvido. Goodfellow et al. (2016) afirmam: Por exemplo, as árvores de decisão costumam ser uma boa opção para problemas de classificação simples, enquanto as máquinas de vetor de suporte ou as redes neurais podem ser uma opção melhor para problemas mais complexos.

A classificação e a regressão tratam de problemas fundamentais da AM e servem como base para muitas outras técnicas e algoritmos. A classificação é o processo de atribuir uma categoria ou classe a um conjunto de dados, enquanto a regressão é o processo de prever um valor numérico com base em um conjunto de dados de entrada. Dominar esses conceitos é importante para entender a lógica por trás da AM.

2.4.6 Algoritmos de classificação

Os algoritmos de classificação são aplicados na AM para associar um conjunto de observações (linhas ou registros) sob a mesma caracterização. Esses algoritmos são do tipo Supervisionado e predizem resultados do tipo de uma classe dentro de categorias com um conjunto de valores binários.

Anteriormente aos valores binários tem-se os rótulos das categorias, por exemplo: **spam**: [“spam”, “não spam”] ou **cães**: [“poodle”, “beagle”, “pastor belga”, “pastor alemão”] ou de **frutas**: [“laranja”, “goiaba”, “banana”] ou **diagnóstico de saúde sobre tumores**: [“maligno”, “benigno”] como também o resultado de uma **operação de cartão de crédito**: [“fraude”, “não fraude”]. A conversão dessas observações (registros) para binário ou números inteiros é um trabalho essencial para o funcionamento dos algoritmos.

Os algoritmos de classificação utilizam os atributos de uma entidade (ou tabela de um banco de dados) para prever a sua classe (Pintea et al., 2023). Portanto, se o algoritmo de AM for aplicado a uma base de dados de estudantes de uma escola, tem-se a separação de “professores”, “assistentes administrativos”, “alunos” e, com base nos atributos de formação, cargo, idade, vínculo de cada rótulo, pode-se tentar predizer a chance de ser um “estudante”.

Desta forma, pode-se dizer que o algoritmo aprende combinações de atributos pertencentes a classes em categorias específicas da base de dados (DataSet), segundo Chung (2020).

2.4.7 Algoritmos de regressão

Para Pintea et al. (2023), a regressão é uma ferramenta que busca modelar relações entre variáveis dependentes e independentes por meio de métodos estatísticos. Não é incomum achar pessoas que pensam em tomar decisões com base em predição, e isso, em AM, é feito por algoritmos de regressão.

2.4.8 Variável: independente (x) e dependente (y)

Uma variável independente, normalmente representada pela letra x, caracteriza uma grandeza que está sendo manipulada durante o processo e que não sofre influência de outras variáveis. Já a variável dependente (meta), normalmente representada pela letra y, caracteriza valores que estão diretamente associados à variável independente.

Os problemas de regressão buscam prever um valor numérico, por exemplo, prever o preço de uma casa com base no endereço, ou valor de ações no mercado, ou as notas de alunos em uma disciplina, como neste estudo, em que o atributo meta é um valor numérico. (Kubat, 2017)

Importante reforçar que as variáveis independentes socioeconômicas, entre outras do ambiente, não são consideradas nos algoritmos citados pelos autores, deixando margem para se pensar que o desempenho do aluno não pode ser considerado apenas pelos dados históricos armazenados em bancos de dados da escola. Em concordância, estão Ofori et al. (2020), quando

citam que o desempenho do aluno é influenciado por fatores socioeconômicos como renda familiar, nível de educação dos pais e situação profissional do aluno ou pais, mas não são considerados ao testar a precisão de vários algoritmos de AM na previsão do desempenho do aluno. Na mesma linha, tem-se o trabalho de Ahmad et al. (2022) sobre a influência por atributos emocionais, como frustração, ansiedade, estresse, expectativa excessiva dos pais e relacionamento dos pais, gerando um desempenho negativo do aluno.

2.5 Consolidação da revisão bibliográfica sobre AM

A revisão bibliográfica realizada não indicou um algoritmo único para a predição do desempenho do aluno. Assim, atingir os objetivos específicos, iniciando em predizer tal desempenho, considerando um vasto volume de dados, possivelmente será realizado por meio de testes de AM até se obterem os algoritmos adequados. Outro aspecto da revisão bibliográfica é que não se achou um consenso sobre como melhorar o desempenho do aluno após a previsão. Da mesma forma, não houve nenhuma menção sobre teorias que pudessem fundamentar estudos ou pesquisas de melhorias para o desempenho do aluno.

Para Yadav e Pal (2012 como citado em Ofori et al., 2020), grande parte da literatura se concentra em prever o desempenho do aluno na resolução de problemas ou na conclusão de cursos. As citações sobre algoritmos que tratam do desempenho do aluno não garantem o melhor entre eles.

Alguns algoritmos de AM, como árvores de decisão, RNA, fatoração de matrizes, filtros colaborativos e algoritmos gráficos probabilísticos, têm sido aplicados para desenvolver algoritmos de previsão (Lakkaraju et al., 2015). Evidencia-se um grande esforço, há mais de uma década, na busca pela previsão do desempenho, sendo o ponto principal aqui a escolha de algoritmos específicos, com a possibilidade de testes conforme o problema, para se encontrar o melhor algoritmo.

Assim, Ofori et al. (2020) insistem que é importante testar a precisão de vários algoritmos de AM que predizem com alta precisão o desempenho do aluno e aquele que é mais apropriado para melhorar o resultado do aprendizado, considerando os efeitos, se possível, dos fatores socioeconômicos e demográficos do aluno no desempenho acadêmico.

Os estudos de Soni et al. (2018) apontam que a aplicação de algoritmos de AM para prever o desempenho dos alunos, com base em suas informações básicas e seu desempenho no período, provou ser uma ferramenta útil para prever desempenhos ruins e bons em vários níveis de ensino. Isso sugere que a AM pode ser uma vantagem sobre as demais formas de algoritmos “mais

tradicionais", porém com um diferencial sobre a capacidade de manipular uma vasta quantidade de informações e a geração de novos *insights* na medida em que novas variáveis são incorporadas ao aprendizado. Isso faz sentido, tendo em vista que a AM realmente tem por finalidade esse objetivo, ou seja, aprender por meio de algoritmos ou outros padrões para a tomada de decisões, fato alinhado com Hussain et al. (2018 como citado em Ofori et al., 2020).

A capacidade de prever o desempenho dos alunos é crucial no sistema educacional, considerando que o principal objetivo é fornecer a melhor educação (Ünal, 2020; Yanamandra & Praasd, 2022). No entanto, não é claro qual algoritmo de AM é melhor para prever o desempenho do aluno e qual é o ideal para melhorar o resultado do aprendizado (Kumar et al., 2017). Faz sentido o dito anterior de que esses algoritmos de aprendizagem são utilizados para o "conhecimento oculto", em relação a grandes quantidades de dados.

O Apêndice C consolida as referências bibliográficas, com autores, sobre algoritmos de AM separadas por previsão do desempenho (foco no aluno), previsão de máquina (foco no algoritmo) e mineração de dados (foco em tecnologia conhecida).

A razão dessa separação, durante o levantamento da revisão bibliográfica, foi entender a motivação do autor, ou seja, quem estava interessado em entender a previsão do desempenho, podendo ser para qualquer finalidade e de preferência do aluno, e aqueles que estavam focados em compreender como os algoritmos funcionavam, denominados aqui previsão de máquina. Por fim, os autores que tinham a fundamentação da mineração de dados e que buscaram algum ajuste técnico para se obter acurácia em resultados.

Entre 2011 e 2016, alguns autores fundamentaram seus trabalhos no entendimento da previsão do desempenho sem testar um algoritmo. Claramente, estavam interessados em desmistificar o que era prever desempenho, mas sem um foco específico, nem mesmo citando um algoritmo em especial, porém embasados em conceitos computacionais que viriam a se tornar algoritmos usáveis e muito conhecidos hoje em dia.

De fato, predominou o estudo do conceito de previsão de desempenho e/ou de máquina, sem considerar a construção ou uso de um algoritmo para uso prático.

Os autores foram ordenados por ano de publicação para se entender como foram interpretados e aplicados os conceitos de previsão de desempenho e de máquina por meio de algoritmos, quando aplicável, em cada período.

Pode-se observar que as aplicações práticas, com os respectivos resultados, foram iniciadas em 2017, indicando que o tema, no aspecto prático, estava criando corpo.

A coluna denominada algoritmo identifica quais deles foram estudados e/ou utilizados pelos autores para a identificação da acurácia.

A acurácia, em AM, é uma medida de desempenho que indica a precisão do algoritmo em fazer previsões corretas em relação às saídas esperadas. Por exemplo, se um modelo fizer 100 previsões e 90 delas estiverem corretas, a precisão será de 90%.

É comumente definida como a proporção de previsões corretas em relação ao total de previsões feitas pelo algoritmo. A acurácia é uma das métricas mais utilizadas para avaliar a *performance* de um algoritmo de AM, mas é importante citar que sua utilização pode ser limitada em problemas com classes desbalanceadas ou custos assimétricos de erro[5]. Além disso, a acurácia não leva em consideração a magnitude dos erros, ou seja, dois algoritmos com a mesma acurácia podem ter desempenhos diferentes em relação à precisão de suas previsões. Por isso, é importante avaliar outras métricas complementares para uma análise completa do desempenho do algoritmo.

A fundamentação teórica trouxe reflexões interessantes sobre como foram usados os algoritmos de AM. Como exemplo, o trabalho de Shahiri et al. (2015), que pesquisaram autores os quais validaram algoritmos com resultados para o que chamaram de atributos, ou seja, quais algoritmos seriam mais adequados para serem aplicados em um estudo sobre os atributos de alunos.

O Apêndice D descreve os atributos e quais algoritmos seriam mais adequados, na visão de Shahiri et al. (2015). A estruturação por atributos sobre o que se espera obter de resultados em relação ao aluno é um bom ponto de partida, não apenas pelo fato de orientar o pesquisador ou o administrador escolar, mas por iniciar o projeto como uma estrutura de requisitos em que se pode compreender melhor a formação do aluno e discuti-la *a posteriori*. Porém, Agrawal et al. (2017) divergem dessa visão argumentado que classificadores mostram diferentes precisões dependendo dos diversos algoritmos usados nele. Portanto, faz-se necessário o teste em algoritmos diferentes para se obter a melhor eficiência.

2.6 Bases teóricas

2.6.1 Teoria da Expectativa da Motivação (TEM)

[5] Custos assimétricos de erro referem-se à situação em que os custos de um erro de classificação dependem do tipo de erro cometido. Em outras palavras, os custos associados a cometer um erro de falsos positivos (previsão incorreta de uma classe positiva) podem ser diferentes dos custos associados a cometer um erro de falsos negativos (previsão incorreta de uma classe negativa).

A TEM, conforme Bandhu et al. (2024), sugere que as pessoas buscam maximizar o prazer e minimizar o desconforto, adotando comportamentos que levem aos melhores resultados ou recompensas. No contexto educacional, a TEM indica que os alunos são mais motivados a melhorar seu desempenho quando percebem valor nos resultados de seus esforços.

A teoria baseia-se em três variáveis principais: expectativa, instrumentalidade e valência, essas variáveis formam elos na cadeia de motivação, e qualquer fragilidade em um desses elos pode comprometer o resultado esperado, exigindo ajustes para restabelecer a motivação.

Expectativa (E): Refere-se à crença de que o esforço leva à melhoria no desempenho. Se o aluno não percebe um impacto positivo de seu esforço, ele tende a não se engajar novamente.

- Ações recomendadas: Verificar se o aluno dispõe de ferramentas e tempo adequados, oferecer monitoria e incentivar o acesso a tutores.
- Aplicação na tese: O uso de algoritmos de AM pode ajudar a prever se o esforço atual resultará em melhorias futuras, atendendo à variável expectativa.

Instrumentalidade (I): Está relacionada à percepção de que o desempenho resulta em recompensas. Se o aluno não perceber que seu esforço traz recompensas, sua motivação diminui, mesmo que ele tenha aprendido novas habilidades.

- Ações recomendadas: Definir recompensas claras e tangíveis, garantindo que o aluno compreenda seu valor e o caminho para alcançá-las.
- Aplicação na tese: O uso de selos (*badges*), físicos ou digitais, é sugerido como uma forma de motivar e recompensar o progresso dos alunos, especialmente em ambientes de aprendizagem *on-line*.

- **Valência (V)**: Representa o valor atribuído às recompensas. Quanto mais o aluno valoriza a recompensa, maior será sua satisfação com o esforço investido.
  - Ações recomendadas: Oferecer reconhecimento significativo, adaptado aos valores e interesses dos alunos.
  - Aplicação na tese: A atribuição de selos adicionais pode ser uma forma interessante de reconhecer publicamente o esforço dos alunos, estimulando engajamento e autoestima.

A equação motivacional proposta pela TEM é:

$$FM = E \times I \times V$$

Se qualquer um dos fatores for zero, a motivação será comprometida.

2.6.1.1 Vantagens e desvantagens da TEM

Vantagens:

- Conexão entre motivação e satisfação.
- O aumento da expectativa de recompensa tende a fortalecer a motivação.
- Foco nas recompensas e na realização de metas.
- Incentiva a relação entre esforço e melhoria de desempenho.

Desvantagens:

- Supõe que o esforço sempre levará à recompensa desejada.
- Não considera fatores como capacidade de aprendizado ou carga de estudo.
- Se o desempenho for inatingível, a motivação é prejudicada.

Embora a TEM ofereça uma abordagem útil para a motivação, ela não deve ser a única teoria utilizada. A TDA é apresentada como complemento para tratar questões relacionadas ao comportamento e aprendizado.

2.6.1.2 Integração com a literatura

A integração entre a TEM e a EDM propõe um modelo que utiliza a análise de dados educacionais para compreender as interações dos alunos e os fatores que influenciam seu desempenho. Ao unir essas abordagens, é possível identificar expectativas, alinhar resultados e ajustar intervenções pedagógicas que aumentem a motivação dos alunos.

Essa integração não apenas ajuda a prever o desempenho dos alunos, mas também personaliza estratégias de ensino, ajustando recompensas de acordo com as motivações individuais. Como resultado, a aplicação conjunta de EDM e TEM contribui para uma melhoria no desempenho acadêmico e para a retenção dos alunos.

2.6.2 Teoria da Dependência do Aprendizado (TDA)

A TDA, desenvolvida por B. F. Skinner em 1971, baseia-se em estudos psicológicos que explicam como as pessoas aprendem novas habilidades e comportamentos por meio de estímulos externos (Rutherford, 2017). A teoria sugere que o comportamento é moldado pela resposta a estímulos no ambiente, o que é particularmente relevante para identificar alunos em risco e intervir em situações de possível evasão escolar (Ahmad et al., 2022). Em ambientes educacionais, essa

teoria é aplicada ao monitoramento e controle de comportamentos para melhorar o desempenho dos alunos, especialmente em escolas que priorizam o apoio a alunos que necessitam de mais atenção (Yanamandra & Prasad, 2022).

Segundo Skinner, a aprendizagem ocorre quando há uma mudança no comportamento em resposta a um estímulo. Essa mudança pode ser provocada por recompensas específicas, que condicionam o indivíduo a repetir o comportamento desejado. Diferentemente de outras abordagens behavioristas, o condicionamento operante de Skinner permite que o indivíduo emita respostas proativamente, em vez de apenas reagir a estímulos externos.

No contexto educacional, a TDA pode ser explorada por meio do uso de reforços positivos, aplicados a ações específicas, como uma estratégia para lidar com previsões de desempenho acadêmico, seja para notas baixas ou altas.

2.6.2.1 Reforço positivo

O reforço positivo é um componente central da Teoria do Estímulo-Resposta. Qualquer estímulo que fortaleça uma resposta específica é considerado um reforçador, podendo ser materializado em percentuais positivos, elogios verbais, ou sentimentos de satisfação e realização. A teoria também reconhece os reforços negativos, que, embora não impliquem punição, resultam na redução de respostas indesejadas. Focar reforços positivos é essencial para promover mudanças comportamentais duradouras em diversos contextos educacionais.

B.F. Skinner propôs um processo em cinco etapas para implementar mudanças de comportamento:

1. **Estabelecimento de metas comportamentais:** Definir claramente os comportamentos que se deseja observar nos alunos. Por exemplo, um professor pode querer que os alunos se acalmem rapidamente no início da aula.
2. **Determinação de formas razoáveis de reforço:** Em vez de utilizar punições, o professor deve identificar como recompensar comportamentos desejáveis, evitando reforços negativos.
3. **Seleção de técnicas para alterar comportamentos:** Decidir como aplicar os reforços positivos e negativos identificados. Por exemplo, recompensar alunos de alto desempenho com estrelas ou pontos pode ser uma técnica aplicável.
4. **Aplicação e registro dos resultados:** Implementar as técnicas escolhidas e monitorar o impacto no comportamento dos alunos, ajustando conforme necessário.

5. **Avaliação e ajuste:** Se uma técnica não for boa, ela deve ser modificada ou substituída por outra que melhor se adapte ao contexto dos alunos.

Estudos mostram que a punição tem pouco efeito positivo em longo prazo e pode, na verdade, reforçar comportamentos indesejados. Portanto, é importante que os reforços sejam pontuais, criativos, claros e aplicados com regularidade, evitando associações com punição.

2.6.2.2 Integração com a literatura

A integração da TDA com a BNCC oferece um *framework* para alinhar as estratégias de reforço positivo com as competências e habilidades definidas pela BNCC. A TDA, ao focar a mudança comportamental por meio de reforços, pode ser aplicada para incentivar os comportamentos esperados dentro das diretrizes curriculares da BNCC.

Por exemplo, a BNCC enfatiza o desenvolvimento de competências socioemocionais e cognitivas, que podem ser reforçadas utilizando as estratégias propostas por Skinner. A aplicação dos reforços positivos, alinhada às metas da BNCC, pode fortalecer a aprendizagem dos alunos ao promover comportamentos que estão diretamente relacionados ao desenvolvimento das competências essenciais para o sucesso acadêmico e pessoal. Essa integração proporciona uma abordagem coerente e sistemática para a implementação de práticas pedagógicas que não apenas melhoram o desempenho dos alunos, mas também contribuem para uma formação integral conforme os padrões estabelecidos pela BNCC.

2.7 Hipóteses de pesquisa

A formulação das quatro hipóteses desta pesquisa se baseia em desafios práticos observados no contexto educacional, buscando trazer soluções para problemas comuns enfrentados pelas instituições de ensino. Em vez de se apoiar apenas em teorias preexistentes, as hipóteses foram desenvolvidas com foco em necessidades identificadas, como a detecção precoce de alunos em risco de evasão, a identificação de disciplinas que mais influenciam no sucesso acadêmico e o impacto dos diferentes perfis comportamentais e estilos de aprendizagem no desempenho. A ideia é validar, de forma empírica, se a personalização e a análise preditiva, mediadas por algoritmos de Aprendizagem de Máquina, podem otimizar a administração educacional, facilitando decisões mais estratégicas e intervenções pedagógicas direcionadas.

As hipóteses foram validadas por um grupo de 6 especialistas.

Assim, cada hipótese representa um esforço para explorar como as tecnologias preditivas podem promover um ensino mais adaptável, eficiente e centrado nas necessidades individuais dos alunos, abordando diretamente os desafios da retenção e do desempenho escolar.

2.7.1 Hipótese A - Score

A Hipótese A busca validar a eficácia dos algoritmos de Aprendizagem de Máquina (AM) na discriminação de alunos "Aptos" e "Não Aptos". A formulação desta hipótese está associada ao uso crescente de AM na educação como uma ferramenta de apoio para a tomada de decisões educacionais. Na prática, administradores e professores buscam métodos que ajudem a identificar, de forma antecipada, alunos que possam precisar de intervenção. A hipótese A tenta responder se o uso de um score preditivo pode ser um instrumento confiável para essa finalidade, ajudando a alocar recursos e direcionar intervenções de maneira mais eficiente.

> **Hipótese A - Score: O score gerado pelo algoritmo de AM tem acurácia suficiente para discriminar os estudantes “Não Apto" e “Apto” para os direcionar os caminhos de aprendizagem.**

2.7.2 Hipótese B - Disciplinas mais influentes

A Hipótese B considera a relevância de disciplinas específicas no desempenho acadêmico dos alunos. Esta hipótese se fundamenta na ideia de que algumas disciplinas têm maior impacto no aprendizado e retenção do aluno, e que identificar essas disciplinas pode permitir intervenções mais direcionadas e estratégias pedagógicas que maximizem os resultados acadêmicos. A hipótese propõe testar se há uma correspondência significativa entre o score preditivo gerado e o desempenho dos alunos em disciplinas específicas, o que justificaria um foco maior em áreas-chave do currículo escolar.

> **Hipótese B – Disciplinas mais influentes: Há disciplinas que apresentam real correspondência com o score gerado pelo algoritmo de AM, e, portanto, têm papel mais relevante para identificar os estudantes “Não Apto" e “Apto”.**

2.7.3 Hipótese C - Psicometria: Perfil Comportamental

A Hipótese C explora a possibilidade de que o perfil comportamental dos alunos (como Analista, Comunicador, Executor e Planejador) influencie o score preditivo gerado pelo algoritmo. A ideia por trás dessa hipótese é que comportamentos e características de personalidade desempenham um papel importante no engajamento e sucesso dos alunos, o que poderia ser refletido nos scores preditivos. Ao entender se esses perfis comportamentais impactam os resultados preditivos, poderíamos refinar as estratégias pedagógicas, adaptando o ensino aos diferentes estilos de comportamento dos estudantes.

> **Hipótese C – Psicometria - Perfil Comportamental**: **O score gerado pelo algoritmo de AM apresenta real diferença conforme os Perfis Comportamentais observados nos participantes do estudo.**

2.7.4 Hipótese D – Psicometria - Sistemas Representacionais

A Hipótese D considera os diferentes sistemas representacionais (Auditivo, Cinestésico, Visual e Digital) dos alunos, partindo do princípio de que estilos de aprendizagem distintos afetam o modo como o aluno absorve o conteúdo educacional e, portanto, seu desempenho. A formulação desta hipótese se fundamenta na diversidade de modos de aprendizado dos estudantes e como adaptar o processo de ensino pode melhorar o desempenho deles. A proposta é verificar se há um impacto significativo dos estilos de aprendizagem nos scores gerados pelos modelos de AM, o que poderia indicar a necessidade de intervenções pedagógicas mais personalizadas de acordo com cada sistema representacional.

> **Hipótese D – Psicometria - Sistemas Representacionais: O score gerado pelo algoritmo de AM apresenta real diferença conforme os Sistemas Representacionais observados nos participantes do estudo.**

3 METODOLOGIA

3.1 O modelo de aprendizado de máquina para prever o desempenho

O ponto alto desta tese está no desenvolvimento de um modelo que utiliza técnicas de AM para prever o desempenho acadêmico dos alunos e sugerir intervenções personalizadas. Esse modelo é apresentado na Figura 4 e ilustra como algoritmos de AM podem transformar dados educacionais em ações concretas, visando otimizar a trajetória de aprendizagem de cada estudante.

