UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

PREVISÃO DA EVASÃO ESTUDANTIL EM DISCIPLINAS INTRODUTÓRIAS DE PROGRAMAÇÃO POR MEIO DE MINERAÇÃO DE DADOS SOCIODEMOGRÁFICOS

ANDRÉ FABIANO SANTOS PEREIRA

MANAUS
2021

ANDRÉ FABIANO SANTOS PEREIRA

PREVISÃO DA EVASÃO ESTUDANTIL EM DISCIPLINAS INTRODUTÓRIAS DE PROGRAMAÇÃO POR MEIO DE MINERAÇÃO DE DADOS SOCIODEMOGRÁFICOS

Dissertação apresentada ao Programa de Pós-Graduação em Informática do Instituto de Computação da Universidade Federal do Amazonas como parte dos requisitos para a obtenção do grau de mestre em Informática.
Orientador: Prof. Dr. Eduardo James Pereira Souto
Coorientador: Prof. Dr. Leandro Silva Galvão de Carvalho

MANAUS
2021

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

Pereira, André Fabiano Santos
P436p Previsão da evasão estudantil em disciplinas introdutórias de programação por meio de mineração de dados sociodemográficos / André Fabiano Santos Pereira . 2021
81 f.: 31 cm.

Orientador: Eduardo James Pereira Souto
Coorientador: Leandro Silva Galvão de Carvalho
Dissertação (Mestrado em Informática) - Universidade Federal do Amazonas.

1. Mineração de dados educacionais. 2. Evasão estudantil. 3. Introdução à programação. 4. Previsão de evasão. I. Souto, Eduardo James Pereira. II. Universidade Federal do Amazonas III. Título

FOLHA DE APROVAÇÃO

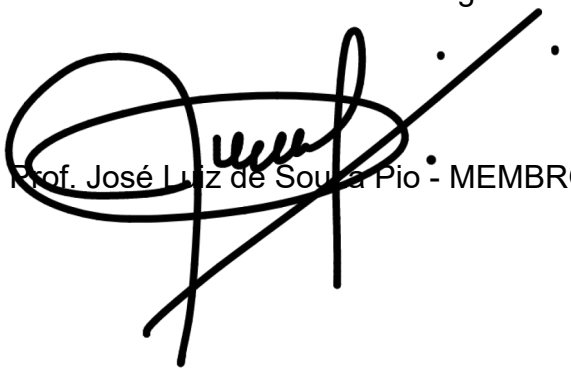
"Previsão da evasão estudantil em disciplinas introdutórias de programação por meio de mineração de dados sociodemográficos"

ANDRÉ FABIANO SANTOS PEREIRA

Dissertação de Mestrado defendida e aprovada pela banca examinadora constituída pelos Professores:

Prof. Eduardo James Pereira Souto - PRESIDENTE

Profa. Rosiane de Freitas Rodrigues - MEMBRO EXTERNO



Prof. José Luiz de Souza Pio - MEMBRO INTERNO

Manaus, 22 de Abril de 2021

Dedico este trabalho a minha querida mãe.

AGRADECIMENTOS

Agradeço a Deus, a minha família e a todos os amigos que estiveram ao meu lado nesta árdua caminhada.

A minha mãe por, mesmo nas horas mais difíceis de sua vida ter investido todos seus recursos que tinha em minha educação

A minha esposa, amiga e companheira, Deyse. Obrigado pela compreensão e parceria neste momento desafiante no qual, diversas vezes, serviu como pai e mãe de nossos filhos.

Aos meus queridos filhos Lucas, Mateus, Samuel e Marina.

Ao meu irmão Leandro Santos que me tem como um exemplo assim como eu o tenho.

Aos meus orientadores, profs. Leandro Galvão e Eduardo Souto, pela orientação e sabedoria compartilhada, apoio e parceria a mim oferecido. Obrigado pela paciência e por ter sempre acreditado que esse sonho seria capaz de ser realizado.

Ao professor Davi Fernandes por todo suporte e informações necessários para a pesquisa.

Ao professor Anselmo Paiva, da minha graduação na UFMA, por - mesmo depois de quase duas décadas após termos sido professor e estudante – de forma diligente me estender a mão e prontamente me recomendar ao PPGI\UFAM.

Aos professores Jose de Souza Pio e Rosiane de Freitas pela imensa contribuição para que este trabalho atingisse esse nível final.

E por fim, e não menos importante, agradeço aos grandes amigos César, Joethe, Ismael, Marcos e Mauro que estiveram presentes nessa batalha em todos os momentos, de domingo a domingo e que por isso mesmo construímos grandes amizades que se expandiram para nossas casas, vidas e família.

Tudo posso naquele que me fortalece.

Filipenses 4.13

RESUMO

A evasão estudantil caracteriza-se como um processo de exclusão do ambiente educacional determinado por fatores motivacionais, estruturais, socioeconômicos, internos e externos às instituições de ensino. A evasão em disciplinas introdutórias de programação, conhecidas como CS1, constitui-se em um desafio frequentemente observado em cursos de ciências exatas e de engenharias. O objetivo deste trabalho é construir um modelo de previsão de evasão de estudantes em disciplinas CS1 destes cursos com uso de dados sociodemográficos, passível de aplicação ainda no início de cada período letivo. A metodologia aplicada foi baseada no processo de mineração de dados CRISP-DM (*Cross-Industry Standard Processo of Data Mining*), com adaptações ao ambiente educacional, para extração do conhecimento e construção do modelo preditivo de evasão baseada na dimensão sociodemográfica dos estudantes. Com o intuito de validar a metodologia proposta, foram realizados experimentos com dados de ex-estudantes de CS1 dos cursos de ciências exatas e de engenharias da Universidade Federal do Amazonas. A previsão de evasão de estudantes nessas turmas mostrou-se viável, sendo construído um modelo preditivo com uso do classificador *AdaBoost* facilmente adaptável, permitindo a condução de iniciativas institucionais e pedagógicas mais eficientes de combate à evasão estudantil.

Palavras-chave: Mineração de Dados Educacionais. Evasão Estudantil. Introdução à Programação. Previsão de Evasão.

ABSTRACT

Student dropout is characterized as an exclusion process from educational environment determined by motivational, structural, socioeconomic factors, internal and external to educational institutions. Dropout occurrences in introductory computer programming classes, known as CS1, is a challenge often observed in courses in sciences and engineering. The present paper aims to build a student‘s predicting dropout model in CS1 classes of these courses with the use of sociodemographic data and suitable to the application of this model even at the beginning of each academic period. The applied methodology was based on the data mining process CRISP-DM (Cross Industry Standard Process of Data Mining), with adaptations to the educational environment, to extract knowledge and build the dropout predictive model using a student sociodemographic data dimension. In order to validate the proposed methodology, experiments were carried out with data from former students of CS1 from the science and engineering courses at UFAM. The prediction of dropout students in these classes proved to be feasible, being built a predictive model easily adaptable using the AdaBoost classifier, allowing the engagement of more efficient institutional and pedagogical initiatives to combat evasion in an attempt that this probability does not materialize.

Keywords: Educational Data Mining. Prediction. Dropout. Introductory Programming.

LISTA DE FIGURAS

Figura 1.1 - Fluxograma de aplicação da metodologia aplicada. 18

Figura 2.1 - IDE do juiz online Codebench. 26

Figura 2.2 - Descoberta de Conhecimento em Bases de Dados.....28

Figura 2.3 - Exemplo de árvore de decisão do modelo preditivo de evasão em CS1. 32

Figura 2.4 - Exemplo de árvores da floresta aleatória. 33

Figura 2.5 - Exemplo de Rede Neural Artificial.....34

Figura 2.6 - Exemplo de classificação com SVM. 36

Figura 2.7 - ilustração de classificação com algoritmo kNN. 37

Figura 2.8 - O processo de mineração de dados CRISP-DM. 40

Figura 2.9 - O processo de mineração de dados CRISP-EDM 41

Figura 3.1 - Processo para a construção do Modelo para previsão do risco de evasão com dados educacionais sociodemográficos. 46

Figura 4.1 - Questionário Sociodemográfico utilizado na pesquisa. 53

Figura 4.2- Ambiente da ferramenta Qlikview (data analytics). 56

Figura 4.3 - Árvore de decisão do modelo baseado no algoritmo Adaboost. 58

Figura 4.4 - Seleção dos atributos preditores no Orange. 59

Figura 4.5 - Estrutura para geração dos modelos preditivos no Orange. 60

Figura 4.6 - Curva Receiver Operating Characteristics (ROC) para comparação dos classificadores. 61

Figura 4.7 - Matriz de confusão do modelo gerado. 63

Figura 4.8 - Distribuição de estudantes *non-majors* em CS1 por gênero. 66

Figura 4.9 - Distribuição de estudantes *non-majors* em CS1 por conhecimento prévio. 67

Figura 4.10 - Distribuição de estudantes *non-majors* em CS1 por origem do nível médio. 68

Figura 4.11 - Distribuição de estudantes *non-majors* em CS1 por faixa etária..... 69

Figura 4.12 - Distribuição de estudantes *non-majors* em CS1 por idade. 69

Figura 4.13 - Distribuição de estudantes *non-majors* em CS1 por estado civil. 70

LISTA DE QUADROS

Quadro 2.1 - Definições de evasão e amplitudes do conceito.....23

Quadro 2.2 - Resumo característico de algoritmos utilizados no trabalho. 38

Quadro 2.3 - Trabalhos relacionados com a metodologia CRISP-DM. 42

Quadro 2.4 - Trabalhos relacionados à evasão em CS1 e atributos utilizados. 43

Quadro 3.1 - Objetivos específicos, técnicas e abordagens do CRISP-EDM/SD.....49

Quadro 4.1 - Síntese de informações sobre a base de dados utilizada. 51

Quadro 4.2 - Cursos vinculados ao ICE e FT utilizados no trabalho 52

Quadro 4.3 - Síntese dos atributos selecionados. 54

Quadro 4.4 - Lista de ferramentas de tratamento de dados utilizadas durante a pesquisa. 57

Quadro 4.5 - Síntese dos atributos selecionados 59

Quadro 4.6 - Resultado das avaliações comparativas dentre os modelos construídos. 62

Quadro 4.7 - Medidas derivadas da Matriz de Confusão (Modelo AdaBoost). 64

Quadro 4.8 - Resultado das avaliações comparativas dentre os modelos construídos. 65

Quadro 4.9 - Perfil sociodemográfico vinculado à evasão estudantil 65

Quadro 4.10 - Índice médio geral de evasão em CS1 no período.....67

Quadro 4.11 - Perfil sociodemográfico vinculado à evasão estudantil obtido a 70

LISTA DE ABREVIATURAS E SIGLAS

AVA	Ambiente Virtual de Aprendizagem
CART	*Classification and Regression Trees*
CBIE	Congresso Brasileiro de Informática na Educação
CRISP-DM	*CRoss-Industry Standard Process for Data Mining*
CRISP-EDM	*CRoss-Industry Standard Process for Educational Data Mining*
EDM	*Educational Data Mining*
FT	Faculdade de Tecnologia
ICE	Instituto de Ciências Exatas
IComp	Instituto de Computação
IPC	Introdução a Programação de Computadores
LA	*Learning Analytics*
L	*Logistic*
RF	*Random Forest*
ROC	*Receiver Operating Characteristic Curve*
SBIE	Simpósio Brasileiro de Informática na Educação
SGBD	Sistemas de Gerenciamento de Bancos de Dados
SMO	*Sequential Minimal Optimization*
SQL	*Sctructure Query Language*
SVM	*Support Vector Machine*
TISE	Conferência Internacional sobre Informática na Educação
UFAM	Universidade Federal do Amazonas
WEKA	*Waikato Environment for Knowledge Analysis*

SUMÁRIO

1 INTRODUÇÃO 14

1.1 Contextualização e motivação 14

1.2 Definição do problema de pesquisa 16

1.3 Justificativa 17

1.4 Objetivos 17

1.5 Metodologia da pesquisa 18

1.6 Organização do documento 19

2 REVISÃO BIBLIOGRÁFICA 21

2.1 Fundamentação teórica 21

2.1.1 Evasão de estudantes 21

2.2 Disciplinas introdutórias de programação (cs1) 23

2.2.1 Juiz online 24

2.2.2 Mineração de dados e mineração de dados educacionais 26

2.2.3 Aprendizagem de máquina 28

2.2.4 Métodos, técnicas e algoritmos 30

2.2.5 Árvores de decisão 30

2.2.6 Floresta aleatória 32

2.2.7 Adaboost 33

2.2.8 Redes neurais 34

2.2.9 Naive bayes 35

2.2.10 Máquinas de vetores de suporte 35

2.2.11 K-vizinhos mais próximos 36

2.3 Trabalhos relacionados 39

2.3.1 Mineração de dados educacionais baseada em CRISP-DM 39

2.3.2 Evasão em disciplinas CS1 42

3 METODOLOGIA DE DESENVOLVIMENTO DO MODELO PREDITIVO DE EVASÃO 46

3.1 Coleta e entendimento dos dados (fase 1) 47

3.2 Preparação dos dados (fase 2) 47

3.3 Predição da evasão (fase 3) 48

3.4 Teste e validação (fase 4) 49

4 MODELAGEM, APLICAÇÕES E RESULTADOS 50

4.1 Coleta e entendimento dos dados 50

4.2 Preparação dos dados 55

4.3 Geração do modelo 55

4.4 Testes e avaliações 62

4.5 Aplicação experimental e resultados 65

4.6 Interpretação dos padrões sociodemográficos de evasão 66

5.1 Contribuições 73

5.2 Limitações 73

5.3 Trabalhos Futuros 74

5.4 Considerações finais 74

REFERÊNCIAS BIBLIOGRÁFICAS 76

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO E MOTIVAÇÃO

A evasão estudantil é uma manifestação complexa observada em ambientes educacionais, um reflexo de múltiplas razões as quais precisam ser compreendidas no contexto socioeconômico, cultural e das instituições de ensino (KORI *et al.*, 2015; SILVA *et al.*, 2018). Caracteriza-se por um processo de exclusão de discentes do cenário educacional, determinado por influências internas e externas a essas instituições tais como motivacionais, estruturais e sociais difíceis de identificá-los e combatê-los (FRITSCH, 2015; GIRAFFA; MORA, 2015).

O Instituto de Pesquisa Econômica Aplicada[1] (IPEA) alerta para o caráter seletivo e excludente da educação brasileira, o qual decorreria prioritariamente da influência de fatores socioeconômicos sobre os estudantes nas diversas fases de sua trajetória educacional: da educação básica ao ensino superior (CORBUCCI, 2014). Nesse contexto, a evasão de estudantes em disciplinas introdutórias de programação - citadas comumente na literatura como *Computer Science 1* (ou CS1) (SANTANA; BITTENCOURT, 2018; PEREIRA *et al.*, 2019; BAZZOCCHI; FLEMMING; ZHANG, 2020) - constitui-se em um desafio frequentemente enfrentado também pelas instituições de ensino superior (DIGIAMPIETRI *et al.*, 2016).

As disciplinas de CS1 costumam fazer parte das grades curriculares de diversos cursos superiores das áreas de ciências exatas e engenharias, ainda que estas sejam uma competência dos cursos de graduação da área de tecnologia da informação, como ciência da computação e sistemas de informação. O termo em inglês *non-majors* é adotado para identificar os estudantes que cursam essas disciplinas introdutórias de programação oriundas de uma área diferente da que se graduam (SANTANA *et al.*, 2017).

O caráter multidisciplinar das diversas áreas de conhecimento tem contribuído para disseminação de CS1 em outros cursos diversos dos de informática, acarretando que profissões distintas passem a colaborar entre si, sendo que a programação de

[1] O Instituto de Pesquisa Econômica Aplicada (Ipea) é uma fundação pública federal vinculada ao Ministério da Economia. Suas atividades de pesquisa fornecem suporte técnico e institucional às ações governamentais para a formulação e reformulação de políticas públicas e programas de desenvolvimento brasileiros. Os trabalhos do Ipea são disponibilizados para a sociedade por meio de inúmeras e regulares publicações eletrônicas, impressas e eventos.

computadores se torna um exemplo de competência cruzada entre essas áreas de atuação (COUTINHO *et al.*, 2017), como as pertencentes às engenharias ou ciências exatas.

Desenvolver programas de computadores é uma competência complexa de se aprender por envolver uma gama de conhecimentos, estratégias e habilidades nos diferentes níveis de domínio do problema (HOED *et al.*, 2018) e que requer conhecimento de lógica, resolução de problemas, compreensão de ferramentas e abordagens para desenho e implementação de programas (GUL et al., 2017). Ensinar programação para *non-majors* costuma ser ainda mais desafiante, haja vista que estes raramente possuem motivação natural para programar, apresentando maior dificuldade no transcurso da disciplina e produzindo quadros não raros de evasão e reprovação em CS1 (SANTANA *et al.*, 2017; SANTANA; BITTENCOURT, 2018).

Desempenhos negativos de *non-majors*, como os casos de evasão escolar, comumente podem estar vinculados a características sociodemográficas destes estudantes (MA *et al.*, 2019). Isso pode significar, que alguns atributos, tais como idade, necessidade de trabalhar e estado civil sejam relevantes nesse contexto (KORI *et al.*, 2015).

Entender a forma pela qual os estudantes interagem e se fazem presentes na família e na sociedade, tais como responsabilidades profissionais, estrutura familiar e financeira, representa o ponto inicial para substituição de uma análise superficial encontrada em boa parte das ações direcionadas à manutenção do quadro de estudantes de disciplinas introdutórias de programação em cursos de graduação das áreas de engenharia e ciências exatas.

A representação de modelos preditivos que possam antecipar o risco de evasão estudantil tem sido objeto de interesse de diversas pesquisas em informática na educação na tentativa de contribuir com um entendimento metodológico da dinâmica de evasão. Técnicas de mineração de dados educacionais vem sendo comumente aplicadas na disponibilização de tais modelos (PETERSEN *et al.*, 2016; QUILLE *et al.*, 2017; PEREIRA *et al.*, 2019; BAZZOCCHI; FLEMMING; ZHANG, 2020). A Mineração de Dados Educacionais (*Educational Data Mining*) procura desenvolver ou adaptar métodos e algoritmos de mineração existentes, de tal modo que se prestem a compreender melhor os dados em contextos educacionais, produzidos principalmente por estudantes e professores, considerando os ambientes virtuais nos quais eles interagem.

A principal contribuição computacional deste trabalho reside na caracterização de um modelo para a predição do risco de evasão de *non-majors* em CS1 por meio de mineração de dados educacionais com uso de características sociodemográficas. Busca validar se a permanência ou não do estudante em CS1 vincula-se às condições e características sociais que o estudante está inserido, tais como acesso a computador ou internet em casa, migração de escola pública ou privada do nível médio para a universidade, estado civil ou existência de filhos.

O trabalho contribui, também, com uma avaliação sobre métodos de classificação para a caracterização e vinculação desses dados ao risco de evasão. Destaca-se, por fim, o aspecto social do trabalho ao produzir uma abordagem alternativa para o controle e acompanhamento da evasão em turmas de CS1, diferenciando-se dos demais trabalhos relacionados tratando o problema da evasão estudantil com base num modelo preditivo de tal risco ocorrer, aplicável *a priori* se utilizando de aprendizagem adquirida e de um perfil proposto a partir de dados estudantis sociodemográficos históricos.

1.2 DEFINIÇÃO DO PROBLEMA DE PESQUISA

O problema tratado neste trabalho é o da previsão de evasão de estudantes em disciplinas introdutórias de programação de computadores em cursos de graduação das áreas de ciências exatas e de engenharias. Detectar o risco de evasão desses estudantes é o primeiro passo para buscar formas de resolvê-la. É necessário dar maior atenção e de maneira antecipada aos que já apresentem tais características que podem levar à evasão escolar.

A evasão estudantil não deve ser explicada em termos individuais. Ao contrário, deve ser vista como uma consequência da influência de subconjuntos de variáveis sociodemográficas os quais parcelas de estudantes estão sujeitas. Esse problema pode ser sintetizado com a seguinte pergunta:

Como prever a evasão de estudantes non-majors em disciplinas introdutórias de programação de computadores a partir de características obtidas em base de dados sociodemográficos?

Este trabalho assume como hipótese que a mineração de dados sociodemográficos em bases de dados educacionais associada a algoritmos de Aprendizagem de Máquina viabiliza a predição da evasão de estudantes *non-majors*

em CS1.

Considera-se que é possível identificar, com aplicação da mineração de dados associada a métodos de classificação, relações entre variáveis sociodemográficas na tarefa de aprender quais destas estão mais vinculadas com uma variável preditora específica binária (evasão ou não do estudante). Enquanto que os dados sociodemográficos funcionam como entrada (variáveis preditivas) de um modelo preditivo (PROVOST; FAWCETT, 2016).

1.3 JUSTIFICATIVA

A evasão de estudantes em cursos de engenharias e ciências exatas nas disciplinas iniciais de computação tem sido um tema recorrente em trabalhos de informática na educação (PASSOS *et al.,* 2017; OLIVEIRA *et al*., 2007). Embora esses trabalhos objetivem minimizar a evasão de estudantes nas fases iniciais do curso, as observações são realizadas normalmente a posteriori no final das disciplinas, quando a evasão já ocorreu e não é mais possível atuar para revertê-la (PETERSEN *et al.,* 2016; HOED *et al*., 2018).

