INSTITUTO FEDERAL DE SANTA CATARINA

JÚLIO CÉSAR BELENKE DOS SANTOS

USANDO MINERAÇÃO DE DADOS PARA PREDIÇÃO DA EVASÃO ESCOLAR

Caçador - SC

02 de Fevereiro de 2021

JÚLIO CÉSAR BELENKE DOS SANTOS

USANDO MINERAÇÃO DE DADOS PARA PREDIÇÃO DA EVASÃO ESCOLAR

Monografia apresentada ao Curso de Sistemas de Informação do Câmpus Caçador do Instituto Federal de Santa Catarina para a obtenção do diploma de Bacharel em Sistemas de Informação.

Orientador: Prof. Cristiano M. Garcia

Coorientador: Prof. Samuel S. Feitosa

Caçador - SC

02 de Fevereiro de 2021

Santos, Júlio César Belenke dos
S237u Usando mineração de dados para predição da evasão escolar / Júlio César Belenke dos Santos ; orientador: Cristiano M. Garcia ; coorientador: Samuel S. Feitosa. -- 2021.
51 f.

Trabalho de Conclusão de Curso (Graduação)-Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, Caçador, 2021.
Inclui bibliografias.

1. Mineração de dados (Computação). 2. Machine Learning 3. Evasão universitária. 4. Evasão escolar. 5. Redes neurais (Computação). I. Garcia, Cristiano M. II. Feitosa, Samuel S. III. Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina – Curso de Sistemas de Informação. IV. Título.

CDD 006.33

Ficha catalográfica elaborada pela Bibliotecária
Renata Ivone Garcia – CRB-14/1417

JÚLIO CÉSAR BELENKE DOS SANTOS

USANDO MINERAÇÃO DE DADOS PARA PREDIÇÃO DA EVASÃO ESCOLAR

Este trabalho foi julgado adequado para obtenção do título de Bacharel em Sistemas de Informação, pelo Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, e aprovado na sua forma final pela comissão avaliadora abaixo indicada.

Caçador - SC, 02 de Fevereiro de 2021:

Prof. Cristiano M. Garcia, Me.
Orientador
Instituto Federal de Santa Catarina
Câmpus Caçador

Prof. Samuel S. Feitosa, Dr.
Coorientador
Instituto Federal de Santa Catarina
Câmpus Caçador

Prof. Paulo Roberto Córdova, Me.
Banca avaliadora
Instituto Federal de Santa Catarina
Câmpus Caçador

Prof. Vitor Sales Rosa, Dr.
Banca avaliadora
Instituto Federal de Santa Catarina
Câmpus São José

Este trabalho é dedicado aos meus pais e meus irmãos.

AGRADECIMENTOS

Em primeiro lugar a Deus, por ter permitido não desanimar durante toda a jornada da graduação e da realização deste trabalho.

Os agradecimentos principais são direcionados para Prof. Vitor, Prof. Paulo e Prof. Ademir. E aos meus orientadores Prof. Cristiano e Prof. Samuel.

Agradecimento ao Gilberto Coutinho da DTIC que auxiliou na obtenção dos dados para este trabalho.

Aos meus pais, irmãos e familiares que me apoiaram e me deram força nos momentos difíceis.

Agradecimento a todos os meus professores que ajudaram no processo de construção de conhecimento, criando assim a base necessária, tanto para os estudos, como para a vida.

Aos meus colegas, amigos e parceiros da graduação e do Instituto Federal de Santa Catarina com quem convivi durante os últimos quatro anos, pelo aprendizado e pela troca de experiências que me permitiram crescer e me tornar uma pessoa melhor.

A Jéssica, João, Christian, Eduardo e Bolinha que partiram desta vida.

A todos aqueles que contribuíram, de alguma forma, meus agradecimentos e gratidão. Obrigado!

"Combati o bom combate,
Acabei a carreira,
Guardei a fé..."
(Bíblia Sagrada, 2 Timóteo 4:7-8)

RESUMO

A Mineração de Dados educacionais é uma área que busca extrair o máximo de conhecimentos úteis que são gerados pela área da Educação. A evasão ainda é um dos desafios a ser combatido no ambiente de Ensino Superior, devido ao fato de ser um problema que possui características locais, de instituição para instituição, população, ambiente estudantil, entre outros, podendo variar bastante, o que requer estudos constantes. Este projeto desenvolveu uma ferramenta com o objetivo de prever a potencial evasão de estudantes por meio de algoritmos de *Machine Learning* no ambiente de Ensino Superior do Instituto Federal de Santa Catarina - IFSC, Câmpus Caçador. Para tal, foram utilizadas as técnicas de Árvore de Decisão e Redes Neurais, sendo que o primeiro apresentou um desempenho melhor, onde a acurácia alcançou 84%, correspondendo a 87% de precisão na detecção de evasão, enquanto o segundo chegou a 82% de acurácia com 78% de precisão. Este trabalho pode servir de base para trabalhos futuros tendo como referência as colunas ou o *dataset* desenvolvido pelo mesmo.

Palavras-chave: Mineração de Dados, Machine Learning, Evasão Escolar, Rede Neural, Árvore de Decisão.

ABSTRACT

Educational Data Mining is an area that seeks to extract the most useful knowledge that is generated by the Education area. Evasion is still one of the challenges to be tackled in the Higher Education Environment, due to the fact that it is a problem that has local characteristics from institution to institution, population, student environment, among others, and can vary a lot, which requires constant studies. This project developed a tool aiming at predicting a potential dropout of students through Machine Learning algorithms in the Higher Education Environment of the Federal Institute of Santa Catarina - IFSC, Campus Caçador. In order to achieve this, we used the Decision Tree and Neural Network technicques, where the first one presented a better performance, when the precision reached 84%, corresponding to 87% accuracy in detecting dropout, while the second reached 82% accuracy with 78% precision. This work can serve as a basis for future work with reference to the columns or the dataset developed by the same.

Keywords: Machine Learning, School Dropout, Neural Network, Decision Tree.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ABNT Associação Brasileira de Normas Técnicas

IFSC Instituto Federal de Santa Catarina

IES Instituição de Ensino Superior

EDM *Educational Data Mining*

KDD *Knowledge Discovery in Databases*

DM *Data Mining*

ML *Machine Learning*

INEP Instituto Nacional de Pesquisas Educacionais Anísio Teixeira

MEC Ministério da Educação

SESU Secretaria de Educação Superior

DTIC Diretoria de Tecnologia da Informação e Comunicação

SQL Linguagem de Consulta Estruturada

PROPPI Pró Reitoria de pesquisa, Pós graduação e Inovação

SIPAC Sistema Integrado de Patrimônio, Administração e Contratos

DEIA Diretoria de Estatística e Informações Acadêmicas

SGBD Sistema Gerenciador de Banco de Dados

NaN Not a Number

SUMÁRIO

1 INTRODUÇÃO

Com intenção de melhorar o futuro de uma nação, a educação é tida como um caminho que, quanto maior o investimento, maior o retorno social (NERI, 2009). A evasão é um dos grandes problemas que afetam negativamente a educação. O foco desta pesquisa envolve o conceito de evasão escolar, quais são as causas de evasão, quais fatores que mais influenciam, entre outros aspectos. Segundo o dicionário Aurelius, "evasão é o ato de desistir de alguma coisa, abandonar alguma coisa". No contexto deste trabalho, a evasão se refere à desistência de um curso. Para evitar ou mitigar a evasão, são necessários mecanismos que auxiliem na análise de quais são as causas de evasão e na tomada de decisão. A educação é tida como o norteador do futuro da nação. Segundo Lobo (2012), existe uma relação ruim entre as instituições de Ensino Superior Brasileiras (IES) e a evasão, pois quando o aluno evade, existe o desperdício de recursos das universidades públicas, enquanto que para as privadas é uma perda de receita por consequência da diminuição de mensalidades.

De acordo com Lobo (2017), as taxas de evasão mantiveram-se constantes durante os últimos 15 anos estudados pelo autor, período compreendido entre 2000 e 2015, alcançando uma taxa de evasão de 22%. Pode-se ressaltar que, as medidas de combate à evasão escolar no Brasil vêm se mostrando ineficientes, pois não houve uma melhora considerável no problema de evasão. Pôde-se observar ainda que, no período de 2011 até 2015, a evasão reduziu em apenas 1% para os cursos de bacharelado.

Técnicas de *Machine Learning* têm sido empregadas em diversas áreas, como medicina (TSUMOTO SHUSAKU E HIRANO, 2010)(RAJESH, 2011), indústria de alimentos (ROPODI; PANAGOU EZ E NYCHAS, 2016), mercado financeiro (ENKE DAVID E THAWORNWONG, 2005) e educação (ROMERO CRISTOBAL E VENTURA, 2013). Diversos trabalhos ao redor do mundo utilizaram técnicas de *Machine Learning* com objetivo de reduzir e predizer a evasão escolar. Há diversos trabalhos encontrados na literatura que atacam este problema, na Turquia (YUKSELTURK; OZEKES; TÜREL, 2014), na Holanda (DEKKER; PECHENIZKIY; VLEESHOUWERS, 2009) e nos Estados Unidos (RAJU; SCHUMACKER, 2015). No Brasil, alguns trabalhos também foram desenvolvidos (RIGO et al., 2014) (DIGIAMPIETRI; NAKANO; LAURETTO, 2016) (MANHÃES et al., 2012) (MACHADO et al., 2015).

Técnicas de *Machine Learning* são utilizadas dentro de processos de *Knowledge Discovery in Databases* (KDD). KDD é o processo de extração de dados que visa a obtenção de informação que não é óbvia, antes desconhecida e com potencial de ser útil. Estes dados vêm de várias fontes e são transformadas em uma base de dados comum. Depois disto são pré-processados e vão para a etapa de mineração de dados, que produz uma saída no formato de regras ou padrão, que devem ser interpretadas para gerar conhecimento novo e potencialmente útil (BRAMER, 2007).

Segundo Roiger (2017), Mineração de Dados é um processo que envolve a exploração de dados e busca por padrões úteis, possibilitando o desenvolvimento de modelos computacionais que representam conhecimento sobre estes dados. Alguns autores definem o processo de Mineração de Dados como sendo sinônimo de KDD. Outros preferem defini-lo como um dos passos dentro de KDD (HAN; PEI; KAMBER, 2011)(GOLDSCHMIDT; BEZERRA; PASSOS, 2015). Neste trabalho, por definição será tratado como o segundo caso: Mineração de Dados como um passo dentro de KDD.

Quando qualquer trabalho relacionado à Mineração de Dados para a área da educação é encontrado, pode-se afirmar que este está dentro do subconjunto denominado de *Educational Data Mining* (EDM). A EDM pode ser descrita como a intersecção de três principais áreas: ciência da computação, educação e

estatística. Na EDM, técnicas de mineração de dados são aplicadas para encontrar padrões em grandes quantidades de dados educacionais, buscando formas de resolver problemas na área de educação. Existem diversos tipos de trabalhos na área de EDM que já foram desenvolvidos, como trabalhos de predição de desempenho de estudantes, predição de evasão (MACHADO et al., 2015), sistema de recomendação em educação (MANOUSELIS et al., 2011), análise de redes sociais na educação (GRUNSPAN; WIGGINS; GOODREAU, 2014), modelo para matrícula (PEDRO et al., 2013), entre outros trabalhos que vêm surgindo sobre este tema (KOEDINGER et al., 2015).

Neste trabalho, busca-se analisar e prever a evasão escolar em um ambiente de Ensino Superior, utilizando o processo de KDD. Este estudo de caso será realizado com os dados do Instituto Federal de Santa Catarina (IFSC) - Câmpus Caçador, levando em conta o Ensino Superior, que atualmente contém dois cursos de graduação: Sistemas de Informação e Engenharia de Produção, nos quais havia, em 2019, por volta de 269 estudantes (MEC, 2020).

1.1 Problema de Pesquisa

Para o problema de pesquisa, pode ser definida a seguinte questão: "Com base nas técnicas e algoritmos de *Machine Learning* mais frequentes na Literatura, como ofertar um protótipo com um modelo computacional para predição de evasão escolar a nível de estudante para os cursos de graduação do Instituto Federal de Santa Catarina - Câmpus Caçador?".

1.2 Hipótese de Pesquisa

A hipótese para a solução do problema apresentado é que é possível desenvolver um protótipo com um modelo computacional para a predição de alunos de graduação do Instituto Federal de Santa Catarina - Câmpus Caçador com grande probabilidade de evadir, usando redes neurais.

1.3 Objetivos

1.3.1 Objetivo Geral

Como objetivo geral deste trabalho, têm-se investigar as possibilidades para oferta um protótipo com um modelo computacional direcionado ao IFSC - Câmpus Caçador para auxiliar na identificação de estudantes de graduação com potencial de evasão.

1.3.2 Objetivos Específicos

- Realizar mapeamento sistemático sobre o tema, a fim de identificar as técnicas/algoritmos de *Machine Learning* mais frequentes para resolução do problema da evasão escolar;
- Fazer a coleta e selecionar os dados da base de dados do IFSC - Câmpus Caçador;
- Realizar a etapa de pré-processamento dos dados obtidos para eliminar inconsistências e preparar os dados para serem utilizados em algoritmos de *Machine Learning*;
- Gerar e avaliar os modelos computacionais, usando os dados coletados e pré-processados, para realizar a classificação de estudantes de graduação como potenciais evasores ou não;
- Desenvolver um protótipo a partir do melhor modelo computacional obtido.