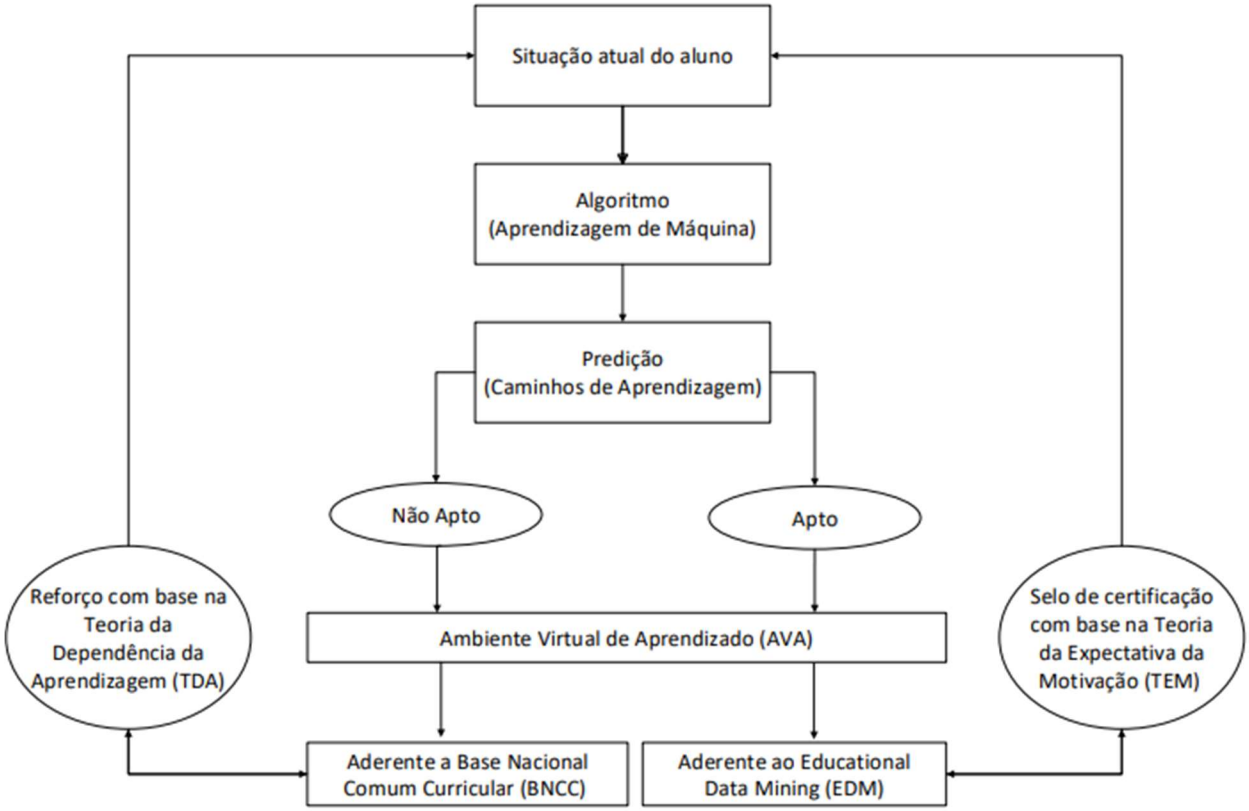
A proposta é criar um sistema adaptativo que analisa dados do histórico acadêmico, características comportamentais e preferências de aprendizado dos alunos. Com base nessas informações, o algoritmo de AM faz uma predição da situação atual dos estudantes, categorizando-os em duas categorias: "Aptos com provável aprovação" ou "Não Aptos e provável abandono". Essa classificação não apenas indica o *status* atual do aluno, mas orienta a administração educacional sobre as melhores ações a serem tomadas para cada grupo.

O modelo integra as teorias educacionais, como a TDA e a TEM, e o uso de tecnologias modernas, como o EDM e ambientes virtuais de aprendizagem (AVA), para proporcionar uma experiência de ensino que é ao mesmo tempo personalizada e eficaz. Cada parte do modelo é estrategicamente desenhada para criar um ciclo de aprendizagem onde o *feedback* contínuo é utilizado para a retenção escolar e para se evitar a evasão de alunos.

A combinação do AM com teorias educacionais e o AVA torna o modelo uma ferramenta aplicável para a administração educacional. Ao prever desafios e antecipar intervenções, ele transforma a experiência de aprendizado de uma abordagem reativa para uma proativa, criando assim oportunidades para que cada aluno possa alcançar seu potencial máximo.

A Figura 4 detalha as diferentes fases do modelo, desde a avaliação da situação atual dos alunos até as ações específicas destinadas a cada categoria identificada (Aptos e Não Aptos). O processo começa pela análise da **Situação Atual do Aluno**, que leva em consideração dados como notas, participação, perfil comportamental e outros fatores que influenciam o desempenho acadêmico.

Figura 4 – *Modelo de Aprendizagem de Máquina para Prever o Desempenho do Aluno*



Fonte: Elaboração própria.

3.1.1 Não Apto

Após a previsão do aluno como "Não Apto" pelo algoritmo, ele é encaminhado para o AVA, onde são definidos caminhos de aprendizado adaptativos, projetados de acordo com suas necessidades e dificuldades. O AVA, que pode ser uma plataforma como Canvas Instructure, oferece estrutura específica para reforço educacional, como atividades personalizadas e novos testes, baseando-se no desempenho e nas áreas de necessidade de cada aluno.

O suporte pedagógico fornecido aos alunos classificados como "Não Aptos" é fundamentado na TDA. A TDA concentra-se em moldar o comportamento dos alunos por meio de estímulos externos, utilizando estratégias de reforço positivo para incentivar mudanças significativas e duradouras. Nesse contexto, a teoria sugere que a aprendizagem ocorre por meio de respostas positivas a estímulos específicos, que promovem comportamentos desejáveis, como o aumento do esforço e a melhora no desempenho.

No AVA, os alunos recebem reforços positivos para cada progresso alcançado em seu percurso de aprendizado. Esses reforços podem ser expressos em elogios, recompensas simbólicas ou percentuais positivos que mostram ao aluno a evolução em seu processo de aprendizagem. Esse

sistema é especialmente importante para incentivar estudantes em risco de evasão, motivando-os a persistir no aprendizado.

A integração da TDA com a BNCC reforça o compromisso de garantir que cada aluno tenha uma educação inclusiva e equitativa. A BNCC enfatiza o desenvolvimento de competências cognitivas e socioemocionais e a necessidade de apoiar estudantes que enfrentam dificuldades de aprendizagem. A TDA, ao aplicar reforços positivos de maneira sistemática, alinha-se perfeitamente às diretrizes da BNCC ao garantir que cada aluno tenha oportunidades iguais de superar barreiras acadêmicas, receber apoio adaptado às suas necessidades e progredir conforme os padrões estabelecidos pela base curricular.

Portanto, o processo de reforço implementado para os alunos "Não Aptos" não apenas aborda suas dificuldades específicas, mas também está estruturado para promover a equidade educacional, proporcionando aos alunos as melhores condições para continuar seus estudos e reduzir as chances de evasão.

3.1.2 Apto

Após o algoritmo identificar o aluno como "Apto", ele é encaminhado para o AVA, onde trilhas específicas de aprofundamento são definidas, incentivando o desenvolvimento contínuo e promovendo a motivação do estudante. A abordagem fundamenta-se na TEM, que sugere que os alunos são mais engajados quando percebem que o esforço empregado resultará em conquistas tangíveis.

No AVA, os alunos "Aptos" seguem caminhos que reforçam essa relação entre esforço e recompensa, promovendo uma evolução contínua no aprendizado. A TEM considera três componentes principais que motivam o aluno: Expectativa, Instrumentalidade e Valência. O AVA garante que essas três variáveis sejam atendidas.

Expectativa: Os alunos devem acreditar que seus esforços resultarão em melhorias no desempenho acadêmico. Isso é reforçado por atividades desafiadoras, porém realizáveis, que demonstram claramente o progresso.

Instrumentalidade: O AVA utiliza selos digitais de certificação como recompensas tangíveis que sinalizam o sucesso do aluno. Isso demonstra que o esforço conduz a recompensas reais, como reconhecimento público de suas habilidades.

Valência: As recompensas, como os selos de certificação, têm valor significativo para os alunos, pois demonstram conquistas relevantes no contexto educacional. Assim, o aluno valoriza mais cada etapa do aprendizado.

A EDM também é utilizada para ajustar e personalizar as trilhas de aprendizado no AVA. A EDM analisa as interações dos alunos, identificando padrões e preferências que ajudam a adequar as atividades conforme o perfil e o ritmo de cada estudante. Essa análise não só permite uma personalização maior do conteúdo, mas também ajuda a alinhar as recompensas com os interesses e necessidades individuais, reforçando a TEM e garantindo que cada aluno sinta que seus esforços são reconhecidos e recompensados.

Assim, a integração da TEM e da EDM no processo de ensino dos alunos "Aptos" assegura um ciclo contínuo de motivação, recompensas e evolução, potencializando o desempenho acadêmico e promovendo a retenção de maneira adaptativa e personalizada.

3.1.3 Monitoramento

O modelo proposto opera como um sistema de monitoramento contínuo e adaptativo, focado na retroalimentação do aprendizado. Isso significa que o desempenho de cada aluno é reavaliado constantemente, e o algoritmo de AM ajusta suas recomendações e intervenções de acordo com a evolução de cada estudante.

Quando um aluno classificado como "Não Apto" recebe reforços no AVA, seu progresso é monitorado para verificar se ele alcança a situação de "Apto". Esse ciclo de retroalimentação não apenas garante que os alunos recebam suporte contínuo, mas também atualiza o modelo conforme novos padrões de dados e comportamentos emergem, proporcionando intervenções personalizadas.

Assim, a retroalimentação cria um processo cíclico de aprendizado, onde cada ciclo aperfeiçoa o próximo, promovendo uma evolução constante e incremental. Essa abordagem torna o uso do AM uma ferramenta dinâmica e proativa, ajustando as ações conforme a realidade de cada aluno. Dessa forma, o modelo responde diretamente à pergunta central da tese: "Como o uso de algoritmos de Aprendizagem de Máquina pode prever o desempenho dos alunos?". Ao reavaliar continuamente o progresso, ele não só prevê o desempenho, mas também intervém de maneira inteligente para aprimorá-lo, focando a trajetória individual e resultados mais eficazes.

3.2 Metodologia de pesquisa

Esta tese seguiu uma orientação estruturada para a metodologia de pesquisa, com critérios garantidores para a qualidade do trabalho. Apresenta-se na Figura 5 essa estrutura, na ordem que foi abstraída e seguida.

Figura 5 - *Estrutura da Metodologia de Pesquisa Aplicada nesta Tese*

Fonte: Elaboração própria.

3.2.1 Abordagem de pesquisa

Esta tese adotou uma abordagem quantitativa, pois ela é a mais adequada para testar hipóteses e explorar as relações entre variáveis no contexto dos algoritmos de AM. A pesquisa quantitativa, ao lidar com dados numéricos, permite uma análise precisa do desempenho dos alunos com base em métricas gerais. Além disso, essa abordagem facilita a generalização dos resultados para uma população maior, ampliando a relevância das conclusões obtidas. Embora a pesquisa tenha se concentrado na abordagem quantitativa, o uso complementar de métodos qualitativos foi considerado para esclarecer pontos específicos, enriquecendo a metodologia aplicada.

3.2.2 Teorias

A tese incorpora duas teorias fundamentais: a TEM e a TDA. Embora essas teorias não tenham sido diretamente aplicadas no levantamento e processamento dos dados, elas são essenciais para o Modelo de Aprendizagem de Máquina proposto. A TEM fornece uma estrutura para entender como a motivação dos alunos é moldada por suas expectativas, pela instrumentalidade de suas ações e pela valência dos resultados que obtêm. Em complemento, a TDA oferece uma perspectiva sobre como o aprendizado prévio dos alunos pode influenciar seu desempenho futuro, contribuindo para a personalização e eficácia das estratégias educacionais propostas.

3.2.3 Nível de pesquisa

O nível de pesquisa foi predominantemente descritivo, focado na caracterização das variáveis, como o desempenho acadêmico e a evasão dos alunos, especialmente a partir da variável STATUS_BIN, que classifica os alunos como "Aptos" e "Não Aptos". Foram analisadas também disciplinas específicas, como ERMS, para entender sua influência no desempenho preditivo.

Paralelamente, o estudo adotou um nível explicativo, ao manipular variáveis como X1 (Não Aptos) e X0 (Aptos), com o objetivo de verificar as reações desses grupos aos scores preditivos. Além disso, foram realizadas correlações entre STATUS_BIN e outras variáveis, permitindo uma compreensão mais profunda das interações entre o desempenho acadêmico e os fatores que influenciam a classificação dos alunos.

3.2.4 Tipo de pesquisa

O tipo de pesquisa adotado foi aplicado, com foco em resultados práticos, voltados para a previsão do desempenho acadêmico e a identificação de alunos em risco de evasão. Por exemplo, a utilização de algoritmos de AM foi aplicada para gerar scores preditivos que podem ser diretamente utilizados por gestores educacionais para identificar alunos "Não Aptos" e tomar decisões de intervenção no ambiente escolar, demonstrando o caráter prático da pesquisa.

3.2.5 Técnica de pesquisa

Diversas técnicas e procedimentos foram utilizados para coletar e analisar os dados desta pesquisa, de acordo com o tipo e objetivo específico do estudo. Algumas dessas técnicas, adequadas ao contexto desta tese, estão detalhadas no Tabela 7. Essas técnicas foram selecionadas para garantir a precisão e a relevância dos dados coletados, permitindo a obtenção de resultados significativos.

Tabela 7 – *Técnicas de Pesquisa Utilizadas nesta Tese*

Técnica de pesquisa	Breve resumo e uso	Autor
Indicação de variáveis	Considera e controla variáveis que podem interferir no objeto de estudo, para evitar comprometimento da pesquisa. **Uso**: Aplicada na separação das variáveis preditoras (X) dos alunos para treinamento e testes, e posterior submissão aos algoritmos.	Lakatos e Marconi (2021, p. 193)
Estudo de campo	Realizado em locais específicos (comunidades, organizações, instituições), inclui técnicas como entrevistas e análise de documentos. **Uso**: Acesso a dados primários da instituição educacional ao longo de 9 meses, utilizados para validar os modelos preditivos.	Lakatos e Marconi (2021, p. 219)

Técnica de pesquisa	Breve resumo e uso	Autor
Levantamento	Coleta de informações de um grupo significativo, permitindo análise quantitativa e obtenção de conclusões com base nos dados. **Uso**: Definição do controle de notas, quantidade de disciplinas cursadas e criação da variável dicotômica STATUS_BIN para classificação dos alunos.	Gil (2019, pp. 56-57)
Observação	Inclui observação assistemática (sem controle) ou sistemática (estruturada), podendo ser participante ou não. **Uso**: Observação do desempenho dos alunos ao longo do curso e análise do impacto dos algoritmos na previsão de desempenho.	Gil (2019, pp. 132-136); Lakatos e Marconi (2021, pp. 226-229)
Participação	Baseia-se em uma resolução empírica e cooperativa de problemas, envolvendo pesquisadores e participantes em um processo colaborativo. **Uso**: Aplicada na conversão de notas dos alunos em percentuais e escalas decimais (0 a 5), facilitando a previsão do desempenho.	Vergara (2010); Gil (2019, pp. 126-131)

Fonte: Elaboração própria.

Durante a coleta de dados, as técnicas de pesquisa foram aplicadas e acompanhadas de maneira contínua para garantir a coerência dos resultados com os objetivos do estudo. A indicação de variáveis foi essencial para organizar o modelo de análise, separando as variáveis preditoras (X) para o treinamento e testes dos algoritmos, evitando interferências que pudessem comprometer os resultados. No estudo de campo, foram coletados dados primários de uma instituição educacional. Esses dados ajudaram na validação e ajuste dos modelos preditivos, adaptando as variáveis conforme surgiam novas informações.

A técnica de levantamento possibilitou a coleta de dados sobre o desempenho acadêmico dos alunos, como notas e disciplinas cursadas, permitindo a criação da variável dicotômica STATUS_BIN, usada para classificar os alunos. Esse levantamento foi ajustado ao longo do tempo com base nos dados coletados. A observação sistemática do desempenho dos alunos durante o curso permitiu analisar o impacto dos algoritmos na previsão de desempenho e fazer ajustes conforme necessário.

Por fim, a participação ativa na pesquisa facilitou a conversão das notas em percentuais e escalas decimais, simplificando a análise e previsão do desempenho. Esse processo colaborativo foi acompanhado de perto, e ajustes foram feitos conforme os resultados eram analisados, permitindo a adaptação contínua dos modelos.

3.3 Coleta de dados

Os dados utilizados nesta pesquisa foram obtidos de uma escola *on-line* especializada em desenvolvimento de *software*, localizada em Joinville, SC. A população estudada, conforme demonstrado na Tabela 8, abrangeu 676 registros iniciais de alunos em abril de 2022, crescendo para 900 registros até novembro de 2022. Os estudantes estão distribuídos por diversas regiões do Brasil e incluem também alunos internacionais, residentes em países como Angola, Portugal, Alemanha e Estados Unidos.

O modelo de ensino da escola era síncrono, com conteúdo gravado disponibilizado em uma plataforma *web*. O curso tinha uma duração total de 9 meses ou 930 horas, distribuídas em 15 disciplinas. Os alunos participavam de encontros quinzenais com os professores, realizados de maneira remota. As disciplinas tinham durações variáveis, de 1 a 8 semanas para conclusão. As provas eram automatizadas, enquanto algumas tarefas específicas eram corrigidas manualmente pelos professores. A admissão de novos alunos ocorria de modo contínuo, permitindo inscrições semanais.

Embora houvesse a possibilidade de que a amostra fosse enviesada — por exemplo, se os alunos que se inscrevessem em semanas diferentes apresentassem características distintas —, esse viés não foi identificado durante o levantamento de dados. A generalização dos dados coletados pode ser aplicada a escolas com modelos de negócio semelhantes, como aquelas que utilizam plataformas síncronas e ciclos de avaliação flexíveis, além de ser adaptável para instituições que operem com avaliações bimestrais.

Tabela 8 – *Fonte de Dados*

Mês e Ano	abr/22	mai/22	jun/22	jul/22	ago/22	set/22	out/22	nov/22
Total registros	676	733	781	808	829	850	864	900

Fonte: Elaboração própria.

A coleta de dados iniciou em abril de 2022 e continuou mensalmente até novembro do mesmo ano, totalizando 900 registros. Para a análise final, foram utilizados apenas os dados de novembro, pois esse período representa um estágio mais avançado dos alunos em diversos pontos do curso. Os dados dos meses anteriores foram utilizados exclusivamente para o treinamento dos algoritmos. Após a remoção de 265 registros com notas zeradas, a base final para predição ficou com 635 alunos, que apresentam *status* variados, como Abandono, Aprovado, Ativo, Cancelado, Desistente e Trancado.

3.4 Dicionário de dados

O Apêndice E apresenta o detalhamento das colunas utilizadas para o armazenamento dos dados dos alunos na sua forma original, antes de serem processados para a AM. Este dicionário de dados contém informações que variam desde detalhes pessoais, como perfil comportamental e sistema representacional, até o desempenho acadêmico dos alunos em diversas disciplinas e seu *status* no curso. As variáveis foram ajustadas conforme a necessidade de análise e predição.

As variáveis *Dummy* de Indicador de Conclusão foram criadas para identificar de maneira binária se o aluno cursou ou não uma determinada disciplina, atribuindo o valor 1 para alunos que completaram a disciplina e 0 para aqueles que não a cursaram. Essa técnica de *dummies* é amplamente utilizada para transformar informações categóricas em dados numéricos, essenciais para o processo de AM.

Além disso, o uso dessas variáveis serve para evitar campos vazios ou a presença de zeros indevidos que poderiam distorcer os resultados do preditor. Ao representar a conclusão de disciplinas de maneira binária, garante-se que o algoritmo consiga interpretar corretamente as informações, eliminando ruídos que possam impactar negativamente a precisão e eficácia da predição de desempenho acadêmico ou risco de evasão.

3.5 Variáveis: Independentes e Dependente

Para garantir a qualidade dos dados, 265 registros de alunos sem notas atribuídas foram excluídos da análise, resultando em um conjunto final de 635 registros. A variável-alvo STATUS_BIN foi transformada em uma variável dicotômica, onde 1 indica evasão (Abandono, Cancelado, Desistente e Trancado) e 0 indica alunos ativos, estruturando o problema como uma classificação binária. Essa variável funciona como o *target* (Y) para os algoritmos de AM, sendo o resultado que o modelo tenta prever com base nas variáveis preditoras.

As principais variáveis preditoras utilizadas incluem:

- Perfil Comportamental (PC): categorias como Analista, Comunicador, Executor e Planejador.
- Sistema Representacional (SR): Auditivo, Digital, Cinestésico ou Visual.
- Gênero: Masculino (M) e Feminino (F).
- Notas nas disciplinas: exemplo: PC_nota, SS_nota, SO_nota, entre outras.

- Cursou/Não cursou: variáveis binárias indicando se o aluno cursou ou não cada disciplina (ex: PC_cursou, SS_cursou).
- N_Disciplinas: número total de disciplinas cursadas por aluno.

As variáveis categóricas, como PC, SR e Gênero, foram convertidas para serem utilizadas nos algoritmos de AM por meio de técnicas como *one-hot encoding*. Esse processo transforma as categorias em variáveis binárias para que os algoritmos consigam interpretá-las corretamente, evitando qualquer distorção nos resultados.

Além disso, após o processo de predição, uma nova coluna chamada Preditor será adicionada, contendo os resultados gerados pelos modelos de AM.

3.6 *Timeline* das etapas da AM

O Apêndice F apresenta um fluxo específico para a metodologia aplicada nesta tese, refletindo um processo de AM que foi adaptado conforme os objetivos de previsão de desempenho acadêmico. Embora siga uma estrutura similar a outros métodos, como os descritos por Jantakun et al. (2022) e Kishor et al. (2022), este fluxo propõe etapas alinhadas às características dos dados educacionais e ao objetivo de personalização do ensino. Assim como outros autores propõem variações em seus processos de AM, o modelo apresentado aqui pode ser ajustado conforme a necessidade de cada aplicação.

3.6.1 Algoritmos e bibliotecas

3.6.1.1 Algoritmos

A escolha do algoritmo mais adequado para prever a *performance* dos alunos foi baseada em uma análise extensa da literatura existente, cobrindo estudos sobre a acurácia de algoritmos entre 2011 e 2024. Sem preconceitos iniciais sobre qual algoritmo seria o melhor, foram identificados 21 algoritmos, incluindo 13 de classificação e 8 de regressão. Essa seleção ampla possibilitou uma análise abrangente e fundamentada em evidências, garantindo que a escolha final fosse embasada em referências bibliográficas e nos resultados obtidos durante o processo de análise.

O Apêndice G apresenta uma seleção desses algoritmos, organizados por técnica e com foco na análise de acurácia, a fim de identificar os melhores modelos. Os algoritmos estão

classificados em diferentes categorias, como Modelos Lineares, Análise Discriminante Linear e Quadrática, Máquinas de Vetores de Suporte, Gradiente Descendente Estocástico, K-Vizinhos Mais Próximos, Naive Bayes, Árvores de Decisão, Métodos de Ensemble e Redes Neurais. Cada algoritmo é representado por sua sigla e pode ser aplicado em tarefas de classificação ou regressão, dependendo da técnica escolhida.