A observação do contexto sociodemográfico de estudantes non-majors utilizada neste trabalho viabiliza a previsão de evasão destes ainda no início semestre letivo de CS1 com uso de técnicas de mineração de dados e de descoberta de conhecimento, ampliando a utilização de dados dos históricos escolares e dispensando a utilização a posteriori de questionários.

1.4 OBJETIVOS

1.4.1 Objetivo geral

O objetivo geral desse trabalho é descrever um modelo de previsão de evasão de estudantes *non-majors* em disciplinas introdutórias de programação

1.4.2 Objetivos Específicos

Para atingir o objetivo geral, os seguintes objetivos específicos foram definidos:

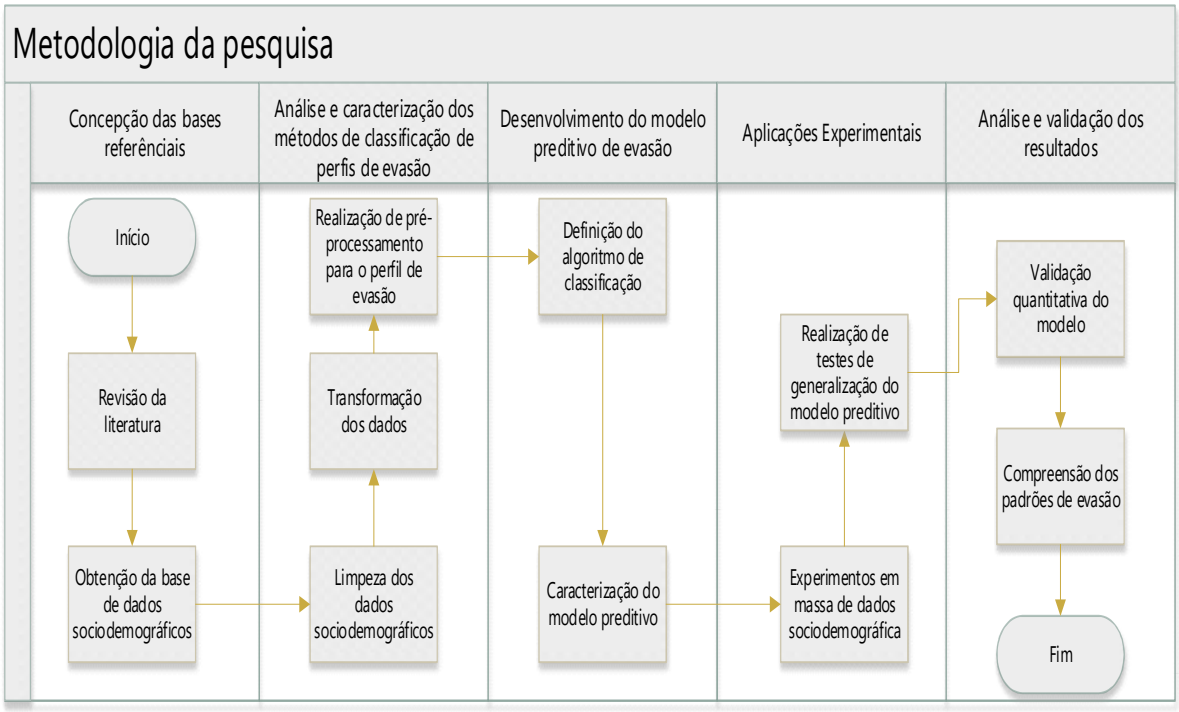
- Caracterizar perfis sociodemográficos que possam indicar riscos de evasão de estudantes em CS1;
- Descrever uma estratégia para avaliar *a priori* o risco de evasão de estudantes *non-majors*;

- Verificar a predição da evasão de estudantes *non-majors* a partir de dados sociodemográficos em base de dados educacionais.

1.5 METODOLOGIA DA PESQUISA

A pesquisa caracteriza-se por uma abordagem quantitativa experimental desenvolvida em cinco fases: (1) concepção das bases referenciais, (2) análise e caracterização dos métodos de classificação de perfis de evasão, (3) desenvolvimento do modelo de predição de evasão, (4) aplicações experimentais e (5) análise e validação dos resultados. A figura 1.1 ilustra o processo de caracterização dos modelos preditivos nos experimentos conduzidos.

Figura 1.1 - Fluxograma de aplicação da metodologia aplicada.



Fonte: Elaborado pelo autor, 2021.

A primeira fase consiste na concepção das bases referenciais da pesquisa a ser conduzida. Como tarefa inicial, é realizada uma revisão exploratória da literatura para conhecimento inicial do tema e identificação de técnicas de mineração de dados para criação modelos preditivos de evasão. Em seguida, deve-se realizar um mapeamento para definir com base na literatura, quais técnicas de mineração de dados poderão ser utilizadas neste trabalho.

Durante a análise e caracterização dos métodos de classificação de perfis de

evasão uma base de dados sociodemográficos de estudantes é anonimizada precisa ser fornecida para posteriormente serem realizados a limpeza, transformação e pré-processamento dos dados, necessários para construção do perfil sociodemográfico preditivo de evasão.

Após isso, a metodologia avança para a fase de desenvolvimento do modelo de predição de evasão, na qual técnicas de mineração de dados são utilizadas com o propósito de que seja definido através da comparação dos classificadores binários com base em sua taxa de acerto aquele que será utilizado na geração do modelo, sendo escolhido o com melhor acurácia. Por fim, com uso do classificador binário escolhido, um modelo preditivo propriamente dito é disponibilizado. É descrito o processo de utilização do modelo a partir da abordagem de mineração de dados educacionais e a sua aplicação na predição da evasão de alunos. Uma análise do modelo é realizada a fim de identificar se algum objetivo não tenha sido atingido ou se todas as etapas foram executadas corretamente a fim de corrigir o que for necessário.

Através das aplicações experimentais o modelo construído é validado através da realização de um conjunto de experimentos sobre uma massa de dados sociodemográfica distinta do utilizado para treinamento do mesmo a fim de que possam ser observados resultados sobre a aplicabilidade do modelo preditivo de evasão.

Na fase de análise e validação dos resultados é obtida a compreensão dos padrões, o modelo é validado e são realizados ajustes partir das considerações acerca dos resultados encontrados.

1.6 ORGANIZAÇÃO DO DOCUMENTO

Além desse capítulo introdutório, essa dissertação está organizada em outros quatro capítulos.

A Revisão Bibliográfica está dividida em duas seções principais: Fundamentação Teórica e Trabalhos Relacionados. A primeira seção apresenta a base teórica para o desenvolvimento do trabalho. Na segunda seção são descritos alguns dos principais trabalhos relacionados que discutem modelos preditivos e evasão de estudantes.

A Metodologia de Desenvolvimento do Modelo Preditivo de Evasão apresenta

o desenvolvimento do modelo preditivo de evasão estudantil. São apresentadas as etapas metodológicas para a estrutura organizacional para a mineração de dados e extração de conhecimento.

Na seção da Modelagem, Aplicações e Resultados são descritos os experimentos e os resultados obtidos com os modelos propostos nestes trabalhos. Além disso, o capítulo fornece uma análise sobre a relação construída entre atributos sociodemográficos e evasão de *non-majors* em CS1.

Por fim, a Conclusão apresenta as considerações finais do trabalho, além de indicar futuras possibilidades para estender esse trabalho.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo são introduzidos os conceitos e métodos necessários para o entendimento deste trabalho e as principais características sobre os trabalhos relacionados a esta pesquisa. O capítulo está dividido em duas seções principais, a fundamentação teórica e os trabalhos relacionados.

2.1 FUNDAMENTAÇÃO TEÓRICA

A busca por inferir o desempenho de estudantes *non-majors* de turmas iniciais de programação vem ganhando força nas áreas de mineração de dados educacionais e análise de aprendizagem (ou *Learning analytics*) - termo utilizado para designar a medição, coleta, análise e relatório de dados sobre os alunos e seus contextos, com o objetivo de entender e aperfeiçoar a aprendizagem e os ambientes em que ela ocorre.

Entender os eventuais motivos pelo qual influenciam os altos índices de evasão destes estudantes é uma questão fundamental para a área de pesquisa educacional de ciência da computação. Nessa seção será explicada toda a teoria envolvida e necessária para a compreensão da presente pesquisa, bem como as técnicas de aprendizagem de máquina utilizadas neste estudo para a construção dos modelos preditivos.

2.1.1 Evasão de Estudantes

Segundo o dicionário Aurélio da Língua Portuguesa[2], o termo evasão**,** do latim *evasione*, é um substantivo feminino que nomeia o ato de evadir-se, de fugir, de escapar, de sumir. É a ação de abandono de alguma coisa.

Diversos conceitos de evasão escolar se multiplicam no campo educacional, mantendo-se, porém, o sentido de que se trata de um problema social complexo, entendido como uma descontinuidade na presença de discentes no meio escolar, como a fuga de estudantes do ambiente educacional (KIRA, 1998) ou a interrupção no ciclo de estudos (GAIOSO, 2005).

[2] https://www.dicio.com.br/aurelio-2

Este trabalho considera como estudante evadido aquele que consta regularmente matriculado no respectivo período letivo em que a disciplina é oferecida e que ao fim do período é reprovado por falta.

Além disso, também considera como evadido aquele estudante reprovado por nota que não tenha participado de pelo menos 75% das avaliações escolares previstas no plano de ensino da disciplina (Resolução UFAM Nº 016/2017 – CONSAD), que pode ser representado pela fórmula

iEvasao = ([terf + term75] / [temd]) x 100

Onde *temd* corresponde ao total de estudantes matriculados na disciplina, *terf* o total de estudantes reprovados por falta, e *term75* o total de estudantes reprovados por média com menos de 75% de avaliações realizadas.

O critério usado para definir a evasão do estudante no ambiente educacional tem variado conforme levantamento realizado na literatura. Almeida (2007) destaca que os conceitos distintos de evasão mudam em razão de critérios escolhidos para categorizar os processos de ingresso e egresso dos estudantes dos eventos de ensino, conforme se pode atestar o quadro 2.1.

Quadro 2.1 - Definições de evasão e amplitudes do conceito.

CONCEITO DE EVASÃO	AMPLITUDE DO CONCEITO	REFERÊNCIAS
Evasão refere-se à desistência definitiva do estudante em qualquer etapa do curso	Não deixa claro se evasão se aplicaria apenas aos estudantes que chegaram a iniciar o curso ou se abrangeria também aqueles que apenas se matricularam e nunca iniciaram o curso	ABBAD; CARVALHO; ZERBINI, 2005
Evasão consiste em estudantes que não completam cursos ou programas de estudo, podendo ser considerada como evasão aqueles estudantes que se matricularam e antes mesmo de iniciar o curso ou programa de estudo.	Especifica que mesmo os estudantes que nunca começaram o curso devem ser considerados no cálculo das taxas de evasão.	MAIA; MEIRELES, 2005
Evasão é entendida como a saída definitiva do estudante de seu curso de origem, sem concluí-lo.	Não foi estabelecido nenhum critério de tempo no curso para saída do estudante.	UTIYAMA; BORBA, 2003

Fonte: Elaborado pelo autor, 2021.

A evasão tem múltiplas razões vinculadas ao contexto social, cultural, político e econômico em que a instituição está inserida (ALMEIDA, 2007), sendo possível ser identificado uma diversidade de novos fatores vinculados a ela e que são resultantes de vários processos sociais e culturais intrínsecos ou alheios ao ambiente educacional.

2.2 DISCIPLINAS INTRODUTÓRIAS DE PROGRAMAÇÃO (CS1)

As disciplinas introdutórias de programação, citadas comumente na literatura como *Computer Science 1* (CS1) (SANTANA; BITTENCOURT, 2018; PEREIRA *et al*., 2019; BAZZOCCHI; FLEMMING; ZHANG, 2020), tem em sua essência o ensino da lógica da programação de computadores e o desenvolvimento do pensamento sistemático e criativo (VIANA; PORTELA, 2019).

Essas disciplinas têm como objetivo prover ao estudante conceitos de lógica de programação para o desenvolvimento de soluções para problemas reais, por meio do uso das estruturas básicas de programação de computadores (COUTINHO *et al*., 2017). Diversas são as denominações atribuídas a essas disciplinas pelas instituições

de ensino: Introdução à Programação de Computadores, Introdução à Ciência da Computação, Lógica de Programação, entre outras (BOSSE; GEROSA, 2015).

As disciplinas de CS1 costumam também fazer parte de diversos cursos de graduação fora da área de computação (SANTANA *et al.,* 2017), principalmente nas áreas de ciências exatas e engenharias.

O ensino de programação de computadores para *non-majors,* assim como no Brasil (BOSSE; GEROSA, 2015; SANTANA *et al.,* 2017), também é tema de estudos desenvolvidos em outros países, como pode ser visto na pesquisa de Dawson *et al.* (2018), da *University of British Columbia* (EUA) e na pesquisa de Petersen *et al.* (2016), da *University of Toronto* (CAN). Em geral, o foco destas pesquisas está em como minimizar os efeitos do alto índice de evasão na disciplina de CS1 (PETERSEN *et al.,* 2016; STADELHOFER; GASPARINI, 2018). Estes trabalhos citam a importância de fazer uso de ferramentas que facilitem o desenvolvimento de programas durante a realização dos cursos, como é o caso dos ambientes de correção automática de códigos, detalhados a seguir.

2.2.1 Juiz Online

É fundamental que estudantes de programação de computadores resolvam o maior número possível de exercícios práticos a fim de que ocorra uma melhor assimilação e entendimento dos conceitos sobre o assunto. Contudo, torna-se inviável para o docente a correção de tarefas de várias turmas com o necessário zelo e tempo hábil (CARVALHO; OLIVEIRA; GADELHA, 2016).

Com base nessa demanda de ensino-aprendizagem[3], surgiu a necessidade de elaboração de ambientes de correção automática de códigos (ACAC), também conhecidos como Juízes *Online* (WASIK *et al.*, 2018), nos quais os estudantes codificam e submetem o código fonte para um veredicto do próprio sistema (FRANCISCO; PEREIRA JÚNIOR; AMBRÓSIO, 2016). O sistema corrige automaticamente o código, gerando um *feedback* para o estudante em tempo real, isto é, se a resposta está certa ou errada (CARVALHO; OLIVEIRA; GADELHA, 2016).

O *Codebench*[4] é um sistema Juiz *Online* criado e mantido por docentes da UFAM para servir de ferramenta de apoio educacional durante atividades práticas de

[3] Ensino-aprendizagem é o nome dado para um complexo sistema de interações comportamentais entre professores e alunos.
[4] http://codebench.icomp.ufam.edu.br/

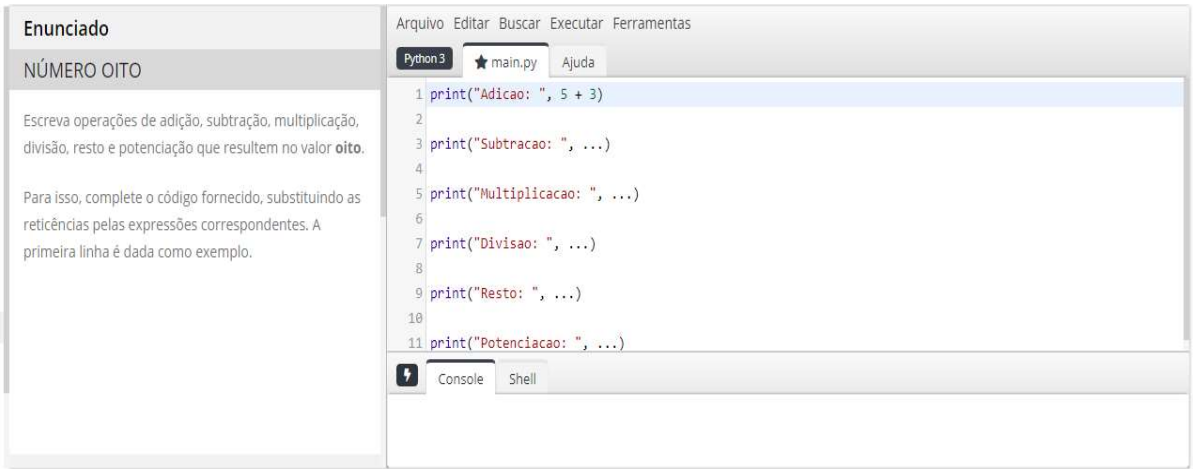
programação em CS1 (PEREIRA *et al.* 2019). A ferramenta é utilizada dentro de uma abordagem híbrida de ensino (*blended learning*). As chamadas *Blended Learning* mesclam o ensino presencial com atividades *online* baseadas em um ambiente virtual de aprendizagem (AVA).

Ambiente virtual de aprendizagem (AVA) é um tipo de sistema que viabiliza o desenvolvimento e distribuição de conteúdos educacionais em meio digital, no qual é possível acompanhar todo o processo de aprendizagem por parte do estudante, bem como acompanhar o desempenho e progresso do mesmo. (WASIK *et al.,* 2018).

A plataforma *Codebench* é, ainda, um ambiente que aplica técnicas de gamificação para motivar o aprendizado (RIBEIRO *et al.*, 2018). Gamificação no processo pedagógico significa adotar a lógica, as regras e o design de jogos (analógicos e/ou eletrônicos) para tornar o aprendizado mais atrativo, motivador e enriquecedor.

A resolução dos problemas de programação, cadastrados no *Codebench,* pode ser implementada utilizando-se diversas linguagens de programação, tais como: C++, Java e *Python*. No contexto dessa pesquisa, o ambiente citado forneceu dados sociodemográficos oriundos de um questionário disponível na ferramenta e preenchido pelos estudantes durante a primeira aula de CS1 do semestre letivo. A figura 2.1 apresenta um exemplo de enunciado de um problema de programação na interface do juiz online *Codebench*.

Figura 2.1 - IDE do juiz online Codebench.



Fonte: Codebench, 2021.

Nesta pesquisa, o papel do Codebench foi o de fornecer dados sociodemográficos a partir de um questionário *online* preenchido pelos estudantes de CS1 na primeira semana de aula. Serviu, ainda, como fonte de dados de notas e resultados nas turmas consideras na pesquisa.

2.2.2 Mineração de dados e mineração de dados educacionais

A Mineração de Dados (MD) é o processo ou a tarefa de descobrir padrões possivelmente desconhecidos em dados e informações e conseguir extrair informação e conhecimento dos mesmos (ALMEIDA SANTANA *et al.,* 2017).

A mineração de dados pode ser definida, ainda, como um conjunto de técnicas de exploração automáticas de grandes massas de dados de forma a descobrir novos padrões e relações que, devido ao volume de dados, não seriam facilmente descobertas manualmente pelo ser humano (KUMAR; SINGH; HANDA, 2017).

É baseada em critérios tais como amostragem, estimativa e teste de hipóteses a partir de estatística. A MD combina métodos tradicionais de análise de dados com algoritmos sofisticados para processamento de grandes volumes de dados, e está vinculada a técnicas de modelagem, algoritmos de busca, reconhecimento de padrões, aprendizagem de máquina e teorias de aprendizagem usadas em inteligência artificial (PROVOST; FAWCETT, 2016).

No âmbito educacional, a Mineração de Dados Educacionais (EDM – do termo em inglês *Educational Data Mining*) se fortaleceu com o advento da aplicação do ensino híbrido em cursos presenciais, semipresenciais e à distância (Hand; Mannila;

Smyth, 2001), o qual busca combinar suporte computacional (sistemas de informação, internet *etc*) ao processo de ensino-aprendizagem, objetivando melhorar o desempenho dos estudantes (SANTOS *et al.*, 2018).

EDM é uma área de pesquisa cujo objetivo é desenvolver métodos para analisar dados volumosos provenientes de fontes relacionadas a um cenário escolar, úteis na descoberta do conhecimento e no direcionamento da tomada de decisão diante de um problema presente no âmbito educacional.

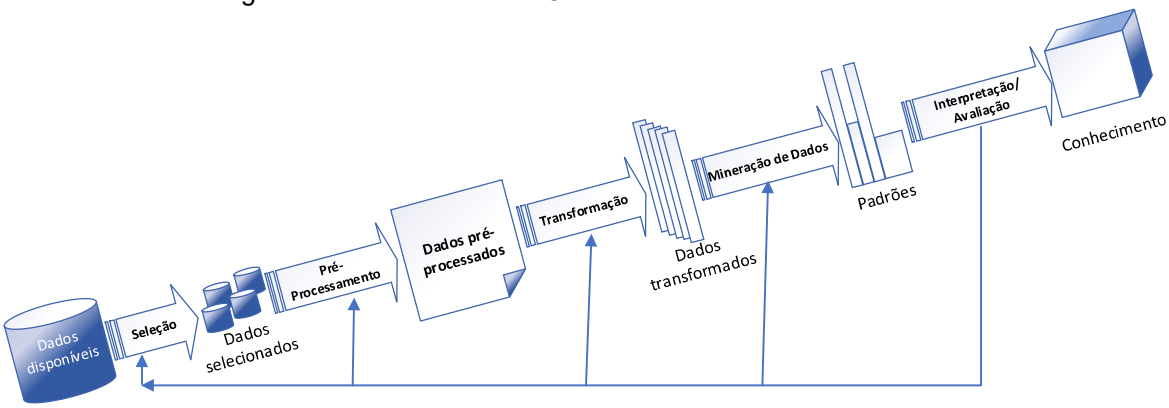
A EDM pode também ser considerada um processo de descoberta de conhecimento sobre dados brutos armazenados por sistemas escolares capazes de nortear desenvolvedores de softwares e pesquisadores que buscam soluções para tomada de decisão no ambiente educacional, no que remete a identificação precoce de comportamentos de aprovação, reprovação ou evasão escolar (SILVA *et al*., 2018).