1.4 Justificativa

Pode-se ressaltar que, embora existam várias pesquisas sobre evasão escolar tentando demonstrar uma maneira de solucionar esse problema, a evasão pode variar de acordo com a cidade ou conforme a IES, e outras características específicas que influenciam sobre o problema. A motivação para este trabalho partiu de pesquisas semelhantes em várias instituições no Brasil e no mundo. Cada Instituição possui suas próprias particularidades. Com isto, se faz necessário a adequação do modelo para cada região ou Instituição diferente em que for aplicado. Ou seja, o modelo que será gerado pode ser útil para o contexto da IES em específico, mas não diretamente para outras localidades. Desta forma, será possível auxiliar a Instituição com objetivo de prevenir as evasões apontadas pelo modelo computacional. O modelo será construído a partir de dados históricos dos alunos na base de dados do sistema acadêmico.

A metodologia abordada nesta pesquisa é a de KDD que é considerada consolidada, como é mostrado por Alban e Mauricio (2019), onde usando termos relacionados à evasão escolar foram encontrados mais de mil trabalhos sobre este problema de pesquisa no contexto de Ensino Superior envolvendo diversas bases de dados (ALBAN; MAURICIO, 2019). Esta metodologia permite processar e analisar dados, o que auxilia o estudo de todas as informações sobre uma grande quantidade de alunos e analisar suas variáveis, o que humanamente seria um trabalho complexo de se fazer, calcular e bastante suscetível a erros.

Todo o resultado obtido neste trabalho pode servir de base para novos projetos que surgirem a respeito deste tema, podendo ser reaproveitado o *dataset*, ou o pré-processamento, os atributos escolhidos. Além disso, os resultados podem ser utilizados em comparações em testes de novos algoritmos.

1.5 Organização do texto

O restante deste texto está organizado da seguinte forma: No Capítulo 2 é apresentada a fundamentação teórica. No Capítulo 3 é apresentada o mapeamento sistemático da literatura. No Capítulo 4 é apresentada a metodologia. No Capítulo 5, é apresentada a linha histórica seguida para a obtenção e o tratamento dos dados. No Capítulo 6 são apresentados os resultados obtidos e o protótipo construído. Por fim, no Capítulo 7 são apresentadas as considerações finais sobre este projeto de conclusão de curso.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são abordados os conceitos de *Knowledge Discovery in Database* (KDD), Mineração de Dados e *Machine Learning* (ML), sendo que estas técnicas e algoritmos são aplicados para auxiliar no tratamento do problema de evasão escolar neste trabalho. Além disso, também será detalhado o problema da evasão escolar.

2.1 Evasão escolar

A evasão escolar é um problema recorrente e um desafio a ser solucionado. Segundo o dicionário Aurelius, "evasão é o ato de desistir de alguma coisa, abandonar alguma coisa". No contexto de uma instituição de ensino superior (IES), pode significar também o ato de desistir de um curso. Isto acaba gerando desperdícios econômicos. Para as instituições públicas pode-se dizer que é, um desperdício de recurso, enquanto que para as privadas é uma diminuição no seu orçamento ou importante perda de receitas sem essas mensalidades dos alunos (FILHO et al., 2007). Segundo o autor, pode-se observar alguns motivos de desistência do estudante sendo estes: (a) a falta de orientação vocacional; (b) imaturidade do estudante; (c) reprovações sucessivas; (d) dificuldades financeiras; (e) falta de perspectiva de trabalho; (f) ausência de laços afetivos na universidade; (g) ingresso na faculdade por imposição familiar; e (h) casamentos não planejados e nascimento de filhos (PERON; BEZERRA; PEREIRA, 2019 apud GAIOSO, 2005).

Umas das formas de analisar informações relevantes sobre a educação no ensino superior no Brasil é através da consulta aos dados do Instituto Nacional de Pesquisas Educacionais Anísio Teixeira (INEP), órgão ligado ao Ministério da Educação (MEC), cujo objetivo é organizar, manter e apresentar informações e estatísticas educacionais. Dentro deste instituto, existem avaliações, exames e indicadores tanto para Educação Superior, como para Educação Básica.

O Ensino Superior brasileiro tem registrado um breve crescimento em 2017 para 2018. Os números de vagas, matrículas e novos cursos vêm aumentando ano após ano, tanto na rede pública como particular. De acordo com o Censo da Educação Superior 2018, publicado pelo INEP, o Brasil possui 2.537 mil instituições de ensino superior (IES), entre públicas e privadas. Conforme o Censo, em 2018 foram ofertados 37.962 mil cursos de graduação. Foram 8.450.755 milhões de matrículas registradas (PERON; BEZERRA; PEREIRA, 2019)(INEP, 2019).

Segundo o INEP, a respeito da Rede Federal de Educação Superior, em 2018, de 33.929 ingressantes em um curso presencial da graduação distante da sua cidade natal, é estimado pelo órgão que no primeiro ano 10,4% desses estudantes haviam desistido do curso, e 4,1% estavam já com a matrícula trancada. Com esses dados é possível analisar que existe uma elevada taxa de evasão no primeiro ano de diversos cursos, por diversos fatores sociais, de distância de casa até a sua instituição, socioeconômicos, etc. Conforme nos indica o relatório do Censo de Educação Superior de 2018, de 362.005 ingressantes, nas Instituições Federais, 64.567 (18%) fizeram o ENEM mais uma vez, e pode-se supor que um destes motivos seja a insatisfação com seu curso, podendo o ENEM ser uma tentativa de troca de curso (INEP, 2019).

De acordo com a Constituição Federal, art. 6º: "São direitos sociais **a educação**, a saúde, a alimentação, o trabalho, a moradia, o lazer, a segurança, a previdência social [...]" (BRASIL, 1988). Ou seja, a educação é um direito garantido pela lei do Brasil, sendo o abandono do ambiente escolar e acadêmico um dos problemas que afetam negativamente esse direito. Pode-se estabelecer uma relação

aqui entre a importância da educação e este problema que é enfrentado. Segundo Neri (2009), a educação constitui o verdadeiro custo de oportunidade para a sociedade: nenhuma nação cresce sem ela.

2.1.1 Tipos de evasão

Segundo SESU (1996), o termo "evasão" no contexto de uma Instituição de Ensino Superior, pode significar a saída definitiva de um estudante do seu curso sem que o mesmo tenha conseguido concluir, ou seja, evasão escolar. Porém, por existirem várias formas na qual o aluno pode evadir-se de um curso. O autor preferiu separar em diversas categorias, como a de evasão de curso, evasão da instituição e evasão do sistema, detalhadas a seguir.

Evasão no curso: O estudante pode evadir do curso por diversas situações: abandono (neste caso refere-se ao aluno não matricular-se), desistência (oficial), transferência ou reopção (mudança de curso), exclusão por norma institucional.

Evasão na instituição: O aluno desliga-se da instituição.

Evasão do sistema: Ele abandona de forma definitiva ou temporária o sistema educacional.

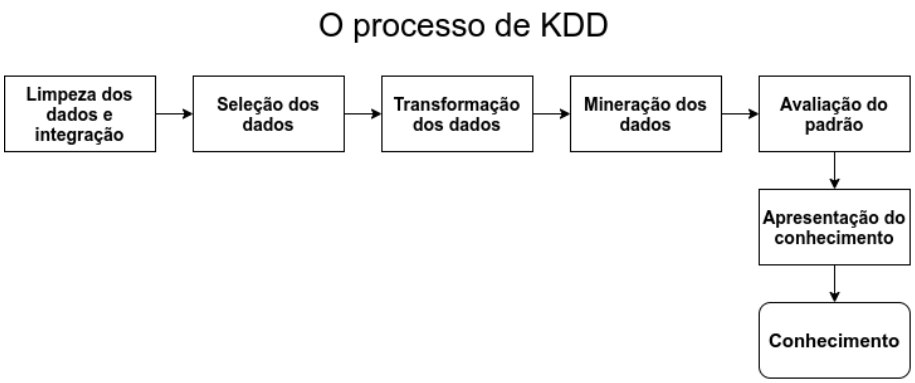
É necessário buscar a melhor alternativa, que pode auxiliar na predição do abandono dos estudantes, e uma destas alternativas emerge por meio da Mineração de Dados Educacionais (EDM). EDM é uma área que integra a estatística, computação e educação para ajudar a analisar e encontrar os fatores que mais influenciam nos problemas da educação, sejam sociais, socioeconômicos, familiares, diversos fatores combinados, entre outros.

Uma das ferramentas que vem sendo utilizada para resolver uma série de problemas em diversas áreas de conhecimento é o processo de KDD. O processo fornece passos que podem auxiliar no encontro de padrões em dados, podendo ser útil para predizer a desistência de estudantes, ajudar a solucionar problemas de outras áreas que envolvam dados de desempenho, dentre outras aplicações.

2.2 Knowledge Discovery in Databases (KDD)

Segundo Han, Pei e Kamber (2011), o processo de KDD é uma sequência de passos que tem por objetivo a descoberta de conhecimento. Importante ressaltar que o KDD é um processo iterativo, significando que a execução do processo em si não ocorre de forma linear do começo ao fim, sendo necessário, por muitas vezes, voltar a estágios anteriores para depois seguir adiante novamente (FAYYAD et al., 1996). A Figura 1 exibe os passos definidos na bibliografia.

Figura 1 – Sequência de passos do KDD



Fonte: Han, Pei e Kamber (2011)

É importante entender o que acontece em cada passo do KDD, conforme mostrado na Figura 1. Os passos são:

1. **Limpeza de dados:** São removidos os dados que estão inconsistentes na base de dados ou fora do padrão, ou se utiliza de alguma técnica para tentar solucionar o problema de dados faltantes. São exemplos de técnicas utilizadas para tratar dados faltantes: (a) preencher o dado faltante manualmente; (b) usar uma constante global para preencher esse dado; (c) trocar o dado que falta por uma medida de tendência central (média, moda ou mediana); (d) usar o valor mais provável para preencher o valor faltante (usar alguma técnica para estimar esse valor); (e) ignorar o valor faltante; (g) usar um atributo de média ou mediana para todos os campos da mesma classe (GARCIA; LEITE; SKRJANC, 2019)(HAN; PEI; KAMBER, 2011).

2. **Integração de dados:** Integração onde várias fontes de dados podem ser combinadas, de forma a enriquecer os dados originais e oferecer maiores possibilidades à obtenção de informações por meio do KDD.

3. **Seleção dos dados:** Os dados relevantes para a solução do problema são extraídos desta base de dados.

4. **Transformação nos dados:** Transformação ou consolidação dos dados em formas que possam ser utilizados pelas técnicas de Mineração de Dados.

5. **Mineração de Dados:** Processo onde são buscados padrões de dados úteis. Não existe nenhum algoritmo específico para a Mineração dos Dados, tendo em vista que essa etapa engloba desde a análise exploratória dos dados até a aplicação dos modelos de *Machine Learning*. Podem ser citados como exemplo de técnicas de *Machine Learning* para classificação: algoritmo de classificação bayesiano, de vizinho mais próximo (i.e. kNN) e árvore de decisão (BRAMER, 2007).

6. **Avaliação dos padrões:** identifica os padrões interessantes entre os apresentados pela etapa anterior de mineração de dados.

7. **Apresentação do conhecimento:** utiliza técnicas de representação e visualização do conhecimento para apresentar o conhecimento adquirido aos usuários.

2.2.1 Mineração de Dados

Segundo Roiger (2017), Mineração de Dados é um processo que envolve a exploração de dados e busca por padrões úteis. O conhecimento é obtido por meio de previsões com o modelo que foi gerado. Pode-se ressaltar que Mineração de Dados é o quinto passo das etapas de KDD. Algumas tarefas de Mineração de Dados que podem ser citadas são a preparação dos dados para mineração, análise exploratória dos dados, modelagem de resposta binária, classificação com algoritmos de *Machine Learning* ou outra técnica, estimação de valores para classes, busca por grupos e associações, aplicação o modelo para novos dados (LINOFF; BERRY, 2011).

Este processo necessita das etapas anteriores como limpeza de dados, seleção dos dados, transformação dos dados, descoberta de padrões, e algumas etapas futuras, estando presente também na avaliação de padrões e apresentação do conhecimento (HAN; PEI; KAMBER, 2011). Mineração de Dados não é uma etapa isolada, englobando atividades desde a análise exploratória, *Feature Engineering*, até a aplicação dos algoritmos de *Machine Learning*.

2.2.1.1 *Machine Learning* (ML)

Machine Learning (ML) é a programação de computadores para utilizar algum critério de desempenho por meio dos dados, podendo ser dados históricos, de experiência passada, ou outros tipos de dados. A etapa de ML gera um modelo que captura a essência dos dados, e a partir disto, é possível realizar predições com dados que ele não conhece para validar o modelo e verificar sua corretude. O modelo gerado pelo ML pode ser tanto preditivo – para fazer predições do futuro –, como pode ser descritivo, para obter conhecimento de dados, ou para ambas as situações (ALPAYDIN, 2020).

A etapa de ML dentro do processo de KDD está ligada aos aspectos de treinamento do modelo, geração do modelo e logo após, avaliação dos padrões encontrados, verificando se há um nível de confiança aceitável. Caso contrário, os passos anteriores a ele são repetidos. *Machine Learning* tem como principal objetivo fazer com que os computadores aprendam padrões - automaticamente por meio de modelos estatísticos -, e até a fazer decisões inteligentes baseadas em dados (HAN; PEI; KAMBER, 2011). A etapa de ML é uma das últimas etapas do processo de KDD, quando o modelo é gerado e depois testado. Com isto, pode-se validar se está correto, aceitável ou se são necessárias modificações.