Por exemplo, na categoria de Métodos de Ensemble, o algoritmo Random Forest Classification (RFC) é utilizado para classificação, enquanto na categoria de Redes Neurais, o Multi-layer Perceptron pode ser aplicado tanto em classificação quanto em regressão, mostrando a flexibilidade de alguns algoritmos.

O Apêndice H oferece uma visão geral das técnicas e algoritmos explorados nesta tese para prever o desempenho dos alunos. Cada técnica é acompanhada de seus algoritmos correspondentes, siglas e uma breve descrição de suas finalidades, facilitando a compreensão do propósito e aplicação de cada abordagem na análise.

3.6.1.2 Bibliotecas

Uma biblioteca, em termos de programação, é um conjunto de códigos e funções pré-escritos que os programadores podem utilizar para realizar tarefas específicas, economizando tempo e esforço. Pense em uma biblioteca como uma caixa de ferramentas: em vez de construir tudo do zero, você usa ferramentas já prontas para resolver problemas rapidamente. No caso do Python, uma linguagem de programação muito popular, existem várias bibliotecas que ajudam a realizar diferentes tarefas, como análise de dados ou cálculos científicos.

Por exemplo, se você quer criar um gráfico ou calcular uma estatística complexa, em vez de programar todas as etapas manualmente, você pode simplesmente usar uma biblioteca que já tenha essas funções prontas. Isso torna o Python uma ferramenta poderosa e acessível para resolver problemas em várias áreas, desde a ciência de dados até o desenvolvimento de *software*.

O Apêndice I apresenta as bibliotecas que serão utilizadas nesta tese, junto com uma breve descrição de cada uma e a sua aplicabilidade no contexto da pesquisa.

3.6.1.3 Importação das bibliotecas

O Google Colaboratory, ou Colab, é uma plataforma oferecida pela Google que permite a execução de códigos de programação diretamente no navegador, sem a necessidade de instalar

softwares específicos no computador. É amplamente utilizada para tarefas de AM e análise de dados, pois oferece um ambiente de desenvolvimento pré-configurado e acessível na nuvem.

Um dos grandes benefícios do Colab é que ele disponibiliza recursos computacionais gratuitos, como GPUs (unidades de processamento gráfico), que são essenciais para lidar com processamentos intensivos. Muitos desses processamentos, especialmente em projetos de AM, podem ser inviáveis de executar em computadores pessoais devido à sua demanda por poder de processamento e memória. Por esse motivo, o Colab destaca-se ao oferecer recursos de alta *performance* na nuvem, permitindo que o usuário execute tarefas complexas de maneira rápida e eficiente, sem precisar de *hardware* especializado.

Essa característica é especialmente vantajosa em comparação a outras plataformas, que podem cobrar pelo uso desses recursos avançados. No Colab, você tem acesso a essas ferramentas de maneira gratuita, o que o torna uma opção acessível tanto para iniciantes quanto para profissionais da área de ciência de dados e AM.

Durante este trabalho, o Colab foi utilizado para executar os códigos necessários para a análise e predição dos dados, aproveitando seus recursos, que seriam inadequados para computadores pessoais, para otimizar o processo de desenvolvimento e análise.

3.6.2 DataSet e DataFrame

3.6.2.1 DataSet

Um DataSet é uma coleção estruturada de dados que serve como base para análises e modelagens preditivas. No contexto desta tese, o DataSet utilizado contém 635 registros, cada um representando um aluno com diversas características, como Perfil Comportamental (PC), Sistema Representacional (SR), e suas notas em disciplinas específicas. A variável-alvo (STATUS_BIN) é dicotômica e classifica os alunos como "Ativos" (0) ou "Evasão" (1).

3.6.2.2 DataFrame

Um DataFrame é uma estrutura de dados, organizada em linhas e colunas, que permite armazenar e manipular dados de modo eficiente, semelhante a uma planilha. No contexto de análise de dados, o DataFrame carrega os dados em memória temporária para serem processados e analisados. Entretanto, é importante destacar que as modificações feitas no DataFrame não

alteram o DataSet original, ou seja, o arquivo inicial de dados permanece intacto, e qualquer mudança ou cálculo é limitado ao ambiente de trabalho até que seja explicitamente salvo.

3.6.3 Limpeza do DataFrame

Nesta etapa do processo, o DataFrame é refinado para incluir apenas as colunas que são diretamente relevantes para os modelos de AM. A limpeza dos dados envolve a remoção de colunas que não contribuem para a análise preditiva, como o nome do aluno, a data de início no curso e a cidade. Esses campos, embora importantes para o gerenciamento de informações administrativas, não são necessários para o processo de modelagem preditiva e poderiam introduzir ruído nos modelos, prejudicando a eficácia das previsões. Portanto, essas colunas são excluídas, garantindo que o DataFrame final contenha apenas os dados essenciais para a construção de modelos de AM precisos e eficientes.

3.6.4 *One-hot encoding* e *dummies*

3.6.4.1 One-hot encoding

One-hot encoding é uma técnica amplamente utilizada na preparação de dados, especialmente para transformar variáveis categóricas em um formato que pode ser interpretado por algoritmos de ML. Essa técnica é particularmente útil para variáveis categóricas nominais, ou seja, aquelas que representam categorias sem uma ordem natural.

Ao aplicar o *one-hot encoding*, cada categoria única de uma variável categórica é convertida em uma nova coluna binária (0 ou 1). Se a variável original possui *N* categorias únicas, essa técnica irá transformá-la em *N* colunas distintas, onde cada coluna indicará a presença (1) ou ausência (0) de uma das categorias originais.

A Tabela 9 exemplifica a saída do DataFrame (df3) após a aplicação do *one-hot encoding* em todas as colunas categóricas. Para simplificação, foram exibidas apenas algumas das colunas.

Tabela 9 – *Exemplo de Saída do DataFrame após Aplicação do One-Hot Encoding*

#	PC_Analista	PC_Comunicador	PC_Executor	PC_Planejador	PC_SemDados
0	false	false	false	false	true
1	true	false	false	false	false
2	true	false	false	false	false

#	PC_Analista	PC_Comunicador	PC_Executor	PC_Planejador	PC_SemDados
3	true	false	false	false	false
4	true	false	false	false	false
5	true	false	false	false	false
6	true	false	false	false	false
7	false	false	false	false	true
8	true	false	false	false	false
9	true	false	false	false	false

Nota. Os valores na tabela são exibidos como False ou True devido à forma como o Pandas, a biblioteca utilizada, representa variáveis booleanas. Internamente, esses valores são interpretados como 0 e 1 pelos algoritmos de AM, onde False é tratado como 0 e True como 1. Isso permite que as colunas binárias sejam corretamente processadas na análise e predição. A conversão é realizada automaticamente pelo Pandas, sem a necessidade de intervenção manual do usuário. Fonte: Elaboração própria.

3.6.4.2 Dummies

No DataSet, as disciplinas não cursadas são representadas por variáveis do tipo "*dummy*", onde 1 indica que a disciplina foi cursada e 0 indica que não foi. Esse formato numérico é essencial para que o algoritmo de AM possa processar essas informações de modo eficiente.

Além de permitir a representação das disciplinas cursadas, as variáveis *dummies* também fornecem *insights* valiosos ao incluir a ausência de disciplinas no modelo preditivo. Por exemplo, se um aluno não cursou a disciplina de "Lógica de Programação" (marcada com 0), o algoritmo pode detectar essa ausência como um possível fator que influencia negativamente o desempenho.

Essa abordagem permite que o modelo de AM aprenda não apenas com as informações sobre as disciplinas que foram concluídas, mas também com as que foram deixadas de lado. A consideração tanto da presença quanto da ausência de disciplinas melhora significativamente a capacidade do modelo de identificar padrões e de fazer predições mais precisas sobre o desempenho dos alunos e possíveis riscos de evasão. Além disso, essa técnica é especialmente útil para evitar que informações ausentes ou incompletas distorçam os resultados, tornando o processo eficiente.

3.6.5 Treinamento e teste (Separação do DataFrame)

O treinamento e teste de um DataFrame é uma técnica fundamental de AM, que envolve a divisão do conjunto de dados em dois subconjuntos: um para treinamento (70%) e outro para teste (30%). Isso funciona da seguinte maneira: o algoritmo é treinado usando o primeiro subconjunto

(conjunto de treinamento) para aprender padrões e relações entre as variáveis, e depois é testado no segundo subconjunto (conjunto de teste) para verificar se ele consegue aplicar o que aprendeu em novos dados que não viu durante o treinamento.

Pense nisso como ensinar alguém a dirigir. Primeiro, você passa a maior parte do tempo treinando a pessoa em um local fechado, sem trânsito (conjunto de treinamento). Depois de ela aprender o básico (como ligar o carro, usar os pedais, virar o volante), você leva essa pessoa para dirigir em uma rua movimentada pela primeira vez (conjunto de teste). Esse é o momento em que você vê se ela consegue aplicar o que aprendeu no treino em um cenário real.

Da mesma forma, o algoritmo aprende com o conjunto de treinamento e, quando é testado no conjunto de teste, pode-se ver se ele consegue generalizar bem o que aprendeu e fazer boas previsões em dados novos.

Em resumo, o conjunto de treinamento é usado para "ensinar" o algoritmo ajustando seus parâmetros, enquanto o conjunto de teste é usado para avaliar quão bem o algoritmo pode aplicar o que aprendeu em dados que ele ainda não viu.

3.6.5.1 Execução do treinamento

Durante a execução do treinamento de um algoritmo de AM, o conjunto de dados é previamente dividido em duas partes: 70% para treinamento e 30% para teste. As variáveis preditoras (ou independentes) incluem informações como perfil comportamental, sistema representacional, gênero, e notas nas disciplinas, entre outras características relevantes. A variável dependente (ou variável Y), chamada STATUS_BIN, indica se um aluno é Ativo (0) ou está em Evasão (1).

No processo de treinamento, o algoritmo aprende com os 70% dos dados, procurando identificar padrões e relações entre as variáveis preditoras e a variável Y. Em seguida, ele tenta usar esse conhecimento para fazer previsões sobre os 30% restantes, que são usados para o teste. O algoritmo gera uma saída chamada Preditor, que é sua tentativa de prever o *status* de cada aluno (se é Ativo ou está em Evasão).

Para avaliar a eficácia do modelo, o pesquisador compara essas previsões feitas pelo algoritmo com os resultados reais dos alunos no conjunto de teste. A chave aqui é que, durante o teste, o algoritmo não tem acesso à variável Y (o verdadeiro *status* dos alunos), o que permite verificar se o algoritmo é capaz de generalizar suas previsões corretamente para novos dados que ele ainda não viu.

Exemplo: Imagine que o algoritmo "aprendeu" com 70% dos alunos que, em geral, alunos com notas baixas em várias disciplinas têm maior chance de evasão. Ao testar esse conhecimento com os 30% restantes, ele tenta prever se os alunos desse grupo vão ou não abandonar o curso, e o pesquisador verifica se essas previsões coincidem com a realidade.

3.6.6 Métricas

O *score* de um algoritmo é uma medida que indica o quão bem ele está desempenhando uma tarefa específica. A forma como esse *score* é calculado pode variar, dependendo do tipo de problema e do algoritmo utilizado. Para esclarecer, apresentam-se dois tipos comuns de problemas: classificação binária e regressão, cada um com métricas específicas.

1. Classificação Binária: Esse é o tipo de problema onde o algoritmo tenta classificar algo em uma de duas categorias. Por exemplo, no caso desta pesquisa, classificam-se os alunos como "Ativos" (0) ou "Em Evasão" (1). Para medir o desempenho de um algoritmo nesse tipo de tarefa, usam-se métricas como:
    - Acurácia: Indica a porcentagem de vezes que o algoritmo acertou, ou seja, quantas previsões estavam corretas em relação ao total de previsões feitas.
    - Precisão: Mede o quanto as previsões positivas feitas pelo algoritmo (como prever que um aluno vai abandonar o curso) estavam realmente corretas.
    - *Recall*: Indica a capacidade do algoritmo de identificar corretamente todos os casos reais de evasão (ou seja, quantos dos alunos que realmente evadiram foram identificados corretamente pelo algoritmo).
    - Regressão: Diferente da classificação binária, nos problemas de regressão, o objetivo é prever um valor contínuo, como a nota de um aluno ou o tempo que ele levará para completar uma tarefa. Para medir o desempenho do algoritmo de regressão, usam-se outras métricas, como:
        - Erro Médio Quadrático (MSE): Mede a média dos erros ao quadrado entre as previsões feitas pelo algoritmo e os valores reais. Quanto menor o valor, melhor o desempenho.
        - Erro Absoluto Médio (MAE): Mede a média dos erros absolutos (sem considerar sinais negativos ou positivos) entre as previsões e os valores reais. Assim como o MSE, quanto menor, melhor.

- Coeficiente de Determinação (R^2): Indica quão bem o modelo se ajusta aos dados reais. Ele varia de 0 a 1, sendo que quanto mais próximo de 1, melhor o modelo explica os dados.

A escolha da métrica depende muito do tipo de problema que você está resolvendo e do que você deseja medir. Usar uma variedade de métricas é importante para ter uma visão mais completa de como o modelo está se comportando e ajustá-lo conforme as necessidades do projeto.

Exemplo: Em um problema de classificação, a acurácia pode ser alta, mas, se o *recall* for baixo, isso significa que o algoritmo está deixando de identificar muitos casos de evasão. Portanto, olhar para diferentes métricas dá uma imagem mais clara e detalhada de como o algoritmo está realmente funcionando.

3.6.6.1 Acurácia em AM

A acurácia é uma métrica que indica quão preciso é um algoritmo de classificação. Ela é calculada pela proporção de exemplos que o algoritmo classificou corretamente em relação ao total de exemplos avaliados. Em outras palavras, a acurácia pode ser entendida como a porcentagem de previsões corretas feitas pelo algoritmo. Quanto maior o valor da acurácia, melhor o desempenho do modelo.

No entanto, é importante destacar que a acurácia, sozinha, pode não ser suficiente para avaliar a qualidade do algoritmo, especialmente quando o conjunto de dados tem uma distribuição desequilibrada de classes. Por exemplo, se em um conjunto de dados a maioria dos alunos estiver ativa e poucos em evasão, um algoritmo pode alcançar uma acurácia alta apenas classificando a maioria dos casos como "Ativo", sem realmente aprender a identificar corretamente os casos de evasão.

A fórmula para calcular a acurácia é:

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

onde:

- VN (Verdadeiros Negativos): Casos corretamente classificados como negativos (por exemplo, alunos que não evadiram).
- VP (Verdadeiros Positivos): Casos corretamente classificados como positivos (por exemplo, alunos que evadiram).
- FP (Falsos Positivos): Casos incorretamente classificados como positivos (quando o algoritmo prevê que o aluno vai evadir, mas ele permanece ativo).

- FN (Falsos Negativos): Casos incorretamente classificados como negativos (quando o algoritmo não prevê que o aluno vai evadir, mas ele acaba evadindo).

A acurácia representa, portanto, a proporção de previsões corretas em relação ao total de previsões feitas. Em geral, valores de acurácia acima de 90% indicam um desempenho muito bom. Valores entre 85% e 90% também podem ser considerados satisfatórios, dependendo do problema e do equilíbrio entre as classes no conjunto de dados.

Por exemplo, se em uma classe houver muito mais alunos "Ativos" do que em "Evasão", o algoritmo pode atingir alta acurácia apenas acertando a maioria dos "Ativos", mas esse resultado pode não ser tão relevante se o foco principal for identificar corretamente os alunos em risco de evasão.

3.6.6.2 Validação cruzada média

A validação cruzada média é uma técnica importante utilizada para avaliar a capacidade de generalização de um modelo de AM. Embora a matriz de confusão forneça uma visão detalhada de como o modelo se comporta em um único conjunto de dados (mostrando erros e acertos de classificação), a validação cruzada média vai além ao testar o modelo em vários subconjuntos dos dados.

O processo funciona da seguinte maneira: o conjunto de dados é dividido em várias partes (ou "*folds*"). O modelo é então treinado em algumas dessas partes e testado em outras. Esse processo se repete até que todas as partes tenham sido usadas tanto para treinamento quanto para teste. A média dos resultados obtidos em cada rodada fornece uma medida mais confiável do desempenho do modelo.

Isso é fundamental para garantir que o desempenho observado não seja apenas fruto de sorte ou de uma combinação específica de dados. Em vez disso, a validação cruzada média permite verificar se o modelo é consistente e capaz de generalizar bem para diferentes situações e dados não vistos. É uma forma de garantir que o modelo funcione de maneira adequada em novos dados, e não apenas nos dados com os quais foi inicialmente treinado.

3.6.6.3 Precisão

A precisão é uma métrica que indica a proporção de previsões positivas que estavam corretas. Ela ajuda a entender quão preciso o modelo foi ao identificar casos positivos, ou seja, aqueles alunos que foram classificados como "Não Aptos".

Por exemplo, se o modelo previu que 10 alunos eram "Não Aptos", mas apenas 7 deles realmente estavam nessa categoria, a precisão do modelo seria de 70%. Isso significa que, em 70% das vezes, quando o modelo disse que um aluno era "Não Apto", ele estava correto.

A precisão é uma métrica especialmente útil em situações em que é importante minimizar falsos positivos — como prever que um aluno vai abandonar o curso quando ele, na verdade, permanece ativo.

3.6.6.4 Recall

O *recall* é uma métrica que mede a capacidade do modelo de identificar corretamente todas as instâncias positivas. Ou seja, ele avalia o quão bem o modelo conseguiu encontrar todos os alunos que realmente eram "Não Aptos".

Por exemplo, se havia 10 alunos que eram "Não Aptos" e o modelo conseguiu identificar 8 deles, o *recall* seria de 80%. Isso significa que, em 80% dos casos, o modelo foi capaz de identificar corretamente os alunos que estavam realmente em risco.

O *recall* é especialmente importante em situações em que é crítico minimizar os falsos negativos, como quando o objetivo é não deixar de identificar alunos que estão em risco de evasão.

3.6.6.5 F1-Score

O F1-Score é uma métrica que combina a precisão e o *recall* em um único valor, proporcionando uma visão mais equilibrada do desempenho do modelo. Essa métrica é especialmente útil quando há um desbalanceamento entre as classes — por exemplo, quando há muito mais alunos "Aptos" do que "Não Aptos".

Enquanto a precisão mede o quão corretas são as previsões positivas (como prever corretamente que um aluno é "Não Apto") e o *recall* mede a capacidade de identificar todos os casos positivos (encontrar todos os alunos "Não Aptos"), o F1-Score tenta equilibrar essas duas métricas. Isso é importante porque, em situações desbalanceadas, focar apenas uma dessas métricas pode fornecer uma visão incompleta do desempenho do modelo.

A fórmula do F1-Score é uma média harmônica entre a precisão e o *recall*, garantindo que o modelo não priorize uma métrica em detrimento da outra. Quanto maior o valor do F1-Score

(que vai de 0 a 1), melhor é o desempenho geral do modelo, considerando tanto a capacidade de fazer previsões corretas quanto a de identificar todos os casos importantes.

Exemplo: Se um modelo tem alta precisão, mas baixo *recall*, o F1-Score será reduzido, refletindo o equilíbrio entre as duas métricas. Isso ajuda a avaliar o modelo de maneira mais justa e completa.

3.6.6.6 AUC-ROC

A AUC-ROC é uma métrica que mede a capacidade do modelo de distinguir entre as classes. ROC significa *Receiver Operating Characteristic*, e a curva ROC mostra o desempenho do modelo em diferentes limiares de decisão. A Área Sob a Curva (AUC) é o valor que resume esse desempenho em um único número.

Se o AUC-ROC for 1, isso significa que o modelo consegue separar perfeitamente as classes, ou seja, ele consegue distinguir corretamente todos os alunos "Aptos" e "Não Aptos" sem cometer erros. Por outro lado, um valor de 0,5 indica que o modelo não está fazendo um trabalho melhor do que um palpite aleatório — como jogar uma moeda para decidir se um aluno é "Apto" ou "Não Apto".

A AUC-ROC é especialmente útil quando se está lidando com desbalanceamento de classes, ou seja, quando uma das classes (como "Não Aptos") é muito menor do que a outra. Ela avalia como o modelo lida com a distinção entre essas classes, independentemente do tamanho de cada uma.

Exemplo: Se o AUC-ROC de um modelo for 0,85, significa que o modelo tem uma boa capacidade de distinguir entre as classes, acertando em 85% dos casos em termos de separação entre os alunos "Aptos" e "Não Aptos".

3.6.6.7 A escolha do melhor algoritmo

Suponha que se esteja tentando prever quais alunos são "Aptos" (0) e "Não Aptos" (1) em um conjunto de dados. Foram feitos testes com três algoritmos: Random Forest Classifier (RFC)**,** Logistic Ridge Regression (LRR) e Gradient Boosting Classifier (GBC). Após calcularmos as principais métricas de desempenho, obtiveram-se os seguintes resultados na Tabela 10:

Tabela 10 – *Exemplo da Escolha do Melhor Algoritmo*

Algoritmo	Acurácia	Precisão	Recall	F1-Score	AUC-ROC
Random Forest Classifier	92%	85%	90%	87,4%	0,94
Logistic Ridge Regression	88%	83%	88%	85,4%	0,90
Gradient Boosting Classifier	93%	87%	91%	88,9%	0,95

Nota. Esses dados são fictícios. Observe sempre os melhores percentuais de cada métrica. Fonte: Elaboração própria.

a) Análise das Métricas

1. Acurácia:
    - A acurácia nos dá uma ideia geral do desempenho. O GBC tem a maior acurácia (93%), seguido de perto pelo RFC (92%), enquanto a LRR teve um desempenho um pouco inferior (88%).
2. Precisão:
    - A precisão nos diz o quanto das previsões positivas foram corretas. O GBC e o RFC apresentaram boas precisões, com o GBC ligeiramente superior (87%) em relação ao RFC (85%). A LRR teve um pouco menos (83%).
3. *Recall*:
    - O *recall* nos mostra quantos dos verdadeiros "Não Aptos" o algoritmo identificou corretamente. Novamente, o GBC (91%) e o RFC (90%) tiveram desempenhos semelhantes, com a LRR um pouco abaixo (88%).
4. F1-Score:
    - O F1-Score é a métrica que equilibra precisão e *recall*. Aqui, o GBC apresentou o melhor valor (88,9%), seguido pelo RFC (87,4%) e, por último, pela LRR (85,4%).
5. AUC-ROC:
    - A AUC-ROC mede a capacidade do modelo de distinguir entre as classes. O GBC teve o melhor valor (0,95), o que indica que ele é o mais capaz de separar alunos "Aptos" de "Não Aptos". O RFC também apresentou um bom valor (0,94), enquanto a LRR ficou com 0,90.

b) Escolha do Melhor Algoritmo

Com base nessas métricas, o GBC parece ser a melhor escolha para esse problema. Ele apresentou os melhores resultados em acurácia, precisão, *recall*, F1-Score e AUC-ROC, superando consistentemente os outros algoritmos.

Por outro lado, o RFC também teve um bom desempenho, ficando muito próximo do GBC. Se a diferença de recursos computacionais for importante (por exemplo, o GBC pode ser mais demorado e custoso de treinar), o RFC poderia ser uma boa alternativa, especialmente devido à sua acurácia e AUC-ROC altas.

A LRR, embora tenha tido um desempenho razoável, foi superada pelos outros dois algoritmos em todas as métricas, o que a torna menos recomendada para esse cenário específico.