Dados de estudantes mantidos pelas instituições de ensino, tais como situação econômica, idade, notas de avaliações, frequência nas disciplinas, viabilizam a realização de diversas formas de análise, podendo dar detalhes sobre o que ocorre durante o processo de ensino-aprendizagem de forma geral (KUMAR; SINGH; HANDA, 2017).

Neste contexto, o processo de extração de conhecimento (também conhecida como KDD, do inglês *Knowledge Discovery in Databases*) tem sido amplamente empregado na área educacional. A KDD é definida como um processo de identificação de padrões válidos, novos, potencialmente úteis e compreensíveis em repositórios de dados (FAYYAD; PIATETSKY; SMYTH, 1996).

É bastante comum ser encontrado o termo “mineração de dados” como sinônimo de KDD. De acordo com Fayyad, Piatetsky e Smyth (1996), KDD refere-se ao processo global de descoberta de conhecimento a partir de dados, enquanto a mineração de dados é uma fase desse processo. A figura 2.2 apresenta uma visão geral das fases do processo de KDD (FAYYAD; PIATETSKY; SMYTH, 1996).

Figura 2.2 - Descoberta de Conhecimento em Bases de Dados.



Fonte: Adaptado de Fayyad; Piatetsky; Smyth, 1996.

A primeira etapa corresponde à seleção de dados de uma instância através da criação de um conjunto ou subconjunto de dados que será o foco da descoberta de novos conhecimentos. Ele deve conter as informações necessárias para que os algoritmos de mineração possam alcançar o objetivo do pesquisador. Na etapa de pré-processamento os dados passam por uma limpeza ou eliminação de ruídos, e que inclui operações básicas para remoção de inconsistências.

Em seguida, os dados são transformados (etapa de transformação) para agregar valor semântico às informações ou características úteis para representar os dados da base. Com os dados pré-processados, então técnicas de mineração de dados são utilizadas com o propósito de encontrar padrões de acordo com o problema investigado. Por fim, na etapa de interpretação e avaliação, é obtida a compreensão dos padrões, incluindo a visualização dos modelos que resumem a estrutura e as informações presentes nos dados.

2.2.3 Aprendizagem de Máquina

A entrega do modelo preditivo de evasão de estudantes *non-majors* proposto neste trabalho foi tratada como um problema de classificação binária de estudantes de acordo com os conceitos de evadido e não evadido e com uso de algoritmos de aprendizagem de máquina supervisionada, conforme conceitos delineados a seguir.

O aprendizado de máquina (em inglês, *machine learning*) é um ramo da inteligência artificial baseado na ideia de que sistemas podem aprender com dados, identificar padrões e tomar decisões com o mínimo de intervenção. Ocorre quando um algoritmo ou um programa de computador tem a capacidade de melhorar seu desempenho nas tarefas que executa com base na experiência passada (PROVOST;

FAWCETT, 2016). Essa experiência ocorre com a utilização de dados e informações colhidas a partir de interações com o mundo real (dados históricos).

O aprendizado de máquina supervisionado busca encontrar o resultado para uma variável explícita ou um alvo (PROVOST; FAWCETT, 2016), como por exemplo: "*Podemos encontrar grupos de estudantes non-majors que tenham probabilidade particularmente elevada de evasão no decurso da disciplina de CS1?*"

Em aprendizagem de máquina, um método de classificação consiste em prever variáveis categóricas (não numéricas) binárias (estudante evadido ou não evadido), classificadas em faixas (níveis de renda, faixas de idade, etc.) ou puramente categóricas não ordenadas (gênero, raça, etc.) (CASTRO; FERRARI, 2016). Alguns métodos populares incluem árvores de decisão, redes neurais e máquina de vetores de suporte.

Por sua vez, o objetivo da previsão é construir um modelo que possa antever alguma situação desconhecida a partir de situações que já ocorreram (dados históricos) (AMARAL, 2016). Para a realização dessa tarefa, é importante conhecer os tipos de dados existentes e que tipo de informação se deseja obter ou prever. Por exemplo, para prever se um estudante vai ou não evadir dos estudos, é necessário verificar quais estudantes já se evadiram ou não e quais características destes estudantes, na época, contribuíram para essa decisão.

Os modelos construídos são utilizados para antever o valor do atributo categórico. A ideia básica é determinar quais características podem prever a qual categoria, entre várias, o atributo preditivo pertence. Essa categoria pode ser binária (por exemplo, o estudante evadiu-se ou não de CS1?) ou está contida em um conjunto de valores categóricos. Algoritmos característicos de cada método funcionam melhor para domínios e problemas específicos (AMARAL, 2016).

A seguir são apresentados os algoritmos de AM supervisionados de geração de modelo de classificação binária utilizados nesta dissertação. Uma apresentação detalhada não se faz necessária, uma vez que há diversas ferramentas de acesso livre que os implementam, mas é importante conhecer o funcionamento de cada um deles, com entendimento das vantagens e desvantagens de cada um destes a fim de que seja possível realizar de forma adequada uma interpretação de seus resultados.

2.2.4 Métodos, técnicas e algoritmos

Após a definição da tarefa a ser utilizada na mineração, deve-se, então, escolher a técnica e/ou algoritmos para realizar o processo de mineração. Para cada tarefa, várias opções podem ser testadas ou combinadas na busca de resultados mais apropriados para o problema tratado. Nesta pesquisa, por se tratar de um estudo que visa prever os riscos de evasão de estudantes *non-majors* em disciplinas introdutórias de programação, foi escolhida a técnica de classificação para o desenvolvimento de tal análise preditiva, por ser uma técnica consagrada na literatura de aprendizagem de máquina e mineração de dados e que tem apresentado resultados satisfatórios nos seus modelos preditivos.

No processo de classificação, duas etapas principais são realizadas: (1) a aprendizagem, na qual dados de treinamento são analisados por um algoritmo classificador, em que são atribuídos os rótulos de classe e o modelo aprendido ou classificador é representado sob a forma de regras de classificação; e (2) a classificação, na qual os dados de teste são usados para estimar a acurácia das regras de classificação. Se a acurácia for considerada aceitável, as regras podem ser aplicadas para a classificação de novos dados (HAN; KAMBER; PEI, 2011).

Os seguintes algoritmos de aprendizagem de máquina foram adotados na etapa de geração do modelo preditivo neste trabalho: *Decision Tree*, *Random Forest*, *Adaboost*, *Neural Network*, *Naive Bayes*, SVM e kNN. Os mesmos serão descritos sucintamente a seguir.

2.2.5 Árvores de Decisão

As Árvores de decisão (*Decision Trees*) são ferramentas que podem ser utilizadas para tomar decisões e inferir valores categóricos. A idéia é um aprendizado indutivo: cria-se uma hipótese baseada em instâncias particulares a qual gera conclusões gerais (SHARMA *et al.*, 2013).

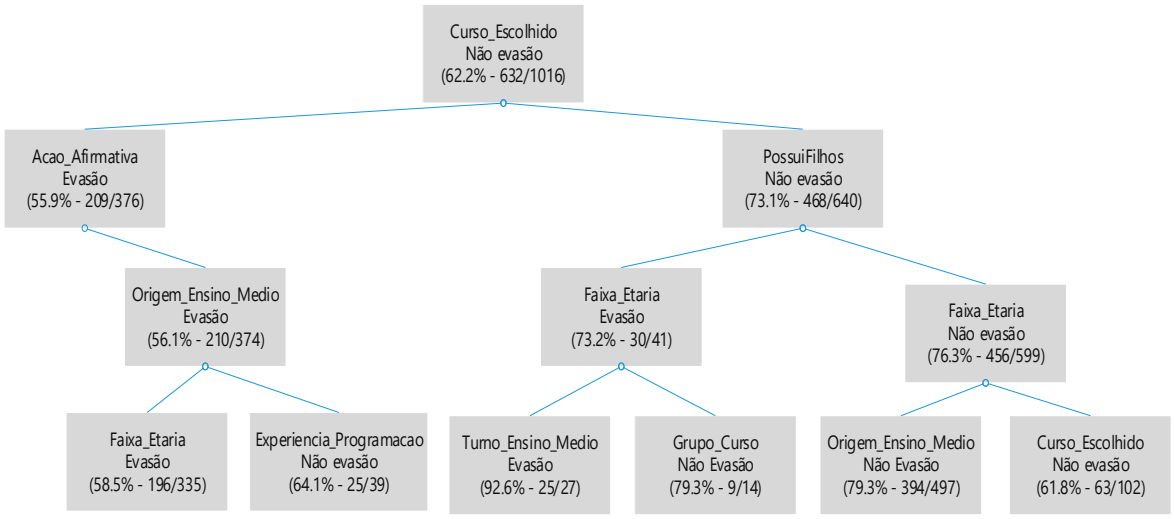
As árvores de decisão são amplamente utilizadas em aprendizagem de máquina pelos bons resultados obtidos com sua aplicação, além da fácil compreensão do processo seguido até a classificação (SHARMA *et al.*, 2013). Tais estruturas tomam como entrada uma situação descrita por um conjunto de atributos e retornam uma decisão, que é a categoria para o valor de entrada (HAND; MANNILA; SMYTH, 2001).

Para construir uma árvore de decisão consistente, o primeiro passo é identificar qual é o atributo mais promissor no que tange a predição, isto é, qual atributo possui maior relação com a classe, a fim de defini-lo como nodo raiz da árvore. Uma forma de realizar essa tarefa é calculando a entropia de cada atributo, isto é, mensurar a impureza ou incerteza de cada atributo. Assim, pode-se verificar a diferença entre a entropia antes e depois da divisão em uma sub-árvore, a fim de calcular o ganho de informação de cada atributo em uma subamostra da base de treino (QUINLAN, 1993). A entropia é um valor entre [0-1] e quanto mais próximo de 1, maior a impureza (SHANNON, 2001).

Objetos são classificados percorrendo um caminho da árvore, seguindo os arcos que contêm valores que correspondem a atributos no objeto. Em uma árvore de decisão a classificação de um caso se inicia pela raiz da árvore, e esta árvore é percorrida até que se chegue a uma folha. Em cada nó de decisão será feito um teste que irá direcionar o caso para uma sub-árvore. Nesse contexto, os nós filhos do nó atual são os possíveis resultados dos testes a serem realizados e os nós folhas o resultado final (SHARMA *et al.*, 2013).

As árvores de decisão podem manipular dados multidimensionais. Sua representação dos conhecimentos adquiridos em forma de árvore é intuitiva e, geralmente, fácil de assimilar pelos seres humanos. As etapas de aprendizagem e de classificação por árvores de decisão são simples e rápidas. Em geral, esses classificadores têm boa precisão (HAN; KAMBER; PEI, 2011). Na figura 2.3 é apresentada uma árvore de decisão gerada no decurso deste trabalho, utilizada para ilustrar critérios relevantes para evasão de um estudante.

Figura 2.3 - Exemplo de árvore de decisão do modelo preditivo de evasão em CS1.



Fonte: Elaborado pelo autor, 2021.

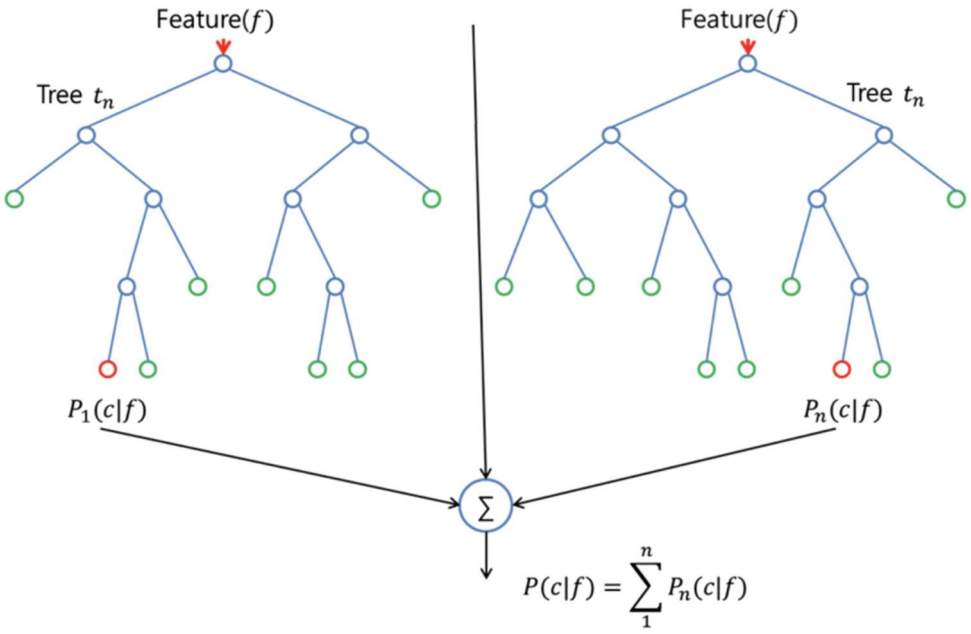
2.2.6 Floresta Aleatória

O algoritmo Floresta Aleatória (*Random Forest*) também é um classificador baseado em árvores de decisão, porém, nesse caso, consiste em várias árvores de decisão combinadas de forma a gerarem apenas um classificador final. Enquanto uma árvore de decisão comum utiliza todos os dados disponíveis para a construção de uma árvore, o *Random Forest* divide os dados aleatoriamente em subconjuntos e cada subconjunto gera uma árvore com atributos selecionados, também, aleatoriamente (BREIMAN, 2001).

Os subconjuntos gerados possuem *n* instâncias de treinamento, selecionadas de forma aleatória na base de dados e possuem *m* atributos selecionados aleatoriamente, para a geração de uma árvore. O nome do algoritmo se deve à maneira aleatória como são escolhidas as análises em cada etapa.

A floresta aleatória pode ser utilizada tanto para a classificação, quanto para a regressão. Ela faz parte do paradigma *ensemble learning* de aprendizado de máquina, onde vários modelos são agrupados com o objetivo de alcançar uma melhor generalização. Sendo assim, as florestas aleatórias são formadas por um conjunto de árvores de decisão. A figura 2.4 apresenta exemplos genéricos das árvores criadas pelo *Random Forest*.

Figura 2.4 - Exemplo de árvores da floresta aleatória.



Fonte: Adaptado de Breiman, 2001.

2.2.7 AdaBoost

O AdaBoost (*Adaptive Boosting)* é um classificador inspirado no método *Boosting* que emprega classificadores bases que se complementam iterativamente, contudo que não exige um grande conjunto de dados de treinamento (FREUND, 1995). O *AdaBoost* é utilizado junto a algoritmos base para cruzar duas ou mais informações sobre dados analisados e, ao comparar com a base de dados de treinamento, reconhecer padrões e fazer as classificações (JAUHARI; SUPIANTO, 2019).

O algoritmo resolveu diversos problemas e dificuldades encontradas anteriormente sobre *Boosting* e atualmente apresenta crescente número de pesquisas sobre a metodologia e suas aplicações por estudos pelo mundo (ZHANG; GAO; HU, 2018; JAUHARI; SUPIANTO, 2019; GAMIE; EL-SEOUD; SALAMA, 2020). Ainda segundo os inventores desse algoritmo, o *AdaBoost* apresenta algumas propriedades específicas que o tornam mais prático de ser utilizado e implementado se comparado aos outros algoritmos predecessores a ele.

Um ponto de atenção sobre o *AdaBoost* é sobre a possibilidade de ruídos no sistema devido à estratégia do algoritmo de enfatizar os dados mais difíceis de serem

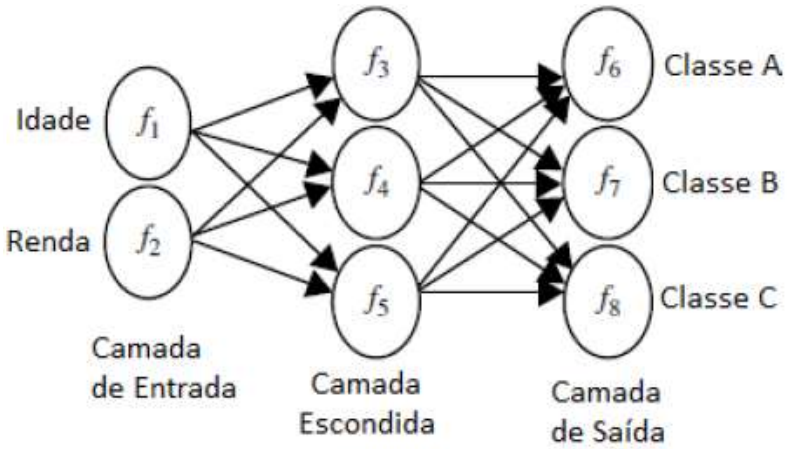
classificados, existindo a possibilidade de eventuais ruídos serem enquadrados nesse grupo e assim prejudicar os resultados gerados.

2.2.8 Redes Neurais

De forma genérica, uma rede neural pode ser vista como um conjunto de unidades de entrada e saída conectadas por camadas intermediárias e cada ligação possui um peso associado. É uma técnica com a qual se busca simular o comportamento dos neurônios humanos. Durante o processo de aprendizado, a rede ajusta esses pesos para conseguir classificar corretamente um objeto.

Necessitam de um longo período de treinamento, ajustes finos dos parâmetros e é de difícil interpretação, não sendo possível identificar de forma clara a relação entre a entrada e a saída. Em contrapartida, as redes neurais conseguem trabalhar de forma que não sofram com valores errados e também podem identificar padrões para os quais nunca foram treinados (ANDERSON, 1995), conforme ilustrado na Figura .

Figura 2.5 - Exemplo de Rede Neural Artificial.



Fonte: Adaptado de Wankhede 2014.

As redes neurais usam neurônios definidos como blocos de construção para construir modelos complexos de dados. Embora existam numerosas variantes de redes neurais artificiais, cada uma pode ser definida em termos pelas seguintes características (LANTZ, 2013):

- Uma função de ativação, que transforma o sinal de entrada de uma rede para um único sinal de saída a ser transmitido pelo restante da rede;
- Uma topologia de rede (ou arquitetura), que descreve o número de neurônios no modelo, o número de camadas e maneira pela qual eles estão ligados;

- O algoritmo de treinamento que especifica os pesos de cada conexão.

2.2.9 Naive Bayes

O classificador *Naive Bayes,* também chamado de classificador probabilístico, é baseado no teorema de *Thomas Bayes* o qual preconiza que é possível encontrar a probabilidade de certo evento ocorrer, dada a probabilidade de outro evento que já tenha ocorrido (TAN *et al.*, 2009).

O *Naive Bayes* parte do princípio de que não existe relação de dependência entre os atributos. É altamente escalável e tem apresentado alta acurácia e boa velocidade quando aplicados a grandes bancos de dados (WITTEN *et* al., 2011).

É conhecido como um classificador ingênuo, mas com vários relatos na literatura sobre sua competitividade para com outros classificadores considerados sofisticados, como os métodos de árvore de decisão e redes neurais. A partir dele, calcula-se a probabilidade de um dado elemento pertencer a uma classe por meio da equação a seguir (BERRAR, 2018).

$$P(c|x_i) = \frac{P(x_i|\, c)P(c)}{P(x_i)}$$

Onde C e Xi são eventos e:

- P(c) e P(Xi) são as probabilidades de c e Xi ocorrerem, sem levar em conta um no outro;
- P(c|Xi) é a probabilidade condicional, é a probabilidade de c dado que Xi é verdadeira; e
- P(Xi|c) é a probabilidade de Xi dado que c é verdadeira.

2.2.10 Máquinas de Vetores de Suporte

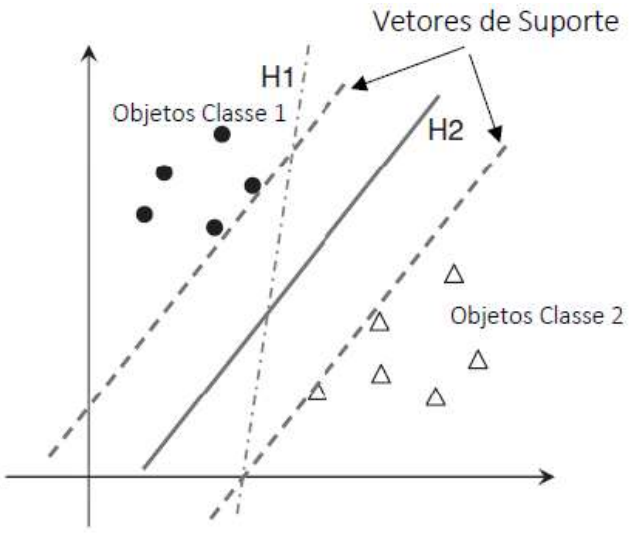
Em uma tarefa de aprendizagem de duas classes, o objetivo da SVM é encontrar a melhor função de classificação para distinguir entre membros das duas classes nos dados de treinamento (HAN; KAMBER; PEI, 2011).

A métrica para o conceito de melhor função de classificação pode ser realizada geometricamente, a partir do conceito de hiperplano, que é uma generalização de um plano em diferentes dimensões. Por exemplo, para uma dimensão, o hiperplano é um simples ponto. Para duas dimensões, o hiperplano é representado por uma reta.