Existem dois tipos de dados que são usados de forma diferente para minerar essas informações úteis. O primeiro é denominado *rotulado*, em que o algoritmo tenta prever o valor desse atributo (também chamado atributo-alvo) dado atributos-preditores desconhecidos para o modelo. A mineração de dados usando dados rotulados é chamada de *aprendizado supervisionado*. Se o atributo-alvo é categórico, e busca-se definir algo como “bom” ou “médio” ou “ruim” ,“gato” ou “cachorro” essa tarefa é chamada de *classificação*. Se o atributo-alvo é numérico, a tarefa é chamada de *regressão*, onde o modelo prevê um número futuro de preços, de venda, volume de chuva, etc (BRAMER, 2007).

Para o segundo tipo de dados, chamados de dados não-rotulados, o aprendizado é conhecido como *aprendizado não-supervisionado*. Ele tem por objetivo-chave extrair a maior parte das informações disponíveis nas relações naturais dos dados. Um exemplo de aprendizado não-supervisionado é a aplicação de algoritmos de *regras de associação*, que por definição analisa os relacionamentos entre os valores dos atributos. Pode-se citar como exemplo a análise de cesta de mercado, onde analisando a compra desses produtos, é possível encontrar regras como “ao comprar produto X, há grande chance de comprar Y” (BRAMER, 2007).

Neste trabalho, são utilizados algoritmos de classificação para prever se o estudante tem a possibilidade (ou não) de evasão de seu curso. Esta tarefa (classificação) é a mais apropriada para este caso, pois é desejável que o atributo-alvo seja se o aluno tem potencial de evadir ou não. Já os atributos analisados podem ser vários dentro do contexto escolar de uma IES, como por exemplo, dados socioeconômicos, escolares ou pessoais. Pode-se citar como exemplo de variáveis a serem utilizadas: frequência, idade, distância da IES, renda familiar, entre outros.

2.2.1.1.1 Classificação

A classificação consiste em separar os dados de modo que eles estejam associados a uma categoria exclusiva conhecida como classe. Cada conjunto de informações deve pertencer exclusivamente e ser associado à apenas uma classe (BRAMER, 2007). Como exemplo de algoritmos temos o Algoritmo de Classificação Bayesiano, Árvore de Decisão, Redes de Crença Bayesiana, Classificação por Retropropagação, Máquinas de Vetores de Suporte, Redes Neurais e outros.

De acordo com o estudo de Alban e Mauricio (2019), o método mais utilizado presente em 22 dos 28 artigos filtrados com técnicas de Mineração de Dados é a técnica/algoritmos de árvore de decisão que foi usado para classificação. Além das árvores de decisão, um algoritmo bastante utilizado que foi

utilizado é a Rede Neural, devido a sua grande capacidade de aproximação de função, sendo que é o segundo método com mais frequência de uso para problemas de aplicação de Mineração de Dados para predição de evasão escolar.

2.2.1.1.2 Rede Neural

O termo Rede Neural remete a um sistema artificial de neurônios que são inspirados nos neurônios biológicos. Este tipo de algoritmo tenta simular o processo que ocorre no cérebro, onde os neurônios estão interconectados e se comunicando com outros neurônios através de sinapses. Isto é, no caso da rede neural, os nós estão conectados e transmitindo informações uns aos outros.

Ela pode abranger desde um simples nó, dado que todo nó corresponde a um neurônio, bem como a um conjunto de nós conectados em uma grande rede. Conforme vai aprendendo com os dados disponíveis, a Rede Neural vai atribuindo pesos a todos os nós. Este peso é multiplicado para cada respectiva entrada e representa a importância atribuída a cada informação (GURNEY, 1997).

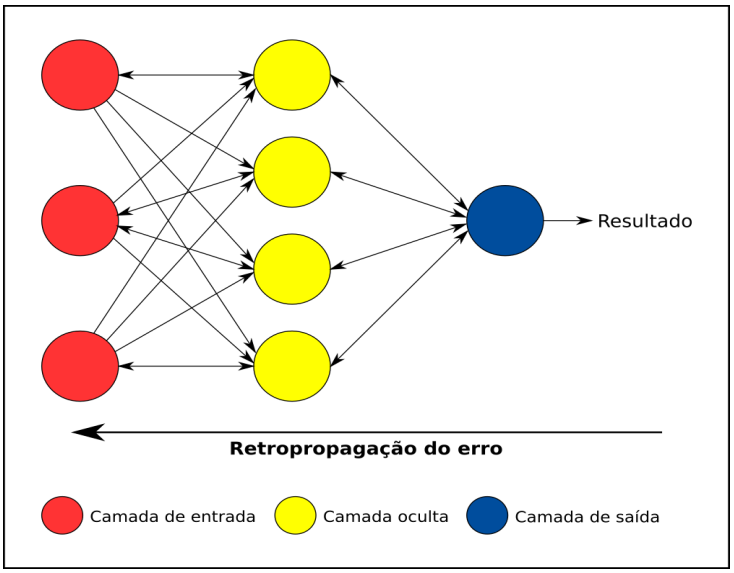
2.2.1.1.3 *Backpropagation* (retropropagação)

O algoritmo de retropropagação, também conhecido como diferenciação de modo reverso, possui por objetivo geral selecionar pesos que forneçam uma estimativa ótima de uma função que modela os dados de treinamento, ou seja, encontrar um conjunto de pesos para minimizar a saída da função de perda DSA (2020).

Ele possui por padrão os seguintes passos de uma estrutura de esqueleto básico, conhecida também como regra do delta ou regra de Perceptron: (a) inicializar os pesos; (b) laço de repetição; (c) recebe padrão para treinamento; (d) realiza predição e compara com o resultado esperado; (e) retropropaga o erro para ajuste dos pesos; (f) ajusta os pesos até que a taxa de erro seja pequena o suficiente de acordo com parâmetro pré-definido (GURNEY, 1997).

Como pode ser visto na Figura 2, dentro de uma Rede Neural temos um modelo onde os neurônios ou nós estão ligados todos por um nó anterior, contando com uma camada de entrada onde são recebidos os dados. A camada oculta que possui este nome por não ser nem de entrada nem de saída, e pode ter várias camadas ocultas. E a camada de saída que gera o valor de saída.

Figura 2 – Conceito de retropropagação nas Redes Neurais



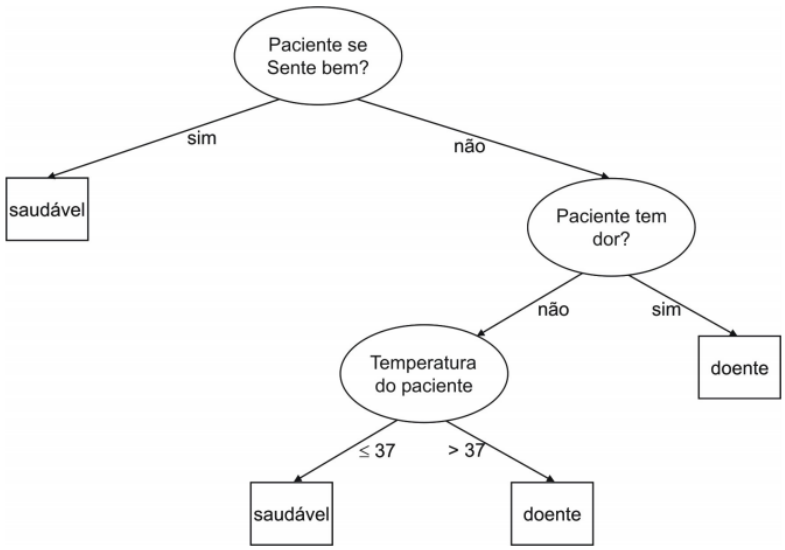
Fonte: DSA (2020)

2.2.1.1.4 Árvore de Decisão

As técnicas de Árvore de Decisão geralmente são usadas para classificação, mas podem ser utilizadas também em outras tarefas. Ela escolhe dentre os atributos possíveis, qual é o melhor atributo para se tornar o nó divisor, por meio de um cálculo que pode ser de entropia, Gini ou outros (HAN; PEI; KAMBER, 2011). São comumente utilizados para ganho de informação com um objetivo de tomada de decisão (PENG; CHEN; ZHOU, 2009).

Uma Árvore de Decisão é uma estrutura onde cada nó-ramo representa uma escolha dentro um número de alternativas, e um nó-folha define um tipo de decisão a ser representada. Geralmente ela é representada por um fluxograma onde são modelados e calculados o peso-valor destas escolhas. O algoritmo de treinamento escolhe atributos para ser os nós-ramos e nós-folhas. A Árvore de Decisão divide cada nó recursivamente de acordo com o algoritmo de aprendizado da Árvore de Decisão. O resultado final é uma Árvore de Decisão na qual cada ramo representa um possível cenário e um valor de saídas (PENG; CHEN; ZHOU, 2009). São exemplos de algoritmos de árvore de decisão ID3 (JIN; DE-LIN; FEN-XIANG, 2009), C4.5 (KORTING, 2006), CART (CRAWFORD, 1989), e outros.

Figura 3 – Um exemplo de Árvore de Decisão



Fonte: Monard e Baranauskas (2003)

Um exemplo de Árvore de Decisão pode ser visto na Figura 3, onde é apresentada uma árvore para o diagnóstico de um paciente seguindo uma rotina padrão. Cada retângulo representa uma classe, ou o diagnóstico dado como doente ou saudável, e cada elipse um teste em um atributo (MONARD; BARANAUSKAS, 2003). Neste exemplo, a primeira verificação ocorre sobre o atributo da sensação da pessoa. Caso ela se sinta bem, ela é classificada como saudável. Caso não se sinta bem, são feitas novas verificações ou testes, e então verificado se o paciente sente dor. Caso ele sinta dor, a pessoa é classificada como doente, se não sente dor, sua temperatura é verificada. A última característica a ser verificada na árvore é a temperatura, se ela estiver acima de 37 graus, a pessoa é classificada como doente. Caso sua temperatura esteja igual ou abaixo de 37 graus, ela é classificada como saudável.

3 ESTADO DA ARTE DA ÁREA PESQUISADA

O processo de pesquisa e seleção dos trabalhos relacionados foi realizado com base em um mapeamento sistemático sobre as pesquisas com propostas para resolver o problema da evasão escolar utilizando a Mineração de Dados. O resultado deste estudo resultou na identificação e seleção dos principais trabalhos de pesquisa no tema deste Trabalho de Conclusão de Curso. Outro objetivo da mapeamento sistemático foi verificar os métodos utilizados para resolver o problema proposto.

3.1 Mapeamento Sistemático da Literatura

Durante a etapa de mapeamento sistemático da literatura é preciso realizar as tarefas de definições de questões de pesquisa e strings de busca, realização da pesquisa de estudos primários relevantes, triagem dos documentos, *keywording* dos resumos, e extração dos dados. A ferramenta Parsifal[1] foi utilizada para registrar o processo de definir a *string* de busca, buscar e salvar os artigos, além de realizar as classificações pertinentes com os critérios selecionados.

A questão de pesquisa utilizada foi "Como a Mineração de Dados tem sido aplicada para prever potencial evasão escolar?", a partir da qual foram extraídos termos e palavras relacionados que nos ajudaram a montar a *string* de busca para realizar as consultas nas bases de dados selecionadas. Podemos visualizar estas palavras com seus sinônimos na Tabela 1.

Tabela 1 – Tabela com palavras-chave e sinônimos

Palavra-chave	**Sinônimos**
Data Mining	KDD, Machine Learning, ML, Knowledge Discovery Database
Predict	avoid, foresee, predicting, prediction
School dropout	college dropout, drop out student, dropping student, scholar evasion, school abandonment, school leavers, school retention, university dropout, university evasion

Fonte: Elaborada pelo autor.

Na Tabela 2, são listadas as bases de dados em que foram pesquisados os artigos juntamente com a *string* de busca utilizada e o número de artigos que foram retornados com esta busca. Como pode ser notado, a mesma *string* de busca foi utilizada para as três bases de dados.

[1] https://parsif.al/

Tabela 2 – Tabela com as bases de dados/artigos

Base de dados	**Artigos**	**String de busca**
ACM Digital Library	13	(“predict” OR “avoid” OR “foresee” OR “predicting” OR “prediction”) AND (“data mining” OR “KDD” OR “knowledge discovery database” OR “machine learning” OR “ML”) AND (“school dropout” OR “college dropout” OR “drop out student” OR “dropping student” OR “scholar evasion” OR “school abandonment” OR “school leavers” OR “school retention” OR “university dropout” OR “university evasion”)
IEEE Digital Library	7	
Scopus	31	

Fonte: Elaborada pelo autor.

3.1.1 Critérios de Exclusão

A pesquisa inicial nas bases de dados utilizando a *string* de pesquisa foi aplicada, excluindo trabalhos que se enquadram nos seguintes critérios de exclusão, conforme a Tabela 3:

Tabela 3 – Tabela com os critérios de exclusão

Critério de exclusão	**Nº de artigos recusados**
O estudo não é um estudo primário	3
O estudo não faz parte da área de pesquisa	15
O estudo não foi aplicado para o ensino superior	6
O estudo é duplicado	7
O autor usou uma base de dados de terceiros, ou de um repositório online. Não possui um *dataset* próprio.	1

Fonte: Elaborada pelo autor.