3.6.7 Predição para todo o DataSet

A fase de predição em uma análise de dados desempenha um papel principal, pois é nesse momento que os *insights* obtidos durante a análise das variáveis são transformados em projeções aplicáveis a situações futuras. Nessa etapa, o modelo treinado é aplicado a todo o conjunto de dados para estimar o valor do Preditor (a variável de interesse).

O objetivo principal dessa fase é utilizar o conhecimento adquirido durante o treinamento do modelo para prever os resultados de todos os alunos no conjunto de dados, não apenas os que foram usados para treinar o modelo. Isso permite avaliar a eficácia do modelo e verificar se ele consegue generalizar bem para novos dados, ou seja, se ele funciona corretamente com dados que não foram vistos durante o treinamento. Além de avaliar a *performance* do modelo, essa etapa oferece uma visão prática sobre como as diferentes variáveis se relacionam e influenciam os resultados. Isso é importante, pois ajuda a entender de maneira mais clara como fatores como perfil comportamental, número de disciplinas cursadas e notas afetam a classificação dos alunos como "Aptos" ou "Não Aptos".

3.6.8 Arquivo com Preditor

A geração do arquivo com o Preditor cria um comparativo entre o valor real da variável STATUS_BIN e o Preditor gerado pelo modelo para cada aluno no conjunto de dados. Nesse arquivo, cada linha representa um aluno, identificado de maneira anônima, e contém colunas que detalham suas informações e o resultado predito pelo modelo.

O principal objetivo desse arquivo é avaliar a precisão das predições em relação ao conjunto de dados original. Ele permite verificar se o modelo está classificando corretamente os alunos como "Aptos" ou "Não Aptos" e identificar discrepâncias entre o valor real e o valor predito.

Essa análise é essencial para identificar resultados discrepantes, como alunos que apresentam comportamento fora do padrão esperado. Conforme estudos como o de Chen et al. (2023), a detecção precoce desses padrões inesperados pode indicar a necessidade de ajustes no modelo.

Esses ajustes, normalmente feitos por meio de novas parametrizações ou refinamento de hiperparâmetros, são fundamentais para aumentar a precisão das predições. Esse processo de ajuste é, contudo, minucioso e exaustivo, exigindo muitas tentativas e análises para encontrar a melhor combinação de parâmetros. Cada pequena alteração nos hiperparâmetros pode ter um impacto significativo no desempenho do modelo, o que torna o trabalho desafiador, mas necessário para alcançar resultados otimizados.

Assim, o arquivo não só avalia o desempenho atual do modelo, mas também serve como uma base sólida para futuros refinamentos, permitindo que o modelo continue a evoluir e melhorar suas previsões de maneira cada vez mais precisa.

3.7 Análises dos resultados

Nesta seção, são apresentados os resultados obtidos a partir da aplicação de estatísticas e dos algoritmos de AM no contexto da gestão educacional. Os dados processados foram analisados com o objetivo de identificar padrões e tendências que pudessem informar decisões administrativas, visando à personalização do ensino e à retenção dos alunos. A análise dos dados foi conduzida em etapas, considerando tanto a acurácia dos algoritmos selecionados quanto a sua aplicação prática em diferentes cenários escolares. Os resultados são organizados de modo a evidenciar como as técnicas de AM, integradas a uma administração educacional moderna, podem contribuir significativamente para apoiar no desempenho acadêmico e para a redução das taxas de evasão escolar.

3.7.1 Detalhamento do DataFrame

O DataFrame 1 contém 635 registros de alunos e 37 colunas, que incluem informações essenciais para a análise de desempenho acadêmico. As colunas capturam tanto variáveis

categóricas, como perfil comportamental (PC), sistema representacional (SR) e gênero, quanto variáveis numéricas, como as notas nas diferentes disciplinas.

Nesta etapa, o objetivo é apresentar a estrutura das colunas do DataFrame 1, sem detalhar seu conteúdo. A variável *target*, chamada "STATUS_BIN", é dicotômica, classificando os alunos como Ativos (0) ou Em Evasão (1). Esse DataFrame ainda não passou por transformações como o *one-hot encoding*, sendo uma versão preliminar que será usada para as primeiras análises.

O Apêndice J ilustra a estrutura desse DataFrame 1, destacando a disposição e o tipo das variáveis que serão utilizadas nas próximas etapas do processamento.

3.7.2 Limpeza de colunas irrelevantes do DataFrame

Nesta etapa de limpeza, foram realizadas operações para refinar o DataFrame e garantir que apenas as colunas relevantes fossem mantidas para o processo de AM. No DataFrame 2, uma das principais modificações foi a remoção da coluna categórica "Status", que continha os rótulos originais dos alunos. Isso foi necessário porque a coluna "STATUS_BIN" já foi criada como a variável alvo (*target*), categorizando os alunos como Ativos (0) ou Evasão (1).

Essa remoção de colunas irrelevantes assegura que o modelo não seja confundido com informações duplicadas ou redundantes, simplificando a estrutura do DataFrame 2 para otimizar o desempenho dos algoritmos.

O Apêndice K ilustra o resultado dessa etapa, mostrando o DataFrame 2 sem a coluna "Status", agora substituída pela variável binária "STATUS_BIN".

3.7.3 *One-hot encoding*

Após a aplicação da técnica de *one-hot encoding*, o DataFrame 3 foi atualizado para conter 45 colunas, das quais 12 são colunas binárias resultantes da conversão das variáveis categóricas originais, como "PC", "SR" e "Gênero". Essas colunas binárias, representadas como valores booleanos ("True" ou "False"), substituem as categorias originais, permitindo que os algoritmos de AM processem as informações de maneira mais eficiente. Esse processo garante que todas as variáveis estejam em um formato numérico consistente, eliminando as limitações impostas por variáveis categóricas, que não são diretamente processáveis por muitos algoritmos de ML.

O Apêndice L ilustra a nova estrutura do DataFrame 3, destacando a importância do *one-hot encoding* na preparação dos dados para a modelagem preditiva, assegurando que o modelo possa interpretar as variáveis de maneira adequada.

Também foi apresentada, no Apêndice L, a visualização da estrutura do DataFrame 3 após a aplicação do processo de *one-hot encoding*, que converte variáveis categóricas em colunas binárias, representadas por valores True/False. Esse processo transforma as variáveis categóricas, como "PC" e "SR", em múltiplas colunas booleanas, como "PC_Analista", "PC_Comunicador", "SR_Auditivo" e "SR_Visual", substituindo as categorias originais por representações numéricas que os algoritmos de AM podem processar de maneira rápida.

Na Figura 6, é possível observar como essas categorias foram expandidas em colunas binárias no DataFrame 3, resultando em um total de 45 colunas. Esse aumento de colunas reflete a desagregação das variáveis categóricas originais, facilitando a análise por algoritmos que dependem de dados numéricos.

A aplicação dessa técnica é indispensável, pois elimina a necessidade de os algoritmos manipularem diretamente variáveis não numéricas, o que poderia dificultar o aprendizado correto. Ao garantir que todas as variáveis estejam em um formato numérico binário, o *one-hot encoding* melhora a eficiência e a precisão da modelagem preditiva, especialmente quando o objetivo é prever o desempenho acadêmico dos alunos com base em múltiplos fatores comportamentais e representacionais.

Figura 6 – *Recorte do Resultado do Processo One-Hot Encoding no DataFrame (3)*

	PC_Analista	PC_Comunicador	PC_Executor	PC_Planejador	PC_SemDados	SR_Auditivo	SR_Cinestésico	SR_Digital	SR_SemDados	SR_Visual	Genero_F	Genero_M
0	False	False	False	False	True	False	False	False	True	False	False	True
1	True	False	False	False	False	False	False	False	False	True	False	True
2	True	False	False	False	False	False	False	True	False	False	False	True
3	True	False	False	False	False	False	False	False	False	True	False	True
4	True	False	False	False	False	False	False	False	False	True	True	False
5	True	False	False	False	False	False	False	False	False	True	False	True
6	True	False	False	False	False	False	False	False	False	True	False	True
7	False	False	False	False	True	False	False	False	True	False	False	True
8	True	False	False	False	False	False	False	False	False	True	True	False
9	True	False	False	False	False	False	False	False	False	True	True	False

Nota. Os valores True e False indicam a presença ou ausência de uma característica específica em cada observação do DataFrame, resultado da aplicação do *one-hot encoding*. No contexto dos dados, True sinaliza que a característica correspondente está presente, enquanto False indica sua ausência. Para os algoritmos de AM, esses valores booleanos são automaticamente interpretados como 1 (para True) e 0 (para False), permitindo que as variáveis categóricas sejam processadas de modo eficiente nos modelos preditivos. Fonte: Elaboração própria.

3.7.4 Treinamento e teste

Na Figura 7, observa-se a distribuição dos registros entre os conjuntos de treino e teste após a divisão do DataFrame 3 processado. Dos 635 registros totais, 444 registros foram alocados para o conjunto de treino (train_x e train_y), enquanto os 191 registros restantes foram reservados para o conjunto de teste (test_x e test_y). Essa divisão segue a proporção recomendada de 70% para treino e 30% para teste, o que garante uma base sólida para o modelo aprender e, posteriormente, ser avaliado de maneira precisa.

O conjunto de treino (train_x) inclui todas as variáveis preditoras (as colunas de informações dos alunos), como o PC, o SR e as notas nas disciplinas. Já o train_y, ou target (STATUS_BIN), contém a variável-alvo que o modelo precisa prever, que indica se o aluno está Ativo (0) ou em situação de Evasão (1). Essa separação de variáveis preditoras e variável-alvo é uma prática fundamental no treinamento de modelos de AM. O modelo é alimentado com as variáveis preditoras para aprender as relações subjacentes entre os dados e, em seguida, prever o STATUS_BIN (Y).

Figura 7 – *Separação para Treinamento e Teste*

```
train_x: 444, train_y: 444, test_x: 191 e test_y: 191 no total de 635 registros
```

Nota. A separação é 70% dos registros para treino e 30% para teste. Fonte: Elaboração própria.

A Figura 8 apresenta um exemplo de recorte dos dados utilizados para o treinamento (train_x), exibindo algumas das variáveis preditoras, como PC e notas em disciplinas. Note que a variável alvo STATUS_BIN não faz parte do train_x, pois o modelo precisa ser treinado sem saber o resultado, aprendendo as relações entre as variáveis. Apenas no conjunto train_y está presente o STATUS_BIN, que o modelo tentará prever após o aprendizado.

Figura 8 – *Recorte dos Dados que Serão Utilizados para Treinamento (train_x)*

	pk	PC_nota	PC_cursou	SS_nota	SS_cursou	SO_nota	SO_cursou	OAC_nota	OAC_cursou	LP_nota	...	PC_Executor	PC_Planejador	PC_SemDados
463	657	0.0	0	0.0	0	0.0	0	0.0	0	0.0	...	False	False	True
188	270	5.0	1	2.0	1	0.0	0	2.7	1	2.0	...	False	False	False
15	21	5.0	1	3.7	1	0.0	0	4.0	1	0.0	...	False	True	False
578	819	5.0	1	5.0	1	0.0	0	5.0	1	4.0	...	True	False	False
211	297	2.3	1	3.7	1	2.3	1	5.0	1	4.3	...	True	False	False

5 rows × 44 columns

Nota. 444 registros com as colunas preditoras separadas para treinamento. Fonte: Elaboração própria.

É importante destacar que a quantidade de dados utilizados nesta tese, com 635 registros, é considerada adequada para o contexto educacional. Embora esse volume de dados não seja comparável a estudos que utilizam milhões de registros, ele é suficiente para fornecer análises significativas e *insights* aplicáveis à gestão educacional. Em pesquisas que buscam prever o desempenho acadêmico, essa quantidade está dentro de uma faixa aceitável.

Na Figura 9, apresenta-se a separação da variável-alvo, STATUS_BIN, em relação ao conjunto de dados de treinamento (train_y). Essa figura 9 destaca que a quantidade de registros permanece igual, evidenciando a importância de manter a consistência nos dados ao dividir entre preditoras e a variável dependente. A presença do STATUS_BIN permite que o modelo aprenda a prever o *status* dos alunos, ao mesmo tempo que as variáveis preditoras oferecem o contexto necessário para o entendimento das relações subjacentes, otimizando o processo de aprendizado do algoritmo.

Figura 9 – *Recorte dos Dados do DataFrame train_y*

	STATUS_BIN
463	1
188	0
15	0
578	0
211	0
...	...
71	0
106	0
270	0
435	0
102	0

444 rows × 1 columns

Nota. 444 registros com a coluna dependente para treinamento. Fonte: Elaboração própria.

Após a separação entre train_x (dados de entrada) e train_y (variável alvo), o próximo passo é aplicar o modelo treinado aos dados de teste, que são representados por test_x e test_y. Embora test_x contenha as mesmas colunas preditoras, test_y possui apenas as saídas correspondentes, mas em menor quantidade. O algoritmo irá prever o *status* dos alunos com base nas informações de test_x, e essas previsões podem ser comparadas com os valores reais de test_y para avaliar a precisão do modelo.

3.7.5 Acurácia: performance ou medida de score

3.7.5.1 Acurácia de todos os algoritmos escolhidos

A Tabela 11 apresenta um comparativo de desempenho entre diversos modelos de AM, avaliados pela métrica de acurácia. Esses modelos foram treinados com um conjunto de dados composto por 444 registros de alunos (70% do total) e testados em um conjunto de 191 registros (30% do total). A acurácia é uma métrica importante porque indica a porcentagem de previsões corretas feitas pelo modelo.

Os resultados demonstram uma variação significativa na acurácia dos modelos, com alguns alcançando valores bastante representativos. Modelos como RFC e GBC tiveram as melhores *performances*, sugerindo que são mais adequados na tarefa de prever o desempenho acadêmico dos alunos. No entanto, além da acurácia, outras métricas como precisão, *recall*, F1-score, e AUC-ROC serão analisadas posteriormente para uma avaliação mais completa do desempenho dos algoritmos.

Tabela 11 – *Métricas de Acurácia*

Técnica	#	Algoritmo	Sigla	Acurácia[a]
Linear Models	1	Linear models (Ridge Regression)	LRR	0.665121
	2	Linear models (Logistic Regression)	LLR	0.937172
	3	Generalized Linear Models (Linear Regression)	LiR	0.656965
Linear and Quadratic Discriminant Analysis	4	Linear and Quadratic Discriminant Analysis	LDA	0.926701
Support Vector Machine	5	Support Vector Machine (Classification)	SVC	0.748691
	6	Support Vector Machine (Regression)	SVR	-
Stochastic Gradient Descent	7	Stochastic Gradient Descent (Classification)	SGDC	0.895287
	8	Stochastic Gradient Descent (Regression)	SGDR	-
Nearest Neighbors	9	Nearest Neighbors (Classification)	NNC	0.759162
	10	Nearest Neighbors (Regression)	NNR	0.213071
Naive Bayes	11	Naive Bayes (Gaussian)	NBG	0.827225
	12	Naive Bayes (Bernoulli)	NBB	0.853403
Decision Trees	13	Decision Tree (Classification)	DTC	0.926701
	14	Decision Tree (Regression)	DTR	0.554778
Ensemble methods	15	Gradient Tree Boosting (Classification)	GBC	0.947643
	16	Gradient Tree Boosting (Regression)	GBR	0.729831
	17	Random Forest (Classification)	RFC	0.952879
	18	Random Forest (Regression)	RFR	0.765073
Neural Network Models	19	Generalized Linear Models (Perceptron)	KNP	0.753926
	20	Neural Network (Classification)	KNC	0.921465
	21	Neural Network (Regression)	KNR	0.600378

Nota. Os algoritmos que não apresentam resultados na tabela foram excluídos devido à predição abaixo do normal. Como esses valores não são adequados para uma análise precisa ou significativa dentro do contexto desta tese, optou-se por não informar seus resultados na tabela. Fonte: Elaboração própria.

3.7.5.2 As cinco melhores acurácias

Na Tabela 12, são apresentados os cinco melhores resultados de acurácia dos algoritmos de AM avaliados. Esses resultados refletem a eficácia das diferentes técnicas utilizadas para prever o desempenho acadêmico dos alunos. Embora a acurácia seja uma métrica interessante, é igualmente importante considerar outros *scores*, como precisão, *recall* e F1-Score, para fortalecer o método.

Tabela 12 – *Os Cinco Melhores Resultados de Acurácia*

Técnica	Algoritmo	Sigla	Acurácia
Ensemble methods	Random Forest (Classification)	RFC	0.952879
Ensemble methods	Gradient Tree Boosting (Classification)	GBC	0.947643
Linear models	Logistic Regression	LLR	0.937172
Neural network models	Neural Network (Classification)	KNC	0.931937
Linear and Quadratic Discriminant Analysis	Linear and Quadratic Discriminant Analysis	LDA	0.926701

Nota. Classificado pela acurácia. Fonte: Elaboração própria.

3.7.5.3 Melhorando o resultado com outros scores

Na Tabela 13, são apresentados os *scores* adicionais dos cinco melhores algoritmos, que incluem métricas como precisão, *recall*, F1-Score e AUC-ROC. Essas métricas fornecem uma visão mais detalhada da *performance* de cada modelo, complementando a acurácia simples com uma análise mais abrangente. A avaliação dessas métricas é para escolher o algoritmo mais adequado, especialmente em situações em que o desbalanceamento de classes pode influenciar os resultados.

Tabela 13 – *Outros Scores para os Cinco Melhores Algoritmos*

Métricas	RFC	GBC	LLR	KNC	LDA
Acurácia (Simples)	0.9528	0.9476	0.9371	0.9319	0.9267
Validação Cruzada Acurácia Média	0.9706	0.9638	0.9571	0.9121	0.9436
Precisão	0.9333	0.9318	0.9285	0.9473	0.9250

Métricas	RFC	GBC	LLR	KNC	LDA
Recall	0.8750	0.8541	0.8125	0.7500	0.7708
F1-Score	0.9032	0.8913	0.8666	0.8372	0.8409
AUC-ROC	0.9595	0.9699	0.9656	0.9701	0.9596

Nota. Observe os maiores resultados para o melhor algoritmo. Fonte: Elaboração própria.

A tabela 13 mostra que o RFC é o algoritmo mais eficiente para o objetivo de predição, com a maior acurácia simples (0.9528), e a validação cruzada da acurácia média (0.9706). Além disso, ele apresenta um desempenho sólido nas métricas de precisão (0.9333) e F1-Score (0.9032), o que demonstra sua capacidade de equilibrar a classificação de alunos corretamente. O alto AUC-ROC (0.9595) também evidencia que o modelo RFC é capaz de distinguir bem entre as classes "Ativo" e "Evasão", tornando-o uma escolha ideal para continuar o trabalho de predição.

3.7.6 Uso do melhor algoritmo

Com base nos resultados obtidos, o RFC foi selecionado para avançar nas predições devido ao seu bom desempenho em várias métricas, como acurácia, precisão, *recall*, F1-Score e AUC-ROC. Esse algoritmo mostrou-se adequado para identificar padrões nos dados e fornecer previsões consistentes.

O RFC será aplicado a todo o DataSet disponível, gerando uma nova coluna chamada "Preditor", que conterá as predições do modelo em relação ao *status* dos alunos, classificados como "Ativo" (0) ou "Evasão" (1). Essa coluna servirá de base para análises estatísticas futuras, permitindo comparações entre diferentes cenários de intervenção.

Embora o RFC tenha apresentado bons resultados, é importante reconhecer que fatores como novos dados ou mudanças nas variáveis podem impactar a qualidade das predições. Assim, ajustes e melhorias no modelo podem ser necessários conforme o trabalho avança, para manter a relevância das previsões e sua aplicabilidade no contexto educacional.

3.7.7 Treinamento e teste com o melhor algoritmo

Após identificar o RFC como o algoritmo de melhor desempenho, foi realizada a etapa de treinamento e teste aplicando exclusivamente o RFC. Nessa fase, utilizam-se 70% dos dados para treinamento e os 30% restantes para teste, seguindo o mesmo procedimento estabelecido anteriormente, porém agora focado apenas no algoritmo escolhido.

Essa etapa visa validar a capacidade do RFC de prever corretamente o desempenho dos alunos ao separar os dados de maneira eficiente. O treinamento permite que o algoritmo aprenda as relações entre as variáveis preditoras (PC, notas, gênero, entre outras) e a variável-alvo STATUS_BIN (ativa ou evasão). No conjunto de teste, o RFC é então avaliado para verificar sua capacidade de fazer previsões com base em novos dados, que não foram utilizados no treinamento.

A Figura 10 ilustra um recorte das predições realizadas durante essa fase de teste. O modelo gera predições para os alunos com base nos padrões identificados durante o treinamento, permitindo avaliar a eficácia do RFC em classificar os alunos como "Ativos" (0) ou em risco de "Evasão" (1).

Figura 10 – *Recorte das Predições Realizadas com o Melhor Algoritmo*

	STATUS_BIN	RFC
523	1	1
250	0	0
428	0	0
228	0	0
208	0	0
..	...	...
75	1	1
92	1	1
333	0	0
68	0	0
296	0	0

[191 rows x 6 columns]

Nota. A coluna RFC será posteriormente mudada para Preditor. Fonte: Elaboração própria.

3.7.8 Predição com todo o DataFrame

Nesta fase, realiza-se a predição utilizando o RFC em 100% dos dados. Diferente das fases anteriores, que envolveram apenas treinamento e teste em subconjuntos dos dados, agora aplica-se o algoritmo a toda a base de dados, composta por 635 registros de alunos. Esse processo resulta na geração da coluna Preditor para todos os alunos, proporcionando uma visão completa e consolidada do desempenho preditivo do modelo.

O grande benefício desta etapa é que o modelo, agora treinado e testado, pode ser utilizado para prever o STATUS_BIN (ativo ou evasão) de todos os alunos, sem exceções. O mesmo processo pode ser escalado para conjuntos de dados maiores, como 10.000 ou mais registros, sem comprometer a integridade ou aplicabilidade do modelo. Novos dados podem ser incorporados em futuros DataFrames, e as predições serão feitas com base no aprendizado anterior do algoritmo.

A Figura 11 ilustra um exemplo dessa predição no DataSet original, onde a coluna Preditor foi gerada para cada aluno. Essa coluna indica o resultado da predição do RFC: 1 = Evasão e 0 = Ativo, conforme o algoritmo aprendeu durante as fases de treinamento e teste.

Figura 11 – *Recorte da Predição no DataSet Original*

PC	SR	Genero	pk	Status	STATUS_BIN	FE_cursou	BE_nota	BE_cursou	N_Disciplinas	Preditor
SemDados	SemDados	M	4	Trancado	1	0	0	0	2	1
Analista	Visual	M	6	Ativo	0	0	0	0	2	0
Analista	Digital	M	7	Ativo	0	0	0	0	9	0
Analista	Visual	M	8	Ativo	0	0	0	0	3	0
Analista	Visual	F	9	Ativo	0	0	4	1	10	0
Analista	Visual	M	10	Ativo	0	0	0	0	3	0
Analista	Visual	M	11	Ativo	0	1	0	0	13	0
SemDados	SemDados	M	12	Cancelado	1	1	0	0	2	1
Analista	Visual	F	13	Ativo	0	1	0	0	11	0
Analista	Visual	F	14	Ativo	0	0	0	0	4	0
Analista	Visual	M	15	Ativo	0	0	0	0	2	0
Analista	Visual	M	16	Ativo	0	0	0	0	6	0
Analista	Visual	M	17	Ativo	0	0	0	0	2	0
Analista	Digital	M	18	Ativo	0	0	0	0	5	0
SemDados	SemDados	M	19	Trancado	1	0	0	0	6	1

Nota. Este é o arquivo original com os dados dos alunos e a última coluna “Preditor” com a predição do melhor algoritmo, indicando o aprendizado obtido por ele, ou seja: 1 = Evasão e 0 = Ativo. Fonte: Elaboração própria.