No caso mais simples de classificação de duas classes, a SVM encontra um hiperplano (chamado de superfície de decisão) que separa as duas classes de dados com a mais ampla margem possível. Isso leva a uma boa precisão com generalização em dados não conhecidos ainda, assim como dá suporte a métodos de otimização especializados que permitem a SVM para aprender a partir de uma grande quantidade de dados (SAMMUT; WEBB, 2011).

Na Figura 2.6, dois hiperplanos (H1 e H2) são definidos e ambos separam corretamente os exemplos nas duas classes, porém o H2 tem uma margem maior do que H1 e seria usado como referência para novas classificações.

Figura 2.6 - Exemplo de classificação com SVM.



Fonte: Adaptado de Sammut e Webb, 2011.

Para um conjunto de dados linearmente separáveis, uma função de classificação linear corresponde a uma separação no hiperplano *f(x)*, que passa pelo meio das duas classes e as separa. Uma vez que essa função é determinada, novos dados *xn* pertencem à classe positiva se f(*xn*) > 0. Como podem existir muitos desses hiperplanos lineares, a SVM garante adicionalmente que a melhor função é encontrada por meio da otimização da margem entre as duas classes (SAMMUT; WEBB, 2011).

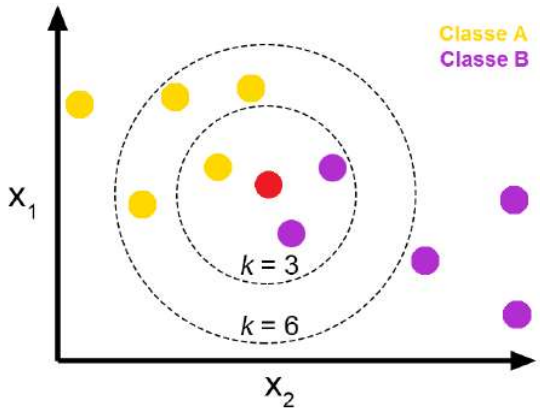
2.2.11 K-Vizinhos mais Próximos

Esse classificador é um algoritmo simples, que armazena todos os casos disponíveis e classifica novos casos com base em uma medida de similaridade (por exemplo, funções de distância) aos casos já armazenados. Tem sido amplamente

utilizado na área de reconhecimento de padrões (HAN; KAMBER; PEI, 2011).

Uma das vantagens deste método é a construção de uma diferente aproximação da função objetivo para cada exemplo do conjunto de dados. De forma geral, a classificação nestas técnicas ocorre a partir da maior quantidade de exemplos vizinhos de uma dada classe, em uma proposta que é conhecida como *K-Nearest Neighbours* (kNN), conforme representa a Figura 2.72.7.

Figura 2.7 - ilustração de classificação com algoritmo kNN.



Fonte: Adaptado de Gou *et al.,* 2019.

A definição do valor de k (número de vizinhos próximos) influencia no processo de classificação de novos dados. No exemplo ilustrado, se *k*=3, o novo elemento seria classificado na classe B, pois teria 2 vizinhos mais próximos (maioria) nessa classe. Para *k*=6, o elemento seria classificado na classe A, com 4 vizinhos mais próximos.

A mensuração do grau de semelhança entre casos nos quais todos os atributos são contínuos pode ser determinada por meio de uma função de avaliação, que de acordo com certos critérios calcula a distância entre os valores destes atributos. A ideia geral desse algoritmo consiste em encontrar os *k* exemplos rotulados mais próximos do exemplo não classificado e, com base no rótulo desses exemplos mais próximos, é tomada a decisão relativa à classe do exemplo não rotulado.

Os algoritmos da família kNN requerem pouco esforço durante a etapa de treinamento. Em contrapartida, o custo computacional para rotular um novo exemplo é relativamente alto, pois, no pior dos casos, esse exemplo deverá ser comparado com todos os exemplos contidos no conjunto de exemplos de treinamento.

No quadro 2.2 é apresentado um resumo sobre as características, vantagens e desvantagens de cada um dos algoritmos utilizados nesta pesquisa.

Quadro 2.2 - Resumo característico de algoritmos utilizados no trabalho.

ALGORITMO DE CLASSIFICAÇÃO	CARACTERÍSTICAS	VANTAGENS	DESVANTAGENS
Decision Tree	Usa o método de ramificação; Combina resultados possíveis em um gráfico.	Fácil entendimento e implementação.	Geralmente muito simples para relações muito complexas.
Random Forest	Encontra a média de muitas árvores, que individualmente são fracas; Divide os dados para cada árvore; Sua combinação é poderosa.	Usa a sabedoria das multidões; Tende a apresentar modelos de alta qualidade.	Pode apresentar lentidão para treinar o modelo; Resultados de difícil entendimento.
Adaboost	Tem seu funcionamento de maneira semelhante ao *Random Forest*	Relativamente robusto para *overfitting* em conjuntos de dados de baixo ruído; Possui apenas alguns hiperparâmetros que precisam ser ajustados para melhorar o desempenho do modelo; Fácil de entender e visualizar.	Facilmente prejudicado por dados ruidosos; Eficiência do algoritmo altamente afetada por *outliers*, pois o algoritmo tenta se ajustar a cada ponto perfeitamente; Em comparação com *Random Forest* tem desempenho pior quando recursos irrelevantes são incluídos.
Neural Network	Simula o funcionamento do cérebro humano; As mensagens são transmitidas de um neurônio para outro; Pode apresentar camadas.	Lida com tarefas extremamente complexas; Tem os melhores resultados em visão computacional.	Muito lento para treinar; Exige muito poder computacional; Resultados de difícil entendimento.
Naive Bayes	Simples e rápido, seu desempenho relativamente maior do que outros classificadores; Precisa apenas de um pequeno número de dados de teste para concluir classificações com boa precisão; Desconsidera completamente a correlação entre as variáveis (*features*), tratando cada uma de forma independente.	Treinamento rápido (varredura única); Caso o problema seja classificar texto ou algo do gênero, é uma das melhores alternativas; Rápido para classificar; Não sensível a características irrelevantes; Lida com dados reais e discretos; Lida bem com dados contínuos.	Se a correlação entre os fatores seja extremamente importante, pode falhar na predição da nova informação; Assume independência das características.
Support Vector Machine (SVM)	Encontra uma linha que melhor separa os diferentes outputs;	Robusto em relação à *overfitting*; Captura relações mais complexas; É efetivo quando se tem mais colunas do que linhas na base de dados;	Não funciona bem com grandes volumes de dados; Resultados de difícil entendimento; Difícil de escolher os valores dos parâmetros do modelo;
k-Nearest Neighbor (kNN)	Compara os valores do output com os demais para determinar quais são semelhantes ou diferentes	Fácil entendimento e implementação; Funciona para outputs numéricos e categóricos; Possui poucos pressupostos e parâmetros ao ser implementados;	Não bem funciona com grandes volumes de dados; Sensível a dados faltantes;

Fonte: Elaborado pelo autor, 2021.

2.3 TRABALHOS RELACIONADOS

Essa seção apresenta alguns dos principais trabalhos relacionados com esta pesquisa. Os trabalhos foram agrupados em duas categorias. Trabalhos que usam o processo CRISP-DM para minerar dados educacionais e trabalhos que tratam evasão de estudantes em disciplinas CS1 considerando dados sociodemográficos.

2.3.1 Mineração de Dados Educacionais Baseada em CRISP-DM

Um consórcio liderado por várias empresas consumidoras e fornecedoras em potencial de serviços de *data mining*, provedores de serviço e de ferramentas de mineração de dados iniciou, na segunda metade da década de 1990, o desenvolvimento do CRISP-DM (Acrônimo de *CRoss-Industry Standard Process for Data Mining*), um modelo de mineração de dados não proprietário, neutro, documentado e disponível livremente (Ramos *et al.*, 2020), diante da necessidade de definição de um paradigma para mineração de dados (DM – Data Mining) que pudesse ser aceita como padrão assim como o *Structured Query Language (SQL)* foi aceita como um padrão para bancos de dados relacionais,

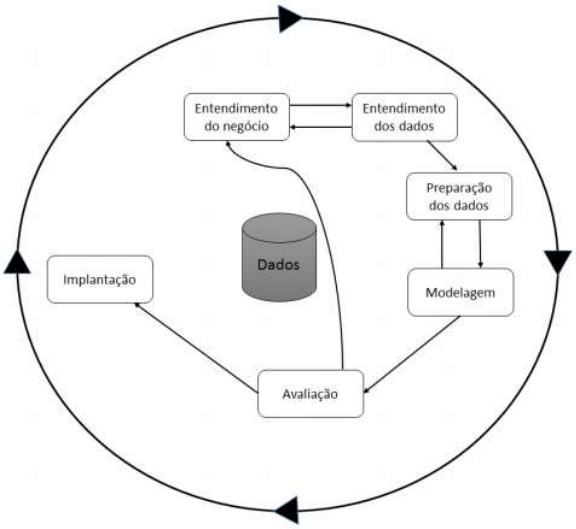
O CRISP-DM é um processo que descreve abordagens comumente usadas em mineração de dados para resolver problemas desse domínio (SHEARER, 2000) e é composto por seis fases organizadas de maneira interativa, conforme mostra a

Figura . Apesar de ser composto por fases, o fluxo não é unidirecional, podendo ir e voltar entre as fases e sua estrutura propõe auxiliar os pesquisadores desde o planejamento até a execução da mineração de dados educacionais (RAMOS *et al.*, 2020), passando pela especificação do processo da descoberta do conhecimento até a apresentação dos resultados alcançados.

Ao final, é feita a interpretação e a avaliação dos resultados obtidos, e o conhecimento descoberto é distribuído conforme tenha se definido no planejamento. Esse processo pode envolver várias iterações e quase sempre é necessário o retorno para fases anteriores.

A figura 2.8 representa a sequência de cada uma das seis fases que compõem o processo utilizado em nosso estudo: Entendimento do negócio, entendimento dos dados, preparação dos dados, modelagem, avaliação e implantação.

Figura 2.8 - O processo de mineração de dados CRISP-DM.



Fonte: Adaptado de Shearer, 2000.

Oreski, Pihir e Konecki (2017) destacam que técnicas de mineração de dados são usadas nas áreas de previsão e classificação em substituição às abordagens estatísticas tradicionais, inclusive na educação. Os autores realizaram uma metodologia abrangente baseada no processo CRISP-DM para mineração de dados educacionais, a fim de obter informações sobre a previsão de desempenho acadêmico.

O estudo explora em que medida as variáveis sociodemográficas como educação anterior, motivação e estilo de aprendizagem podem ajudar na identificação de estudantes com problemas de desempenho. Com base no método da árvore de decisão foi identificado o perfil do típico de estudante bem-sucedido academicamente. O estudo conclui que o desempenho acadêmico era significativamente influenciado pela formação anterior do estudante.

Castro, Espitia e Montilla (2018) utilizam uma metodologia dentro da perspectiva da descoberta de conhecimento em bancos de dados para a análise de desistência de estudantes baseada no modelo CRISP-DM a fim de identificar padrões de comportamento de tais estudantes. Os autores deste trabalho buscaram dar suporte à tomada de decisão e à criação de planos de ação por parte das instituições de ensino superior para reduzir o alto índice de evasão de estudantes.

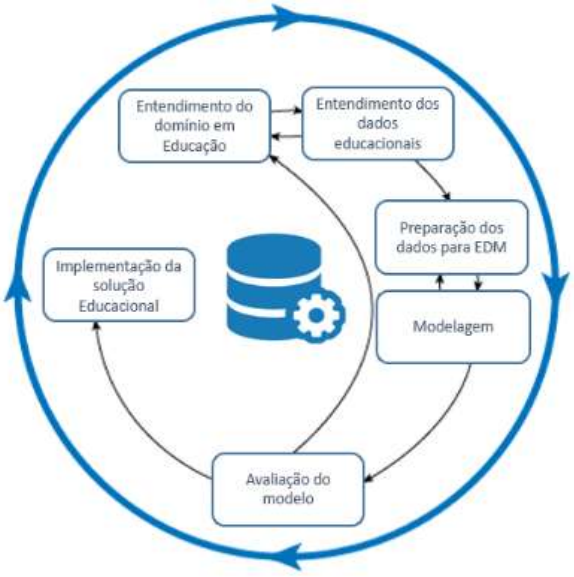
Lima e Fagundes (2020) realizam uma análise de correlação de fatores educacionais do censo escolar às ocorrências de evasão utilizando uma metodologia baseada nas fases do CRISP-DM. Os autores construíram um modelo preditivo

usando métodos de regressão linear baseado em fatores educacionais e econômicos dos estudantes.

Rodrıguez-Maya *et al.* (2017) utilizam um processo de extração de conhecimento baseado no modelo CRISP-DM. O trabalho propõe um modelo preditivo de evasão em universidades mexicanas baseado em uma tarefa de classificação com atributos acadêmicos de estudantes e notas do vestibular.

Ramos *et al.* (2020) descrevem um processo de adoção do consolidado modelo de mineração de dados CRISP-DM em cenários de dados educacionais, deixando ainda mais evidente a sua utilização nesses contextos. Os autores fornecem uma proposta de adaptação denominada CRISP-EDM. Apesar de seguir integralmente as seis fases do modelo original, o modelo proposto apresentou algumas particularidades quanto (1) o entendimento do domínio, (2) o entendimento dos dados e (3) a preparação dos dados, todas funcionando como uma adaptação para o contexto educacional, conforme ilustrado na Figura .

Figura 2.9 - O processo de mineração de dados CRISP-EDM



Fonte: Adaptado de Ramos *et al.,* 2020.

Por fim, para prever a evasão de estudantes, Utari, Warsito e Kusumaningrum (2020) utilizam dados acadêmicos de estudantes que se matricularam na universidade entre os anos de 2008 e 2012. O estudo utiliza o algoritmo *Random Forest* para prever a evasão em um processo com fases baseadas no CRISP-DM. Como resultado da pesquisa, o algoritmo de classificação conseguiu fornecer resultados de precisão de

até 93,43%. O quadro 2.3, apresentado a seguir, resume os principais trabalhos discutidos nesta seção.

Quadro 2.3 - Trabalhos relacionados com a metodologia CRISP-DM.

REFERÊNCIAS	MÉTODO
Castro; Espitia; Montilla, 2018	Aplica o modelo CRISP-DM em um processo KDD para a análise de evasão de estudantes, identificando padrões de comportamento destes
Lima; Fagundes, 2020	Propõe modelos de previsão do abandono escolar levando em consideração fatores educacionais e econômicos usando métodos de regressão, utilizando como metodologia as fases do CRISP-DM
Oreski; Pihir; Konecki, 2017	Aplicação do modelo CRISP-DM em dados oriundos de um ambiente educacional
Ramos *et al.,* 2020	Propõe uma adaptação do Modelo CRISP-DM para mineração de dados educacionais
Rodrıguez-Maya *et al.,* 2017	Apresenta um modelo preditivo baseado em uma tarefa de classificação. A fase de extração de conhecimento é baseada em um processo de DM, o Processo Padrão Cross Industry para Data Mining (CRISP-DM).
Utari; Warsito; Kusumaningrum, 2020	Utiliza o método CRISP-DM para predição de desempenho educacional aplicando o algoritmo Random Forest

Fonte: Elaborado pelo autor, 2021.

2.3.2 Evasão em Disciplinas CS1

A utilização de técnicas de mineração de dados aplicada à educação, ainda pode ser vista como um assunto recente. Há dúvidas, inclusive, sobre quais evidências ou atributos devem ser utilizados e sobre quais técnicas são mais adequadas (PETERSEN *et al.,* 2016). O quadro 2.4 sintetiza os estudos dos autores sobre o tema.

Quadro 2.4 - Trabalhos relacionados à evasão em CS1 e atributos utilizados.

REFERÊNCIAS	MÉTODO	ATRIBUTOS/OBSERVAÇÕES
BALMES, 2017	Pesquisa de correlação para determinar até que ponto dois fatores estão relacionados	Resultados do vestibular apenas em matemática e seu desempenho em todos os cursos de programação do primeiro ao quarto ano.
CASANOVA *et al.,* 2018	Algoritmos de classificação para predição de evasão em função do desempenho acadêmico	Foram analisadas as seguintes variáveis: área disciplinar do curso, idade, sexo, média das notas para entrar na universidade, se o curso ou a universidade foram a primeira escolha do estudante e nota média no primeiro ano.
DIGIAMPIETRI *et al.,* 2016	Modelo baseado no classificador *Rotation Forest* para identificação de estudante com risco elevado de evasão	Histórico escolar nas disciplinas do primeiro ano do curso de graduação.
GIRAFFA; MORA, 2015	Entrevista, análise estatística	Aplicação de questionário para inferir as possíveis causas relacionadas ao cancelamento da disciplina por parte de estudantes.
HOED *et al.,* 2018	Pesquisa por meio de questionários online	Foi observada influência de vários fatores no abandono, principalmente institucionais e profissionais e no conhecimento matemático. Em suma, Cursos nas principais áreas de Ciências, Matemática e Computação, que exigem níveis mais altos de matemática e abstração algorítmica, apresentam taxas de evasão mais altas.
PETERSEN *et al.,* 2016	Entrevista, análise estatística	E-mails aos estudantes que haviam desistido do curso, convidando-os a participarem de uma entrevista (questionário). Carga de trabalho e mudança de prioridades pessoais, ausência de experiência anterior de programação e dependência financeira de terceiros influenciaram diretamente na decisão de evasão.
QUILLE *et al.,* 2017	Algoritmos de classificação	Atributos demográficos, tais com o sexo e estado civil.
SANTANA *et al.,* 2017	Métodos predeterminados das pesquisas quantitativas com métodos emergentes das pesquisas qualitativas	Insucesso de *non-majors* é influenciado por diferentes aspectos, como linguagem. utilizada, nível escolar, tamanho das turmas e motivação.
SOUSA *et al.,* 2015	Modelo preditivo de classificação	Notas do Enem relacionadas com habilidades acadêmicas.

Fonte: Elaborado pelo autor, 2021.

Várias são as pesquisas que buscam a identificação precoce de fatores que possam contribuir para a previsão de evasão em CS1 (BALMES, 2017; QUILLE *et al.,* 2017; BAZZOCCHI; FLEMMING; ZHANG, 2020).

Conforme citado por vários pesquisadores (CASANOVA *et al.,* 2018; HOED *et al.,* 2018; PEREIRA FD *et al.* 2019), os motivos que possam desencadear o problema da evasão são diversos e estes contribuem para tal fato ocorrer de maneira rotineira,

incluindo a ausência de aptidão para programação e conhecimento mínimo prévio, priorização às disciplinas mais importantes na grade curricular, incompatibilidade do plano de ensino proposto para a disciplina pela instituição em relação ao perfil e expectativas pessoais (SANTANA *et al.,* 2017; SIGURDSON; PETERSEN, 2019) ou um contexto social não favorável à aprendizagem do estudante (PETERSEN *et al.*, 2016).

Giraffa e Mora (2015) aplicaram um questionário para inferir as possíveis causas relacionadas à desistência de estudantes na disciplina. O questionário foi aplicado em estudantes que desistiram de cursar CS1 pelo menos uma vez nos três semestres analisados. Os autores concluíram que 74% dos estudantes declararam que o critério “falta de tempo para estudar” foi um fator importante ou decisivo para o cancelamento da disciplina e que para 21% dos estudantes desistentes a ausência de conhecimento prévio de raciocínio-lógico matemático oriundo do ensino fundamental e médio foi fator decisivo para a evasão da disciplina.

Sousa *et al.* (2015) utilizam as notas do Exame Nacional do Ensino Médio (ENEM) relacionadas com habilidades acadêmicas, demonstrando que o processo do vestibular serve como preditor de desempenho para o primeiro ano universitário e que a nota da redação é boa preditora para o modelo.

Digiampietri *et al.* (2016) apresentam um método de geração de classificadores *Rotation Forest* para identificação do estudante com elevado risco de evasão com base no histórico escolar nas disciplinas do primeiro ano do curso de graduação. Os autores concluem que caso o estudante não perceba a relevância de uma disciplina ou qualquer outra atividade exigida em sua formação e/ou atuação profissional, ele facilmente poderá perder o interesse pela disciplina, incorrendo em reprovações e evasões.

Petersen *et al.* (2016) realizaram um estudo em um grande curso CS1 em uma instituição norte-americana, cuja disciplina é obrigatória para curso de Ciência da Computação. O estudo foi realizado durante o outono de 2015 e nesse período foi enviado e-mails aos estudantes que haviam desistido do curso, convidando-os a participarem de uma entrevista, sendo que como forma de incentivo, estes seriam compensados financeiramente. Um total de 327 estudantes evadidos recebeu o convite.

A análise dos dados demonstrou que carga de trabalho e mudança de prioridades pessoais, ausência de experiência anterior de programação e

dependência financeira de terceiros influenciaram diretamente na decisão de evasão.

Casanova *et al.* (2018) examina as decisões de estudantes universitários de permanecer ou abandonar os estudos, criando diferentes grupos preditivos em função do desempenho acadêmico. Foram analisadas as seguintes variáveis: área disciplinar do curso, idade, sexo, média das notas para entrar na universidade, se o curso ou a universidade foram a primeira escolha do estudante e nota média no primeiro ano.