A pesquisa começou com 51 artigos no total, buscando nas três bases de dados. Após a realização dos critérios de exclusão sobraram 19 artigos. Três (3) artigos foram eliminados no critério de “o estudo não é um estudo primário”, que significa que o artigo geralmente é uma revisão sistemática ou derivada. Pelo critério de o “estudo não faz parte da área de pesquisa”, 15 trabalhos foram rejeitados, o que indica relação com uma área próxima, porém diferente da proposta. Com relação aos últimos critérios, foram removidos 7 artigos duplicados, sendo 6 artigos que não foram aplicados com foco para o ensino superior, e sim para outros como ensino fundamental ou médio. Por fim, 1 artigo usou ou elaborou o estudo por meio de um *dataset* pronto, ou que não era próprio.

3.1.2 Critérios de Inclusão

Os seguintes critérios de inclusão foram definidos:

- Nova tecnologia para prever evasão escolar;
- Processo, método ou técnica para previsão automática de evasão escolar;
- Sistema para prever evasão escolar usando Mineração de Dados.

Todos os 19 artigos selecionados se enquadram em um ou mais dos critérios de inclusão apresentados. Na Tabela 4, são apresentados os artigos selecionados:

Tabela 4 – Tabela com os artigos selecionados

ID	Título do artigo	Autores/Autor(a)
A1	Data Mining Applied in School Dropout Prediction	Amelec Viloria, Jesús García Gualiny, William Niebles Núñez, Hugo Hernández Palma e Leonardo Niebles Núñez
A2	Early Prediction of University Dropouts – A Random Forest Approach	Andreas Behr, Marco Giese, Herve Teguim, Katja Theune
A3	Predictive models for imbalanced data: A school dropout perspective	Thiago Barros, Plácido A.Souza Neto, Ivanovitch Silva, Luiz Affonso Guedes
A4	A multinomial and predictive analysis of factors associated with university Dropout [Un análisis multinomial y predictivo de los factores asociados a la deserción universitaria]	Tatiana Fernández-Martin, Martin Solis, María Teresa Hernández-Jiménez, Tania Elena Moreira-Mora
A5	Ensemble regression models applied to dropout in higher education	Paulo M. da Silva, Marilia N. C. A. Lima, Wedson L. Soares, Iago R. R. Silva, Roberta A. de A. Fagundes, Fernando F. de Souza
A6	Supervised learning in the context of educational data mining to avoid university students dropout	Kelly J. de O. Santos, Angelo G. Menezes, Andre B. de Carvalho, Carlos A. E. Montesco
A7	Predictive modelling of student dropout using ensemble classifier method in higher education	Nindhia Hutagaol, Suharjito Suharjito
A8	Integration of data technology for analyzing university dropout	Amelec Viloria, Jholman Garcia Padilla , Carlos Vargas-Mercado, Hugo Hernández Palma, Nataly Orellano Llinas, Monica Arrozola David
A9	University dropout: A prediction model for an engineering program in bogotá, Colombia	Andres Acero, Juan Camilo Achury, Juan Morales Piñero
A10	Predicting early dropout students is a matter of checking completed quizzes:The case of an online statistics module	Josep Figueroa-Cañas, Teresa Sancho-Vinuesa
A11	Educational data mining: An application of regressors in predicting school dropout	Rafaella Leandra Souza do Nascimento, Ricardo Batista das Neves Junior, Manoel Alves de Almeida Neto, Roberta Andrade de Araújo Fagundes
A12	Using academic analytics to predict dropout risk in engineering courses	Johnny Lima, Paulo Alves, Maria João Pereira, Simone Almeida
A13	Prediction of university dropout through technological factors: A case study in Ecuador	Mayra Susana Alban Taipe, David Mauricio Sánchez
A14	Data-driven system to predict academic grades and dropout	Sergi Rovira, Eloi Puertas, Laura Igual
A15	Automatic feature selection for desertion and graduation prediction: A chilean case	B. Peralta, T. Poblete, L. Caro
A16	Prediction of university desertion through hybridization of classification algorithms	Carol Francia Rocha, Yuliana Flores Zelaya, David Mauricio Sánchez, Armando Fermín Pérez

ID	Título do artigo	Autores/Autor(a)
A17	Dropout prediction at university of genoa: A privacy preserving data driven approach	Luca Oneto, A Siri, G Luria, Davide Anguita
A18	Optimization of Weight Backpropagation with Particle Swarm Optimization for Student Dropout Prediction	Eka Yulia Sari, Kusrini, Andi Sunyoto
A19	Applying Data Mining Techniques to Predict Student Dropout: A Case Study	Boris Perez, Camilo Castellanos, Dario Correal

Fonte: Elaborado pelo autor.

Todos os trabalhos se referem à maneiras de auxiliar na predição da evasão escolar por meio da Mineração de Dados ou aplicação de técnicas de *Machine Learning*.

3.2 Análise dos Trabalhos Selecionados

A última etapa do Mapeamento Sistemático da Literatura foi a extração de dados dos trabalhos selecionados. Os algoritmos mais utilizados em grande parte dos trabalhos são o algoritmo de Naïve Bayes e derivados (A5, A6, A7, A8, A14, A16, A19), máquinas de vetores de suporte (A4, A6, A14), regressão logística (A4, A5, A13, A14, A15, A19), rede neural (A3, A4, A6, A8, A16), árvore de decisão (A1, A3, A4, A5, A6, A7, A8, A10, A12, A13, A15, A16, A19), floresta aleatória (A2, A4, A6, A14, A17, A18), e K-ésimo vizinho mais próximo (A6, A7, A8).

Alguns autores preferiram utilizar ferramentas para realizar o processo de KDD, como a ferramenta WEKA (A1, A8). Outros algoritmos que foram usados em menor frequência são *balanced bagging* (A1), regressão do vetor de suporte e regressão quantílica (A11), *K-means* (A12), reforço adaptativo (A14).

Os *datasets* foram fornecidos pela Diretoria de Tecnologia da Informação e Comunicação (DTIC). Os dados obtidos são do IFSC - Câmpus Caçador, com informações dos cursos solicitados e os dados dos alunos devidamente anonimizados. Com base na análise dos artigos estudados, as técnicas identificadas foram estudadas de forma mais aprofundada, de modo que seja aproveitado o mapeamento da área de pesquisa, especialmente o assunto sobre Redes Neurais e Árvores de Decisão.

4 METODOLOGIA

O projeto foi iniciado com uma pesquisa exploratória da bibliografia, onde foi realizado um mapeamento sistemático da literatura para identificar as principais técnicas, métodos e algoritmos para realizar a predição da evasão escolar por meio do uso do processo de mineração de dados. Além disso, também foram estudados os conceitos mais utilizados sobre evasão escolar com foco no ensino superior.

A partir disto, foi feita a solicitação do *dataset* contendo dados dos estudantes de graduação do IFSC Câmpus Caçador devidamente anonimizados. Após a obtenção do *dataset*, foi possível realizar as etapas de limpeza, seleção, transformação, e mineração dos dados, para enfim aplicar os métodos de *Machine Learning*. Estes dados foram solicitados à própria Instituição, mais especificamente em comunicação com a Diretoria de Tecnologia da Informação e Comunicação (DTIC), conforme descrito em detalhes na Seção 5.1.

A seguir, são apresentadas as etapas necessárias para o desenvolvimento e avaliação do trabalho. De acordo com o processo do KDD, muitas vezes é preciso codificar os dados para que seja possível superar possíveis limitações de técnicas de *Machine Learning*. Algumas das técnicas são apresentadas rapidamente em 4.1. Além disso, como parte da criação do modelo de *Machine Learning*, é preciso argumentar sobre as possibilidades de teste do mesmo, explicado na Seção 4.2. Também é necessário mencionar algumas das métricas e ferramentas visuais de avaliação do modelo, que são fornecidas na Seção 4.4.

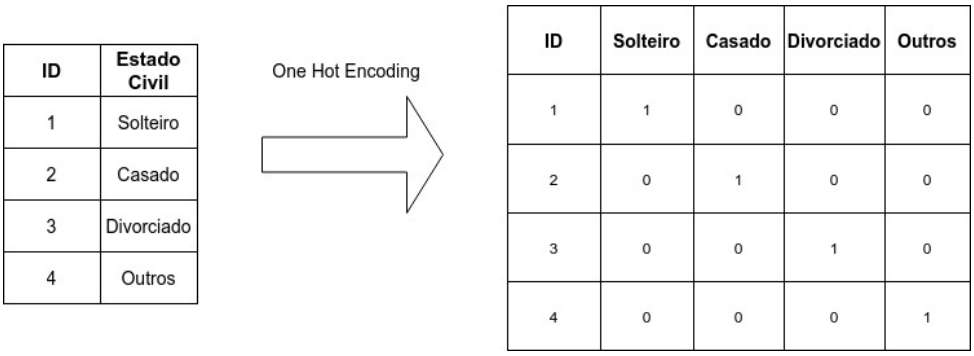
4.1 Codificação dos Dados

Ao trabalhar com dados e técnicas de *Machine Learning*, o responsável deve conhecer as limitações e algumas técnicas de tratamento de dados. Por exemplo, em casos onde há dados textuais (categóricos) e a técnica de *Machine Learning* escolhida suporta apenas dados do tipo numérico, existem métodos de codificação dos dados, que tem como objetivo transformar os dados de categóricos em numéricos, ou numéricos em categóricos. Pode-se citar alguns exemplos de técnicas de codificação de variáveis categóricas como: *One Hot Encoding*, *Ordinal Encoding*, *Binary Encoding*, entre outros (POTDAR; PARDAWALA; PAI, 2017).

Quando há uma coluna/característica/atributo que tem uma escala nominal qualitativa, ou seja, os valores presentes para esta coluna não possuem ordem ou sequência, os dados funcionam como categorias. Assumindo a característica “estado_civil” que é usada neste trabalho para o conjunto de dados dos estudantes. Esta pode possuir como possíveis valores: “solteiro”, “casado”, “divorciado”, “outros”. Ao converter estes valores categóricos para números, não se pode criar uma sequência hierárquica, porque logicamente “solteiro” não é maior que “casado” e “casado” não é maior que “solteiro”. Em outras palavras, seria incorreto substituir o valor “solteiro” por 0 e “casado” por 1, pois, ao modelo, isso significaria que “casado” é maior que “solteiro”.

Uma conversão incorreta pode acabar afetando negativamente o desempenho do modelo. Logo, a maneira correta de realizar esta conversão é utilizando a técnica *One Hot Encoding*, que transforma cada possível valor de “estado_civil” em uma nova coluna. Outro atributo que foi feita a conversão com o uso da técnica *One Hot Encoding* foi o “tipo_raca_descricao” que descreve qual raça determinado estudante se auto declara ou pertence. Caso haja correspondência, a técnica atribui o valor 1 para a coluna específica e 0 para as outras colunas (POTDAR; PARDAWALA; PAI, 2017). A Figura 4 apresenta o exemplo sobre a técnica descrita.

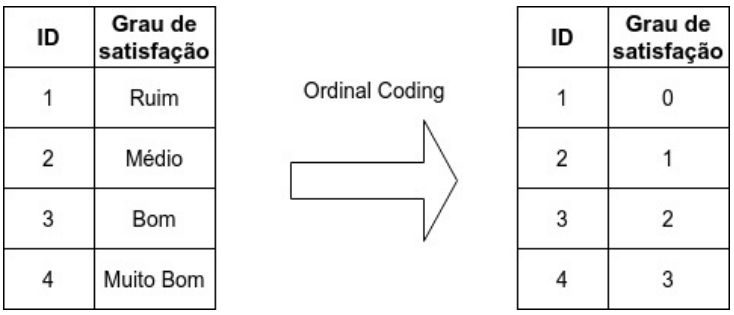
Figura 4 – Exemplo de uso da técnica *One Hot Encoding*



Fonte: Elaborada pelo autor.

O método *Ordinal Encoding* é usado para casos em que as variáveis estão em escala nominal quantitativa, ou seja, situações em que é possível estabelecer uma ordem ou sequência numérica. Como exemplo, suponha uma variável que possui os registros: “Ruim”, “Médio”, “Bom” e “Ótimo”, e que haja a necessidade de conversão para valores numéricos. Neste caso, utilizando *Ordinal Encoding*, pode-se observar na Figura 5 que os valores foram convertidos para uma escala, sendo “Ruim” convertido para o valor 0, “Médio’ para 1, “Bom” para 2 e “Muito bom” para 3, determinando assim uma sequência (POTDAR; PARDAWALA; PAI, 2017).

Figura 5 – Exemplo de uso do *Ordinal Encoding*



Fonte: Elaborada pelo autor.

Para o caso de valores do atributo-alvo (classe), é realizada apenas a etapa de transformação dos dados de categórico para numérica utilizando *Ordinal Encoding*. Isso se deve ao fato de que se trata de uma classificação binária (existindo apenas as classes “Evasão” e “Não evasão”). Assim, não é necessário criar uma nova coluna e sim substituir as respectivas classes para 0 ou 1.