3.7.9 Pontos de dados para a administração escolar

Os pontos de dados no processamento de AM referem-se a cada unidade de informação analisada pelo algoritmo para aprender e prever resultados. No contexto deste estudo, o algoritmo RFC foi escolhido após a avaliação de 21 algoritmos diferentes nos DataSet mensais, passando por um processo detalhado de análise de desempenho em cada disciplina. Essa avaliação culminou na aplicação do RFC ao DataSet consolidado de novembro, que processou todas as colunas de dados associadas a cada registro. Esse processo envolveu a análise de todas as variáveis de cada aluno, o que resultou em um volume significativo de pontos de dados.

A Tabela 14 ilustra a distribuição dos registros de alunos utilizados para treinamento e teste ao longo dos meses de abril a novembro de 2022. O número total de pontos de dados processados chegou a 10.724.265, um volume considerável que reflete o esforço computacional necessário para essa análise. À medida que o volume de dados cresce, especialmente em ambientes educacionais com muitos alunos e disciplinas, o custo de processamento também aumenta.

Ao utilizar plataformas profissionais pagas para o processamento de AM, como as oferecidas por serviços de nuvem, o custo pode ser calculado com base em fatores como o número de pontos de dados, o tempo de processamento ou o uso de recursos como GPUs. Portanto, é fundamental que os administradores escolares considerem esses custos ao planejar a implementação de modelos de predição em larga escala.

A análise dos custos de execução torna-se uma parte estratégica da gestão escolar, especialmente em ambientes que desejam adotar tecnologias preditivas para otimizar a gestão acadêmica. Esse planejamento permitirá que o administrador mantenha a viabilidade do projeto, mesmo à medida que mais dados são coletados e utilizados.

Tabela 14 – *Registros de Alunos Utilizados para Treinamento e Teste e seus Pontos de Dados*

Descrição	abr/22	mai/22	jun/22	jul/22	ago/22	set/22	out/22	nov/22	Médias
Treinamento	473	513	547	566	580	595	605	630	564
Teste	203	220	234	242	249	255	259	270	242
Total registros	676	733	781	808	829	850	864	900	805
[a] Pontos de dados para treino e teste de algoritmos (21)	638.820	692.685	738.045	763.560	783.405	803.250	816.480	850.500	760.843
[b] Pontos de dados mensais com o Preditor [(1)]	30.420	32.985	35.145	36.360	37.305	38.250	38.880	40.500	36.231
[c] Pontos de dados por disciplina (15) e Preditor (1)	456.300	494.775	527.175	545.400	559.575	573.750	583.200	607.500	543.459
[d] Total de pontos de dados	1.125.540	1.220.445	1.300.365	1.345.320	1.380.285	1.415.250	1.438.560	1.498.500	1.340.533

Nota. A fonte do texto da tabela foi diminuída para melhor visualização.
[a] Foram 45 colunas, multiplicadas por 676 registros, multiplicado por 21 algoritmos.
[b] Foram 45 colunas, multiplicadas por 676 registros, multiplicado por 1 algoritmo.
[c] Foram 45 colunas, multiplicadas por 676 registros, multiplicado por 15 disciplinas.
[d] Total = 10.724.265 pontos de dados. Fonte: Elaboração própria.

3.8 Estatística

Nesta seção, o objetivo foi validar as relações entre as variáveis do estudo e os *scores* gerados pelo algoritmo preditivo, a fim de responder às hipóteses levantadas ao longo da pesquisa. A principal meta foi avaliar como diferentes fatores se correlacionam com os resultados preditivos e, assim, verificar a eficácia do modelo na identificação de estudantes classificados como "Aptos" e "Não Aptos" ou, alternativamente, como "Ativos" e "Evasão".

Para alcançar essa validação, foi aplicada uma variedade de testes estatísticos. O Teste de Kappa foi utilizado para medir a concordância entre variáveis e o *score* preditivo, garantindo uma análise precisa sobre o quanto as classificações do algoritmo estão alinhadas com os dados reais dos alunos. Além disso, foram conduzidas análises de regressão linear e geométrica, que permitiram identificar a força e a significância das associações entre variáveis como o número de disciplinas cursadas, os perfis comportamentais e os sistemas representacionais.

Esses métodos estatísticos forneceram *insights* importantes sobre o impacto de cada variável no desempenho preditivo do modelo. Por exemplo, as disciplinas cursadas revelaram-se uma variável importante para determinar a previsão de sucesso ou evasão. Da mesma forma, as análises dos PCs e dos SRs forneceram evidências de que alguns grupos de alunos podem ter maior ou menor probabilidade de sucesso com base nesses fatores. Ao validar ou refutar as hipóteses levantadas, essa análise estatística consolidou o modelo e ajudou a identificar áreas para melhorias

e ajustes finos, com o objetivo de otimizar a personalização do ensino e as intervenções pedagógicas voltadas a alunos em risco de evasão.

Essas hipóteses são apresentadas na Tabela 15, que serve como base para a análise subsequente dos dados e a validação ou rejeição das hipóteses.

3.8.1 Hipóteses

A Tabela 15 apresenta a consolidação dos resultados de quatro hipóteses testadas na tese. Cada hipótese é apresentada integralmente, junto à indicação de "Aceita" ou "Rejeitada". As hipóteses analisam a relação entre o *score* gerado pelo algoritmo e diversos aspectos, como a identificação dos estudantes "Não Aptos" e "Aptos", a influência das disciplinas e outras variáveis observadas. A tabela organiza essas informações de maneira clara, permitindo uma visão geral dos resultados obtidos.

Tabela 15 – *Resultados das Hipóteses*

Hipótese	Descrição	Resultado
A	O *score* gerado pelo algoritmo tem acurácia suficiente para discriminar os estudantes “Não Aptos" e “Aptos” para direcioná-los aos caminhos de aprendizagem.	Aceita
B	Há disciplinas que apresentam real *correspondência* com o *score* gerado pelo algoritmo e, portanto, têm papel mais relevante para identificar os estudantes “Não Apto" e “Apto”.	Aceita
C	O *score* gerado pelo algoritmo apresenta real *diferença* conforme os Perfis Comportamentais observados nos participantes do estudo.	Rejeita
D	O *score* gerado pelo algoritmo apresenta real *diferença* conforme os Sistemas Representacionais observados nos participantes do estudo.	Aceita

Fonte: Elaboração própria.

3.8.2 Testes

3.8.2.1 Teste de Kappa da Hipótese A – Score

A Hipótese A avaliou se o *score* gerado pelo algoritmo é capaz de diferenciar entre estudantes "Não Aptos" e "Aptos" para direcioná-los aos caminhos de aprendizagem.

O Teste de Kappa, também conhecido como Kappa de Cohen, é uma estatística que mede a concordância entre duas classificações, ajustando-se pela chance de concordância. É amplamente utilizado em pesquisas e contextos educativos para avaliar a eficácia de um algoritmo de classificação. Um valor de Kappa próximo a 1 indica alta concordância entre as classificações, enquanto valores próximos a 0 sugerem que a concordância é similar ao acaso. Esse teste é

essencial para validar a precisão de predições em relação a categorias, como estudantes "Aptos" e "Não Aptos".

A Figura 12 apresenta os resultados do Teste de Kappa, que avalia a concordância entre as classificações dos estudantes "Aptos" e "Não Aptos". Os dados indicam uma concordância observada de 0.9575, sugerindo uma alta consistência entre as classificações. O valor de Kappa de 0,8901 reforça essa ideia, indicando que o algoritmo possui a capacidade de discriminação. Com um p-valor inferior a 0.0001, é possível aceitar a hipótese, confirmando que o preditor é eficiente para classificar os estudantes corretamente.

Figura 12 – *Teste Kappa para Scores*

```
- Teste de Kappa -
-------------------------------------------------
Dados Fornecidos
-------------------------------------------------
     A = 455    B = 15
     C = 12     D = 153

RESULTADO
-------------------------------------------------
Concordância observada: 0.9575
Concordância esperada.: 0.6131
Kappa.................: 0.8901
Z (Kappa).............: 22.4317
(p-valor).............: < 0.0001
-------------------------------------------------
```

Fonte: Elaboração própria, utilizando o BioStat 5.3.

O p-valor indica a probabilidade de observar resultados extremos sob a hipótese, com valores abaixo de 0.05 sugerindo significância estatística. O valor de Z reflete a quantos desvios-padrão o resultado está da média, com valores acima de 1.96 indicando significância a 5%. Juntos, esses parâmetros avaliam a confiabilidade dos resultados obtidos, essencial para a validação da hipótese em testes estatísticos.

Para a Hipótese B, focaram-se somente alunos que cursaram a disciplina, analisando tanto os aprovados quanto os reprovados. O Teste de Kappa foi utilizado para avaliar a correspondência entre as disciplinas cursadas e o *score* gerado pelo algoritmo. A hipótese propõe que há uma relação real entre as disciplinas e os resultados, o que pode impactar diretamente a identificação dos alunos "Não Aptos" e "Aptos".

3.8.2.2 Teste de Kappa para disciplinas mais influentes

O Teste de Kappa foi aplicado separadamente para cada uma das 15 disciplinas cursadas, com o objetivo de avaliar a correspondência entre o desempenho dos alunos e os *scores* preditivos

gerados pelo algoritmo. Essa abordagem permitiu uma análise detalhada de como cada disciplina influencia as classificações dos alunos. Os resultados, na Tabela 16, revelaram que a disciplina de Engenharia de Requisitos e Modelagem de Software (disc.ERMS) apresentou uma correspondência estatisticamente significativa, indicando um papel relevante na discriminação entre os alunos "Não Aptos" e "Aptos". Embora as demais disciplinas não tenham mostrado concordância significativa, a identificação dessa disciplina-chave facilita a compreensão dos padrões de desempenho e a eficácia do algoritmo em contextos específicos. Esses achados servirão como base para uma avaliação final abrangente e para direcionar possíveis intervenções pedagógicas.

Tabela 16 – *Resultado do Teste de Kappa para cada Disciplina*

Disciplina	Concordância Observada	Concordância Esperada	Kappa	Z (Zappa)	p-valor	Replicabilidade
disc.ERMS	0.8077	0.6981	0.3631	6.6902	< 0.0001	Fraca
disc.FE	0.6789	0.6872	-0.0263	1.3423	< 0.0001	Fraca
disc.PDP	0.8031	0.8148	-0.0629	1.1789	0.1192	Fraca
disc.EOVS	0.9497	0.9510	-0.0255	0.3429	0.3658	Fraca
disc.SQL	0.9628	0.9631	-0.0092	0.1821	0.4278	Fraca
disc.MA	0.9071	0.9032	0.0404	0.6244	0.2662	Fraca
disc.OAC	0.7716	0.7586	0.0542	1.5208	0.0642	Fraca
disc.MD	-	-	-	-	-	-
disc.LP	0.7068	0.6880	0.0603	1.3270	0.0923	Fraca
disc.SO	0.2367	0.2389	-0.0029	0.2650	0.3955	Fraca
disc.PC	0.8889	0.8800	0.0741	1.4447	0.0743	Fraca
disc.IP	0.7561	0.7652	-0.0386	0.8808	0.1892	Fraca
disc.SS	0.9136	0.9172	-0.0442	0.8996	0.1842	Fraca
disc.BE	-	-	-	-	-	-
disc.git	-	-	-	-	-	-

Nota. As disciplinas **disc.MD**, **disc.BE** e **disc.git** não apresentaram resultados no Teste de Kappa devido à insuficiência de dados ou à falta de variabilidade nas respostas. Isso impossibilitou a realização da análise estatística adequada utilizando o *software* BioStat 5.3. Fonte: Elaboração própria, usando o BioStat 5.3.

Com base nos resultados obtidos das 15 disciplinas analisadas, conclui-se que a hipótese foi confirmada, principalmente devido à disciplina de Engenharia de Requisitos e Modelagem de Software (disc.ERMS). Essa disciplina apresentou uma concordância observada de 0.8077 e um coeficiente Kappa de 0.3631, com um p-valor inferior a 0.0001. Esses resultados indicam uma concordância significativa entre o *score* gerado pelo algoritmo e os resultados dos alunos,

evidenciando o papel relevante dessa disciplina na identificação de estudantes "Aptos" e "Não Aptos".

Para as demais disciplinas, embora algumas tenham apresentado p-valores baixos, os coeficientes Kappa foram negativos ou próximos de zero, indicando que não há concordância significativa nessas áreas.

Em síntese, a presença de pelo menos uma disciplina com correspondência significativa confirma a hipótese. Isso sugere que intervenções pedagógicas direcionadas à disciplina ERMS podem contribuir para aprimorar o modelo e melhorar a identificação dos estudantes. A revisão contínua dos métodos de ensino nessa área é recomendada para fortalecer ainda mais o modelo proposto.

3.8.2.3 Teste de Qui-Quadrado

Com base nos resultados apresentados na Tabela 17, que exibe a contingência dos scores gerados pelo algoritmo em relação aos PCs — Analista, Comunicador, Executor e Planejador —, realizou-se uma análise para verificar a existência de uma associação significativa entre os *scores* preditivos e os diferentes perfis comportamentais dos participantes do estudo.

Tabela 17 – *Contingência – Teste de Qui-Quadrado – Hipótese C*

	Resultado
Tabela de Contingência	2 x 4
Qui-Quadrado	3.206
Graus de Liberdade	3
p-valor (p)	0.3609

Nota. Fonte: Elaboração própria, usando o BioStat 5.3.

A aplicação do teste do Qui-Quadrado resultou em um valor de 3.206 com 3 graus de liberdade e um p-valor de 0.3609. Como esse p-valor é superior ao nível de significância-padrão de 0,05, não há evidência estatística suficiente para aceitar a hipótese. Portanto, conclui-se que não há diferenças estatisticamente significativas entre os *scores* gerados pelo algoritmo e os PCs observados.

Esse resultado sugere que os PCs — Analista, Comunicador, Executor e Planejador — não influenciam de maneira significativa os *scores* gerados pelo algoritmo. Consequentemente, os PCs dos estudantes não são um fator determinante na classificação entre "Não Aptos" e "Aptos" nesse modelo específico.

3.8.2.4 Distribuição de participações

Para complementar a análise da Hipótese C, foi elaborada a Tabela 18, que apresenta a distribuição dos estudantes classificados como "Ativo" e "Evasão" em relação aos diferentes PCs — Analista, Comunicador, Executor e Planejador. A Tabela 18 fornece tanto os números absolutos quanto os percentuais de participação de cada perfil nos dois grupos de estudantes.

Tabela 17 - *Distribuição de Estudantes "Ativo" e "Evasão" por Perfil Comportamental*

Perfil Comportamental	Ativo	Evasão	Total	Ativo	Evasão
Analista	143	17	160	89%	11%
Comunicador	168	16	184	91%	9%
Executor	64	11	75	85%	15%
Planejador	92	7	99	93%	7%

Fonte: Elaboração própria.

A análise dos dados da Tabela 18 revela que a proporção de estudantes "Ativo" é elevada em todos os PCs, variando de 85% a 93%. Da mesma forma, a proporção de estudantes em "Evasão" é relativamente baixa em todos os perfis, variando de 7% a 15%. Observa-se que não há diferenças acentuadas nas porcentagens de "Ativo" e "Evasão" entre os diferentes PCs.

Essas proporções semelhantes indicam que a distribuição de estudantes "Ativo" e "Evasão" é bastante homogênea entre os PCs. Por exemplo, o perfil Planejador possui a maior taxa de estudantes "Ativo" (93%), enquanto o perfil Executor apresenta a menor taxa (85%), mas a diferença entre eles é de apenas 8 pontos percentuais. Essa pequena variação não é suficiente para indicar uma diferença significativa.

A Hipótese D busca determinar se o *score* gerado pelo algoritmo varia significativamente conforme os SRs observados nos participantes do estudo. A hipótese propõe que existe, de fato, uma diferença relevante, indicando que o *score* pode variar de acordo com os SRs dos estudantes.

3.8.2.5 Teste de Qui-Quadrado

Para aprofundar a análise da Hipótese D (psicometria SR), foi consolidada a distribuição dos estudantes classificados como "Ativo" e "Evasão" em relação aos diferentes SRs — Auditivo, Cinestésico, Visual e Digital. A Tabela 19 apresenta a contingência que ilustra essa distribuição, permitindo visualizar como cada SR está associado aos *scores* preditivos gerados pelo algoritmo.

Tabela 18 – *Contingência – Teste de Qui-Quadrado – Hipótese D*

	Resultado
Tabela de Contingência	2 x 4
Qui-Quadrado	19.529
Graus de Liberdade	3
p-valor (p)	0.0002

Fonte: Elaboração própria, usando o BioStat 5.3.

O valor de p = 0.0002 é significativamente menor que o nível de significância-padrão de 0,05, indicando que há evidência estatística suficiente para aceitar a hipótese, concluindo que o *score* gerado pelo algoritmo apresenta diferenças estatisticamente significativas conforme os SRs observados nos participantes do estudo.

Para aprofundar a análise da Hipótese D (psicometria SR), foi elaborada a Tabela 20, que apresenta a distribuição dos estudantes classificados como "Ativo" e "Evasão" em relação aos diferentes SRs — Auditivo, Cinestésico, Visual e Digital. A Tabela 20 consolida os dados dos preditores do algoritmo para cada SR, permitindo uma comparação detalhada das taxas de retenção e evasão entre os diferentes perfis.

Tabela 19 – *Distribuição de Estudantes "Ativo" e "Evasão" por Sistemas Representacionais*

Sistema Representacional	Ativo	Evasão	Total	Ativo	Evasão
Auditivo	87	5	92	94,57%	5,43%
Cinestésico	110	5	115	95,65%	4,35%
Visual	96	3	99	96,97%	3,03%
Digital	163	30	193	84,46%	15,54%

Fonte: Elaboração própria.

A Tabela 20 revela diferenças significativas nas taxas de "Ativo" e "Evasão" entre os diferentes SRs: o Sistema Visual apresenta a maior taxa de estudantes "Ativo" (96,97%) e a menor taxa de "Evasão" (3,03%), indicando que estudantes com preferência visual tendem a permanecer mais engajados e ativos no curso; o Sistema Cinestésico também exibe uma alta taxa de "Ativo" (95,65%) e baixa taxa de "Evasão" (4,35%), sugerindo que estudantes que aprendem por meio de experiências práticas e movimento mantêm-se envolvidos nas atividades acadêmicas; o Sistema Auditivo possui uma taxa de "Ativo" ligeiramente menor (94,57%) e uma taxa de "Evasão" de 5,43%, ainda assim demonstrando alta retenção entre os estudantes auditivos; já o Sistema Digital destaca-se por ter a menor taxa de "Ativo" (84,46%) e a maior taxa de "Evasão" (15,54%) entre todos os SRs, o que indica que estudantes com preferência pelo sistema digital apresentam maior propensão à evasão.

Isso confirma a influência significativa dos SRs nos *scores* gerados pelo algoritmo e nas taxas de "Ativo" e "Evasão" dos estudantes. Esses achados reforçam a relevância de considerar os perfis de aprendizagem ao elaborar estratégias educacionais e modelos preditivos. Ao atender às necessidades específicas de cada SR, é possível melhorar o engajamento, reduzir a evasão e promover o sucesso acadêmico dos estudantes.

3.9 Caminhos de aprendizagem

O modelo de AM proposto nesta tese sugere a aplicação de caminhos de aprendizagem no AVA para monitorar e apoiar o progresso de alunos em risco, conectando teorias educacionais, como a TDA e a TEM, com práticas de gestão de ensino. Esse processo contínuo de monitoramento e apoio visa garantir uma intervenção adaptativa que promova o sucesso acadêmico e a retenção dos estudantes.

Os caminhos de aprendizagem apresentados no Apêndice M mostram as diferentes trajetórias que os alunos podem seguir, dependendo de suas classificações como "Aptos" ou "Não Aptos". Essas trajetórias são ajustadas conforme a avaliação de desempenho atual, e o Canvas Instructure, utilizado nesse contexto, permite um aprofundamento ainda maior dos caminhos de aprendizado conforme as necessidades de reforço são identificadas.

Alunos Não Aptos – Provável Evasão ou Reprovação:

- Revisão e Reforço: Alunos classificados com Nota até 1 são direcionados para revisar conteúdos-base e reforçar o último conteúdo estudado. Esse reforço está em consonância com a TDA, que enfatiza a necessidade de estímulos contínuos e reforços positivos para promover mudanças comportamentais desejáveis, de acordo com as diretrizes de equidade da BNCC.
- Testes Contínuos e Progressão: Para progredirem, esses alunos participam de novos testes de múltipla escolha. Conforme demonstram melhoria, são encaminhados para etapas que envolvem a revisão e o reforço adicionais até atingirem a Nota 3.

Alunos Aptos – Provável Aprovação:

- Aprofundamento do Conteúdo: Alunos considerados aptos avançam para o aprofundamento dos tópicos atuais da grade curricular. Esse aprofundamento está alinhado

à TEM, onde os alunos são incentivados a buscar novos desafios com a garantia de reconhecimento e recompensas significativas.

- Selos de Certificação: Após demonstrarem sucesso nas etapas de aprofundamento, os alunos recebem selos de certificação, atuando como recompensas e reforçando a instrumentalidade e valência dos esforços, de acordo com a TEM. O Canvas facilita esse reconhecimento contínuo e adaptativo.

Os caminhos de aprendizagem são ajustados continuamente, formando um ciclo contínuo. Assim, cada intervenção no AVA não apenas oferece reforço ou recompensa, mas também alimenta o próximo ciclo de aprendizado e monitoramento, garantindo que alunos "Não Aptos" recebam o apoio necessário até se tornarem "Aptos" e possam avançar em sua jornada educacional.

Com essa abordagem adaptativa e fundamentada em teorias educacionais, o modelo promove um suporte eficiente e personalizado. Ele garante que as intervenções sejam rápidas e ajustadas de acordo com o desempenho de cada aluno, criando um ambiente de aprendizagem que evolui constantemente, visando sempre o sucesso do estudante.

3.10 IA generativa integrada com caminhos de aprendizagem

Durante o desenvolvimento do Modelo de Aprendizagem de Máquina, inicialmente, o uso da IA generativa para criar conteúdos pedagógicos de reforço não estava previsto. O plano original envolvia que professores e coordenadores elaborassem esses materiais. No entanto, a revisão da literatura e os testes práticos revelaram o grande potencial da IA generativa, especialmente no contexto de criação de conteúdos personalizados e testes de múltipla escolha, com gabaritos e feedbacks tanto para respostas corretas quanto incorretas. Para esses testes, foi utilizado o ChatGPT-4, mas a expectativa é que futuras versões dessa tecnologia tragam ainda mais benefícios.

Um exemplo dessa aplicação está descrito no Apêndice M, que demonstra como os caminhos de aprendizagem no Canvas Instructure foram automatizados para encaminhar alunos aos módulos específicos de reforço. Os resultados mostraram que, especialmente em disciplinas como Engenharia de Requisitos e Modelagem de Software (ERMS), onde muitos estudantes apresentaram dificuldades, a IA gerou conteúdos detalhados e testes de múltipla escolha alinhados com os temas tratados, como Modelagem de Requisitos e Casos de Uso.