O estudo proposto por Balmes (2017) aponta que estudantes de CS1 acham os cursos de programação difíceis no programa porque estes exigem conhecimento sobre estruturas, sintaxe, pensamento crítico e a capacidade de resolver problemas de programação. Por outro lado, a matemática é considerada significativa no tratamento dos cursos de programação porque melhora a capacidade lógica dos estudantes para resolver problemas de programação. Foram utilizadas diversas variáveis, como os resultados do vestibular apenas em matemática e seu desempenho em todos os cursos de programação do primeiro ao quarto ano.

Por fim, o trabalho de Quille *et al. (*2017) desqualificou antigo estereótipo vinculado ao gênero do estudante, demonstrando estatisticamente que homens e mulheres não diferem na capacidade de permanência na disciplina de CS1. O quadro 4 apresentada uma tabela resumo dos métodos atributos encontrados na literatura e que tratam de evasão em CS1.

3 METODOLOGIA DE DESENVOLVIMENTO DO MODELO PREDITIVO DE EVASÃO

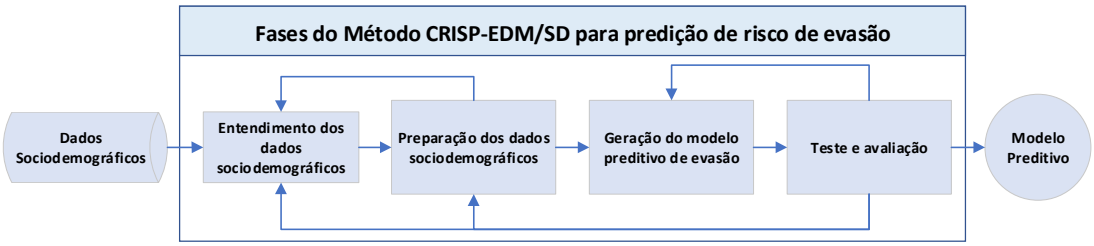
Esse capítulo descreve o processo de construção do modelo preditivo de evasão de estudantes *non-majors* em disciplinas introdutórias de programação de computadores com uso de dados sociodemográficos em ambiente educacional, CRISP-EDM/SD, construído a partir do Modelo CRISP-EDM (RAMOS *et al.* 2020).

O modelo é construído a partir de dados extraídos de 1.016 estudantes de turmas de CS1 da Universidade Federal do Amazonas (UFAM), sendo utilizadas 16 turmas do 1º e 2º semestres do ano de 2017 e do 1º semestre de 2018.

As disciplinas de Introdução à Programação de Computadores (IPC) e Introdução à Ciência da Computação (ICC), realizadas pelos estudantes *non-majors* da UFAM seguem uma metodologia híbrida de ensino, mesclando o ensino presencial com o uso de ferramentas tecnológicas de apoio ao aprendizado. No caso em questão, foi utilizado para resolução das listas de exercícios um ambiente de correção automática de código chamado *Codebench*, desenvolvido pela UFAM (CARVALHO; FERNANDES; GADELHA, 2016).

O modelo preditivo de evasão classifica os estudantes de acordo com os conceitos de evadido e não evadido por meio de uma análise quantitativa da previsão de evasão. Para a construção do modelo, diversos algoritmos de aprendizagem de máquina supervisionada são avaliados, inclusive os abordados durante a revisão da literatura deste trabalho (ver seção 2.2). O modelo comporta quatro fases distintas, conforme ilustrado na figura 3.10, as quais serão detalhadas a seguir.

Figura 3.1 - Processo para a construção do Modelo para previsão do risco de evasão com dados educacionais sociodemográficos.



Fonte: Elaborada pelo autor com adaptações de Ramos *et al.,* 2020.

Há possibilidade de retorno a etapas anteriores para ajustes e verificações nos procedimentos adotados como, por exemplo, os modelos inicialmente obtidos não

apresentem resultados relevantes em suas métricas de avaliação ou caso alguma variável se mostre pouco significativa no modelo. A seguir, é descrito cada uma das quatro fases do modelo e são detalhadas as tarefas realizadas em cada uma das mesmas.

3.1 COLETA E ENTENDIMENTO DOS DADOS (FASE 1)

A coleta de dados é baseada em qualquer fonte de dados que contenha informações sociodemográficas de estudantes. A fonte dos dados pode ser, por exemplo, dados de um sistema acadêmico, de um ambiente virtual de aprendizagem ou uma planilha eletrônica com informações sociodemográficas dos alunos. Além disso, este é o estágio de familiarização com os dados do problema e identificação da qualidade destes. A compreensão dos dados é essencial para a elaboração de modelos preditivos eficientes. Deve-se entender as características e limitações das bases de dados, o histórico, sua composição, seu tipo e se os dados disponíveis são suficientes para entendimento do problema proposto.

3.2 PREPARAÇÃO DOS DADOS (FASE 2)

Esta etapa corresponde à realização da seleção e integração dos dados, tratamento dos dados brutos (remoção de ruídos) e identificação das variáveis que serão trabalhadas. É necessário transformar dados não estruturados em estruturados, fazer tratamento de dados faltantes, converter diferentes tipos de dados de acordo com a necessidade, entender se os dados são categóricos, contínuos, se devem ser normalizados ou não, dentre muitas outras tarefas.

Nesta etapa também é realizada a preparação dos dados (bases de dados de treino, validação e teste) que serão usadas para geração e avaliação dos modelos preditivos. A base de treino é utilizada para treinar os algoritmos e, geralmente, é a base que tem a quantidade majoritária de instâncias. Uma base de treino construída equivocadamente refletirá nos modelos gerados ocasionando *underfitting* (não aprendeu a resolver o problema) ou *overfitting* (não aprendeu a generalizar o problema ou somente funciona com a base de treino caracterizando um vício). Modelos com *underfitting* ou *overfitting* não são confiáveis e possuem problemas de acurácia. Já a base de validação é usada para validar os modelos que estão sendo gerados durante o treinamento.

Diante disso, a preparação dos dados é essencial para obtenção de modelos

com bons resultados. Nessa fase, devem ser realizadas quatro tarefas:

- ***Seleção de dados:*** é definida como o processo de determinação do tipo e fonte de dados apropriados, bem como dos instrumentos adequados para coletar dados. A seleção de dados precede a prática real de coleta de dados. Deve-se considerar as informações disponíveis em bases de dados educacionais, sendo necessário integrar todas as bases em uma única para a realização da previsão.
- ***Limpeza de dados:*** é o processo de detecção e correção (ou remoção) de registros corrompidos ou imprecisos de um conjunto de registros, tabela ou banco de dados e se refere à identificação de partes incompletas, incorretas, imprecisas ou irrelevantes dos dados e, em seguida, substituindo, modificando, ou excluindo os dados sujos ou grosseiros.
- ***Derivação de dados:*** Derivação de novos atributos que serão úteis para o modelo. Talvez nem todos os dados necessários estejam à disposição. É possível que se tenha que criar novos dados para seu modelo. Por exemplo, talvez seja necessário um campo ou coluna no conjunto de dados que diga qual dia da semana representa uma determinada data;
- ***Integração de dados:*** essa tarefa é necessária quando for necessário juntar duas fontes de dados diferentes.

3.3 PREDIÇÃO DA EVASÃO (FASE 3)

Fase onde é realizada a predição propriamente dita de evasão de estudantes *non-majors* baseado em dados sociodemográficos. Podem ser construídos livremente vários modelos com base em várias técnicas de modelagem diferentes. Esta fase tem quatro tarefas:

- ***Seleção de técnicas:*** Definição de quais algoritmos devem ser utilizados para fornecimento do modelo preditivo.
- ***Gerar plano de teste:*** Enquanto se aguarda a abordagem de modelagem, é necessário dividir os dados em conjuntos de treinamento, teste e validação.
- ***Construção do modelo:*** Construção do modelo propriamente dito com base nos dados disponíveis aplicados aos algoritmos escolhidos.
- ***Avaliação do modelo:*** O resultado desta etapa frequentemente leva a iterações de ajuste do modelo até que os melhores modelos sejam

encontrados. A fase dura até que um modelo suficientemente bom seja encontrado. A avaliação é realizada por meio de técnicas que buscam identificar capacidade de previsão das técnicas e modelos de mineração de dados usando métricas que possam caracterizar o desempenho da predição antes de incorporá-lo modelo.

3.4 TESTE E VALIDAÇÃO (FASE 4)

Na fase de teste e avaliação devem ser realizadas considerações acerca dos resultados. Se necessário, pode-se voltar para a primeira etapa, caso seja entendido que o objetivo do projeto ainda não tenha sido alcançado. Análise do trabalho realizado a fim de identificar se algo foi esquecido, se todas as etapas foram executadas corretamente e corrigir o que for necessário.

A seguir no quadro 3.1 são listados os objetivos específicos, as técnicas e abordagens aplicadas em cada uma das fases do modelo.

Quadro 3.1 - Objetivos específicos, técnicas e abordagens do CRISP-EDM/SD.

Fase	Objetivos específicos	Técnica/Abordagem
Fase 01	Caracterizar o problema da evasão em CS1. Obter uma base de dados educacional sociodemográfica. Identificar as diversas variáveis que possam compor um perfil de evasão.	• Questionário Sociodemográfico • *Definição e mapeamento das variáveis sociodemográficos*
Fase 2	Coleta e tratamento dos dados representativos do perfil de evasão de estudantes.	• *Extração dos dados na base de dados* • Limpeza e transformação dos dados
Fase 3	Definir e validar um modelo preditivo da evasão de estudante, a partir do uso de EDM, com base nas variáveis sociodemográficas.	• Seleção de métodos, técnicas e algoritmos de aprendizagem de máquina • Planejamento e execução de testes nos dados existentes
Fase 4	Validar o modelo. Realizar ajustes a partir das análises sobre os resultados. Descrever o processo de utilização do modelo a partir da abordagem de mineração de dados educacionais e a sua aplicação na predição da evasão de alunos	• Análise quantitativa • Avaliação dos resultados • Revisão do processo

Fonte: Elaborado pelo autor, 2021.

4 MODELAGEM, APLICAÇÕES E RESULTADOS

A modelagem tem por objetivo disponibilizar o modelo preditivo de evasão por meio das técnicas e métodos de mineração de dados usando as informações educacionais e sociodemográficas, conforme Metodologia de Desenvolvimento do Modelo Preditivo de Evasão descrito no capítulo 3.

4.1 COLETA E ENTENDIMENTO DOS DADOS

A Faculdade de Tecnologia (FT) e o Instituto de Ciências Exatas (ICE) da Universidade Federal do Amazonas (UFAM) são responsáveis por oferecer, na cidade de Manaus, cursos de graduação em diversas áreas técnicas tais como: Estatística, Engenharia Civil, Engenharia Elétrica, Engenharia de Materiais e Engenharia Química.

A Universidade Federal do Amazonas (UFAM) oferece disciplinas introdutórias de programação de computadores (CS1) para estudantes de onze cursos de engenharias e ciências exatas. Na maioria das vezes, ela está presente na grade curricular do primeiro ano desses cursos e, semestre após semestre, apresenta elevados índices de reprovação e evasão (PEREIRA FD *et al.,* 2019). Atualmente a metodologia de ensino é híbrida (CARVALHO; FERNANDES; GADELHA, 2016) e as aulas se dividem em dois momentos:

- **Presencial**: o professor é o elemento central da aula, ele apresenta o conteúdo aos estudantes, que dispõem de computadores para acompanhar os slides e executar exercícios de exemplo. As avaliações também são presenciais e realizadas por meio do computador.
- ***Online***: o estudante tem a flexibilidade para decidir onde irá realizar os exercícios práticos, que pode ser em laboratório ou em qualquer outro lugar que tenha um computador conectado à internet. Os exercícios são liberados após a apresentação do conteúdo e podem ser realizados até a data da avaliação. O estudante que decidir realizá-los em laboratório tem o apoio de um tutor e fácil acesso ao professor.

As disciplinas de introdução a programação são compostas por 60 horas de aulas, sendo 30h de carga teórica e outras 30h de prática - é dividida em sete módulos de oito horas, onde cada um destes módulos possui uma avaliação parcial aplicada após os estudantes terem resolvido exercícios de fixação. Ainda existe um módulo

introdutório de quatro horas para apresentação da disciplina e motivação da turma. A linguagem de programação adotada na disciplina é o *Python 3.x*

O estudante é considerado aprovado caso a média final seja maior ou igual a cinco e tenha frequência superior a 75% do total de aulas. Caso contrário, serão considerados reprovados por falta ou por nota, conforme o caso. Por outro lado, o estudante é considerado como evadido caso não tenha realizado pelo menos 75% das avaliações escolares previstas no plano de ensino da disciplina, ainda que conste como reprovado por falta ou por nota no boletim acadêmico (Resolução UFAM Nº 016/2017 – CONSAD.

Foram obtidos os dados anonimizados de estudantes de 11 cursos de graduação presencial nas áreas de engenharia e ciências exatas em que a disciplina de Introdução à Programação (CS1) era um componente curricular obrigatório. Nos experimentos foram utilizadas 16 turmas do 1º e 2º semestres dos anos de 2016 e 2017 e do 1º semestre de 2018, totalizando 1.016 registros de matrícula, entre os quais 872 correspondiam a estudantes únicos. O quadro 4.1 apresenta uma síntese dos dados coletados da amostra pesquisada.

Quadro 4.1 - Síntese de informações sobre a base de dados utilizada.

ITEM	CARACTERÍSTICAS
População-alvo	Estudantes de graduação das áreas de engenharia e ciências exatas
Limitação de escopo	Disciplinas introdutórias de programação (*CS1*)
Fonte de dados	Juiz *online* e sistema de controle acadêmico
Períodos letivos avaliados	2016/1, 2017/1 e 2018/1
Total de registros de matrícula	1.016
Cursos de graduação	11 cursos de ciências exatas e engenharia
Atributos selecionados	13 atributos (ver Tabela 4.4)
Variável dependente	Situação final na disciplina (evasão ou não evasão)

Fonte: elaborado pelo autor, 2021.

Os cursos aqui considerados são vinculados ao Instituto de Ciências Exatas (ICE) ou à Faculdade de Tecnologia (FT) da UFAM (quadro 4.2). Frisa-se que as disciplinas citadas seguem uma metodologia híbrida de ensino, mesclando o ensino presencial com o uso de ferramentas tecnológicas de apoio ao aprendizado. No caso em questão, foi utilizado para resolução das listas de exercícios um ambiente de

correção automática de código chamado *Codebench*, desenvolvido pela própria UFAM.

Quadro 4.2 - Cursos vinculados ao ICE e FT utilizados no trabalho

ITEM	CURSO DE GRADUAÇÃO
1	Matemática Licenciatura Matutino
2	Matemática Bacharelado Diurno
3	Matemática Licenciatura Noturno
4	Física Licenciatura Noturno
5	Física Bacharelado Noturno
6	Engenharia de Materiais
7	Estatística
8	Engenharia Química
9	Engenharia de Petróleo e Gás
10	Engenharia Mecânica
11	Engenharia de Produção

Fonte: elaborado pelo autor, 2021.

Atributos utilizados na pesquisa foram extraídos do cadastro geral de estudantes da UFAM e de um questionário socioeconômico (figura 4.1) preenchido nas primeiras aulas de CS1 no *Codebench*. A variável dependente é representada pela palavra "*status_estudante*" e é dividida, no máximo, em duas classes: evadido e não evadido.

Figura 4.1 - Questionário Sociodemográfico utilizado na pesquisa.

1) Ensino Médio
a) Qual o nome da escola onde você cursou o último ano do Ensino Médio? _____
b) Qual o tipo de escola onde você predominantemente estudou no Ensino Médio?
( ) Pública Convencional ( ) Particular Convencional ( ) Técnica
c) Em que turno você predominantemente cursou o Ensino Médio?
( ) Matutino ( ) Vespertino ( ) Noturno ( ) Integral
d) Em que ano você concluiu o Ensino Médio? _____
2) Experiência prévia em programação
a) Das linguagens de programação listadas abaixo, qual delas você tem algum conhecimento?
( ) Nunca programei ( ) C ( ) C++ ( ) Python ( ) Java ( ) Scratch ( ) Outra
b) Possui computador pessoal (PC, desktop ou notebook) na sua casa?
( ) Sim ( ) Não
3) Experiência de trabalho ou estágio
a) Você já trabalhou ou estagiou antes de iniciar esta graduação?
( ) Sim ( ) Não
4) Graduação prévia
a) Você já iniciou algum curso de graduação antes do atual?
( ) Sim ( ) Não
5) Informações pessoais
a) Qual o ano em que você nasceu? _____
b) Qual é o seu sexo?
( ) Masculino ( ) Feminino
c) Qual o seu estado civil?
( ) Solteiro ( ) Casado ( ) Viuvo
d) Você tem filho(s)?
( ) Sim ( ) Não

Fonte: Elaborada pelo autor, 2021.

Foram definidos os atributos a serem extraídos, realizada limpeza e transformação dos dados selecionados, extração dos dados da camada de interesse e avaliação dos dados minerados a fim de gerar o conhecimento sobre os mesmos. Os atributos coletados foram comparados ao atributo de saída correspondente ao Resultado de cada estudante em CS1: evasão ou não evasão.

As informações de interesse, extraídas de dados oriundos do *Codebench*, para cada ex-estudante com registro de evasão na disciplina CS1, e consideradas pelo presente trabalho estão descritas no quadro 4.2.

Quadro 4.2 - Síntese dos atributos selecionados.

ITEM	**ATRIBUTO**	**DESCRIÇÃO**
1	A01_sexo	Sexo (gênero) do estudante
2	A02_estado_civil	Estado civil do estudante
3	A03_experiencia_programacao	Experiência em alguma linguagem de programação
4	A04_faixa_etaria	Faixa etária a qual pertence o estudante
5	A05_filhos	Sim ou não
6	A06_origem_ensino_medio	Categoria (pública, privada, etc.) a qual pertence à escola de nível médio do estudante
7	A07_vaga_acao_afirmativa	Identificação de estudante pertencente a algum tipo de cota por ação afirmativa
8	A08_experiencia_trabalho	Estudante já trabalhou algum dia
9	A09_curso	Nome do curso de ciências exatas e de engenharias do estudante
10	A10_grupo_curso	Ciências Exatas ou Engenharias
11	A11_acesso_internet	Possui acesso à internet
12	A12_pc_casa	Possui Computador em casa
13	A13_turno_ensino_medio	Matutino, vespertino, noturno ou integral

Fonte: Elaborado pelo autor, 2021.

Na obtenção de dados brutos, foram extraídos registros de resultados finais em CS1 de estudantes e ex-estudantes de cursos de graduação das áreas de engenharia e ciências exatas, disponíveis na base de dados do sistema de gestão educacional da instituição. Foram extraídos, ainda, dados brutos de informações socioeconômicas dos estudantes a partir da base de dados do *Codebench*. Os históricos de resultados finais em CS1 extraídos estão vinculados a estudantes ativos e egressos do quadro de discentes da universidade.

4.2 PREPARAÇÃO DOS DADOS

Durante a preparação dos dados, foi realizada uma cópia das tabelas necessárias para o servidor local, durante consultas codificadas com a linguagem SQL. Houve restrições quanto à utilização de Visões no Banco de Dados e a conexão com a ferramenta *HeidiSql*[5].

O sistema inicialmente possuía dados de todos os usuários do *Codebench*, o que incluía dados de professores e administradores além dos dados dos estudantes. Uma seleção de tuplas foi realizada, considerando informações apenas referentes à turma em questão, assim como os usuários com o perfil de acesso intitulado — *"estudante"*.

Para viabilizar o processo de mineração de dados, algumas transformações nos dados foram necessárias. Os dados registrados na base do *Codebench* continham informações limitadas quanto às informações sociodemográficas e fez-se necessária a complementação de dados oriundos de outro conjunto de dados. Por essa razão, dados complementares foram obtidos com o auxílio de planilhas retiradas do sistema de gerenciamento acadêmico da organização, o que tornou o processo de integração uma transformação imprescindível.

Foram selecionadas 13 variáveis sociodemográficas, sendo algumas resultadas de derivações, como é o caso do atributo *A04_faixa_etaria*, criado a partir do atributo *data_nascimento*. Além disso, ruídos causados por rotinas administrativas no sistema também foram removidos. Os dados foram organizados em uma tabela única que permitisse uma visão integrada dos registros, em que cada linha representasse a totalidade das informações dos estudantes e as colunas apresentassem cada um de seus atributos.