4.2 Configuração do Treinamento

Overfitting é um dos problemas mais comuns e recorrentes na etapa de treinamento de algoritmos de *Machine Learning*. Este problema pode ocorrer após o treinamento de um modelo no qual foi obtido uma alta performance. Porém, ao aplicar dados desconhecidos ao modelo, o modelo tem uma performance muito abaixo do esperado. Com isto, é possível perceber que o modelo não tem poder de generalização. Informalmente, é como se o modelo tivesse “decorado” os dados de treinamento. Uma das formas de testar a capacidade de generalização de um modelo é através da separação dos dados em conjuntos de treinamento e teste. Os dados de teste são desconhecidos pelo modelo, que deve tentar classificar ou prever

estes dados desconhecidos. Assim, é possível avaliar seu desempenho por meio das métricas apropriadas (DOMINGOS, 2012).

A parte de treinamento do modelo é realizada pela separação dos dados em conjuntos de treinamento e teste, onde o modelo, após treinado, precisa classificar corretamente a parcela de dados que foi definida para teste. Por exemplo, se houver a definição de 70/30, 70% do *dataset* será utilizado para o treinamento do modelo e 30% será definido como teste para verificar as taxas de acerto daquele modelo recém-treinado. Os dados são misturados antes da divisão, de forma que a sequência dos dados não interfira no processo de treinamento.

Existem diversos tipos de separação dos dados em conjuntos de treinamento e teste, sempre representando uma proporção para o treinamento e teste. Podem ser citadas algumas estratégias usadas por pesquisadores, como 50/50, 60/40, 75/25, 70/30, 80/20, 90/10, validação cruzada, entre outras estratégias de separação. A mais comumente utilizada pelos pesquisadores é a separação dos dados de 70/30 (ANIFOWOSE; KHOUKHI; ABDULRAHEEM, 2017). A estratégia de separação utilizada por este trabalho será a de validação cruzada, a qual é recomendada para *datasets* pequenos, ou seja, aqueles que possuem um número reduzido de registros (BRAMER, 2007).

A validação cruzada é um processo de separação dos dados que tem por objetivo separar os dados em partes determinadas e avaliar a performance do classificador. Na validação cruzada tradicional, todo o dataset é dividido em K partes de tamanho igual que são também chamadas de *folds*, e estes vários *folds* são divididos (treino e teste) e pode-se avaliar por meio da média das métricas dos modelos gerados (BRAMER, 2007).

4.3 *Feature Engineering*

Na área de *Machine Learning* e Mineração de Dados, uma *feature* é definida como um atributo ou variável e é usada para descrever algum aspecto individual, para representar algum tipo de informação. Outros sinônimos para *features* são características e atributos. Como exemplo de *feature*, podem ser utilizados o estado civil, cor dos olhos, sexo, média de notas, entre outras (DONG; LIU, 2018).

Feature Engineering é um processo que usa do conhecimento do domínio para encontrar ou extrair variáveis úteis dos dados. Estas variáveis podem representar um ganho de informação para o modelo de *Machine Learning*. Elas podem ser úteis para distinguir e caracterizar diferentes grupos de elementos e também para melhorar o desempenho do modelo. Dentro deste universo, existe um conceito muito usado chamado *Feature Transformation* que possui como objetivo construir novas variáveis a partir de variáveis já existentes (DONG; LIU, 2018).

Alguns exemplos de *Feature Engineering* usados neste trabalho e que serão descritos em detalhes nas próximas seções são: o cálculo da idade a partir da data de nascimento; a média das aprovações do aluno nos últimos períodos cursados, representando o "status" do aluno; a média de faltas calculada de forma similar, dentre outros. Todas estas transformações têm por objetivo melhorar o desempenho e trazer informações em formatos aceitos pelo modelo.

4.4 Validação do modelo

Os modelos foram avaliados por meio de métricas de desempenho do algoritmo de *Machine Learning* para classificação. As métricas mais comuns para este tipo de tarefa são: acurácia, revocação, precisão, entre outras. Algumas outras técnicas podem ser utilizadas no auxílio da avaliação, como matriz de confusão, que mostra visualmente onde o modelo está classificando com mais taxas de erro e onde está

acertando mais. Como um exemplo desta situação, supondo um modelo para classificar cães e gatos, a matriz de confusão auxilia a verificar visualmente os casos como era para ser classificado como "cão" e foi classificado como "gato", e vice-versa. Em outras palavras, uma matriz de confusão auxilia na identificação de falsos-positivos e falsos-negativos, que podem ser extremamente prejudiciais a depender do problema a ser atacado. Uma explicação mais detalhada é fornecida na Subseção 4.4.1.

4.4.1 Matriz de Confusão

A matriz de confusão é uma tabela que permite visualizar de uma maneira mais clara a quantidade de erros e acertos de cada classe, indicando se um registro foi classificado corretamente ou não. As linhas da tabela correspondem as classificações corretas, e as colunas representam as classificações preditas pelo modelo.

No caso de um problema de classificação binária, a matriz de confusão terá duas linhas e duas colunas, dispondo os valores de verdadeiro-negativos e falso-positivos, e verdadeiro-positivos e falso-negativos. Na Tabela 6 é demonstrado um exemplo, onde na diagonal principal estão dispostos respectivamente os verdadeiro-positivos e verdadeiro-negativos como classificações corretas das classes; e na diagonal oposta há os falso-positivos e falso-negativos como uma classificação incorreta por parte do algoritmo. Levando em consideração o contexto de COVID-19 e supondo um modelo projetado para a classificação de casos de COVID-19, um falso-positivo está relacionado à pessoa não estar contaminada porém ser classificada como contaminada; um falso-negativo seria uma pessoa estar contaminada mas ser classificada como saudável.

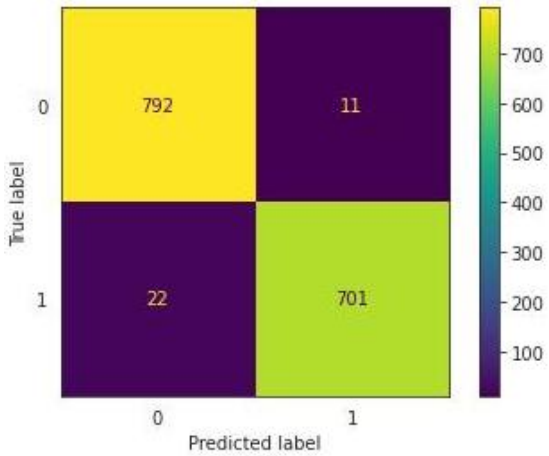
Figura 6 – Tabela explicativa sobre a Matriz de Confusão.

Classificação correta	Classificado como	
	+	-
+	Verdadeiro Positivo (VP)	Falso Negativo(FN)
-	Falso Positivo (FP)	Verdadeiro Negativo(VN)

Fonte: Bramer (2007)

No caso da utilização do *dataset* de evasão escolar, é importante encontrar os casos que tem potencial de evadir para reversão da tendência de evasão. Se o modelo classifica um estudante (que na realidade tem potencial de evadir) como não-evasor, haverá um falso-negativo. Isso é considerado um problema pois o modelo não estaria servindo ao principal propósito do modelo que é prever a evasão com o máximo de precisão possível. Além disso, o estudante passaria despercebido pelo modelo e não seria englobado por qualquer mecanismo que fosse implementado para reversão de evasão. Já no caso de um aluno não ter tendência de evasão e ser classificado como evasor, não haveria tanto problema, pois o estudante já não tem a tendência de evadir. Importante notar que falsos-positivos e falsos-negativos podem ter importâncias dependentes do problema atacado.

Figura 7 – Matriz de Confusão referente a predição de evasão escolar.



Fonte: Gerada a partir da biblioteca *Scikit-Learn* (PEDREGOSA et al., 2011).

Na Figura 7 pode-se observar um exemplo de uma matriz de confusão, assumindo 0 como a classe de "não-evasor" e 1 como a classe de "evasor". Nas colunas, 0 e 1 representam as classes que estão corretas. Nas linhas, 0 e 1 representam as classes preditas pelo modelo. Interpretando a imagem, o modelo classificou 792 estudantes como "não-evasores" corretamente, e 701 foram classificados como "evasores" do mesmo modo. Os falso-negativos e falso-positivos são apresentados na diagonal oposta, onde 11 estudantes não evadiram e foram classificados como "evasores" e 22 estudantes evadiram mas foram colocados como "não-evasores".

4.4.2 Métricas de desempenho

Para avaliar o desempenho do modelo são utilizadas métricas de desempenho de classificação, sendo as mais comuns: revocação, acurácia, precisão e *F1 Score*. Dependendo do contexto e aplicação essas métricas podem assumir outros nomes (BRAMER, 2007).

$$Acurácia = \frac{VP + VN}{T} \tag{4.1}$$

A Acurácia se refere à proporção de instâncias que foram classificadas corretamente no modelo em relação ao número total de instâncias (onde cada instância é um exemplo). Na fórmula de acurácia (Equação 4.1), VN se refere aos "Verdadeiros-Negativos" e T se refere ao total de instâncias que foram preditas pelo modelo (BRAMER, 2007). Importante notar que: (a) $T = VP + VN + FP + FN$, onde FN significa "Falsos-Negativos" e FP significa "Falsos-Positivos"; (b) falsos-negativos e falsos-positivos não são considerados no cálculo da Acurácia. Logo, quanto maior a proporção de erros (ou seja, quanto maior FP e FN), menor o valor da Acurácia. Porém, quanto maior o valor da Acurácia, melhor, pois significa mais acertos por parte do modelo.

$$Precisão = \frac{VP}{VP + FP} \tag{4.2}$$

A Precisão se propõe a responder a seguinte pergunta: "entre aqueles que foram classificados como de uma determinada classe, quantos realmente são?". Os significados dos termos foram mencionados anteriormente. A Precisão é calculada individualmente em relação à cada classe. Pela Equação, pode-se

perceber que, quando maior a existência de falso-positivos, menor a Precisão. Para a Precisão, obviamente, quanto maior o valor obtido, melhor.

$$Revocação = \frac{VP}{VP + FN} \tag{4.3}$$

A Revocação se refere à frequência com que o classificador encontra os exemplos de uma determinada classe. Na Equação 4.3, FN foi explicado anteriormente. A Revocação também é calculada em relação à cada classe. Através da Equação, pode-se perceber que, quanto menor a presença de falso-negativos, maior a Revocação. Logo, para Revocação, quanto maior o valor, melhor.

$$F1\_Score = \frac{2 * Precisão * Revocação}{Precisão + Revocação} \tag{4.4}$$

O F1_Score, demonstrado na Equação 4.4, é também conhecido como média harmônica, representa uma média que combina a precisão e revocação, usando as métricas anteriores no seu cálculo (BRAMER, 2007).

4.5 Desenvolvimento do Protótipo

Para o desenvolvimento do protótipo de predição da evasão escolar foram aplicadas técnicas de Rede Neural e Árvore de Decisão sobre os dados dos estudantes, utilizando a linguagem Python e um conjunto de bibliotecas para trabalhar com aprendizagem de máquina. Realizada a avaliação dos resultados, estes foram coletados e descritos na conclusão deste trabalho.

Para facilitar a utilização do modelo de *Machine Learning*, foi desenvolvida uma aplicação *Web* utilizando a biblioteca *Streamlit*[1]. As aplicações envolvendo esta biblioteca vão desde detecção de objetos em tempo real, aplicações geográficas, *debuggers* para Deep Learning, entre outros. Também é compatível com vários *frameworks* populares de ML, como Scikit Learn, Keras, Seaborn, PyTorch, e outros. Neste trabalho foram desenvolvidas interfaces gráficas intuitivas para facilitar a interação com o modelo de *Machine Learning* desenvolvido.

[1] https://www.streamlit.io/

5 COLETA E TRATAMENTO DOS DADOS

5.1 Coleta dos dados

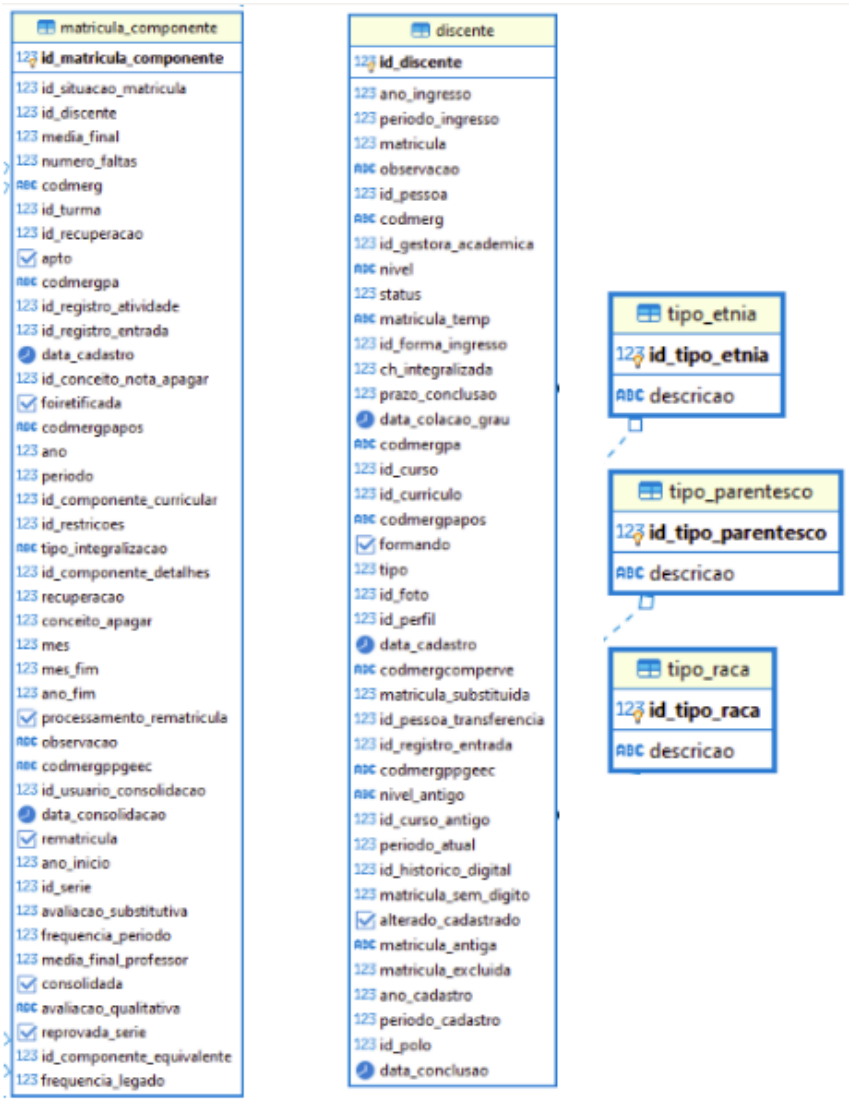
Nesta seção é descrita a etapa de coleta dos dados, isto é, como foi possível a obtenção do conjunto de dados. Este trabalho objetivou a obtenção dos dados institucionais dos estudantes dos cursos de graduação do IFSC - Câmpus Caçador, para posterior aplicação das etapas de KDD, como forma de auxiliar na predição da evasão escolar. Neste processo, diversas comunicações com diferentes setores do IFSC foram necessárias para obter a liberação e extração dos dados.