Essa abordagem tornou possível oferecer uma personalização de ensino que respondia diretamente às necessidades de cada aluno. O conteúdo gerado pela IA foi adicionado ao Canvas

por meio da funcionalidade "Caminhos do Domínio", que automatiza o encaminhamento de alunos a novos conteúdos e atividades. Assim, para estudantes com dificuldades, por exemplo, em Testes e Qualidade de Software, a plataforma direcionou automaticamente esses alunos a materiais de reforço e testes adicionais adaptados às suas lacunas.

O uso da IA gerou resultados adequados, mas a qualidade do conteúdo produzido dependeu muito do material base disponível. Quando o material original era superficial, os conteúdos gerados pela IA refletiam essa limitação, mostrando que uma base de qualidade é fundamental.

A configuração dos caminhos de aprendizagem no Canvas, embora eficiente, envolveu um esforço manual considerável, especialmente para parametrizar cada trilha de aprendizado. Para otimizar essa implementação, uma possível melhoria seria a integração via API, o que reduziria significativamente a necessidade de intervenção manual e maximizaria a eficiência da IA generativa.

Além disso, a utilização do ChatGPT foi discutida na apresentação dos artigos relacionados a esta tese, envolvendo feedback e debates com coordenadores e professores universitários, que destacaram o impacto positivo na produção de conteúdos educacionais, especialmente em um contexto de limitação de recursos humanos.

Assim, a integração da IA generativa com os caminhos de aprendizagem não apenas automatiza o suporte aos alunos, como também garante um conteúdo de reforço alinhado com as disciplinas e as necessidades específicas dos estudantes. Essa abordagem enriquece o Modelo de Aprendizagem de Máquina proposto, proporcionando um ciclo contínuo de melhoria e personalização educacional.

3.11 Discussões

3.11.1 Análise dos resultados

Os resultados desta tese demonstram a aplicação de algoritmos de AM no contexto da administração educacional, com o objetivo de prever o desempenho acadêmico dos alunos e identificar padrões associados à evasão. A análise das hipóteses levantadas trouxe descobertas importantes sobre os fatores que influenciam a eficácia dos modelos preditivos.

A Hipótese A, que avaliou a capacidade do score preditivo em distinguir entre estudantes "Aptos" e "Não Aptos", apresentou resultados positivos. O teste de Kappa indicou alta concordância entre as classificações, confirmando que o algoritmo é eficiente em realizar essa

distinção. Essa alta acurácia valida o modelo como uma ferramenta útil para identificar alunos em diferentes caminhos de aprendizagem, direcionando-os para intervenções adequadas.

A Hipótese B focou o impacto das disciplinas sobre os scores preditivos. Os resultados indicaram que a disciplina Engenharia de Requisitos e Modelagem de Software (ERMS) exerce uma influência significativa no desempenho preditivo, sugerindo que essa disciplina é fundamental para distinguir alunos "Aptos" de "Não Aptos". Outras disciplinas não apresentaram o mesmo impacto, o que reforça a importância de identificar áreas específicas que afetam o sucesso dos alunos.

Em relação à Hipótese C, que investigou os PCs (Analista, Comunicador, Executor e Planejador), a hipótese foi rejeitada. Isso indica que, estatisticamente, os perfis comportamentais não influenciam significativamente os scores preditivos. Embora tenham sido observadas variações descritivas em perfis como "Executor" e "Analista", essas diferenças não se traduziram em relevância estatística para o modelo preditivo.

A Hipótese D, que analisou os SRs (Auditivo, Cinestésico, Visual e Digital), revelou diferenças significativas. O sistema Visual foi o que apresentou a maior retenção de alunos "Ativos", enquanto o sistema Digital teve a maior taxa de evasão. Esses resultados sugerem que os estilos de aprendizagem exercem um impacto importante no desempenho acadêmico e podem ser considerados na formulação de intervenções pedagógicas mais focadas.

Em resumo, os resultados indicam que, embora os PCs e o número de disciplinas cursadas não tenham mostrado influência significativa, os SRs e disciplinas específicas, como ERMS, desempenham um papel importante no desempenho preditivo. Isso sugere que uma abordagem mais personalizada, que leve em consideração os estilos de aprendizagem e o desempenho em disciplinas-chave, pode ser eficiente na melhoria das predições do desempenho acadêmico e na redução da evasão escolar.

3.11.2 Implicações práticas

A personalização do ensino, mediada pela tecnologia, é uma das principais contribuições práticas desta tese. A integração de algoritmos de AM com plataformas educacionais, como o *Canvas Instructure*, possibilita a criação de rotas de aprendizado adaptativas, otimizando o desempenho acadêmico dos alunos. Esse modelo personalizado permite intervenções mais cirúrgicas, além de reduzir custos ao automatizar processos como a configuração de atividades corretivas e a gestão de alunos em risco de evasão.

A utilização de ferramentas como a BNCC e os métodos de EDM permite que as instituições ajustem continuamente suas práticas educacionais de acordo com diretrizes nacionais, ao mesmo tempo que se beneficiam da retroalimentação dos dados preditivos. Esse ajuste contínuo garante que as previsões do modelo de AM se adaptem às necessidades em constante evolução dos alunos, promovendo uma educação mais centrada.

3.11.3 Limitações do estudo

Uma limitação relevante foi a ausência de variáveis relacionadas à frequência dos alunos, visualização de conteúdo e participação em fóruns de discussões. A inclusão dessas informações seria fundamental para um estudo mais profundo sobre o comportamento dos alunos em plataformas *on-line*, ajudando a validar teorias como a TDA e a TEM. A integração de dados comportamentais adicionais poderia aumentar a precisão dos modelos preditivos e permitir intervenções pedagógicas mais personalizadas.

Outro ponto a ser considerado é a possibilidade de ampliar o volume de dados analisados, incluindo novas variáveis que abordem aspectos socioeconômicos e psicológicos. Essa expansão permitiria uma compreensão mais abrangente dos fatores que influenciam o desempenho acadêmico, potencialmente promovendo soluções ainda mais personalizadas para atender às necessidades dos alunos.

3.11.4 Perspectivas futuras

A integração de modelos preditivos com sistemas de gestão educacional, como o Canvas Instructure, oferece um grande potencial para o futuro da administração educacional. A automatização de processos reduz a necessidade de configurações manuais, facilitando o uso dos algoritmos em larga escala e permitindo a personalização do ensino de modo mais eficiente e acessível.

Outra perspectiva promissora é o uso de IA generativa na criação de conteúdo e avaliações personalizadas. A construção de Large Language Models (LLMs) específicos para o contexto educacional brasileiro pode transformar o ensino, proporcionando a geração de materiais e estratégias pedagógicas mais adequadas às necessidades dos alunos e alinhadas com as diretrizes educacionais locais. O avanço dessas tecnologias não só facilitará a personalização do aprendizado, mas também contribuirá para uma maior equidade no acesso a recursos educacionais de alta qualidade.

3.11.5 Reflexão final sobre o uso da IA na educação

O uso crescente da IA na administração educacional oferece novas oportunidades para a criação de soluções integradas, combinando modelos preditivos, algoritmos generativos e sistemas de gestão. Essa abordagem não só melhora o monitoramento e a previsão do desempenho acadêmico, mas também possibilita intervenções proativas para reduzir a evasão e promover a excelência no aprendizado.

Ao integrar dados provenientes de fontes heterogêneas — como frequência, desempenho acadêmico e participação em atividades extracurriculares —, a IA amplia as possibilidades de personalização do ensino. Ferramentas de IA podem se conectar diretamente a sistemas de gestão educacional, criando um ecossistema de ensino adaptativo que responde de maneira mais eficiente e ágil às necessidades dos alunos e às mudanças no ambiente educacional. Essa sinergia entre IA e educação posiciona as instituições de ensino para enfrentar os desafios de um cenário educacional em constante evolução.

4 CONCLUSÃO

O objetivo desta tese foi aplicar o uso de algoritmos de Aprendizagem de Máquina (AM) para prever o desempenho acadêmico dos alunos, como descrito no título, e desenvolver um modelo preditivo capaz de validar essa abordagem. O foco esteve tanto na identificação de riscos de evasão quanto na criação de intervenções personalizadas, com a finalidade de melhorar o percurso educacional. Além disso, foi proposta a integração dessas técnicas preditivas em uma solução educacional, visando à personalização do ensino, otimização dos recursos pedagógicos e modernização da administração educacional.

Os resultados mostraram que o modelo preditivo proposto foi eficiente em identificar padrões associados ao sucesso acadêmico e à evasão. O algoritmo Random Forest Classification (RFC) emergiu como o mais adequado, atingindo uma acurácia de 0.9529, superando outros algoritmos, como o Gradient Boosting Classifier (GBC), com 0.9476, e a Regressão Logística (LLR), com 0.9371. Isso fez do RFC o modelo principal para categorizar alunos como "Aptos" ou "Não Aptos", proporcionando uma base sólida para intervenções pedagógicas.

Respondendo à pergunta central — "Como o uso de algoritmos de AM pode prever o desempenho do aluno?" —, esta pesquisa demonstrou que algoritmos de AM são eficazes em prever o desempenho acadêmico e em identificar alunos em risco de evasão, permitindo a aplicação prática de intervenções baseadas em dados, de forma precisa.

A análise das hipóteses revelou nuances importantes:

- **Hipótese A**: Confirmou e validou a eficácia do modelo em distinguir entre alunos "Aptos" e "Não Aptos" por meio do score preditivo.
- **Hipótese B**: Destacou a relevância de disciplinas como ERMS no desempenho preditivo, sugerindo que o foco em disciplinas específicas pode aprimorar a personalização do ensino.
- **Hipótese C**: O resultado gerado pelo algoritmo não apresenta real diferença conforme os PCs observados nos participantes do estudo.
- **Hipótese D**: Validou que há evidência estatística suficiente para aceitar a hipótese. Embora os Perfis Comportamentais (PCs) não tenham impactado significativamente os scores, os sistemas representacionais, especialmente o Visual, mostraram maior retenção de alunos "Ativos", enquanto o Digital apresentou maior taxa de evasão, indicando a necessidade de adaptações pedagógicas aos estilos de aprendizagem.

Um dos principais avanços práticos desta tese foi o desenvolvimento de um modelo de personalização do ensino baseado em dados preditivos. O modelo proposto não se limita a uma análise estática, mas opera como um sistema dinâmico de monitoramento contínuo e adaptativo, criando um ciclo de retroalimentação. Ao integrar algoritmos de AM com plataformas educacionais, como o Canvas Instructure, foi possível criar rotas de aprendizado adaptativas, otimizando o desempenho acadêmico e automatizando processos de gestão de risco.

Neste modelo, cada aluno é monitorado em tempo real. Quando identificado como "Não Apto", o aluno é encaminhado para atividades específicas de reforço, enquanto alunos classificados como "Aptos" são direcionados para trilhas que promovem aprofundamento e motivação contínua. Esse ciclo de retroalimentação, fundamentado nas teorias da Expectativa da Motivação (TEM) e da Dependência do Aprendizado (TDA), contribui para intervenções mais personalizadas, ajustadas continuamente ao progresso de cada aluno, e alinhadas com os princípios de equidade da Base Nacional Comum Curricular (BNCC).

A solução proposta está em sintonia com as práticas de gestão educacional moderna, promovendo a adaptabilidade e a centralização de dados para decisões estratégicas. A integração com a BNCC e o uso de técnicas de Educational Data Mining (EDM) permitiram que as instituições ajustassem continuamente suas práticas de ensino, retroalimentando os modelos preditivos para otimização constante.

Em relação aos vieses, todos os dados foram de fonte primária e processados conforme as melhores práticas. Não houve manipulação indevida dos dados, assegurando integridade e confiabilidade nos resultados. A metodologia seguiu padrões rigorosos para mitigar possíveis vieses, garantindo que as predições fossem justas e equilibradas para diferentes perfis de alunos.

Embora o estudo tenha enfrentado limitações, como a necessidade de uma base de dados mais extensa e variada, as contribuições são significativas para o campo da educação personalizada. O uso de IA generativa e a automação em larga escala, com base em dados preditivos, oferecem um caminho promissor para a modernização da gestão educacional, com potencial para melhorar a eficácia pedagógica e reduzir a evasão escolar.

Em conclusão, a Tese evidencia o uso efetivo de algoritmos de Aprendizagem de Máquina para prever o desempenho dos alunos, cria e testa um sistema dinâmico e proativo. Não apenas permite prever o desempenho com precisão, mas também ajusta intervenções em tempo real, proporcionando uma experiência educacional mais personalizada e eficiente, garantindo que cada aluno receba o suporte necessário ao longo de sua jornada educacional. Isso valida o modelo como um componente central para a previsão e melhoria do desempenho acadêmico e a retenção de estudantes.

REFERÊNCIAS

Abdi, A. (2016). *Three types of Machine Learning Algorithms*. https://doi.org/10.13140/RG.2.2.26209.10088

Agrawal, S., Vishwakarma, S. K., & Sharma, A. K. (2017). Using data mining classifier for predicting student's performance in UG level. *International Journal of Computer Applications*, *172*(8), 39–44. https://doi.org/10.5120/ijca2017915201

Ahmad N., Hassan, N., Jaafar, H., & Enzai, N. I. M. (2021). Students' performance prediction using artificial neural network. *IOP Conference Series: Materials Science and Engineering*, *1176*(1), 012020. https://doi.org/10.1088/1757-899X/1176/1/012020

Ahmad, S., Aoun, N. B., Ali, G., El-Affendi, M. A., & Anwar, M. S. (2023). Multi-clustered mathematical model for student cognitive skills prediction optimization. *IEEE Access*, *11*, 65371–65381. https://doi.org/10.1109/ACCESS.2023.3285612

Ahmad, S., El-Affendi, M. A., Anwar, M. S., & Iqbal, R. (2022). Potential future directions in optimization of students' performance prediction system. *Computational Intelligence and Neuroscience*, *2022*, 1–26. https://doi.org/10.1155/2022/6864955

Akuma, S., & Abakpa, H. (2021). Predicting undergraduate level students' performance using regression. *Nigerian Annals of Pure and Applied Sciences*, *4*(1), 129–139. https://doi.org/10.46912/napas.224

Alamri, R., & Alharbi, B. (2021). Explainable student performance prediction models: A systematic review. *IEEE Access*, *9*, 33132–33143. https://doi.org/10.1109/ACCESS.2021.3061368

Allen-Zhu, Z. (2017). Katyusha: The first direct acceleration of stochastic gradient methods. *Journal of Machine Learning Research*, *18*(221), 1–51. https://www.jmlr.org/papers/v18/16-410.html

Al-Shabandar, R., Hussain, A., Laws, A., Keight, R., Lunn, J., & Radi, N. (2017). Machine learning approaches to predict learning outcomes in Massive open online courses. *2017 International Joint Conference on Neural Networks (IJCNN)*, 713–720. https://ieeexplore.ieee.org/document/7965922

Anderson, H., Boodhwani, A., & Baker, R. (2019). *Assessing the fairness of graduation predictions*. Educational Data Mining. https://www.semanticscholar.org/paper/Assessing-the-Fairness-of-Graduation-Predictions-Anderson-Boodhwani/d0fe8bf3d51932385e6047d8274d8d1efc9a5a90

Bandhu, D., Mohan, M. M., Nittala, N. A. P., Jadhav, P., Bhadauria, A., & Saxena, K. K. (2024). Theories of motivation: A comprehensive analysis of human behavior drivers. *Acta Psychologica*, *244*, 104177. https://doi.org/10.1016/j.actpsy.2024.104177

Belachew, E. B., & Gobena, F. A. (2017). Student performance prediction model using machine learning approach: The case of Wolkite University. *International Journal of Advanced Research in Computer Science and Software Engineering*, *7*(2), 46–50. https://doi.org/10.23956/ijarcsse/V7I2/01219

Base Nacional Comum Curricular. (2022). *Base Nacional Comum Curricular*. http://basenacionalcomum.mec.gov.br/abase

Bonab, M. N., Tanha, J., & Masdari, M. (2024). A semi-supervised learning approach to quality-based web service classification. *IEEE Access*, *12*, 50489–50503. https://doi.org/10.1109/ACCESS.2024.3385341

Buenaño-Fernández, D., Gil, D., & Luján-Mora, S. (2019). Application of machine learning in predicting performance for computer engineering students: A case study. *Sustainability*, *11*(10), Artigo 10. https://doi.org/10.3390/su11102833

Butt, N. A., Mahmood, Z., Shakeel, K., Alfarhood, S., Safran, M., & Ashraf, I. (2023). Performance prediction of students in higher education using multi-model ensemble approach. *IEEE Access*, *11*, 136091–136108. https://doi.org/10.1109/ACCESS.2023.3336987

Charania, M. R. (2024). *Can assessments be used to eliminate inequities in education? AI could help*. eCampus News. https://www.ecampusnews.com/teaching-learning/2024/07/17/assessments-eliminate-inequities-education-ai/

Chen, W., Xie, X., Wang, J., Pradhan, B., Hong, H., Bui, D. T., Duan, Z., & Ma, J. (2017). A comparative study of logistic model tree, random forest, and classification and regression tree models for spatial prediction of landslide susceptibility. *CATENA*, *151*, 147–160. https://doi.org/10.1016/j.catena.2016.11.032

Chen Z., Cen, G., Wei, Y., & Li, Z. (2023). Student performance prediction approach based on educational data mining. *IEEE Access*, *11*, 131260–131272. https://doi.org/10.1109/ACCESS.2023.3335985

Chung, M. K. (2020). *Introduction to logistic regression* (arXiv:2008.13567). arXiv. https://doi.org/10.48550/arXiv.2008.13567

Cortex. (2022). *Maturidade digital nas empresas brasileiras: Conheça os setores que mais investem em tecnologia*. https://pages.cortex-intelligence.com/pesquisa-maturidade-digital-nas-empresas-brasileiras

Gil, A. C. (2019). *Métodos e técnicas de pesquisa social*. Atlas Editora.

Goodfellow, I., Yoshua, B., & Courville, A. (2016). *Deep learning*. https://www.deeplearningbook.org/

Hussain, M., Zhu, W., Zhang, W., Abidi, S. M. R., & Ali, S. (2019). Using machine learning to predict student difficulties from learning session data. *Artificial Intelligence Review*, *52*(1), 381–407. https://doi.org/10.1007/s10462-018-9620-8

Hussain, S., Muhsion, Z. F., Salal, Y. K., Theodoru, P., Kurtoğlu, F., & Hazarika, G. C. (2019). Prediction model on student performance based on internal assessment using deep learning. *International Journal of Emerging Technologies in Learning (iJET)*, *14*(8), 4. https://doi.org/10.3991/ijet.v14i08.10001

Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira. (2023). MEC e Inep divulgam resultados do Censo Escolar 2023.

https://www.gov.br/inep/pt-br/assuntos/noticias/censo-escolar/mec-e-inep-divulgam-resultados-do-censo-escolar-2023

Jantakun, K., Jantakun, T., & Jantakoon, T. (2022). The architecture of system for predicting Student Performance Based on Data Science Approaches (SPPS-DSA Architecture). *International Journal of Information and Education Technology*, *12*(8), 778–785. https://doi.org/10.18178/ijiet.2022.12.8.1684

JavaTpoint. (2021). *Life cycle of machine learning*. https://www.javatpoint.com/machine-learning-life-cycle

Jawed, S., Faye, I., & Malik, A. S. (2024). Deep learning-based assessment model for real-time identification of visual learners using raw EEG. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *32*, 378–390. https://doi.org/10.1109/TNSRE.2024.3351694

Jayaprakash, S., Balamurugan, E., & Chandar, V. (2018). *Predicting students academic performance using Naive Bayes Algorithm*. BlueCrest College Accra, Ghana. https://www.academia.edu/download/44648345/Predicting_Students_Academic_Perfomace_using_Naive_Bayes_Algorithm.pdf

Joseph, S., Dada, E., Mshelia, D., & Hamidu Alkali, A. (2019). Students Graduation on Time Prediction Model Using Artificial Neural Network. *IOSR Journal of Computer Engineering (JCE). 21*, 28–35. https://www.iosrjournals.org/iosr-jce/papers/Vol21-issue3/Series-1/E2103012835.pdf

Kabakchieva, D. (2012). Student Performance prediction by using data mining classification algorithms. *International Journal of Computer Science and Management Research*, *1*(4), 686–690. https://www.semanticscholar.org/paper/Student-Performance-Prediction-by-Using-Data-Mining-Kabakchieva/1f817e320e0f00fe8225821ce923f85980c1bdc9

Karthikeyan, K., & Kavipriya, P. (2017). On improving student performance prediction in education systems using enhanced data mining techniques. *International Journal of Advanced Research in Computer Science and Software Engineering*, *7*(5), 935–941. https://www.academia.edu/61258144/On_Improving_Student_Performance_Prediction_in_Education_Systems_using_Enhanced_Data_Mining_Techniques?auto=download

Keshtkar, F. (2016). *Predicting risk of failure in online learning platforms using machine learning algorithms for modeling students' academic performance*. https://www.academia.edu/67071946/Predicting_Risk_of_Failure_in_Online_Learning_Platforms_Using_Machine_Learning_Algorithms_for_Modeling_Students_Academic_Performance

Kibby, B. (2022). *Higher-ed needs hyper-personalization.* eCampus News. https://www.ecampusnews.com/teaching-learning/2022/11/16/higher-ed-needs-hyper-personalization/

Kim, B.-H., Vizitei, E., & Ganapathi, V. (2018). *GritNet: Student performance prediction with deep learning* (arXiv:1804.07405). arXiv. https://doi.org/10.48550/arXiv.1804.07405

Kishor, K., Sharma, R., & Chhabra, M. (2022). Student performance prediction using technology of machine learning. In D. K. Sharma, S.-L. Peng, R. Sharma, & D. A. Zaitsev (Orgs.), *Micro-electronics and telecommunication engineering* (pp. 541–551). Springer Nature. https://doi.org/10.1007/978-981-16-8721-1_53

Koutina, M., & Kermanidis, K. L. (2011). *Predicting postgraduate students' performance using machine learning techniques*. In L. Iliadis, I. Maglogiannis, & H. Papadopoulos (IFIP), (Vol. 364, pp. 159–168). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-23960-1_20

Kubat, M. (2017). *An introduction to machine learning*. Springer International Publishing. https://doi.org/10.1007/978-3-319-63913-0

Kumar, M., Singh, A. J., & Handa, D. (2017). Literature survey on educational dropout prediction. *International Journal of Education and Management Engineering*, *7*(2), Artigo 2. https://doi.org/10.5815/ijeme.2017.02.02

Lacic, E., Kowald, D., Reiter-Haas, M., Slawicek, V., & Lex, E. (2017). Beyond accuracy optimization: On the value of item embeddings for student job recommendations. *Beyond Accuracy Optimization*. https://arxiv.org/abs/1711.07762

Lakatos, E. M., & Marconi, M. de A. (2021). *Fundamentos de metodologia científica* (9ª ed.). Atlas. https://www.meulivro.biz/metodologia/3358/fundamentos-de-metodologia-cientifica-lakatos-marconi-8-ed-pdf/

Lakkaraju, H., Aguiar, E., Shan, C., Miller, D., Bhanpuri, N., Ghani, R., & Addison, K. L. (2015). A machine learning framework to identify students at risk of adverse academic outcomes. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1909–1918. https://doi.org/10.1145/2783258.2788620