4.3 GERAÇÃO DO MODELO

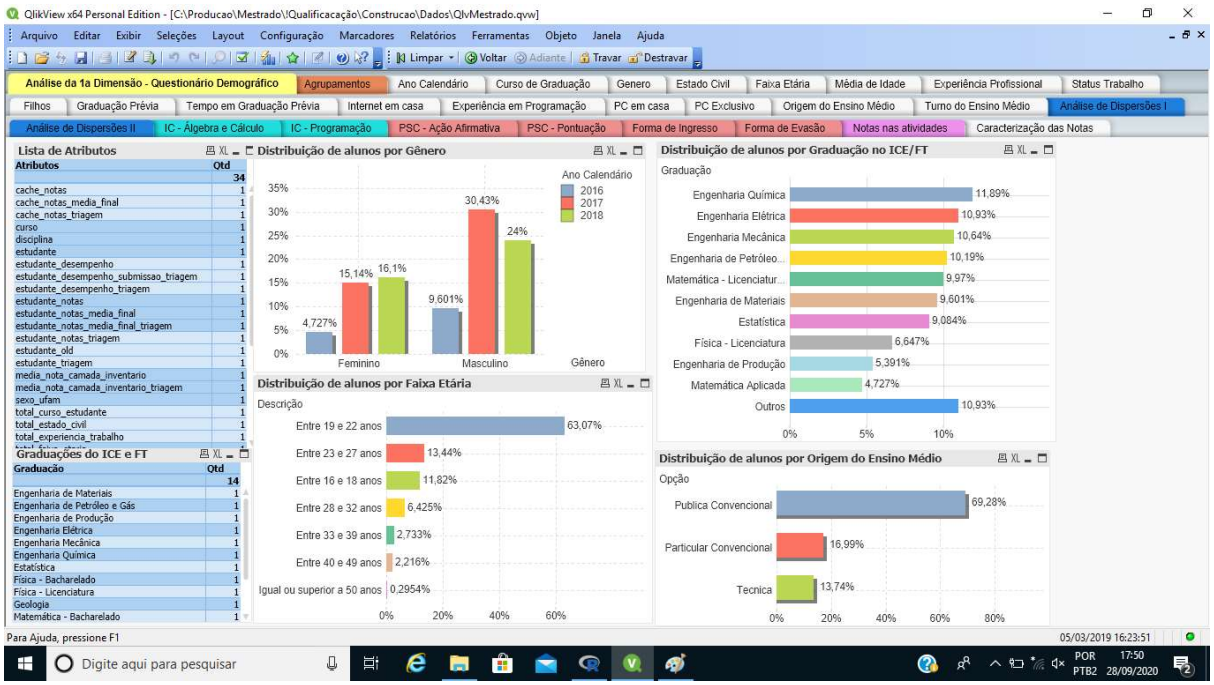
O objetivo proposto para esse experimento esteve focado na geração de um modelo para previsão de uma variável categórica e binária, o atributo "status_estudante", podendo gerar registros como evadido ou não-evadido. Para tanto, algoritmos de classificação foram aplicados para construção dos modelos de predição. Além disso, visando a generalização do modelo a ser obtido, classificadores

[5] https://www.heidisql.com

foram treinados utilizando a base de dados coletada particionada em 66% para tal objetivo.

Para a realização da geração do modelo foram utilizadas ferramentas de coleta, tratamento e análise de dados, de geração de modelos com algoritmos de aprendizagem de máquina. O pacote de software *Weka*[6] foi utilizado na geração dos modelos preditivos. Utilizou-se o *QlikView*[7] para tratamento e apresentação de dados. A Figura 4.2 apresenta uma imagem do ambiente de desenvolvimento (IDE) da ferramenta *QlikView*.

Figura 4.2 - Ambiente da ferramenta Qlikview (data analytics).



Fonte: Elaborada pelo autor, 2021.

[6] https://www.cs.waikato.ac.nz/ml/weka/
[7] https://www.qlik.com/pt-br

No quadro 4.3, é possível observar todas as ferramentas aplicadas.

Quadro 4.3 - Lista de ferramentas de tratamento de dados utilizadas durante a pesquisa.

Ferramentas e Recursos	**Objetivo**	**Referências**
Mysql	Scripts e SGBD	https://www.mysql.com
Orange	Mineração de dados e aprendizagem de máquina	https://orangedatamining.com/
Python	Scripts, Processamento de dados e aprendizagem de máquina	https://www.python.org
Weka	Seleção de atributos	https://www.cs.waikato.ac.nz/ml/weka/
R	Análise de dados	https://www.r-project.org
QlikView	Visualização de dados e Interpretação de dados	https://www.qlik.com/pt-br

Fonte: Elaborado pelo autor, 2021.

Foram realizados experimentos com dados de ex-estudantes *non-majors* da disciplina de CS1 da UFAM. Salienta-se que foram contempladas todas as etapas da abordagem, desde o levantamento das variáveis a partir de questões exploradas no questionário sociodemográfico até a validação dos modelos de MD empregados.

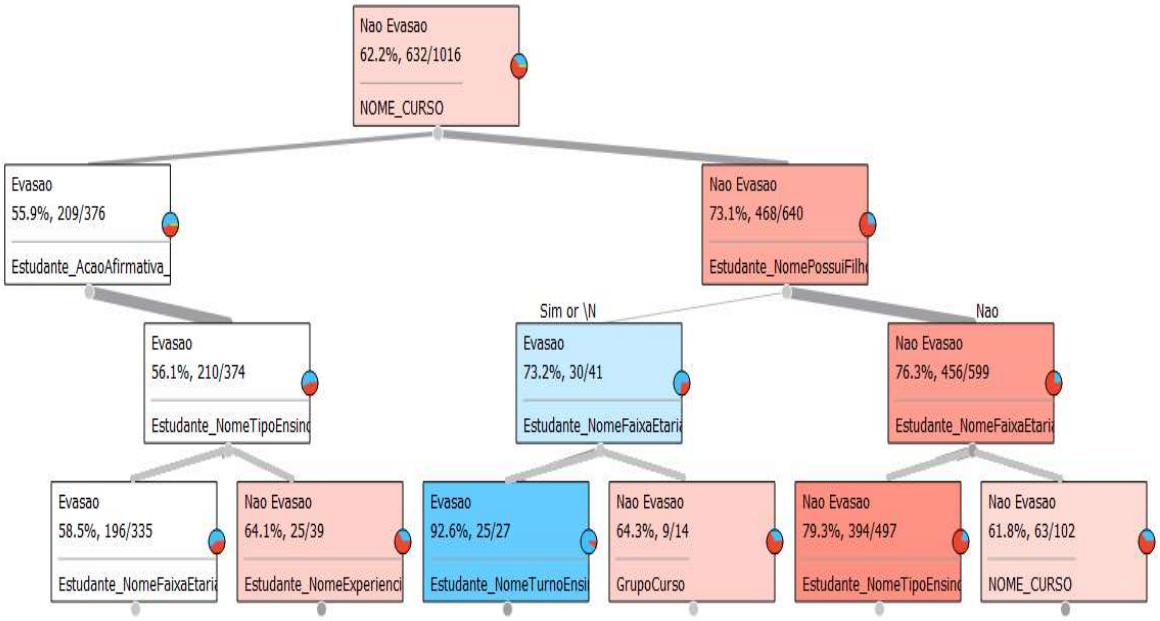
A previsão foi tratada como um problema de classificação binária (RAMOS *et al.*, 2018), no qual, dadas as informações do histórico de um estudante, o classificador tenta identificar se esse estudante irá ou não se evadir da disciplina. Os resultados esperados para este estudo são: o estudante conseguiu concluir a disciplina (independentemente de ser aprovado ou não) ou o estudante evadiu-se da disciplina.

O escopo do estudo compreendeu dados históricos sobre estudantes de turmas realizadas entre 2017 e 2018, impossibilitando adaptações no sistema de captura dos dados a serem utilizados. Fez-se necessária a utilização apenas dos dados que o ambiente dispunha, assim como uma pré-seleção das fontes de dados mais promissoras a serem exploradas nos passos seguintes.

Técnicas de mineração de dados educacionais foram aplicadas em informações provenientes de duas bases de dados educacionais: histórico de avaliações com resultado final de um sistema de juiz *online* e registros de discentes no sistema de controle acadêmico. Para tanto, foi necessário um estudo do ambiente do *Codebench* e como este armazena os dados mantidos pelo mesmo.

O algoritmo *Adaboost* (JAUHARI; SUPIANTO, 2019) foi usado como base para geração de uma árvore de decisão (*decision tree*) cujo resultado pode ser observado na Figura , onde os nós em laranja representam a estimativa de conclusão e os nós em azul e branco a de evasão. O objetivo desta tarefa foi dispor de uma análise mais aprofundada sobre quais atributos sociodemográficos de estudantes possuíam vínculos mais relevantes com evasão em CS1. A diferença nos tons de cores é devido à divisão do nó pai. Quanto mais escura a cor, maior o ganho de informação na estimativa (menor a entropia).

Figura 4.3 - Árvore de decisão do modelo baseado no algoritmo Adaboost.



Fonte: Elaborada pelo autor, 2021.

O quantitativo de atributos utilizados durante a pesquisa (quadro 4.4) é compatível com os valores encontrados em outros estudos nacionais e internacionais sobre evasão catalogados, conforme pode ser verificado no capítulo de trabalhos relacionados.

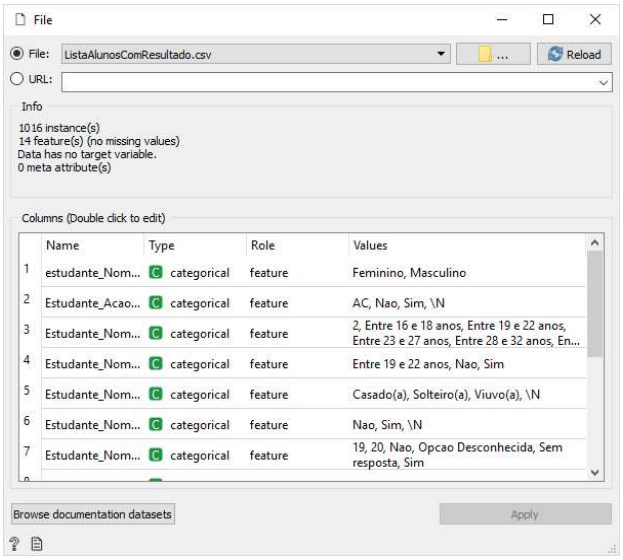
Quadro 4.4 - Síntese dos atributos selecionados

ITEM	ATRIBUTO	VALOR
1	A01_sexo	Masculino ou feminino
2	A02_estado_civil	Solteiro, casado, divorciado e viúvo
3	A03_experiencia_programacao	Experiência em alguma linguagem de programação (sim ou não)
4	A04_faixa_etaria	16 a 18 anos – 19 a 22 anos – 23 a 27 anos – 28 a 32 anos – 33 a 39 anos – 40 a 49 anos – acima de 50 anos
5	A05_filhos	Possuir ou não filhos
6	A06_origem_ensino_medio	Público convencional – privado convencional – médio técnico
7	A07_vaga_acao_afirmativa	Ampla concorrência - Cota Independente de renda - Cota com renda baixa – Outros
8	A08_experiencia_trabalho	Possui experiência laboral (sim ou não)
9	A09_curso	Um dos 11 cursos analisados de ciências exatas e engenharia
10	A10_grupo_curso	Ciências Exatas ou Engenharias
11	A11_acesso_internet	Possui acesso à internet (sim ou não)
12	A12_pc_casa	Computador em casa (sim ou não)
13	A13_turno_ensino_medio	Matutino - Vespertino - Noturno – Integral

Fonte: Elaborado pelo autor, 2021.

A seguir é apresentada a lista dos atributos preditores na ferramenta de aprendizagem *Orange* (figura 4.4) de máquina para compor o modelo e vinculados ao quadro 5.

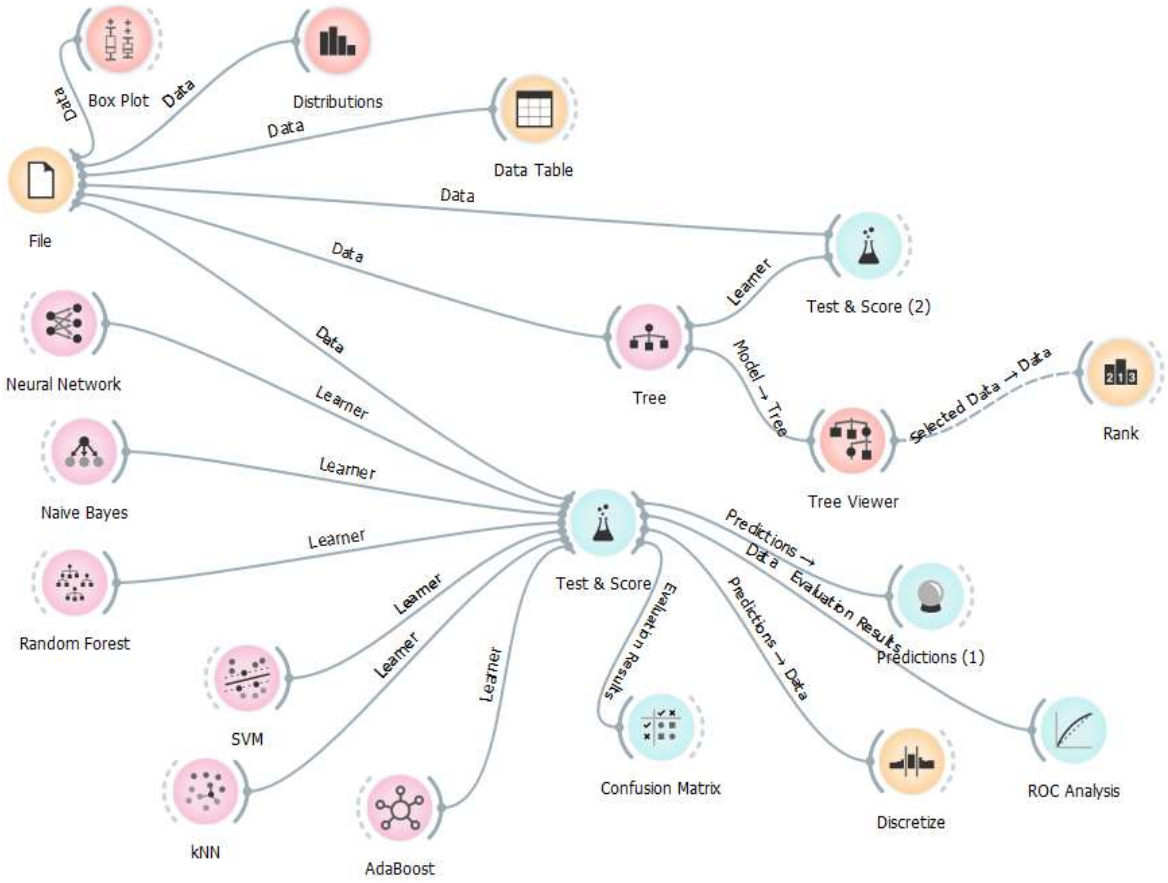
Figura 4.4 - Seleção dos atributos preditores no Orange.



Fonte: Elaborada pelo autor, 2021.

A figura 4.5 apresenta toda a estrutura construída para geração dos modelos preditivos de evasão de *non-majors* no Orange.

Figura 4.5 - Estrutura para geração dos modelos preditivos no Orange.



Fonte: Elaborada pelo autor, 2021.

Modelos gerados com os sete classificadores baseados em algoritmos de aprendizagem de máquina supervisionada citados nesta pesquisa foram avaliados a fim de identificar aquele com melhor efetividade preditiva. Com uso da métrica área sobre curva ROC (*Receiver Operating Characteristics* ou Característica de Operação do Receptor), permitiu-se demonstrar o quão bom cada modelo criado pode distinguir entre a condição binária buscada (evadido e não evadido).

A construção do gráfico de curva ROC é baseada na taxa de verdadeiros positivos e na taxa de falsos positivos, onde a construção do gráfico é feita por meio da plotagem dos falsos positivos no eixo das ordenadas (*eixo x*) e verdadeiros positivos no eixo das abscissas (*eixo y*).

A Curva ROC possui dois parâmetros:

- Taxa de verdadeiro positivo *(True Positive Rate)*, que é dado por *true*

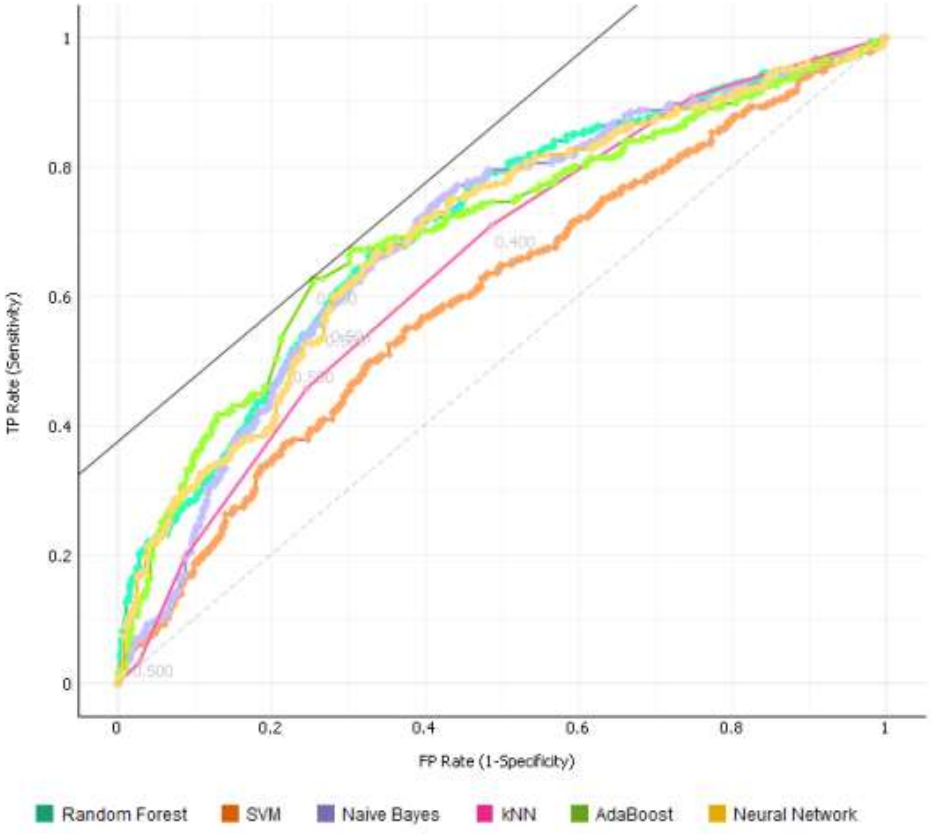
positives / (true positives + false negatives)

- Taxa de falso positivo *(False Positive Rate)*, que é dado por *false positives / (false positives + true negatives)*

Uma curva ROC traça "*True Positive Rate vs. False Positive Rate*" em diferentes limiares de classificação (veja figura abaixo).

A Figura 4.6 exibe a área sob a curva ROC utilizada para gerar uma representação gráfica do desempenho dos modelos de classificação gerados neste trabalho, evidenciando a curva do melhor classificador (Adaboost) na cor verde e do pior (SVM) na cor laranja.

Figura 4.6 - Curva Receiver Operating Characteristics (ROC) para comparação dos classificadores.



Fonte: Elaborada pelo autor, 2021.

A Curva ROC permitiu evidenciar os valores para os quais exista maior otimização da sensibilidade em função da especificidade que corresponde ao ponto em que se encontra mais próxima do canto superior esquerdo do diagrama, uma vez que o índice de positivos verdadeiros é "um" e o de falsos positivos é "zero".

O modelo alcançou 72% de efetividade no processo preditivo de evasão dos estudantes em CS1 com uso do classificador *Random Forest* no conjunto de testes, sendo que nenhum outro obteve resultados superiores ao primeiro para este conjunto de dados, conforme detalhado no Quadro .

Quadro 4.5 - Resultado das avaliações comparativas dentre os modelos construídos.

MÉTODO	CURVA ROC
Random Forest	0.720
AdaBoost	0.701
Neural Network	0.701
Naive Bayes	0.703
KNN	0.653
SVM	0.622

Fonte: Elaborado pelo autor, 2021.

4.4 TESTES E AVALIAÇÕES

Antes de proceder à aplicação experimental do modelo construído, é importante avaliá-lo e rever a sua construção para ter certeza que este atinge adequadamente os objetivos planejados (RAMOS *et al.,* 2020).

Os testes e avaliações realizados tiveram o objetivo de garantir que a instituição de ensino pudesse eventualmente utilizar os resultados obtidos e os conhecimentos descobertos. Outro objetivo era o de determinar se havia algum problema importante sobre a massa de dado que não tivesse sido suficientemente considerado (RAMOS *et al.,* 2020).

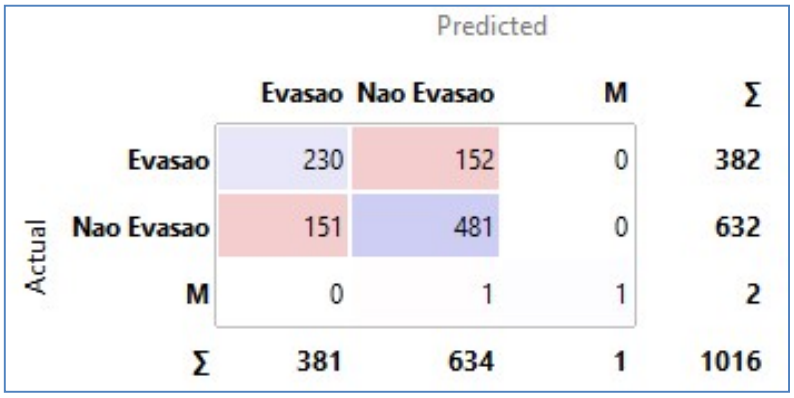
Além da métrica da área sobre a curva ROC, a avaliação da acurácia e da precisão foram utilizadas para avaliar a precisão preditiva dos modelos gerados. Para entender os erros gerados pelo classificador, foi feita a construção da matriz de erros denominada matriz de confusão (*Confusion Matrix*).

Matriz de Confusão é uma ferramenta muito usada para avaliações de modelos de classificação em Aprendizado de Máquina. É uma tabela 2x2 que mostram as frequências de classificação para cada classe do modelo. A partir desta matriz foi possível obter métricas de qualidade para a avaliação do desempenho dos algoritmos de classificação.