A seleção das colunas foi feita a partir do estudo dos artigos escolhidos no mapeamento sistemático, onde foram identificadas as informações mais utilizadas para a predição da evasão escolar. As informações selecionadas foram: nível de estudo e ocupação dos pais, sexo, idade, performance acadêmica no primeiro ano, frequência das aulas, data de matrícula, ano de inscrição, mora com (pais, sozinho), componentes da família (mãe, pai ou ambos), tipo de casa (própria, alugada, outra), trabalha atualmente (sim, não), renda familiar, entre outras. Algumas dessas informações não estavam disponíveis no banco de dados em que foram consultado os dados.

Após a seleção das colunas para servir de base, foi necessário identificar o processo para obtenção do *dataset* sobre os estudantes. A Coordenadoria de Informática e a Coordenação de Pesquisa do Câmpus foram contatados por e-mail, sugerindo o contato com a Diretoria de Estatística e Informações Acadêmicas (DEIA). Após duas semanas de espera, foi recebido um e-mail instruindo sobre os processos e justificando a demora no retorno devido a reestruturação que se passava no setor. O retorno da DEIA informou que a Pró-Reitoria de Pesquisa possui regulamentações sobre a utilização de dados institucionais em pesquisas, sendo necessário o encaminhamento de solicitação de pesquisa à Pró-Reitoria de Pesquisa, Pós-graduação e Inovação (PROPPI). A partir disso, foi enviado um novo e-mail à PROPPI solicitando informações sobre como proceder para obtenção dos dados. A resposta obtida é que havia sido iniciado um processo no Sistema Integrado de Patrimônio, Administração e Contratos (SIPAC) e era preciso aguardar uma autorização da Direção geral do Câmpus Caçador.

Obtida a autorização para realização da pesquisa e utilização dos dados, foi abordada a questão com vários setores do IFSC como secretária acadêmica, diretoria, coordenaria de pesquisa e outros sobre como proceder para conseguir os dados acadêmicos. Neste período, em contato com o coordenador da Diretoria de Tecnologia da Informação e Comunicação (DTIC), foi repassado o contato de um técnico especialista da área, o qual indicou a necessidade de consulta direta ao banco de dados institucional. Este profissional encaminhou os diagramas de entidade-relacionamento de quatro esquemas de dados: “comum”, “ensino”, “graduacao” e “public”. O Sistema Gerenciador de Banco de Dados (SGBD) utilizado pelo IFSC é o PostgreSQL e, após estudar estes diagramas, foram elencadas as informações mais importantes a serem extraídas dos vários bancos de dados e também organizadas as informações sobre as tabelas para entender como seria feita esta consulta.

Figura 8 – Trecho de um dos diagramas Entidade-Relacionamento.



Fonte: Elaborada pelo Autor.

A Figura 8 é um recorte de um dos diagramas de entidade-relacionamento que não está disponível em sua completude por motivos de segurança.

Por fim, foi desenvolvida um *script* SQL (Linguagem de Consulta Estruturada) para obtenção dos dados de acordo com os diagramas recebidos. Houveram várias versões deste *script* enviadas ao técnico especialista, que executava o *script* e retornava uma amostra do retorno para ser validada. Após quatro versões do *script* SQL os dados foram obtidos em um arquivo com formato '.csv', de onde se iniciou a exploração e tratamento dos dados. Na Figura 9, é exibido um trecho do *script* SQL desenvolvido para a consulta no banco de dados do IFSC.

Figura 9 – Trecho do *script* SQL para obtenção dos dados.

```
SELECT p.sexo,p.id_pessoa ,p.municipio_naturalidade_outro,
p.cep AS pessoa_cep,p.cidade,p.uf,p.segundograuanoconclusao,
p.data_nascimento,p.segundograucidade,
...
FROM comum.pessoa AS p
INNER JOIN public.discente AS d on p.id_pessoa = d.id_pessoa
LEFT JOIN comum.tipo_parentesco as tp on tp.id_tipo_parentesco =
p.id_tipo_parentesco
...
WHERE
    d.id_gestora_academica = 2217 -- ID do Campus Caçador
    and c.id_curso in (2399185,2399176) -- Filtro por curso
```

Fonte: Elaborada pelo autor.

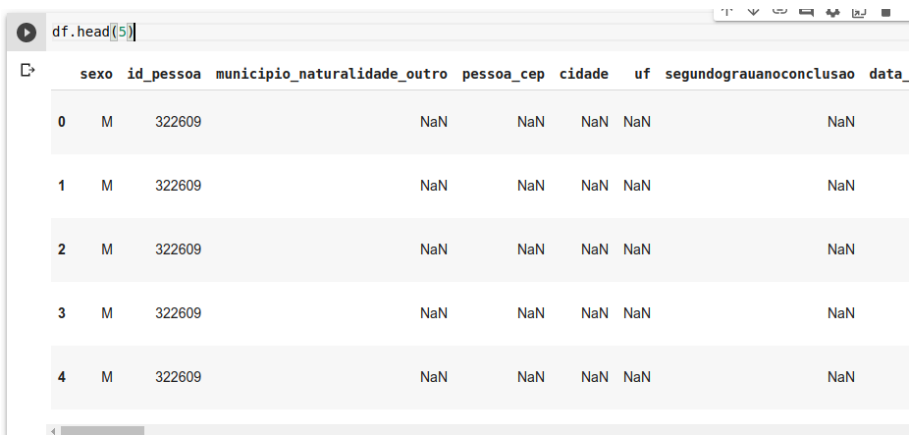
Estes foram os passos para a obtenção dos dados que vão desde solicitar a autorização até o desenvolvimento da consulta SQL. Na próxima seção serão abordados os ajustes e tratamentos realizados nos dados para prepará-los para a aplicação dos modelos de aprendizagem de máquina.

5.2 Tratamento dos dados

Esta seção aborda a etapa de tratamento de dados, que inclui a limpeza dos dados, transformação dos dados e *Feature engineering*. Em um contexto geral, para a manipulação, tratamento, leitura e visualização dos dados e para a aplicação de algoritmos de *Machine Learning*, foi utilizada a linguagem de programação Python em conjunto com as bibliotecas Pandas[1] e Scikit-Learn[2].

A etapa de tratamento dos dados consistiu em realizar transformações nos mesmos, substituição de dados faltantes, realização de *feature transformation* e verificação de balanceamento dos dados. Na Figura 10 pode-se observar um trecho do estado inicial dos dados recebidos anterior ao tratamento realizado. Como pode-se notar, existia uma quantidade excessiva de valores NaN (*not a number*) em algumas colunas, o que foi corrigido posteriormente.

Figura 10 – Estado inicial dos dados.



```
df.head(5)
```

	sexo	id_pessoa	municipio_naturalidade_outro	pessoa_cep	cidade	uf	segundograuanoconclusao	data_
0	M	322609	NaN	NaN	NaN	NaN	NaN	
1	M	322609	NaN	NaN	NaN	NaN	NaN	
2	M	322609	NaN	NaN	NaN	NaN	NaN	
3	M	322609	NaN	NaN	NaN	NaN	NaN	
4	M	322609	NaN	NaN	NaN	NaN	NaN	

Fonte: Elaborada pelo autor.

A primeira questão observada ao ter uma pré-visualização dos dados foi a existência de dois

[1] https://pandas.pydata.org/
[2] https://scikit-learn.org/stable/

problemas: dados desagrupados e repetições de informações sobre os alunos. Inicialmente, tomou-se o cuidado de não agrupar os dados, pois o agrupamento poderia causar a perda de informações, como por exemplo as notas por semestre.

O *dataset* inicialmente possuía 50 colunas e 70786 registros. Foi verificado que existiam muitas colunas que não possuíam nenhum registro, e após a exclusão destas, restaram 34 colunas. Informações como os tipo de movimentação e semelhantes foram excluídas do conjunto de dados devido ao fato de estarem atrapalhando os agrupamentos. Na Tabela 5 pode-se verificar as 34 colunas restantes, e os casos de dados faltantes que são todos aqueles que tem menos de 70786 possuem algum dado que é representado no Python como NaN.

Tabela 5 – Colunas presentes no *Dataset* obtido.

Nome da coluna	**Nº Registros**	**Tipo**
sexo	70786	object
id_pessoa	70786	int64
pessoa_cep	3216	float64
segundograuanoconclusao	51557	float64
data_nascimento	70027	object
segundograucidade	28910	float64
tipo_raca_descricao	70785	object
estado_civil_descricao	53995	object
tipo_parentesco_descricao	8015	object
endereco_contato_cep	70786	object
endereco_residencial_cep	33605	object
ano_ingresso	70786	int64
periodo_ingresso	70786	int64
matricula	70784	float64
prazo_conclusao	70718	float64
ano_cadastro	46871	float64
curso_nome	70786	object
status_discente_descricao	70786	object
forma_ingresso_descricao	70786	object
forma_ingresso_nivel	70786	object
forma_ingresso_realiza_processo_seletivo	70786	bool
media_final	51599	float64
numero_faltas	55249	float64
ano	67273	float64
periodo	67273	float64
rematricula	28058	object
frequencia_legado	2075	float64
ano_inicio	28	float64
componente_curricular_detalhes_nome	70751	object
situacao_matricula_descricao	70751	object
situacao_matricula_ativo	70751	object
matricula_valida_no_semestre	70751	object
solicitacao_trancamento_matricula_situacao	893	float64
solicitacao_trancamento_matricula_justificativa	128	object

Fonte: Elaborada pelo autor.

A etapa de limpeza de dados é uma etapa que consiste em eliminar ruídos, dados duplicados ou inseridos incorretamente, remoção de redundâncias, tratamento de valores faltantes, criação de padronização para o algoritmo de *Machine Learning*, entre outros. Um problema observado nos dados é o caso de disciplinas que armazenavam informação equivalente porém com valores distintos.

Observando os nomes de todas as disciplinas presentes no banco de dados, existiam algumas que estavam com a mesma nota, no mesmo período e do mesmo aluno, mas em algumas linhas estavam com nomes diferentes (porém equivalentes), representando a mesma disciplina. Isso pode ser ruim, pois, ao realizar a operação de agrupamento para obter algum dado como a soma ou a média, as disciplinas (embora sejam as mesmas) serão consideradas como duas disciplinas diferentes, podendo impactar negativamente no tratamento dos dados. Na Tabela 6, as disciplinas que estavam com o valor "METODOLOGIA CIÊNTÍFICA" foram mudadas para "METODOLOGIA CIENTÍFICA", "COE38101" alteradas para "COMUNICAÇÃO E EXPRESSÃO I", "GESTÃO PARA SUSTENTABILIDADE" alteradas para "GESTÃO DA SUSTENTABILIDADE". Com isto uma padronização na coluna "componente_curricular_detalhes_nome" foi obtida.

Tabela 6 – Padronização nas nomenclaturas das disciplinas.

Como estava	**Alterado para**
METODOLOGIA CIÊNTÍFICA	METODOLOGIA CIENTÍFICA
COE38101	COMUNICAÇÃO E EXPRESSÃO I
GESTÃO PARA SUSTENTABILIDADE	GESTÃO DA SUSTENTABILIDADE

Fonte: Elaborada pelo autor.

As novas colunas que foram criadas a partir de *Feature Engineering* foram: (a) a idade, gerada a partir da coluna de data de nascimento; (b) após a contabilização das colunas em que o aluno foi aprovado e reprovado por meio do agrupamento em id, ano e período foi criada a coluna "status" que é calculada pelo número de disciplinas em que o aluno foi aprovado, dividido pela total de disciplinas que o aluno cursou no semestre; (c) as colunas "penultimo_status" e "ultimo_status" tem seu valor recebido do "status" no penúltimo e último semestres do aluno respectivamente; (d) a média do "status" que se refere a média geral do "status' no qual o cálculo foi descrito anteriormente; (e) "diferenca_anos", que contabiliza a diferença entre o ingresso no ensino superior e o término do ensino médio em anos; (f) onde foi feito um agrupamento pelo id, ano e período e calculado a média a partir do número de faltas, para posteriormente calcular a média no penúltimo e último semestre, gerando as colunas "penultimo_faltas" e "ultimo_faltas". Ainda poderiam ser geradas outras colunas, porém devido a falta de muitos registros (isto pode gerar um modelo que não representa muito a realidade) foi optado por não serem criadas ou usadas algumas das informações presentes no *dataset*.