Lamb, R., Neumann, K., & Linder, K. A. (2022). Real-time prediction of science student learning outcomes using machine learning classification of hemodynamics during virtual reality and online learning sessions. *Computers and Education: Artificial Intelligence*, *3*, 100078. https://doi.org/10.1016/j.caeai.2022.100078

Lau, E. T., Sun, L., & Yang, Q. (2019). Modelling, prediction, and classification of student academic performance using artificial neural networks. *SN Applied Sciences*, *1*(9), 982. https://doi.org/10.1007/s42452-019-0884-7

Libâneo. (2010). *O sistema de organização e gestão da escola*. http://acervodigital.unesp.br/handle/123456789/32

Meirelles, F. S. (2024). *Uso de TI nas empresas* (35ª edição anual). FGVcia – Centro de Tecnologia e Informação Aplicada da Escola de Administração de Empresas da Fundação Getulio Vargas. www.fgv.br/cia/pesquisa

Mitchell, T. (1977). *Machine learning*. McGraw Hill. https://www.cs.cmu.edu/~tom/mlbook.html

Muliono, R., Hakim Lubis, J., & Khairina, N. (2020, abril 2). *Analysis K-nearest neighbor algorithm for improving prediction student graduation time*. https://jurnal.polgan.ac.id/index.php/sinkron/article/view/10480

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. The MIT Press. https://books.google.com.br/books?hl=pt-BR&lr=&id=RC43AgAAQBAJ&oi=fnd&pg=PR7&dq=Machine+learning:+a+probabilistic+perspective+Kevin+P+Murphy&ots=unhucBSv17&sig=Nd4hB3Fkme7W7URg-KkvxcULnvM&redir_esc=y#v=onepage&q=Machine%20learning%3A%20a%20probabilistic%20perspective%20Kevin%20P%20Murphy&f=false

Nohuddin, P., Zainol, Z., & Nordin, A. (2018). Monitoring students performance using self organizing map trend clustering. *ULFAQAR International Journal of Defence Science, Engineering & Technology*. *1*, 50–56. https://www.researchgate.net/publication/330701203_Monitoring_Students_Performance_using_Self_Organizing_Map_Trend_Clustering

Obsie, E. Y., & Adem, S. A. (2018). Prediction of student academic performance using neural network, linear regression and support vector regression: A case study. *International Journal of Computer Applications*, *180*(40), 39–42. https://www.semanticscholar.org/paper/Prediction-of-Student-Academic-Performance-using-A-Obsie-Adem/1c7e1e9478cefd58283c9844bb13f7d5a00e6146

Ofori, F., Maina, E., & Gitonga, R. (2020). Using machine learning algorithms to predict students´ performance and improve learning outcome: A literature based review. *Journal of Information and Technology*, *4*(1), Artigo 1. https://stratfordjournals.org/journals/index.php/Journal-of-Information-and-Techn/article/view/480

Olaniyi, A. S., Kayode, S. Y., Abiola, H. M., Tosin, S.-I. T., & Babatunde, A. N. (2017). Student's performance analysis using decision tree algorithms. *Computer Science Series*, *15*(1), 55–62. https://anale-informatica.tibiscus.ro/download/lucrari/15-1-07-Olaniyi.pdf

Oloruntoba, S., & Akinode, J. L. (2017). *Student academic performance prediction using support vector machine*. https://www.semanticscholar.org/paper/STUDENT-ACADEMIC-PERFORMANCE-PREDICTION-USING-Oloruntoba-J.L.Akinode/8aefc7f2fc7e42d3f055ee2209b66d095f5bf01b

Oyerinde, O. D., & Chia, P. A. (2017). Predicting students' academic performances: A learning analytics approach using multiple linear regression. *International Journal of Computer Applications*, *157*(4), 37–44. https://ijcaonline.org/archives/volume157/number4/26822-2017912671/

Ozyurt, O., Ozyurt, H., & Mishra, D. (2023). Uncovering the educational data mining landscape and future perspective: A comprehensive analysis. *IEEE Access*, *11*, 120192–120208. https://doi.org/10.1109/ACCESS.2023.3327624

Pal, V., & Bhatt, V. (2019). *Performance prediction for post graduate students using artificial neural network*. https://www.researchgate.net/publication/342170431_Performance_Prediction_for_Post_Graduate_Students_using_Artificial_Neural_Network

Palkar, A., Dias, C. C., Chadaga, K., & Sampathila, N. (2024). Empowering glioma prognosis with transparent machine learning and interpretative insights using explainable AI. *IEEE Access*, *12*, 31697–31718. https://doi.org/10.1109/ACCESS.2024.3370238

Pandey, M., & Sharma, V. K. (2013). A decision tree algorithm pertaining to the student performance analysis and prediction. *International Journal of Computer Applications*, *61*(13), 1–5. https://doi.org/10.5120/9985-4822

Pelima, L. R., Sukmana, Y., & Rosmansyah, Y. (2024). Predicting university student graduation using academic performance and machine learning: A systematic literature review. *IEEE Access*, *12*, 23451–23465. https://doi.org/10.1109/ACCESS.2024.3361479

Pesquisa Nacional por Amostra de Domicílios (PNAD). (2023). O sistema educacional brasileiro. IBGE. https://loja.ibge.gov.br/pnad-continua-educac-o-2023.html

Pintea, S. L., Lin, Y., Dijkstra, J., & van Gemert, J. C. (2023). *A step towards understanding why classification helps regression* (arXiv:2308.10603). arXiv. https://doi.org/10.48550/arXiv.2308.10603

Pojon, M. (2017). *Using machine learning to predict student performance*. University of Tampere, Faculty of Natural Sciences. https://www.semanticscholar.org/paper/Using-Machine-Learning-to-Predict-Student-Pojon/4a50f0917142467b1b3ce48d126378a808c4abfc

Rajalaxmi, R. R., Natesan, P., Krishnamoorthy, N., & Ponni, S. (2019). Regression model for predicting engineering students academic performance. *International Journal of Recent Technology and Engineering (IJRTE). 7*, 71–75. https://www.researchgate.net/publication/333102911_Regression_Model_for_Predicting_Engineering_Students_Academic_Performance

Rodriguez-Galiano, V. F., Ghimire, B., Rogan, J., Chica-Olmo, M., & Rigol-Sanchez, J. P. (2012). An assessment of the effectiveness of a random forest classifier for land-cover classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, *67*, 93–104. https://doi.org/10.1016/j.isprsjprs.2011.11.002

Rutherford, A. (2017). B. F. Skinner and technology's nation: Technocracy, social engineering, and the good life in 20th-century America. *History of Psychology*, 20, 290–312. https://doi.org/10.1037/hop0000062

Sekeroglu, B., Dimililer, K., & Tuncal, K. (2019). Student performance prediction and classification using machine learning algorithms. *Proceedings of the 2019 8th International Conference on Educational and Information Technology*, 7–11. https://doi.org/10.1145/3318396.3318419

Sevetlidis, V., Pavlidis, G., Mouroutsos, S. G., & Gasteratos, A. (2024). Dense-PU: Learning a density-based boundary for positive and unlabeled learning. *IEEE Access*, *12*, 90287–90298. https://doi.org/10.1109/ACCESS.2024.3420453

Shahiri, A. M., Husain, W., & Rashid, N. A. (2015). A review on predicting student's performance using data mining techniques. *Procedia Computer Science*, *72*, 414–422. https://doi.org/10.1016/j.procs.2015.12.157

Shaziya, H., Kavitha, G., & Zaheer, R. (2015). *Text categorization of movie reviews for sentiment analysis*. https://www.semanticscholar.org/paper/Text-Categorization-of-Movie-Reviews-for-Sentiment-Shaziya-Kavitha/78a33ae26418bcaea2be8e4d966bc77f6aaf45e6

Shovon, H. I., & Haque, M. (2012). An approach of improving student's academic performance by using k-means clustering algorithm and decision tree. *International Journal of Advanced Computer Science and Applications*, *3*(8), 146–149. https://doi.org/10.14569/IJACSA.2012.030824

Skittou, M., Merrouchi, M., & Gadi, T. (2024). Development of an early warning system to support educational planning process by identifying at-risk students. *IEEE Access*, *12*, 2260–2271. https://doi.org/10.1109/ACCESS.2023.3348091

Soni, A., Kumar, V., Kaur, R., & Hemavathi, D. (2018). Predicting student performance using data mining techniques. *International Journal of Pure and Applied Mathematics*, *119(12)*, 221–227. https://www.acadpubl.eu/hub/2018-119-12/articles/7/1591.pdf

Sultana, J., Usha, R. M., & Farquad, H. (2019). *Student's performance prediction using deep learning and data mining methods*. https://www.researchgate.net/publication/335234927_Student%27s_Performance_Prediction_using_Deep_Learning_and_Data_Mining_methods

Swamy, M. N., & Hanumanthappa, M. (2012). Predicting academic success from student enrolment data using decision tree technique. *International Journal of Applied Information Systems*, *4*(3), 1–6. https://www.ijais.org/archives/volume4/number3/278-0654/

Tjandra, E., Kusumawardani, S. S., & Ferdiana, R. (2022). Student performance prediction in higher education: A comprehensive review. *AIP Conference Proceedings*, *2470*(1), 1–10. https://doi.org/10.1063/5.0080187

Ulinnuha, N., Sa'dyah, H., & Rahardjo, M. (2017). *A study of academic performance using random forest, artificial neural network, Naïve Bayesian, and logistic regression*. https://www.semanticscholar.org/paper/A-Study-of-Academic-Performance-using-Random-Forest-Ulinnuha-Sa%E2%80%99dyah/509053ce692561ae9fb402412c7d1d8b94f6547e

Ünal, F. (2020). data mining for student performance prediction in education. *Data Mining—Methods, Applications and Systems*. IntechOpen. https://doi.org/10.5772/intechopen.91449

Vergara, S. C. (2010). *Métodos de pesquisa em administração*. Atlas.

Vives, L., Cabezas, I., Vives, J. C., Reyes, N. G., Aquino, J., Cóndor, J. B., & Altamirano, S. F. S. (2024). Prediction of students' academic performance in the programming fundamentals course using long short-term memory neural networks. *IEEE Access*, *12*, 5882–5898. https://doi.org/10.1109/ACCESS.2024.3350169

World Economic Forum. (2018, setembro 17). *World Economic Forum*. https://www.weforum.org/agenda/2018/09/future-of-jobs-2018-things-to-know/

Yadav, S. K., & Pal, S. (2012). *Data mining: A prediction for performance improvement of engineering students using classification* (arXiv:1203.3832). arXiv. https://doi.org/10.48550/arXiv.1203.3832

Yanamandra, D., & Prasad, G. (2022). A feasibility study for the student performance prediction using machine learning. *International Journal of Innovative Research in Computer and Communication Engineering*, *11*(7), 535–538. https://doi.org/10.17148/IJARCCE.2022.117100

Yang, Y., & Xiao, Y. (2024). A deep reinforcement learning-based optimal transmission control method for streaming videos. *IEEE Access*, *12*, 53088–53098. https://doi.org/10.1109/ACCESS.2024.3386176

Yeganova, L., Kim, W., Kim, S., & Wilbur, W. J. (2014). Retro: Concept-based clustering of biomedical topical sets. *Bioinformatics (Oxford, England)*, *30*(22), 3240–3248. https://doi.org/10.1093/bioinformatics/btu514

Zhang, B. J., & Katsoulakis, M. A. (2023). *A mean-field games laboratory for generative modeling* (arXiv:2304.13534). arXiv. https://doi.org/10.48550/arXiv.2304.13534

APÊNDICES

APÊNDICE A – *Ambientes e Ferramentas para Aprendizagem de Máquina*

Fabricante	Solução	Descritivo
Microsoft	*Azure Machine Learning*	É uma plataforma de nuvem que permite que os desenvolvedores criem, treinem e implantem algoritmos de IA. A Microsoft está constantemente fazendo atualizações e melhorias em suas ferramentas de AM e anunciou recentemente alterações no *Azure Machine Learning*, aposentando o *Azure Machine Learning Workbench*.
IBM	Watson *Machine Learning*	O Watson *Machine Learning* é um serviço de nuvem da IBM que usa dados para colocar em produção algoritmos de *machine learning* (ML) e *deep learning*. Essa ferramenta de AM permite que os usuários realizem treinamento e pontuação, duas operações fundamentais de AM. O IBM Watson é mais adequado para construir aplicativos de AM por meio de conexões de Application Programming Interface (API).
Google	TensorFlow	É uma biblioteca de *software* de código aberto para programação de fluxo de dados. O resultado do TensorFlow é uma estrutura de AM. Essa ferramenta de AM é relativamente nova no mercado e está evoluindo rapidamente. A fácil visualização de redes neurais do TensorFlow é provavelmente o recurso mais atraente para os desenvolvedores.
Amazon	Amazon *Machine Learning*	É um serviço gerenciado para construir algoritmos de ML e gerar previsões. O Amazon *Machine Learning* inclui uma ferramenta de transformação automática de dados, simplificando ainda mais a ferramenta de ML para o usuário. Além disso, a Amazon também oferece outras ferramentas de ML, como o Amazon SageMaker, que é uma plataforma totalmente gerenciada que facilita a utilização de algoritmos de ML por desenvolvedores e cientistas de dados.
The University of Waikato	WEKA	Weka é uma coleção de algoritmos de AM para tarefas de mineração de dados. Ele contém ferramentas para preparação de dados, classificação, regressão, agrupamento, mineração de regras de associação e visualização.
RapidMiner	RapidMiner Studio Educational 9.10	É uma plataforma de *software* de ciência de dados desenvolvida pela empresa de mesmo nome que fornece um ambiente integrado para preparação de dados, AM, aprendizado profundo, mineração de texto e análise preditiva. Ele é usado para aplicativos comerciais, bem como para pesquisa, educação, treinamento, prototipagem rápida e desenvolvimento de aplicativos, e suporta todas as etapas do processo de AM, incluindo preparação de dados, visualização de resultados, validação e otimização de algoritmo.
Linguagem de Programação Python	O Python Software Foundation	O Python Software Foundation é uma organização dedicada ao avanço da tecnologia de código aberto relacionada à linguagem de programação Python. As bibliotecas para AM utilizadas em Python são: ▪ Numpy ▪ Scipy ▪ Scikit-learn ▪ Theano ▪ TensorFlow ▪ Keras ▪ PyTorch ▪ Pandas ▪ Matplotlib

Nota. Esta tese utilizou a linguagem de programação Python Software Foundation com a biblioteca *SciKit-learn*. Fonte: Elaboração própria, com base em consolidado de pesquisas.

APÊNDICE B – *Exemplo de Algoritmos por Tipos e Técnicas*

Tipos	Técnicas
Supervisionado	Regressão Linear, Regressão Logística, Árvores de Decisão, Random Forest, SVM, Redes Neurais, KNN e Naive Bayes.
Não Supervisionado	K-means, DBSCAN, Principal Component Analysis (PCA), Análise de Componentes Independentes e Agrupamento Hierárquico.
Por Reforço	Q-Learning, SARSA, Actor-Critic e Deep Reinforcement Learning.
Semisupervisionado	Self-Training, Co-Training, Multi-View Learning e Label Propagation.
Aprendizado Profundo	RNA, Redes Neurais Convolucionais (CNN), Redes Neurais Recorrentes (RNN), Autoencoder, Redes Generativas Adversariais (GANs), entre outros.
Modelos Lineares	Regressão Linear, Regressão Logística, Análise Discriminante Linear, Regressão Ridge, Lasso, Elastic Net, Generalized Linear Models, Perceptron e Passive-Aggressive Algorithms.
Redes Neurais	MLP, CNN, RNN, LSTM, Autoencoders, CNN e RNN
Processamento de Linguagem Natural	Bag-of-Words, TF-IDF, Word2Vec, GloVe, Sentiment Analysis e Named Entity Recognition.
AM Baseada em Grafos	PageRank, Propagação de Rótulo, Random Walks e Spectral Clustering.

Nota. A lista de algoritmos é muito mais extensa, a consolidação neste apêndice representa as principais. Fonte: Elaboração própria, com base em pesquisas consolidadas.

APÊNDICE C – *Consolidação do Referencial: Previsão do Desempenho e de Máquina e da Mineração de Dados*

Autor(es)	Objetivo do(s) autor(es)	Algoritmo	Acurácia
Koutina & Kermanidis, 2011	Previsão do desempenho	Sem aplicação prática.	-
Rodriguez-Galiano et al., 2012	Previsão de máquina	Algoritmo estudado: Random Forest.	-
Swamy & Hanumanthappa, 2012	Previsão do desempenho	Algoritmo estudado: Decision Tree.	-
Yadav & Pal, 2012	Previsão do desempenho	Sem aplicação prática.	-
Pandey & Sharma, 2013	Previsão do desempenho	Algoritmo estudado: Decision Tree.	-
Yeganova et al., 2014	Previsão de máquina	Algoritmo estudado: Stochastic Gradient Descent (SGD) e Support Vector Machine (SVM).	-
Lakkaraju et al., 2015	Previsão do desempenho	Algoritmo estudado: Decision Tree, Artificial Neural Network, Matrix Factoring, Collaborative Filters, and Probabilistic Graphical Models.	-
Shaziya et al., 2015	Previsão de máquina	Algoritmo estudado: Naive Bayes.	-
Keshtkar, 2016	Previsão do desempenho	Algoritmo estudado: Logistic Model Tree (LMT).	-
Al-Shabandar et al., 2017	Previsão de máquina	Algoritmo estudado: Random Forest, Linear Discriminant Analysis.	98,8%
Agrawal et al., 2017	Previsão do desempenho	Algoritmo estudado: Decision Tree, Random Forest, Naive Bayes e Rule Induction.	90%
Allen-Zhu, 2017	Previsão de máquina	Algoritmo estudado: SGD e SVM.	-
Belachew & Gobena, 2017.	Previsão do desempenho	Algoritmo estudado: Naive Bayes, Artificial Neural Network, SVM e Multi-layer Perceptron.	95,7%
Chen et al., 2017	Previsão do desempenho	Algoritmo estudado: Artificial Neural Network, Naive Bayes, K-Nearest Neighbor (KNN), Decision Tree, Random Forest, SVM e Logistic Regression.	66,1%
Kumar et al., 2017	Previsão do desempenho	Algoritmo estudado: Decision Tree, Neural Network, KNN, SVM, Random Forest, Logistic Regression, Linear Discriminant Analysis, Multiple Regression e Self-Organized Mapping.	-
Olaniyi et al., 2017	Previsão do desempenho	Algoritmo estudado: Decision Tree.	-
Oloruntoba & Akinode, 2017	Previsão do desempenho	Algoritmo estudado: SVM.	97%
Oyerinde & Chia, 2017	Previsão do desempenho	Algoritmo estudado: Multiple linear regression.	89%
Pojon, 2017	Previsão do desempenho	Algoritmo estudado: Linear Regression, Decision Tree e Naive Bayes.	98%
Lacic et al., 2017	Previsão do desempenho	Algoritmo estudado: SVM.	-
Ulinnuha et al., 2017	Previsão do desempenho	Algoritmo estudado: Random Forest, Neural Network, Naive Bayes e Logistic Regression.	-
Jayaprakash et al., 2018	Previsão do desempenho	Algoritmo estudado: Naive Bayes.	92,2%
Obsie & Adem, 2018	Previsão do desempenho	Algoritmo estudado: Neural Network, Linear Regression e Support Vector Regression (SVR).	97,63% e 98,05%
Nohuddin et al., 2018	Previsão do desempenho	Algoritmo estudado: Self Organizing Map (SOM) Trend Clustering.	-

Autor(es)	Objetivo do(s) autor(es)	Algoritmo	Acurácia
Soni et al., 2018	Previsão do desempenho	Sem aplicação prática.	-
Kim et al., 2018	Previsão do desempenho	Algoritmo estudado: Artificial Neural Network (GritNet).	-
Anderson et al., 2019	Previsão do desempenho	Algoritmo estudado: SVM, Decision Tree, Logistic Regression e Stochastic Gradient Descent (SGD).	82,4%
Joseph et al., 2019	Mineração de dados	Algoritmo estudado: Neural Network.	-
Buenaño-Fernández et al., 2019	Desempenho do aluno	Algoritmo estudado: Literatura Empírica e Decision Tree.	-
S. Hussain et al., 2019	Previsão do desempenho	Algoritmo estudado: Artificial Neural Network.	95,34%
M. Hussain et al., 2019	Previsão do desempenho	Sem aplicação prática.	-
Lau et al., 2019	Previsão do desempenho	Algoritmo estudado: Artificial Neural Network, SGD e SVM.	84,8%
Rajalaxmi et al., 2019	Previsão do desempenho	Algoritmo estudado: (Método estatístico) Regressão linear múltipla.	-
Sekeroglu et al., 2019	Mineração de dados	Algoritmo estudado: SVR.	87,7%
Sultana et al., 2019	Previsão de máquina	Algoritmo estudado: Random Forest.	-
Pal & Bhatt, 2019	Previsão do desempenho	Algoritmo estudado: Artificial Neural Network, Linear Regression, Random Forest e Artificial Neural Network (ANN).	97,4%
Pelima et al., 2024	Previsão do desempenho	Support Vector Machine (SVM), Random Forest, Logistic Regression, KNN, Artificial Neural Network/Multi-Layer Perceptron (ANN/MLP).	90%
Vives et al., 2024	Previsão do desempenho	Long Short-Term Memory (LSTM), Deep Neural Network (DNN), Decision Tree (DT), Random Forest (RF), Logistic Regression (LR), Support Vector Classifier (SVC), KNN.	98,3%
Skittou et al., 2024	Previsão do desempenho	KNN, SVM, Random Forest, SGD.	99,5%

Nota: Dependendo do caso de uso, a precisão de um modelo de AM geralmente é considerada de ruim a excelente. É importante observar que esses intervalos são apenas diretrizes gerais. A acurácia de um modelo de AM varia de acordo com o problema específico que está sendo resolvido e a qualidade dos dados usados para treinar o modelo. Fonte: Pesquisas consolidadas.

[1] Ruim: de 0% a 50%

[2] Média: de 50% a 75%

[3] Boa: de 75% a 90%

[4] Excelente: de 90% a 100%

APÊNDICE D – *Atributos Utilizados por Shariri et al. (2015)*

Atributo	Algoritmo
Avaliação interna com base no *Cumulative Grade Point Average* (CPGA)	SVM
Avaliação interna do aluno, CGPA e Atividades extracurriculares	Naive Bayes, KNN e SVM
CGPA, Demografia[6], Antecedentes do ensino médio, Bolsa de estudo e Interação nas redes sociais	Naive Bayes
Avaliação interna e externa	Artificial Neural Network
Avaliação externa, Demografia e Antecedentes do ensino médio	Artificial Neural Network e Naive Bayes
Demografia e Antecedentes do ensino médio	Artificial Neural Network
CGPA, Demografia, Antecedentes do ensino médio, Bolsa de estudo e Interação nas redes sociais	Artificial Neural Network
Avaliação interna	Decision Tree, Artificial Neural Network e KNN
CGPA, Demografia, Antecedentes do ensino médio, Interação nas redes sociais, Avaliação interna e Atividades extracurriculares	Decision Tree
CGPA, Demografia, Atividades extracurriculares e Avaliação externa	Decision Tree
Fatores psicométricos[7], Atividades extracurriculares e *Soft skills*	Decision Tree
CGPA	Artificial Neural Network e Naive Bayes
Demografia, Antecedentes do ensino médio, Avaliação interna e Atividades extracurriculares	Decision Tree
Avaliação externa	Decision Tree e Artificial Neural Network
Fatores psicométricos	Decision Tree, Artificial Neural Network, KNN e SVM

Fonte: Adaptado e traduzido com base em Shahiri, A. M., Husain, W., & Rashid, N. A. (2015). A review on predicting student's performance using data mining techniques. *Procedia Computer Science*, *72*, 414–422. https://doi.org/10.1016/j.procs.2015.12.157

[6] A demografia do aluno como o ramo da geografia que se dedica ao estudo das populações e a relação delas com a sociedade.