O funcionamento da Matriz de confusão é simples: consideram-se os valores positivos que o modelo julgou positivos (verdadeiros positivos), valores positivos que o modelo julgou negativos (falsos negativos), valores negativos que o sistema julgou como negativos (verdadeiros negativos) e valores negativos que o sistema julgou positivos (falsos positivos).

A matriz de confusão gerada representou os níveis de precisão preditiva atingidos pelo modelo preditivo baseado no algoritmo *AdaBoost* (Jauhari; Supianto, 2019), disposta na Figura , observa-se que esse modelo classificou corretamente 711 (70.1%) instâncias e incorretamente 303 (29.9 %), enquanto que o classificador *Random Forest* atingiu apenas 67,9% nesse critério.

Figura 4.7 - Matriz de confusão do modelo gerado.



Fonte: Elaborada pelo autor, 2021.

Nesse trabalho, a condição binária está considerada da seguinte forma: evasão é um caso positivo procurado e não evasão, por sua vez, é um caso negativo. No Quadro 4.6, são apresentadas algumas medidas derivadas da Matriz de Confusão do modelo gerado com o algoritmo *AdaBoost*.

Quadro 4.6 - Medidas derivadas da Matriz de Confusão (Modelo AdaBoost).

Medida	Entendimento	Proporção (A / B)	Valor
Acurácia	Proporção de predições corretas: sem levar em consideração o que é positivo e o que é negativo.	(total de acertos / total de dados no conjunto)	70,12%
Sensibilidade	Proporção de verdadeiros positivos: a capacidade do sistema em predizer corretamente a condição para casos que realmente a têm.	(acertos positivos / total de positivos)	60,21%
Especificidade	Proporção de verdadeiros negativos: a capacidade do sistema em predizer corretamente a ausência da condição para casos que realmente não a têm.	(acertos negativos / total de negativos)	76,11%
Eficiência	Média aritmética da Sensibilidade e Especificidade. Quando um modelo é muito sensível à positivos, tende a gerar muitos falso-positivos, e vice-versa. Assim, um método de decisão perfeito (100 % de sensibilidade e 100% especificidade) raramente é alcançado, e um balanço entre ambos deve ser atingido.	(sensibilidade + especificidade) / 2	68,16%
Preditividade Positiva	Proporção de verdadeiros positivos em relação a todas as predições positivas.	(acertos positivos / total de predições positivas)	60,21%
Preditividade Negativa	Proporção de verdadeiros negativos em relação a todas as predições negativas.	(acertos negativos / total de predições negativas)	76,11%

Fonte: Elaborado pelo autor, 2021.

Para a métrica precisão obteve-se um valor de 0.701 (70,1%) com o *AdaBoost*, enquanto que o classificador *Random Forest* atingiu apenas 67,6%. Essa métrica determina o percentual de registros que são positivos no grupo que o classificador previu como classe positiva. Quanto maior o percentual de precisão, menor será o número de erros falsos positivos pelo classificador, ou seja, quanto maior a precisão menor será a quantidade de alunos que estão em situação de risco, mas que o classificador errou e os classificou como alunos em uma situação de desempenho satisfatório.

Os resultados indicam que o modelo preditivo de evasão dos estudantes com uso do classificador *AdaBoost* representa aquele com melhor resultado dentre todos os avaliados, conforme detalhado no quadro 4.7.

Quadro 4.7 - Resultado das avaliações comparativas dentre os modelos construídos.

MÉTODO	ACURÁCIA	PRECISÃO	CURVA ROC
AdaBoost	0.701	0.701	0.701
Random Forest	0.679	0.676	0.720
Neural Network	0.669	0.679	0.701
Naive Bayes	0.663	0.663	0.703
KNN	0.640	0.630	0.653
SVM	0.608	0.080	0.622

Fonte: Elaborado pelo autor, 2021.

4.5 APLICAÇÃO EXPERIMENTAL E RESULTADOS

Para validar o modelo construído foi realizado um conjunto de experimentos sobre uma base de dados constituída por todo o conjunto de disciplinas ofertadas aos cursos descritos na Seção 4.1.

Um conjunto comparativo baseado em características extraídas de bases de dados educacionais da literatura também foi compilado. O perfil caracterizado é formado pelas variáveis sociodemográficas listada no quadro 4.8 e que compõem uma matriz de características vinculadas à decisão de evasão de estudantes em CS1 e que serão comparadas com o conjunto que compõe o perfil de evasão gerado neste trabalho.

Quadro 4.8 - Perfil sociodemográfico vinculado à evasão estudantil

VARIÁVEL SOCIODEMOGRÁFICA	DESCRIÇÃO	REFERÊNCIA
Necessidade de trabalhar	Estudante já inserido no mercado de trabalho com necessidade de conciliar suas obrigações profissionais com estudo	Giraffa; Mora, 2015; Petersen *et al.*, 2016
Conhecimento prévio	Existência de conhecimento prévio em conceitos básicos ou em linguagens de programação de computadores	Giraffa; Mora, 2015; Petersen *et al.,* 2016; Balmes, 2017
Opção inicial do curso	Se a graduação na qual o estudante está inscrito foi a primeira escolha do mesmo	Casanova *et al*., 2018
Falta de tempo	Falta de tempo aplicado ao estudo ou por obrigações no trabalho, ou afazeres domésticos ou qualquer outro motivo	Giraffa; Mora, 2015

Fonte: Elaborado pelo autor, 2021.

Com base no conjunto de variáveis sociodemográficas descrito anteriormente e encontrados na literatura, inferiu-se dois fatores fundamentais que influenciam a

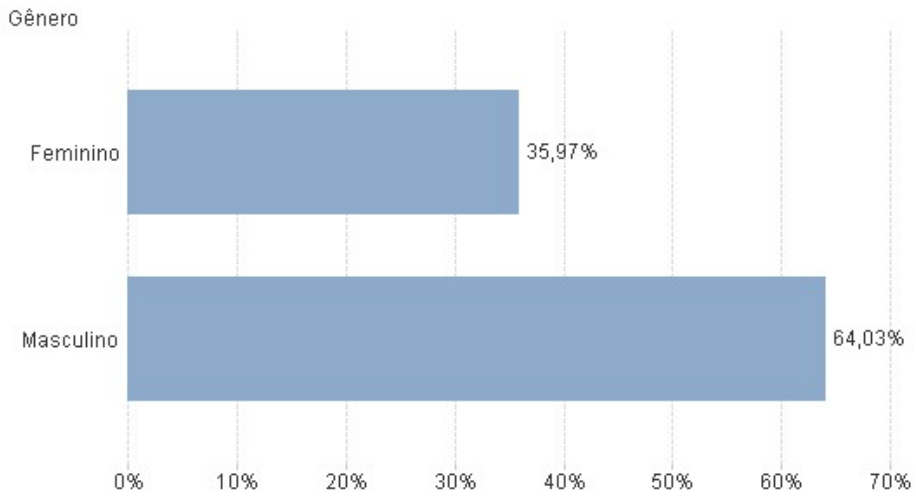
decisão de evasão do estudante em CS1. O primeiro fator vinculante é nitidamente o tempo que o estudante dispõe para dedicar a suas obrigações na disciplina, quer seja por necessidade de trabalho quer seja por afazeres domésticos.

O segundo fator que se sobressaiu nesta caracterização está vinculado à origem ou base educacional deste estudante, que envolve tanto os conhecimentos acumulados previamente (raciocínio lógico, programação, etc.) a sua participação como discente em CS1 quanto a sua base do ensino médio ter lhe proporcionado (ou não) a condição de cursar a sua primeira escolha para graduação, podendo isto está incorrendo em problemas motivacionais ou até mesmo ausência de uma estrutura cognitiva[8] básica que lhe cause dificuldades no aprendizado da programação (o que não faz parte do escopo deste estudo).

4.6 INTERPRETAÇÃO DOS PADRÕES SOCIODEMOGRÁFICOS DE EVASÃO

As características sociodemográficas dos estudantes foram interpretadas quantitativamente a fim de ser identificados padrões sociodemográficos de estudantes *non-majors* que possam levar à evasão em CS1. Nos dados analisados das turmas de 2016 a 2018, do total de estudantes, 64,9% eram homens e 35,1% eram mulheres (Figura)

Figura 4.8 - Distribuição de estudantes *non-majors* em CS1 por gênero.

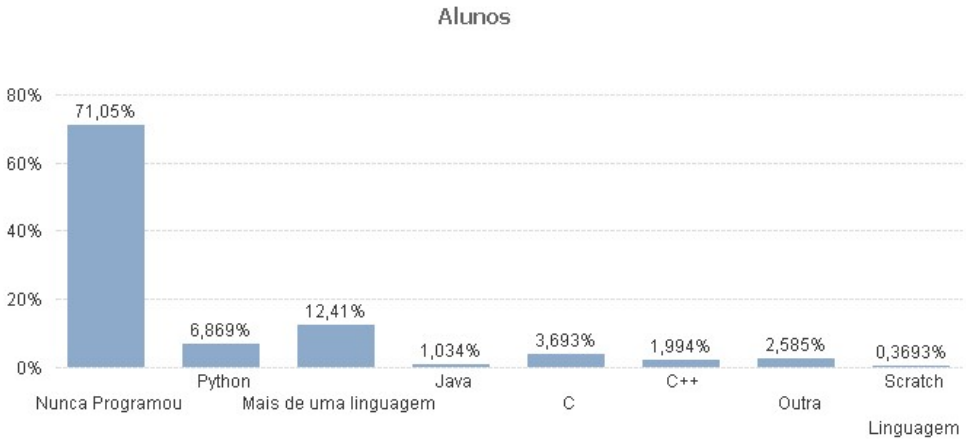


Fonte: Elaborado pelo autor, 2021.

[8] A estrutura cognitiva é o conteúdo total e organizado de idéias de um dado indivíduo; ou, no contexto da aprendizagem de certos assuntos, refere-se ao conteúdo e organização de suas idéias naquela área particular de conhecimento.

Dentre esses estudantes, 71,05% não possuíam experiência prévia em programação (Figura) e 69,2% era originário do ensino médio público convencional (Figura). Esse grupo obteve um índice de evasão de 37,3%.

Figura 4.9 - Distribuição de estudantes *non-majors* em CS1 por conhecimento prévio.



Fonte: Elaborada pelo autor, 2021.

A média geral de evasão de *non-majors* em CS1, para o período avaliado, e conforme conceitos e metodologia aplicada na massa pesquisada foi de 40,5%, sendo que o curso com maior evasão foi o de Licenciatura em Matemática Matutino com 61,8% de evasão, seguido de Bacharelado em Matemática Diurno (licenciatura) e Licenciatura em Matemática Noturno. O curso de Engenharia de Produção apresentou o menor número de evasão entre todos, conforme pode ser verificado no quadro 4.9.

Quadro 4.9 - Índice médio geral de evasão em CS1 no período.

Curso de Graduação	**% Evasões**	**% Aprovações**	**% Reprovações**
Matemática Licenciatura Matutino	61,8	31,9	6,30
Matemática Bacharelado Diurno	58,8	34,9	6,30
Matemática Licenciatura Noturno	55,4	43,4	1,20
Física Licenciatura Noturno	54,2	35,8	10,00
Física Bacharelado Noturno	52,8	25,4	21,80
Engenharia de Materiais	34,0	44,0	22,00
Estatística	33,3	60,2	6,50
Engenharia Química	27,9	62,8	9,30
Engenharia de Petróleo e Gás	22,8	67,6	9,60
Engenharia Mecânica	22,8	68,4	8,80
Engenharia de Produção	22,2	70,4	7,4

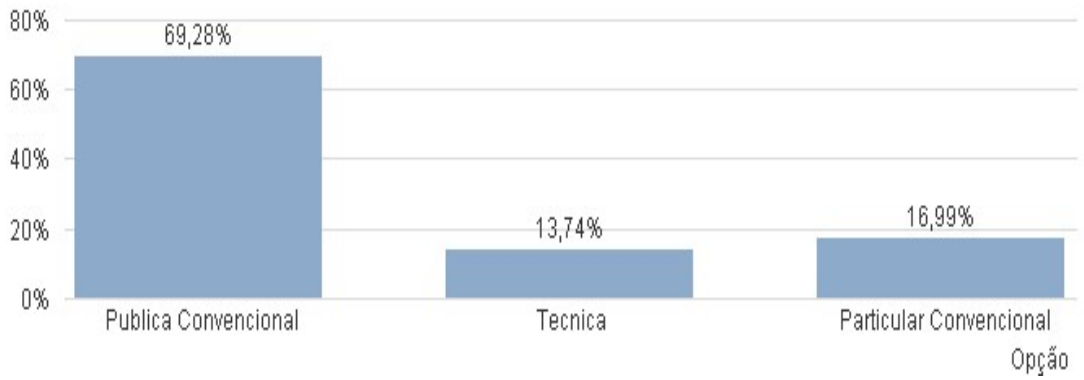
Fonte: Elaborado pelo autor, 2021.

Do total de mulheres, 36,1% evadiram-se da disciplina antes do fim do período letivo, número muito próximo ao encontrado entre os homens. Ao combinar os mesmos atributos utilizados anteriormente para os homens, os números de evasão permaneceram próximos: 38,5%.

A combinação dos atributos A06_origem_ensino_medio (técnico) com o sexo feminino ou masculino e a A03_experiencia_programacao ou não trouxe ganhos significativos ao modelo, sendo que nesses casos os estudantes provenientes de escolas técnicas (públicas ou privadas) obtiveram ocorrências de desistência extremamente minimizadas. Além disso, a ausência de filhos para estudantes do sexo feminino e masculino interferiu significativamente os índices de evasão, elevando os percentuais para acima de 57% em ambos os casos.

Nesses mesmos grupos de evadidos (homens ou mulheres com filhos) que 8,6% dos estudantes possuíam origem do ensino médio técnico. Do total de estudantes evadidos neste conjunto, 81,5% advinha de escolas públicas convencionais e 9,9% de escolas particulares.

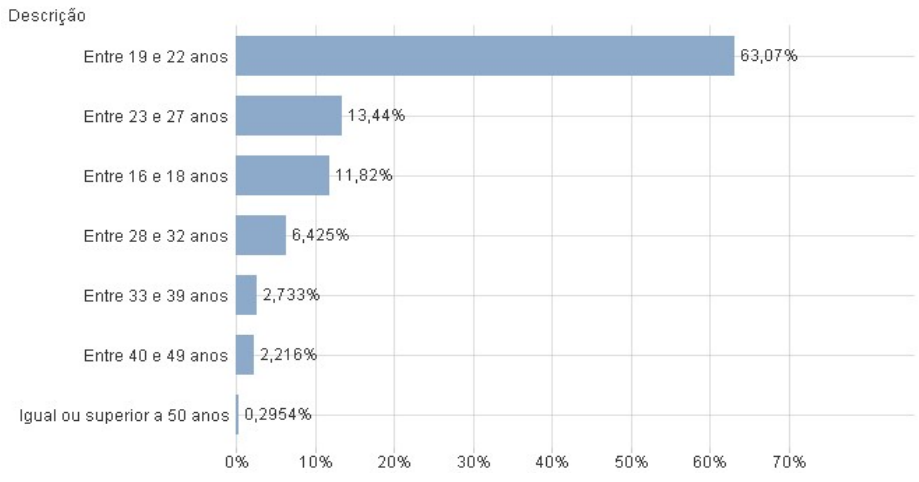
Figura 4.10 - Distribuição de estudantes *non-majors* em CS1 por origem do nível médio.



Fonte: Elaborado pelo autor, 2021.

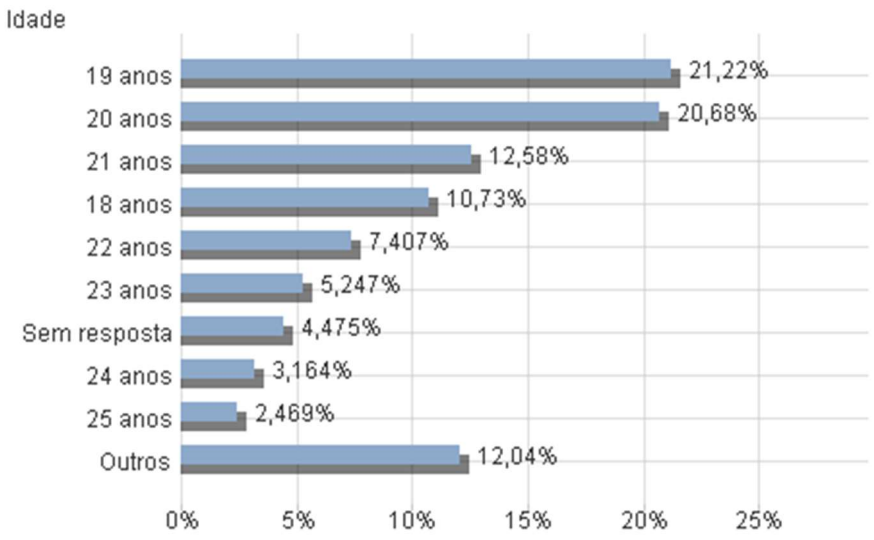
Foi verificado que os estudantes do sexo masculino de maior idade (figuras 4.11 e 4.12) têm maior probabilidade de desistir do que os homens mais jovens, enquanto a idade não é fator importante para evasão das mulheres. Além da idade dos estudantes, verificou-se que os estudantes non-majors que ingressam na universidade imediatamente após o ensino médio têm uma menor probabilidade de desistência em CS1.

Figura 4.11 - Distribuição de estudantes *non-majors* em CS1 por faixa etária.



Fonte: Elaborado pelo autor, 2021.

Figura 4.12 - Distribuição de estudantes *non-majors* em CS1 por idade.

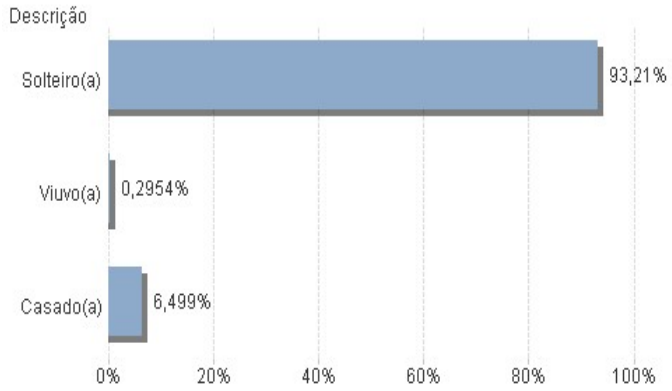


Fonte: Elaborada pelo autor, 2021.

O estado civil (figura 4.13) influenciou a ocorrência de evasão, mas há algumas diferenças entre estudantes do sexo masculino e feminino aqui. Por um lado, os homens casados têm maior probabilidade de interromper os estudos por um curto

período do que os homens que não são casados, mas a probabilidade de desistir é menor do que parar por um curto período.

Figura 4.13 - Distribuição de estudantes *non-majors* em CS1 por estado civil.



Fonte: Elaborada pelo autor, 2021.

Por outro lado, as informações mostram que as mulheres casadas também são mais propensas a parar por um curto período, mas não são significativamente mais propensas a parar do que a desistir. Mulheres com crianças pequenas têm mais probabilidade de desistir do que parar e homens com crianças pequenas têm menor probabilidade de desistir.

A verificação numérica aqui realizada buscou trazer uma interpretação quantitativa sobre a predição da evasão de estudantes *non-majors* a partir de dados sociodemográficos na base de dados educacionais avaliada, demonstrando as nuances censitárias por traz do modelo preditivo construído.

Quadro 4.11 - Perfil sociodemográfico vinculado à evasão estudantil obtido.

Variável sociodemográfica	**Descrição**
A04_faixa_etaria	Faixa etária a qual pertence o estudante
A05_filhos	Sim ou não
A06_origem_ensino_medio	Categoria (pública, privada, etc.) a qual pertence à escola de nível médio do estudante
A09_curso	Nome do curso de ciências exatas e de engenharias do estudante
A13_turno_ensino_medio	Matutino, vespertino, noturno ou integral

Fonte: Elaborado pelo autor, 2021.

Assim como feito com base na revisão da literatura, também foi realizado um perfil sociodemográfico de estudante atrelado a um risco maior de evasão em CS1,

desta vez baseado nas 13 variáveis disponíveis no *dataset* utilizado neste trabalho. E assim como no primeiro perfil sociodemográfico, as variáveis de maior correlação à evasão (quadro 16), ainda que distintas daquelas do perfil encontrado na literatura, também estão vinculadas às dimensões tempo e base educacional prévia dentre as características sociodemográficas.

5 CONCLUSÃO

Neste trabalho, um modelo preditivo de evasão de estudantes *non-majors* de turmas de CS1 foi construído com base em um método de mineração de dados educacionais sociodemográficos (CRISP-EDM/SD). Um achado importante dessa pesquisa está no conjunto de evidências baseadas nos dados destes estudantes vinculados ao relacionamento dos mesmos com o meio social que o cercam, o qual foi rotulado como perfil sociodemográfico de evasão estudantil.

A pesquisa conseguiu entregar um modelo facilmente adaptável, propício à generalização e à aplicação em outros contextos educacionais, inclusive em outras instituições. O modelo de evasão de estudantes *non-majors* em CS1 alcançou 70,1% de acurácia e precisão com uso algoritmo *AdaBoost* utilizando uma base de dados educacionais da UFAM. A obtenção a priori do risco de evasão foi obtido a partir da aplicação do modelo na primeira semana de aula. O modelo tende a ficar cada vez mais acurado ao longo do semestre letivo com a redução de avaliações restantes.