Quando uma coluna tem todos os seus valores nulos não é possível reaproveitar ou aplicar uma técnica de correção de limpeza de dados. Porém quando apenas alguns registros estão vazios ou NaN (*Not a Number*), o tratamento destes valores faltantes é possível. Para tratar o caso dos dados NaN, algumas técnicas comumente utilizadas são: substituição pela mediana dos valores da coluna, substituição pela média, predição dos valores por meio de algoritmos de regressão, exclusão dos registros, entre outras (GARCIA; LEITE; SKRJANC, 2019). Para este trabalho, especificamente nas colunas "idade", "penultimo_status", "penultimo_faltas" e "diferenca_anos", os valores NaN foram substituídos pela média.

A obtenção da informação que indica se uma pessoa evadiu ou não foi feita por meio da coluna de "status" do discente, com o nome de "status_discente_descricao" que possui os seguintes valores possíveis: "ATIVO", "TRANCADO", "ATIVO - FORMANDO", "DESATIVADO", "CANCELADO", "EXCLUÍDO". Como é necessário uma coluna relativa à evasão (i.e. classe-alvo), contendo os valores "SIM" se o aluno evadiu e "NAO" se ele não evadiu, foi necessário criar uma nova coluna a partir de "status_discente_descricao". Para definir o valor para a nova coluna "evadiu", foi utilizado o seguinte critério: onde o status do discente for "ATIVO" ou "ATIVO - FORMANDO", o valor recebido é "NAO";

enquanto para as outras situações, o valor recebido é “SIM”.

Os algoritmos de *Machine Learning* selecionados (Árvore de Decisão e Rede Neural Artificial) possuem restrições, sendo necessário realizar uma transformação nos dados de categórico para numérico. Neste caso, para a rede neural é necessário converter todos as colunas categóricas para numéricas. As árvores de decisão necessitam que apenas as colunas sem a classe sejam convertidas para valores numéricos.

6 RESULTADOS

Neste capítulo serão descritos os modelos de *Machine Learning* implementados para o problema proposto neste trabalho, além de sumarizar os principais resultados obtidos.

6.1 Modelo

Após a definição das colunas a serem utilizadas pelo modelo e depois da etapa de *Feature engineering* – que adiciona novas colunas também –, o próximo passo executado foi o de agrupar os dados para cada linha representar apenas um aluno, auxiliando para o treinamento do modelo. Na etapa seguinte foi criada uma cópia dos dados para o *dataset* denominado "df_alunos". Neste *dataset* final foi preciso ainda realizar algumas transformações, como o tratamento dos dados faltantes, conversão de dados de categóricos para numéricos.

Por fim, foi realizada a etapa de validação do modelo, visando avaliar se as métricas de desempenho estão em conformidade com um bom resultado. Todos estes passos compreenderam o processo de KDD, que funciona de forma iterativa e incremental, ou seja, algumas das etapas anteriores foram executadas mais de uma vez, de modo a aprimorar os resultados oferecidos pelo modelo desenvolvido.

Após o agrupamento dos dados, restaram 12 colunas com informações sobre alunos e as disciplinas cursadas por eles. Como os dados são anonimizados, cada aluno é representado por um identificador numérico, o que permitiu o agrupamento de informações mantendo a integridade dos dados. Ao final, o *dataset* englobou 380 registros, sendo que cada linha representa um único aluno. Na Tabela 7 são apresentadas as colunas com informações contidas no *dataset* final. Os tipos int64 se referem a variáveis do tipo inteiro, os tipos float64 se referem ao conjunto dos números reais e o tipo object se referem a objetos de classe como *String* e outros.

Tabela 7 – Tabela com as informações após tratamento dos dados.

Nome da coluna	**Nº Registros**	**Tipo**
id_pessoa	380	int64
status	380	float64
ultimo_status	380	float64
penultimo_status	380	float64
media_status	380	float64
penultimo_faltas	380	float64
ultimo_faltas	380	float64
estado_civil_descricao	380	object
diferenca_anos	380	float64
evadiu	380	int64
idade	380	float64
tipo_raca_descricao	380	object

Fonte: Elaborado pelo autor.

As colunas "estado_civil_descricao" e "tipo_raca_descricao" precisaram ser convertidas para numéricas por meio da técnica *One Hot Enconding*, conforme comentado na Seção 5.2 sobre tratamento de dados, devido ao fato dos algoritmos de Árvore de Decisão e Rede Neural só aceitarem dados com tipos numéricos. No caso da Rede Neural, para o atributo-alvo (classe) com o caso de classificação binária,

também foi necessária a conversão de categórica para numérica, definindo 0 como indicativo da "não-evasão" e 1 para o caso do aluno ter se evadido ("evasão"). Os casos de dados faltantes foram substituídos pela média nas colunas "penultimo_status", "penultimo_faltas", "idade" e "diferenca_anos".

6.2 Evolução do modelo - Resultados

Inicialmente, para a etapa de treinamento e testes dos modelos, foi adotada a estratégia de separação dos dados 70/30, onde as colunas utilizadas foram: (a) "status", calculada pela taxa de aprovação do estudante durante todo seu tempo no curso, representado pela seguinte fórmula: $status = \frac{disciplinas\_aprovadas}{disciplinas\_cursadas}$; (b) "penultimo_status", que tem seu valor calculado semelhante ao "status", porém relativo apenas ao penúltimo semestre ativo do aluno; (c) "ultimo_status", que tem seu valor calculado semelhante ao "status", porém relativo apenas ao último semestre ativo do aluno; e (d) "media_status", que se refere à taxa de aprovação média por semestre. E o atributo-alvo denominado "evadiu", que se refere à classe. As seções seguintes abordarão a evolução das métricas resultantes da execução dos modelos para os algoritmos de Rede Neural e Árvore de Decisão.

6.2.1 Rede Neural

Os parâmetros estabelecidos para a Rede Neural são os parâmetros padrão do Scikit Learn, com exceção do *random_state*, que recebe o valor 12 para que os resultados se mantenham iguais a cada execução. Qualquer valor poderia ter sido utilizado para este parâmetro, desde que não fosse *None*, o que tornaria os resultados não-reproduzíveis. Outro parâmetro que foi definido foi o número de iterações (ou épocas) para o algoritmo de *Machine Learning* convergir o modelo (*max_iter*). Foram feitos alguns testes com valores maiores e não foram obtidos resultados melhores. Com isto, foi definido este número de iterações. Na Figura 11 é possível observar a criação do objeto da Rede Neural, onde a variável que está armazenando está com o nome de "mlp".

Figura 11 – Criação do classificador que utiliza Rede Neural.

```
mlp = MLPClassifier(max_iter= 1000, random_state = 12)
```

Fonte: Elaborado pelo autor

Após a criação do objeto do classificador, foi realizado o treinamento do modelo utilizando os dados que foram previamente separados para esta etapa. Na Figura 12 pode-se observar o treinamento do modelo com o método "fit", passando como parâmetros o "x_treino_d" e "y_treino_d".

Figura 12 – Treinamento da Rede Neural com o conjunto de dados definido para treino.

```
mlp.fit(x_treino_d,y_treino_d)
```

Fonte: Elaborado pelo autor

Logo após ao treinamento, foi aplicado ao modelo um conjunto de dados de teste (sem as respectivas classes), para avaliar se o modelo é capaz de predizer a evasão/não-evasão de alunos com dados desconhecidos. A Figura 13 apresenta esta etapa em código em Python com o método *predict*, sendo passados os dados de teste e armazenados na variável "y_val".

Figura 13 – Predição da Rede Neural com o conjunto de dados definido para teste.

```
y_val = mlp.predict(x_teste_d)
```

Fonte: Elaborado pelo autor

Para avaliar o desempenho do modelo inicial, que utilizou-se das quatro colunas mencionadas acima, foi realizada a comparação entre os valores preditos pelo modelo para com os valores esperados. A seguir, a Tabela 8 apresenta todas as execuções (com adições e alterações em colunas do *dataset*) utilizando Rede Neural, visando melhorar o desempenho das métricas. É importante obter um equilíbrio entre a precisão e a revocação, pois ter uma precisão alta e ignorar a revocação – que leva em consideração a quantidade de falsos-negativos e falsos-positivos – pode levar a uma avaliação incorreta do modelo.

Tabela 8 – Tabela com a evolução das métricas da Rede Neural

Nº execução	Coluna alteradas ou adicionadas	Precisão	Revocação	Acurácia	Classe
1	Primeira execução	0.71	0.76	0.69	"Não evadiu"
		0.67	0.61		"Evadiu"
2	Adição da "idade"	0.77	0.81	0.76	"Não evadiu"
		0.75	0.71		"Evadiu"
3	Adição de "estado_civil" e "raça"	0.80	0.76	0.76	"Não evadiu"
		0.72	0.76		"Evadiu"
4	Adição de "diferenca_anos"	0.85	0.81	0.82	"Não evadiu"
		0.78	0.82		"Evadiu"
5	Adição de "penultima_faltas" e "ultima_faltas"	0.77	0.95	0.82	"Não evadiu"
		0.92	0.65		"Evadiu"
6	Remoção da coluna "raça"	0.82	0.86	0.82	"Não evadiu"
		0.81	0.76		"Evadiu"

Fonte: Elaborado pelo autor.

A partir do resultado da primeira execução, constatou-se a necessidade de realizar algumas modificações no conjunto de dados com o intuito de melhorar a precisão e revocação na classe de evasores, uma vez que esta é a classe mais importante no problema deste trabalho. A primeira coluna que foi adicionada para verificar se havia uma mudança sobre o modelo foi a coluna da idade dos estudantes, junto da separação de dados em treinamento e teste para 90/10 devido ao pequeno volume de dados. A próxima adição de coluna foi sobre o estado civil e a descrição da raça do aluno, o que acabou melhorando a precisão da classe dos "não-evasores" e a métrica de revocação em prever os alunos "evasores".

A adição da "diferenca_anos", que é definida pela diferença de anos entre o ano de ingresso na instituição e o ano de conclusão do ensino médio, apresentou uma melhora considerável para o modelo, o fazendo acertar mais. Isto poder ser verificado ao observar o valor obtido de acurácia.

As últimas colunas adicionadas e validadas foram as colunas que representam média de faltas do último e penúltimo período. O objetivo ao adicionar estas duas colunas foi detectar tendência de crescimento de número de faltas. Com isto, houve uma melhora na taxa de erro da Rede Neural e também nas métricas, porém acabou deixando as métricas entre as duas classes de evasão ou não desequilibradas, como a revocação da classe dos "evasores" baixa. A última mudança foi retirar a coluna que representa raça, obtendo um modelo mais equilibrado entre precisão e revocação e com a mesma acurácia.

Uma preocupação constante ao se trabalhar com modelos de *Machine Learning* é a possibilidade de *overfitting*, situação em que o modelo é incapaz de generalizar os resultados para dados não vistos

na fase de treinamento, principalmente em *datasets* com pequeno volume de dados, como é o caso deste trabalho. Por esta razão, optou-se por utilizar também a técnica de validação cruzada, que divide os dados em vários blocos, aplicando o modelo diversas vezes e calculando a média entre as principais métricas para a validação. Isto difere das divisões tradicionais de 70/30, 80/20 ou 90/10, que (geralmente) aplicam a divisão de forma aleatória uma única vez, o que pode influenciar no treinamento do modelo em termos de *overfitting* ou *underfitting*. Considerando a validação cruzada, com a técnica de Redes Neurais foi possível obter uma precisão de 85%, com uma média de revocação de 72% e média de acurácia de 81% . Usando o *dataset* preparado neste trabalho baseado nas colunas utilizadas na sexta execução que obteve os melhores resultados.

O melhor modelo gerado neste trabalho para previsão de potenciais casos de evasão é o da quarta execução devido à sua revocação ser melhor do que a dos outros casos, e possuir uma acurácia igual a da quarta, quinta e sexta execução de acordo com a tabela de evolução das métricas. Este modelo é o modelo utilizado pelo protótipo.

6.2.2 Árvore de Decisão

Os parâmetros utilizados para a Árvore de Decisão são os parâmetros que vêm por padrão na biblioteca do Scikit Learn, com exceção do parâmetro *random_state*. A seguir é mostrado a evolução das métricas da Árvore de Decisão. Na Figura 14, é possível observar a criação do objeto Árvore de Decisão, onde a variável que está armazenando está com o nome de "clf".

Figura 14 – Árvore de Decisão com os parâmetros

```
clf = DecisionTreeClassifier(random_state= 3)
```

Fonte: Elaborado pelo autor

A próxima etapa após a configuração da Árvore de Decisão, acontece o treinamento com o conjunto de dados de treino definidos na separação do *dataset*. Na figura 15 pode-se observar o treinamento do modelo com a chamada do método *fit* passando como parâmetros o "x_treino_a" e "y_treino_a"

Figura 15 – Treino da Árvore de Decisão

```
clf = clf.fit(x_treino_a, y_treino_a)
```

Fonte: Elaborado pelo autor

Depois da etapa de treinamento, foi repassado ao classificador da Árvore de Decisão "clf" o conjunto de dados de teste sem a classe, para ele predizer a classe dos dados desconhecidos. Na Figura 16, pode-se visualizar esta etapa em código em Python com a chamada ao método *predict*, sendo passados os dados de teste armazenados na variável "resultado_arvore".