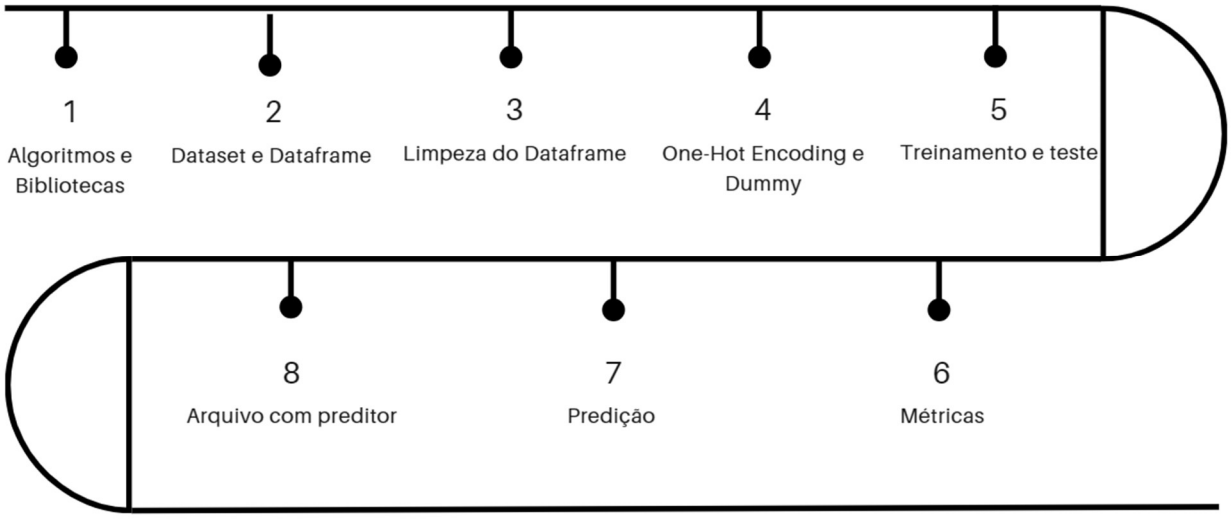
[7] A psicometria é a parte da Psicologia que busca mensurar as características psicológicas dos indivíduos.

APÊNDICE E – *Dicionário de Dados das Colunas do DataSet*

#	Coluna	Dicionário
1	PC	**Perfil Comportamental**: Analista, Comunicador, Executor e Planejador.
2	SR	**Sistema Representacional**: Auditivo, Digital, Cinestésico e Visual.
3	Genero	**Gênero**: Masculino (M) e Feminino (F).
4	pk	***Primary Key***: Chave primária para identificar os alunos (ID único).
5	Status *	***Status* do Aluno**: Composto por Abandono, Ativo, Cancelado, Desistente e Trancado.
6	STATUS_BIN	**Variável Binária** combinando os *status* de evasão (1) e ativo (0). Usada para a AM como variável-alvo.
7	Nome *	**Nome Completo do Aluno** (não utilizada no modelo de AM).
8	Cidade *	**Cidade** atual do aluno (não utilizada no modelo de AM).
9	DataInicio *	**Data de Início no Curso** (formato dd/mm/aaaa) (não utilizada no modelo).
10	PC_nota	**Nota** na disciplina de Pensamento Crítico (número contínuo).
11	PC_cursou	**Indicador de Conclusão**: 1 se a disciplina Pensamento Crítico foi cursada, 0 caso contrário.
12	SS_nota	**Nota** na disciplina de Soft Skill (número contínuo).
13	SS_cursou	**Indicador de Conclusão** para Soft Skill.
14	SO_nota	**Nota** na disciplina de Sistemas Operacionais.
15	SO_cursou	**Indicador de Conclusão** para Sistemas Operacionais.
16	OAC_nota	**Nota** na disciplina de Organização e Arquitetura de Computadores.
17	OAC_cursou	**Indicador de Conclusão** para Organização e Arquitetura de Computadores.
18	LP_nota	**Nota** na disciplina de Lógica de Programação.
19	LP_cursou	**Indicador de Conclusão** para Lógica de Programação.
20	IP_nota	**Nota** na disciplina de Introdução à Programação.
21	IP_cursou	**Indicador de Conclusão** para Introdução à Programação.
22	GIT_nota	**Nota** na disciplina de Git e GitHub.
23	GIT_cursou	**Indicador de Conclusão** para Git e GitHub.
24	MD_nota	**Nota** na disciplina de Modelagem de Dados.
25	MD_cursou	**Indicador de Conclusão** para Modelagem de Dados.
26	SQL_nota	**Nota** na disciplina de Structured Query Language (SQL).
27	SQL_cursou	**Indicador de Conclusão** para SQL.
28	MA_nota	**Nota** na disciplina de Metodologias Ágeis.
29	MA_cursou	**Indicador de Conclusão** para Metodologias Ágeis.
30	PDP_nota	**Nota** na disciplina de Proteção de Dados Pessoais.
31	PDP_cursou	**Indicador de Conclusão** para Proteção de Dados Pessoais.
32	EOVS_nota	**Nota** na disciplina de Estrutura Organizacional e Visão Sistêmica.
33	EOVS_cursou	**Indicador de Conclusão** para Estrutura Organizacional e Visão Sistêmica.
34	ERMS_nota	**Nota** na disciplina de Engenharia de Requisitos e Modelagem de Software.
35	ERMS_cursou	**Indicador de Conclusão** para Engenharia de Requisitos e Modelagem de Software.
36	FE_nota	**Nota** na disciplina de Front-End.
37	FE_cursou	**Indicador de Conclusão** para Front-End.
38	BE_nota	**Nota** na disciplina de Back-End I.
39	BE_cursou	**Indicador de Conclusão** para Back-End I.
40	N_Disciplinas	**Número de Disciplinas Cursadas** pelo aluno.

Nota. As colunas marcadas com (*) não foram utilizadas para a AM, resultando em 37 colunas processadas, das quais 30 são disciplinas e variáveis relacionadas ao desempenho acadêmico, além da variável N_Disciplinas. A variável STATUS_BIN (y) é dicotômica, representando 1 para evasão e 0 para ativo, utilizada como variável-alvo em modelos de AM. Fonte: Elaboração própria.

APÊNDICE F – *Timeline das Etapas*



Nota. Cada etapa apresentada foi executada no Google Colab. Fonte: Elaboração própria.

APÊNDICE G – *Agrupamento de Algoritmos Selecionados para Análise de Acurácia*

Técnica	#	Algoritmo	Sigla	Criador	Ano
Linear Models (S)	1	Ridge regression	LRR	Arthur E. Hoerl and Robert W. Kennard	1970
	2	Logistic regression	LLR	David Cox e Outros	1958
	3	Linear Regression	LiR	Adrien-Marie Legendre	1805
Linear and Quadratic Discriminant Analysis (S)	4	Dimensionality Reduction using Linear Discriminant Analysis	LDA	Ronald A. Fisher	1936
Support Vector Machine (S)	5	Classification	SVC	Vapnik e Alexey Ya	1964
	6	Regression	SVR	Vapnik e Outros	1997
Stochastic Gradient Descent (S)	7	Classification	SGDC	Herbert Robbins & Monroe Swets	1951
	8	Regression	SGDR	Herbert Robbins & Monroe Swets	1951
Nearest Neighbors (S)	9	Nearest Neighbors Classification	NNC	Evelyn Fix e Joseph Hodges	1951
	10	Nearest Neighbors Regression	NNR	Evelyn Fix e Joseph Hodges	1951
Naive Bayes (S)	11	Gaussian Naive Bayes	NBG	Thomas Bayes	1968
	12	Bernoulli Naive Bayes	NBB	Thomas Bayes	1968
Decision Trees (S)	13	Classification	DTC	J. Ross Quinlan	1980
	14	Regression	DTR	J. Ross Quinlan	1980
Ensemble Methods (S)	15	Gradient Tree Boosting (Classification)	GBC	Jerome H. Friedman	2001
	16	Gradient Tree Boosting (Regression)	GBR	Jerome H. Friedman	2001
	17	Random Forest (Classification)	RFC	Tin Kam Ho	1995
	18	Random Forest (Regression)	RFR	Tin Kam Ho	1995
Neural Network Models (S)	19	Multi-layer Perceptron	KNP	David E. Rumelhart, Geoffrey Hinton e Ronald J. Williams	1980
	20	Classification	KNC	Warren McCulloch e Walter Pitts	1943
	21	Regression	KNR	Warren McCulloch e Walter Pitts	1943

Nota. O apêndice inclui o ano da criação de cada algoritmo, fornecendo contexto histórico e destacando a evolução até o aprendizado de máquina contemporâneo. Fonte: Elaboração própria.

APÊNDICE H – *Descritivo dos Algoritmos Selecionados para Estudo*

Técnica	#	Algoritmo	Sigla	Explicação
Modelos Lineares	1	Ridge Regression	LRR	Técnica de regressão linear que inclui uma penalização L2 para evitar *overfitting*.
	2	Logistic Regression	LLR	Algoritmo de classificação que modela a probabilidade de uma classe binária.
	3	Linear Regression	LiR	Modelo de regressão linear simples que prediz uma variável dependente contínua com base em uma ou mais variáveis independentes.
Análise Discriminante Linear e Quadrática	4	Dimensionality Reduction using Linear Discriminant Analysis	LDA	Técnica de redução de dimensionalidade que maximiza a separação entre múltiplas classes.
Máquinas de Vetores de Suporte	5	Classification	SVC	Algoritmo de classificação que encontra o hiperplano ótimo para separar classes em um espaço de alta dimensão.
	6	Regression	SVR	Algoritmo de regressão que usa suporte vetores para prever valores contínuos.
Gradiente Descendente Estocástico	7	Classification	SGDC	Técnica de classificação que otimiza o modelo usando gradiente descendente estocástico.
	8	Regression	SGDR	Técnica de regressão que utiliza gradiente descendente estocástico para otimização.
Vizinhos Mais Próximos	9	Nearest Neighbors Classification	NNC	Algoritmo de classificação que classifica um ponto de dados com base na maioria dos seus k vizinhos mais próximos.
	10	Nearest Neighbors Regression	NNR	Algoritmo de regressão que prediz o valor de um ponto de dados com base na média dos seus k vizinhos mais próximos.
Naive Bayes	11	Gaussian Naive Bayes	NBG	Algoritmo de classificação baseado no teorema de Bayes, assumindo uma distribuição normal dos dados.
	12	Bernoulli Naive Bayes	NBB	Algoritmo de classificação para dados binários baseado no teorema de Bayes.
Árvores de Decisão	13	Classification	DTC	Algoritmo de classificação que utiliza uma estrutura de árvore de decisão para classificar dados.
	14	Regression	DTR	Algoritmo de regressão que usa uma árvore de decisão para prever valores contínuos.
Métodos de ***Ensemble***	15	Gradient Tree Boosting (Classification)	GBC	Técnica de classificação que constrói um modelo combinando múltiplas árvores de decisão por meio do método de *boosting*.
	16	Gradient Tree Boosting (Regression)	GBR	Técnica de regressão que combina múltiplas árvores de decisão para formar um modelo confiável, minimizando o erro por *boosting*.
	17	Random Forest (Classification)	RFC	Algoritmo de classificação que cria uma "floresta" de árvores de decisão e combina seus resultados para melhorar a precisão.
	18	Random Forest (Regression)	RFR	Algoritmo de regressão que utiliza múltiplas árvores de decisão para prever valores contínuos, combinando seus resultados para maior precisão.
Redes Neurais	19	Multi-layer Perceptron	KNP	Algoritmo de rede neural que consiste em múltiplas camadas de neurônios, usado para aprendizado supervisionado.
	20	Classification	KNC	Modelo de rede neural para classificação de dados em múltiplas classes.
	21	Regression	KNR	Modelo de rede neural para prever valores contínuos em um conjunto de dados.

Fonte: Elaboração própria.

APÊNDICE I – *Bibliotecas do Python Utilizadas nesta Tese*

Biblioteca	Descrição	Uso Principal
Pandas	Biblioteca de análise de dados de código aberto que oferece estruturas de dados flexíveis e eficientes para manipulação de dados em tabelas e séries temporais.	Importação, limpeza, manipulação de dados e análises exploratórias.
NumPy	Biblioteca fundamental para computação científica que oferece suporte a *arrays* e matrizes multidimensionais, além de funções matemáticas complexas.	Operações matemáticas com *arrays* e matrizes, processamento de imagens, ML, física.
Matplotlib	Biblioteca de visualização de dados que oferece ferramentas para criar gráficos e visualizações de alta qualidade.	Criação de gráficos de linhas, barras, dispersão, histogramas e outros tipos de gráficos.
Scikit-learn	Biblioteca de AM que oferece uma ampla variedade de algoritmos para classificação, regressão, *clustering*, entre outras tarefas.	Treinamento de algoritmos de AM, avaliação de desempenho e previsões.
Seaborn	Biblioteca de visualização de dados baseada no Matplotlib que facilita a criação de gráficos estatísticos atraentes.	Criação de gráficos estatísticos, visualização de dados.
Scikit-plot	Biblioteca de visualização de dados para AM que fornece funções de plotagem para avaliar a *performance* dos modelos.	Avaliação de desempenho dos modelos de AM.
sklearn.metrics	Módulo de métricas do Scikit-learn que fornece funções para avaliar a *performance* dos modelos de AM.	Avaliação de precisão, *recall*, MSE, R2, relatórios de classificação.
sklearn.preprocessing	Módulo do Scikit-learn que fornece funções para pré-processamento de dados, como escalonamento e codificação.	Pré-processamento de dados, escalonamento, codificação.
sklearn.model_selection	Módulo do Scikit-learn que fornece funções para divisão de dados em conjuntos de treino e teste.	Divisão de dados em conjuntos de treino e teste.
utils (sklearn)	Funções utilitárias do Scikit-learn para operações diversas.	Funções diversas para manipulação de dados e modelos.

Fonte: Elaboração própria.

APÊNDICE J – *DataFrame (1) sem Transformações*

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 635 entries, 0 to 634
Data columns (total 37 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   PC              635 non-null    object
 1   SR              635 non-null    object
 2   Genero          635 non-null    object
 3   pk              635 non-null    int64
 4   Status          635 non-null    object
 5   STATUS_BIN      635 non-null    int64
 6   PC_nota         635 non-null    float64
 7   PC_cursou       635 non-null    int64
 8   SS_nota         635 non-null    float64
 9   SS_cursou       635 non-null    int64
 10  SO_nota         635 non-null    float64
 11  SO_cursou       635 non-null    int64
 12  OAC_nota        635 non-null    float64
 13  OAC_cursou      635 non-null    int64
 14  LP_nota         635 non-null    float64
 15  LP_cursou       635 non-null    int64
 16  IP_nota         635 non-null    float64
 17  IP_cursou       635 non-null    int64
 18  Git_nota        635 non-null    int64
 19  Git_cursou      635 non-null    int64
 20  MD_nota         635 non-null    float64
 21  MD_Cursou       635 non-null    int64
 22  SQL_nota        635 non-null    float64
 23  SQL_cursou      635 non-null    int64
 24  MA_nota         635 non-null    float64
 25  MA_cursou       635 non-null    int64
 26  PDP_nota        635 non-null    float64
 27  PDP_cursou      635 non-null    int64
 28  EOVS_nota       635 non-null    float64
 29  EOVS_cursou     635 non-null    int64
 30  ERMS_nota       635 non-null    float64
 31  ERMS_cursou     635 non-null    int64
 32  FE_nota         635 non-null    float64
 33  FE_cursou       635 non-null    int64
 34  BE_nota         635 non-null    int64
 35  BE_cursou       635 non-null    int64
 36  N_Disciplinas   635 non-null    int64
dtypes: float64(13), int64(20), object(4)
memory usage: 183.7+ KB
```

Nota. As colunas indicadas como irrelevantes no dicionário de dados (seção 3.4) foram retiradas antes dessa etapa, exceto a coluna Status. Fonte: Elaboração própria.

APÊNDICE K – *Resultado da Limpeza de Colunas Irrelevantes do DataFrame (2)*

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 635 entries, 0 to 634
Data columns (total 36 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   PC             635 non-null    object
 1   SR             635 non-null    object
 2   Genero         635 non-null    object
 3   pk             635 non-null    int64
 4   STATUS_BIN     635 non-null    int64
 5   PC_nota        635 non-null    float64
 6   PC_cursou      635 non-null    int64
 7   SS_nota        635 non-null    float64
 8   SS_cursou      635 non-null    int64
 9   SO_nota        635 non-null    float64
 10  SO_cursou      635 non-null    int64
 11  OAC_nota       635 non-null    float64
 12  OAC_cursou     635 non-null    int64
 13  LP_nota        635 non-null    float64
 14  LP_cursou      635 non-null    int64
 15  IP_nota        635 non-null    float64
 16  IP_cursou      635 non-null    int64
 17  Git_nota       635 non-null    int64
 18  Git_cursou     635 non-null    int64
 19  MD_nota        635 non-null    float64
 20  MD_Cursou      635 non-null    int64
 21  SQL_nota       635 non-null    float64
 22  SQL_cursou     635 non-null    int64
 23  MA_nota        635 non-null    float64
 24  MA_cursou      635 non-null    int64
 25  PDP_nota       635 non-null    float64
 26  PDP_cursou     635 non-null    int64
 27  EOVS_nota      635 non-null    float64
 28  EOVS_cursou    635 non-null    int64
 29  ERMS_nota      635 non-null    float64
 30  ERMS_cursou    635 non-null    int64
 31  FE_nota        635 non-null    float64
 32  FE_cursou      635 non-null    int64
 33  BE_nota        635 non-null    int64
 34  BE_cursou      635 non-null    int64
 35  N_Disciplinas  635 non-null    int64
dtypes: float64(13), int64(20), object(3)
memory usage: 178.7+ KB
```

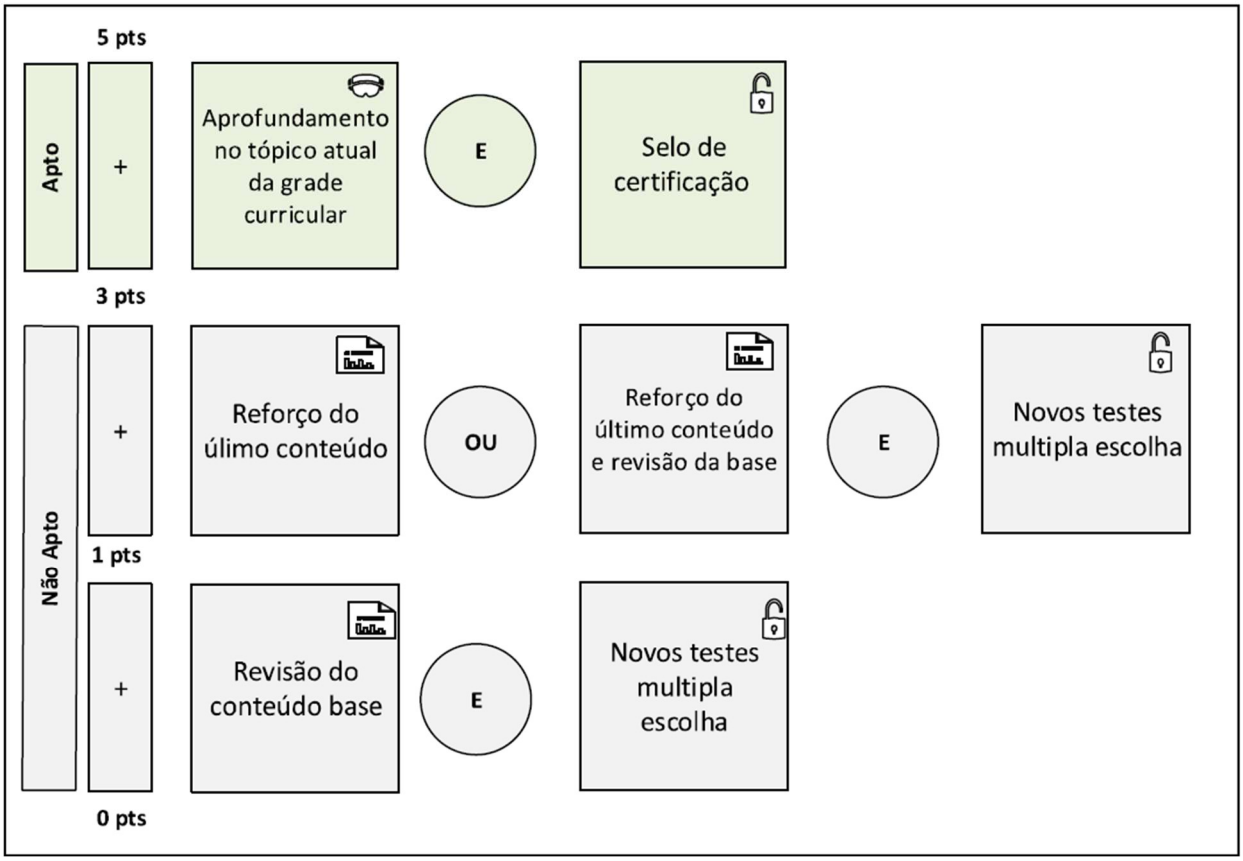
Nota. Retirada da coluna categórica Status, restando 36 colunas. Fonte: Elaboração própria.

APÊNDICE L – *Resultado do* One-Hot Encoding *no DataFrame (3)*

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 635 entries, 0 to 634
Data columns (total 45 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   pk              635 non-null    int64
 1   STATUS_BIN      635 non-null    int64
 2   PC_nota         635 non-null    float64
 3   PC_cursou       635 non-null    int64
 4   SS_nota         635 non-null    float64
 5   SS cursou       635 non-null    int64
 ...
 32  N_Disciplinas   635 non-null    int64
 33  PC_Analista     635 non-null    bool
 34  PC_Comunicador  635 non-null    bool
 35  PC_Executor     635 non-null    bool
 36  PC_Planejador   635 non-null    bool
 37  PC_SemDados     635 non-null    bool
 38  SR_Auditivo     635 non-null    bool
 39  SR_Cinestésico  635 non-null    bool
 40  SR_Digital      635 non-null    bool
 41  SR_SemDados     635 non-null    bool
 42  SR_Visual       635 non-null    bool
 43  Genero_F        635 non-null    bool
 44  Genero_M        635 non-null    bool
dtypes: bool(12), float64(13), int64(20)
memory usage: 171.3 KB
```

Nota. As reticências entre a coluna 5 e 32 indicam a montagem da figura com o propósito de diminuir o tamanho da imagem. Após a coluna 32, são as novas colunas. Fonte: Elaboração própria.

APÊNDICE M – *Caminhos de Aprendizado com Base no Canvas Instructure*



Fonte: Elaboração própria.