A abordagem foi validada por meio da construção de um modelo de predição de evasão e da aplicação deste sobre uma base de dados educacionais com dados sociodemográficos de estudantes dos cursos da Faculdade de Tecnologia e do Instituto de Ciências Exatas da UFAM. Os resultados mostram a viabilidade do modelo haja vista a acurácia atingida para turmas iniciais de programação na primeira semana de aula.

O modelo viabilizou a identificação de um perfil sociodemográfico de evasão estudantil aplicáveis a outros modelos preditivos. Os resultados obtidos mostram que modelos construídos a partir deste perfil se comportam de forma eficiente na tarefa da predição pretendida.

Este trabalho teve como resultado, ainda, uma análise quantitativa gerada a partir da avaliação dos dados processados tendo em vista que todo o conhecimento adquirido deve ser organizado e apresentado de uma forma que a instituição possa usá-lo efetivamente dentro dos processos de tomada de decisão.

5.1 CONTRIBUIÇÕES

Como contribuição desse estudo, tem-se a propositura e validação de um modelo preditivo de evasão de estudantes *non-majors* com uso de algoritmos de classificação binária e técnicas de descoberta de conhecimento em uma base de dados sociodemográfica, capaz de apoiar a tomada de decisões educacionais estratégicas frente à problemática de evasão de tais estudantes.

Os resultados apresentados nesta dissertação estão presentes parcialmente em dois artigos apresentados em congressos nacionais de informática na educação.

O primeiro artigo intitulado “Analisando a influência de atributos demográficos no desempenho de estudantes em disciplinas introdutórias de programação de computadores” (PEREIRA; CARVALHO; SOUTO, 2019) foi selecionado e apresentado nos Anais do XXVI Workshop sobre Educação em Computação (WEI, 2019).

Já o segundo artigo construído no decorrer da elaboração dos trabalhos que deram fruto a esta dissertação é intitulado “Predição de evasão de estudantes non-majors em disciplina de introdução à programação” (PEREIRA; CARVALHO; SOUTO, 2019a) foi selecionado e apresentado nos Anais do Workshop de Ciência de Dados Educacionais (WCDE) do Congresso Brasileiro de Informática na Educação (CBIE).

5.2 LIMITAÇÕES

As maiores limitações do presente trabalho estão relacionadas à base de dados utilizada. Primeiramente, em termos de validação externa, foram utilizados dados de 1.016 estudantes do 1º e 2º semestres do ano de 2017 e do 1º semestre de 2018 de cursos de ciências exatas e de engenharia da UFAM. Assim, é importante que o presente método seja replicado com outras bases de dados educacionais de outras instituições de ensino, a fim de validar a robustez do método.

Com a viabilidade da aplicação do modelo logo no início de cada período letivo, o mesmo pode contribuir para condução de iniciativas institucionais e pedagógicas mais eficientes de combate à evasão. Espera-se que esta dissertação motive outros estudantes da UFAM a estudarem e pesquisarem sobre o assunto com a finalidade de realizarem mais contribuições para a previsão apresentada, para a EDM e consequentemente para o ensino da disciplina de CS1.

5.3 TRABALHOS FUTUROS

Considerando a comprovação de que casos de evasão de estudantes *non-majors* em CS1 estejam vinculados a aspectos sociodemográficos, como faixa etária e existência de filhos e origem do ensino médio, abre-se a possibilidade de aplicação do mesmo modelo de predição em dados de estudantes de outros cursos e disciplinas a fim de avaliar o desempenho do mesmo modelo nesse novo contexto.

Pretende-se, da mesma forma, replicar esse estudo em outras instituições de ensino, com um formato de avaliação diferente. Além disso, é importante aplicar o método apresentado e verificar o resultado da intervenção do professor à medida que se tem identificado o risco antecipado de evasão.

Por fim, o impacto gerado no mundo em decorrência da pandemia causada pelo SARS-COV-2, também conhecido como COVID-19[9], trouxe consigo uma série de consequências tais como isolamento e distanciamento social a fim de conter a contaminação do novo coronavírus, fato que culminou em inúmeros impactos transversais (SENHORAS, 2020), inclusive na condução do ensino-aprendizagem, os quais poderão ter seus efeitos explorados no campo socioeconômico dos estudantes *non-majors* de CS1 em novas turmas da disciplina nos próximos semestres.

5.4 CONSIDERAÇÕES FINAIS

No decurso do trabalho, a identificação dos alunos que apresentam risco de evasão através do uso técnicas de mineração de dados mostrou-se viável. A qualidade dos resultados iniciais abre a possibilidade de investigações futuras, como por exemplo, a modelagem de uma ferramenta de auxílio acadêmico que identifique quais alunos apresentam maior risco de abandonar os estudos de CS1.

É possível aumentar a acurácia dos modelos através do uso de bases de dados maiores capazes de serem aplicadas técnicas de aprendizagem profunda em substituição à abordagem de AM ora aplicada.

Por esses motivos, sustenta-se a intenção de aprofundamento desta pesquisa e desenvolvimento de um modelo preditivo de evasão em CS1, com base em atributos socioeconômicos, a fim de que, entre outras questões, possam ser abordadas

[9] COVID-19 (do inglês Coronavirus Disease 2019) é uma doença infecciosa causada pelo coronavírus da síndrome respiratória aguda grave 2 (SARS-CoV-2) e causador de uma grave pandemia no ano de 2020 e que perdura de maneira severa em 2021.

políticas institucionais de apoio a estudantes com risco de evasão da disciplina ou evasão acadêmica.

Espera-se que esta dissertação motive outros estudantes a pesquisarem sobre o assunto com a finalidade de realizarem mais contribuições para a previsão apresentada, para a EDM e posteriormente para o combate à evasão em CS1.

REFERÊNCIAS BIBLIOGRÁFICAS

ABBAD, G.; CARVALHO, R.; ZERBINI, T. Evasão em curso à distância via internet: explorando variáveis explicativas. *In*: Encontro da Anpad, 29., 2005, Campinas. Anais […] Campinas: ANPAD, 2005.

AHA, D. W.; KIBLER, D.; ALBERT, M. K. Instance-based learning algorithms. *Machine learning*, Springer, v. 6, n. 1, p. 37–66, 1991.

ALMEIDA, O. d. *Evasão em cursos a distância: validação de instrumento, fatores influenciadores e cronologia da desistência. 2007. 177 f*. Tese (Doutorado) — Dissertação (Mestrado em Administração) –Universidade de Brasília, Brasília-DF, 2007.

AMARAL, F. *Aprenda mineração de dados: teoria e prática*. Rio de Janeiro: Alta Books, 2016.

AMERI, S. *et al*. Survival analysis based framework for early prediction of student dropouts. In: *Proceedings of the 25., ACM International on Conference on Information and Knowledge Management*, 2016, p. 903–912.

BALMES, I. L. Correlation of mathematical ability and programming ability of the computer science students. *Asia Pacific Journal of Education, Arts and Sciences*, v. 4, n. 3, p. 85–88, 2017.

BARCZAK, A. L.; JOHNSON, M. J.; MESSOM, C. H. Empirical evaluation of a new structure for adaboost. In: *Proceedings of the 2008 ACM symposium on Applied computing*, 2008, p. 1764–1765.

BAZZOCCHI, R.; FLEMMING, M.; ZHANG, L. Analyzing cs1 student code using code embeddings. In: *Proceedings of the 51., ACM Technical Symposium on Computer Science Education*. 2020. p. 1293–1293.

BOSSE, Y.; GEROSA, M. A. Reprovações e trancamentos nas disciplinas de introdução à programação da universidade de são paulo: Um estudo preliminar. In: SBC. *Anais do XXIII Workshop sobre Educação em Computação*. 2015. p. 426–435.

BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, n. 1, p. 5–32, 2001.

BRITO, D. M. et al. Identificação de estudantes do primeiro semestre com risco de evasão através de técnicas de data mining. *Nuevas Ideas en Informática Educativa TISE*, p. 459–463, 2015.

CARVALHO, L. S. et al. Ensino de programação para futuros não-programadores: contextualizando os exercícios com as demais disciplinas de mesmo período letivo. In: SBC. *Anais do XXIV Workshop sobre Educação em Computação*. 2016. p. 121–130.

CASANOVA, J. R. et al. Factors that determine the persistence and dropout of university students. *Psicothema, 30*, 2018.

CASTRO, L. et al. Applying crisp-dm in a kdd process for the analysis of student attrition. In: SPRINGER. *Colombian Conference on Computing*. 2018. p. 386–401.

CASTRO, L. N. d.; FERRARI, D. G. *Introdução à mineração de dados: conceitos básicos, algoritmos e aplicações*. São Paulo: Saraiva, 2016.

CORBUCCI, P. R. *Evolução do acesso de jovens à educação superior no Brasil*. Brasília: IPEA, 2014.

COSTA, E. B. *et al.* Evaluating the effectiveness of educational data mining techniques for early prediction of students' academic failure in introductory programming courses. *Computers in Human Behavior*, Elsevier, v. 73, p. 247–256, 2017.

COUTINHO, E. F.; LIMA, E. T. de; SANTOS, C. C. Um panorama sobre o desempenho de uma disciplina inicial de programação em um curso de graduação. *Revista Tecnologias na Educação*, v. 19, n. 9, p. 1–15, 2017.

DIGIAMPIETRI, L. A.; NAKANO, F.; LAURETTO, M. de S. Mineração de dados para identificação de alunos com alto risco de evasão: Um estudo de caso. *Revista de Graduação USP*, v. 1, n. 1, p. 17–23, 2016.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. *AI magazine*, v. 17, n. 3, p. 37–37, 1996.

FERNANDES, K. C.; OLIVEIRA, A. B. D. Estudo da evasão de discentes do curso de graduação em ciência da computação utilizando educational data mining (edm). *Anais do Salão Internacional de Ensino, Pesquisa e Extensão*, v. 12, n. 2, 2020.

FRANCISCO, R. E.; JÚNIOR, C. P.; AMBRÓSIO, A. P. Juiz online no ensino de programação introdutória-uma revisão sistemática da literatura. *In*: Simpósio Brasileiro de Informática na Educação (SBIE), 27., 2016. Uberlândia. Anais [...]. Uberlândia: SBIE, 2016. p. 11.

FRITSCH, R. A problemática da evasão em cursos de graduação em uma universidade privada. *Reunião Nacional da ANPEd, 37. Ed.*, 2015.

GAIOSO, N. P. d. L. *O fenômeno da evasão escolar na educação superior no Brasil*. 20 p. Tese (Doutorado) — Dissertação (Mestrado em Educação) – Universidade Católica de Brasília, 2005.

GALVÃO, L.; FERNANDES, D.; GADELHA, B. Juiz online como ferramenta de apoio a uma metodologia de ensino híbrido em programação. *In*: Simpósio Brasileiro de Informática na Educação (SBIE), 27., 2016. Uberlândia. Anais [...]. Uberlândia: SBIE, 2016. p. 140.

GAMIE, E. A.; EL-SEOUD, M.; SALAMA, M. A. Comparative analysis for boosting classifiers in the context of higher education. *International Journal of Emerging Technologies in Learning*, v. 15, n. 10, 2020.

GIRAFFA, L.; MORA, M. d. C. Evasão na disciplina de algoritmo e programação: um estudo a partir dos fatores intervenientes na perspectiva do aluno. *In*: Congresso Latino-Americano sobre o Abandono da Educação Superior (CLABES), 3., 2015, Rio Grande do Sul. *Anais* [...]. Rio Grande do Sul, 2015.

GUL, S. *et al.* Teaching programming: A mind map based methodology to improve learning outcomes. *In*: IEEE, *2017, [S.l.].* Anais[...] *International Conference on Information and Communication Technologies (ICICT)*, 2017. p. 209–213.

HAN, M. J.; PEI, J. *Data mining: concepts and techniques: concepts andtechniques*. Amsterdam: Elsevier, 2011.

HAND, D. J.; MANNILA, H.; SMYTH, P. *Principles of data mining (adaptive computation and machine learning)*: MIT Press, 2001.

HOED, R. M.; LADEIRA, M.; LEITE, L. L. Influence of algorithmic abstraction and mathematical knowledge on rates of dropout from computing degree courses. *Journal of the Brazilian Computer Society*, SpringerOpen, v. 24, n. 1, p. 1–16, 2018.

JAUHARI, F.; SUPIANTO, A. A. Building student's performance decision tree classifier using boosting algorithm. *Indones. J. Electr. Eng. Comput. Sci*, v. 14, n. 3, p. 1298–1304, 2019.

JÚNIOR, J. G. d. O.; NORONHA, R. V.; KAESTNER, C. A. A. Método de seleção de atributos aplicados na previsão da evasão de cursos de graduação. *Revista de Informática Aplicada*, v. 13, n. 2, 2017.

KIRA, L. F. *A evasão no ensino superior: o caso do Curso de Pedagogia da Universidade Estadual de Maringá (1992-1996)*. Tese (Doutorado) — Dissertação de Mestrado] Universidade Metodista de Piracicaba—Pós-Graduação, 1998.

KORI, K. *et al*. First-year dropout in ict studies. *In*: Global Engineering Education Conference (EDUCON), 8, 2015, Estônia. Anais[...]. Estônia: IEEE, 2015. p. 437–445.

KUMAR, M.; SINGH, A.; HANDA, D. Literature survey on educational dropout prediction. *International Journal of Education and Management Engineering*, Modernm*Education and Computer Science Press*, v. 7, n. 2, p. 8, 2017.

LANTZ, B. *Machine learning with R*. Birmingham: Packt publishing, 2013.

MA, L. *et al*. Social influences and dropout risks related to college students' academic performance: Mathematical insights. *Int. J. Nonlinear Sci*, v. 27, n. 1, p. 31–42, 2019.

MAIA, M. d. C.; MEIRELLES, F. d. S. Evasão nos cursos à distância e sua relação com as tecnologias de informação e comunicação. In: Encontro da Associação Nacional de Pós-Graduação e Pesquisa em Administração, 29, 2005, Brasilia. Anais[...]. Brasilia: ANPAD, 2005.

OLIVEIRA, V. *et al.* Rendimento dos alunos de engenharia nas disciplinas do núcleo de conteúdos básicos da ufjf. *In*: *Congresso Brasileiro de Educação em Engenharia, 35,* 2007, Curitiba. Anais[...]. Curitiba: COBENGE, 2007.

ORESKI, D.; PIHIR, I.; KONECKI, M. Crisp-dm process model in educational setting. *Economic and Social Development: Book of Proceedings*, Varazdin Development and Entrepreneurship Agency (VADEA), p. 19–28, 2017.

PASSOS, A. A. *et al*. Perfil e desempenho acadêmico do estudante de engenharia em disciplinas do ciclo básico. *Revista de Ensino de Engenharia*, v. 36, n. 2, p. 16–26, 2017.

PEDREGOSA, F. *et al*. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, JMLR. org, v. 12, p. 2825–2830, 2011.

PEREIRA, A.; CARVALHO, L.; SOUTO, E. Analisando a influência de atributos demográficos no desempenho de estudantes em uma disciplina de introdução à programação. *In*: Sociedade Brasileira de Computação, 27, 2019, Belém. *Anais[...].* Workshop sobre Educação em Computação. Belém: WEI, 2019. p. 360–369.

PEREIRA, A. F. S.; CARVALHO, L. S. G. de; SOUTO, E. Predição de evasão de estudantes non-majors em disciplina de introdução à programação. *In*: Workshops do Congresso Brasileiro de Informática na Educação, 8, 2019. p. 178.

PEREIRA, F. D. *et al*. Early performance prediction for cs1 course students using a combination of machine learning and an evolutionary algorithm. In: International Conference on Advanced Learning Technologies - IEEE, 19, *2019. Anais [...]* Brasília: ICALT*,* 2019. v. 2161, p. 183–184.

PETERSEN, A. et al. Revisiting why students drop cs1. In: *Koli Calling International Conference on Computing Education Research, 16,* 2016. p. 71–80.

PROVOST, F.; FAWCETT, T. *Data Science for Business*. 2. ed. Rio de Janeiro: Alta Books, 2016.

QUILLE, K.; CULLIGAN, N.; BERGIN, S. Insights on gender differences in cs1: A multi-institutional, multi-variate study. *In*: ACM *Conference on Innovation and Technology in Computer Science Education*, 2017. p. 263–268.

QUINLAN, J. R. *C4.5: Programs for Machine Learning*. San Francisco: Kaufmann Publishers, 1993.

RAMOS, J. L. C. *et al*. Crisp-edm: uma proposta de adaptação do modelo crisp-dm para mineração de dados educacionais. *In*: SBC, 31, 2020. *Anais[...] Simpósio Brasileiro de Informática na Educação,* 2020. p. 1092–1101.

RAMOS, J. L. C. *et al.* Um estudo comparativo de classificadores na previsão da evasão de alunos em EaD. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. 2018. v. 29, n. 1, p. 1463.

RIBEIRO, R. B. *et al*. Gamificaçao de um sistema de juiz online para motivar alunos em disciplina de programaçao introdutória. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. 2018. v. 29, n. 1, p. 805.

RISH, I. *et al*. An empirical study of the naive bayes classifier. In: *IJCAI 2001 workshop on empirical methods in artificial intelligence*. 2001. v. 3, n. 22, p. 41–46.

RODRIGUEZ-MAYA, N. E. *et al*. Modeling students' dropout in mexican universities. *Research in Computing Science*, v. 139, p. 163–175, 2017.

ROMERO, C.; VENTURA, S. Educational data mining and learning analytics: An updated survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Wiley Online Library, v. 10, n. 3, p. e1355, 2020.

SAMMUT, C.; WEBB, G. I. *Encyclopedia of machine learning*. : Springer Science & Business Media, 2011.

SANTANA, B. L.; BITTENCOURT, R. A. Increasing motivation of cs1 non-majors through an approach contextualized by games and media. In: IEEE. *2018 IEEE Frontiers in Education Conference (FIE)*. 2018. p. 1–9.

SANTANA, B. L.; FIGUERÊDO, J. S. L.; BITTENCOURT, R. A. Motivaçao de estudantes non-majors em uma disciplina de programaçao. In: SBC. *Anais do XXV Workshop sobre Educação em Computação*. 2017.

SANTOS, C. A. M. dos et al. Cemtral: uma nova metodologia híbrida de ensino e aprendizagem. *Revista Brasileira de Aprendizagem Aberta e a Distância*, v. 18, n. 1, p. 18–18, 2019.

SENHORAS, E. M. Coronavírus e educação: análise dos impactos assimétricos. *Boletim de Conjuntura (BOCA)*, v. 2, n. 5, p. 128–136, 2020.

SHARMA, G. et al. Decision tree analysis on j48 algorithm for data mining. *Proceedings of international journal of advanced research in computer science and software engineering*, v. 3, n. 6, 2013.

SHEARER, C. The crisp-dm model: the new blueprint for data mining. *Journal of data warehousing*, THE DATA WAREHOUSE INSTITUTE, v. 5, n. 4, p. 13–22, 2000.

SHUKLA, A.; DHIR, S. Tools for data visualization in business intelligence: case study using the tool qlikview. In: *Information Systems Design and Intelligent Applications*. : Springer, 2016. p. 319–326.

SIGURDSON, N.; PETERSEN, A. A survey-based exploration of computer science student perspectives on mathematics. In: *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. 2019. p. 1032–1038.

SILVA, L. G. da *et al*. Dinâmicas de evasão na educação superior brasileira. *Examen: Política, Gestão E Avaliação Da Educação*, v. 2, n. 2, p. 100–127, 2018.

STADELHOFER, L. E.; GASPARINI, I. Ensino de algoritmos e lógica de programação para os diferentes cursos: Um mapeamento sistemático da literatura. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. 2018. v. 29, n. 1, p. 108.

UTARI, M.; WARSITO, B.; KUSUMANINGRUM, R. Implementation of data mining for drop-out prediction using random forest method. In: IEEE. *2020 8th International Conference on Information and Communication Technology (ICoICT)*. 2020. p. 1–5.

UTIYAMA, F.; BORBA, S. d. F. P. Uma ferramenta de apoio ao controle da evasão de alunos em cursos à distância via internet. In: *Congresso Brasileiro de Computação*. 2003. v. 3.

VIANA, G. A.; PORTELA, C. dos S. O uso de softwares educativos para introdução de lógica de programação no ensino de base e superior. *Informática na educação: teoria & prática*, v. 22, n. 1, 2019.

WANKHEDE, S. B. Analytical study of neural network techniques: Som, mlp and classifier-a survey. *IOSR Journal of Computer Engineering*, v. 16, n. 3, p. 86–92, 2014.

WASIK, S. *et al*. A survey on online judge systems and their applications. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 51, n. 1, p.1–34, 2018.

WITTEN, I. H. *et al*. Practical machine learning tools and techniques. *Morgan Kaufmann*, p. 578, 2011.

ZHANG, S.; GAO, M.-W.; HU, Q.-H. Research and application of adaboost based prediction of student's academic achievement. In: ATLANTIS PRESS. *3rd Annual International Conference on Education and Development (ICED 2018)*. 2018. p. 218–223.

ZHOU, W. *et al.* The framework of a new online judge system for programming education. In: *Proceedings of ACM Turing Celebration Conference-China*. 2018. p. 9–14.