Figura 16 – Predição da Árvore de Decisão

```
resultado_arvore = clf.predict(x_teste_a)
```

Fonte: Elaborado pelo autor

Na Tabela 9, são listadas todas execuções junto das modificações no conjunto de dados e os

valores das métricas de desempenho obtidas ao longo das alterações. Estão também a precisão e revocação para cada classe e a acurácia referente ao modelo.

Tabela 9 – Tabela com a evolução das métricas na Árvore de Decisão

Nº execução	Coluna alteradas ou adicionadas	Precisão	Revocação	Acurácia	Classe
1	Primeira execução	0.69	0.68	0.66	"Não evadiu"
		0.62	0.63		"Evadiu"
2	Adição da "idade"	0.65	0.71	0.63	"Não evadiu"
		0.60	0.53		"Evadiu"
3	Adição de "diferenca_anos"	0.74	0.81	0.74	"Não evadiu"
		0.73	0.65		"Evadiu"
4	Adição de "penultima_faltas" e "ultima_faltas"	0.82	0.86	0.82	"Não evadiu"
		0.81	0.76		"Evadiu"
5	Adição de "estado_civil" e "raça"	0.82	0.86	0.82	"Não evadiu"
		0.81	0.76		"Evadiu"
6	Remoção da "raça"	0.83	0.90	0.84	"Não evadiu"
		0.87	0.76		"Evadiu"

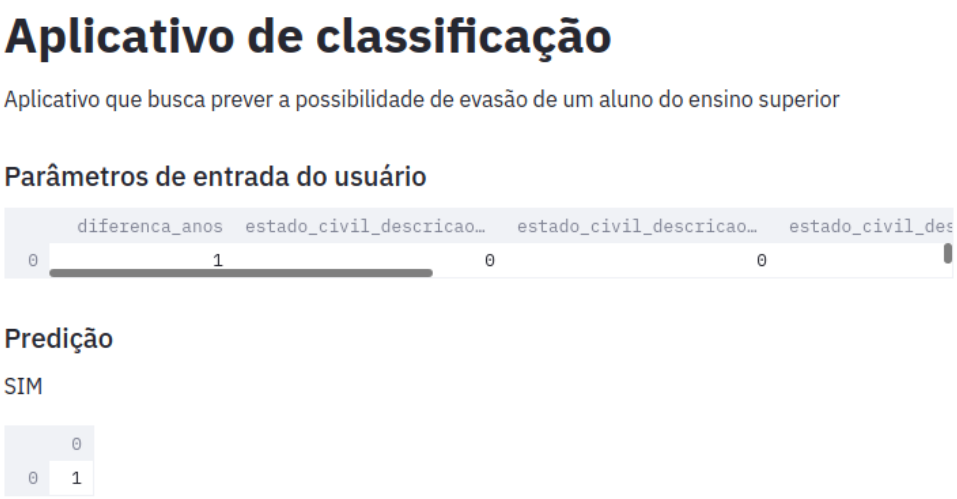
Fonte: Elaborado pelo autor

Na Tabela 9, pode-se verificar a primeira execução com as mesmas cinco primeiras colunas mencionadas anteriormente, onde estão várias métricas que são geradas pela biblioteca do *Scikit-Learn*. Ao observar os resultados obtidos com esta execução pode-se notar que obteve-se uma precisão e uma revocação baixos, necessitando buscar maneiras de melhorar o desempenho destas métricas. Nas próximas execuções para a Árvore de Decisão, foram adicionadas as colunas de "idade", "diferenca_anos", "penultima_faltas" e "ultima_faltas" como citado anteriormente na seção da Rede Neural houve uma progressão a cada adição de colunas nas métricas. Com exceção do caso do estado civil e raça juntos que não obtiveram uma melhora na performance do modelo.

A última modificação que foi a remoção da coluna com os dados de raça dos alunos e houve uma grande melhora, gerando o melhor modelo entre todas as execuções como pode-se verificar pelas métricas com uma precisão, revocação e acurácia maiores do que os outros e este foi o modelo escolhido para este trabalho. Utilizando a validação cruzada para o modelo com as colunas da sexta execução, a média de precisão dos modelos gerados na Árvore de Decisão é de 77%, a média de revocação é 74% e a média de acurácia é de 78%.

Na Figura 17, é mostrado o aplicativo desenvolvido no *StreamLit* que busca prever a possibilidade de evasão de um estudante de cursos superiores do Instituto Federal de Santa Catarina utilizando técnicas de *Machine Learning*.

Figura 17 – Conteúdo do StreamLit dizendo se o aluno evadiu

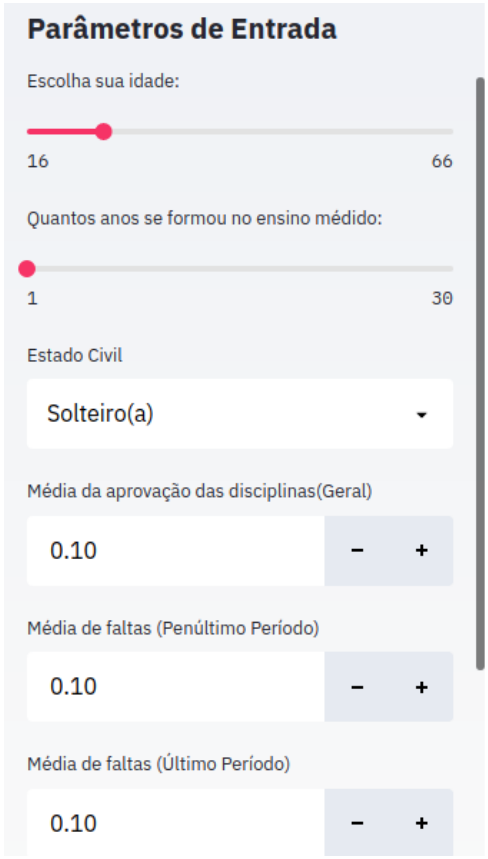


Fonte: Elaborada pelo autor.

Este aplicativo permite visualizar de forma gráfica o resultado, como visto na imagem os "Parâmetros de entrada do usuário", neste caso ele mostra os parâmetros selecionados ou digitados pelo usuário com os seus respectivos valores. Também pode exibir informações sobre a probabilidade daquelas informações serem determinada classe, exibir as classes e principalmente da a resposta do algoritmo para a questão do algoritmo se o aluno evadiu ou não.

Na Figura 18 está a parte da interface *Web* em que os parâmetros de entrada para o modelo são definidos. Estes campos possuem uma interatividade maior, como barra para ajuste de valores com possibilidade de digitá-los. É necessário preencher todos os campos para que o modelo possa realizar a predição.

Figura 18 – Menu Lateral para selecionar as opções

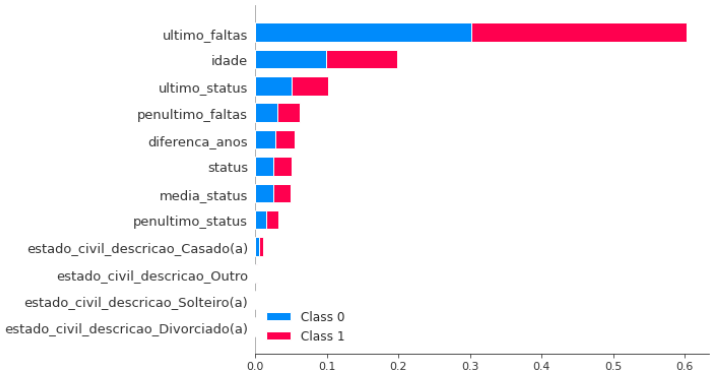


Fonte: Elaborada pelo autor.

A Figura 19 foi gerada através da biblioteca *Shap*[1]. Esta biblioteca auxilia na explicação do modelo de Árvore de Decisão, exibindo de forma gráfica as colunas mais importantes e como elas impactam cada classe. A informação que mais afetou positivamente o modelo para ambas as classes foi a de "ultimas_faltas", seguida pelas colunas em grau de importância: "idade" e "ultimo_status". As duas colunas com maior contribuição é a idade do aluno e o número de faltas do último semestre, e as que possuem menos são alguns tipos de estado civil que foram definidos como "Solteiro(a)", "Divorciado(a)" e "Outros" que possuem pouca ou zero contribuição. A utilização da biblioteca *Shap* apenas com o modelo de Árvore de Decisão se deve ao fato de modelos gerados por esta técnica serem explicáveis por padrão. Modelos de Rede Neural são conhecidos por serem "caixa-preta", significando que são necessárias outras técnicas para obter uma explicação do modelo.

[1] https://shap.readthedocs.io/en/latest/

Figura 19 – Gráfico com importância das colunas da Árvore de Decisão



Fonte: Elaborada pelo autor.

Como pode-se observar na Tabela 10, o segundo modelo (Árvore de Decisão) obteve performance melhor considerando a acurácia de 84% e a precisão 87%, enquanto o primeiro (Rede Neural) obteve uma revocação melhor, sendo 82% da Rede Neural em comparação com 78% da Árvore de Decisão. Em relação às médias dos modelos obtidas com a validação cruzada, a Rede Neural obteve precisão e acurácia maiores e revocação menor.

Tabela 10 – Desempenho dos melhores modelos da Rede Neural e Árvore de Decisão

Rede Neural				Árvore de Decisão			
Nº Execução	Precisão	Revocação	Acurácia	Nº Execução	Precisão	Revocação	Acurácia
4	0.78	0.82	0.82	6	0.87	0.78	0.84

Fonte: Elaborado pelo autor.

Neste capítulo foram comentados os resultados da aplicação das técnicas de Rede Neural e Árvore de Decisão. A técnica de Árvore de Decisão mostrou-se melhor com a configuração da sexta execução. As colunas utilizadas neste modelo foram: “diferenca_anos”, “estado_civil_descricao_Casado(a)”, “estado_civil_descricao_Divorciado(a)”, “estado_civil_descricao_Outro”, “estado_civil_descricao_Solteiro(a)”, “idade”, “media_status”, “penultimo_faltas”, “penultimo_status”, “status”, “ultimo_faltas”, “ultimo_status”. As características que mais contribuíram para auxiliar na predição de provável evasão foram a média de faltas do último semestre, a idade e a média de aprovação no último semestre. De posse destas informações, o setor pedagógico pode definir ações de forma a, após identificada a probabilidade de evasão, tentar evitá-la.

7 CONCLUSÕES

O principal objetivo deste trabalho foi investigar as possibilidades para oferta de um protótipo com um modelo computacional para predição de potenciais casos de evasão, onde as técnicas aplicadas foram a Rede Neural e a Árvore de Decisão. A aplicação do modelo para um contexto institucional é importante devido ao fato das características locais do Instituto Federal de Santa Catarina - Câmpus Caçador e dos seus cursos. Um ponto importante sobre o modelo é que ele ajuda a entender (por meio dos explicadores dos modelos) as variáveis mais importantes para este cenário, além da ferramenta desenvolvida para auxiliar no problema da evasão.

A hipótese de pesquisa, – a criação de um protótipo com um modelo computacional usando redes neurais para predição de evasão de alunos de graduação no Câmpus Caçador –, foi verificada e pode-se dizer que é possível. Porém, para o problema deste trabalho, a técnica que ofereceu melhores resultados foi a Árvore de Decisão. Desta forma, foi desenvolvida uma ferramenta utilizando o modelo que apresentou a melhor acurácia para prever casos de evasão (Árvore de Decisão). Devido ao fato do baixo volume de dados e complexidade do problema, o modelo não conseguiu se aproximar dos 100%.

Os objetivos deste trabalho foram cumpridos, criando um protótipo com um modelo para prever a potencial evasão (ou não) dos estudantes por meio dos dados institucionais, que pode (futuramente) apresentar melhores resultados, ao incorporar novos dados históricos.

Este trabalho possui abertura para trabalhos futuros partindo do que foi desenvolvido, reaproveitando as variáveis utilizadas, o modelo de predição, as novas colunas criadas, etc. Além disso, é possível realizar a aplicação de novos algoritmos, contrastando os resultados nas diversas etapas apresentadas por este trabalho. Outra contribuição importante no âmbito da instituição foi a criação da consulta SQL nos diversos esquemas de bancos de dados e a organização das informações resultantes em um *dataset* que também pode servir de base para novos trabalhos. Como última contribuição deste trabalho, há a ideia das colunas que mais ajudaram o modelo a classificar os casos de evasão, que seria a média de faltas no último período e a idade.

REFERÊNCIAS

ALBAN, M.; MAURICIO, D. Predicting university dropout through data mining: a systematic literature. *Indian Journal of Science and Technology*, v. 12, n. 4, p. 1–12, 2019. Citado 2 vezes nas páginas 16 e 20.

ALPAYDIN, E. *Introduction to machine learning*. [S.l.]: MIT press, 2020. Citado na página 20.

ANIFOWOSE, F.; KHOUKHI, A.; ABDULRAHEEM, A. Investigating the effect of training–testing data stratification on the performance of soft computing techniques: an experimental study. *Journal of Experimental & Theoretical Artificial Intelligence*, Taylor & Francis, v. 29, n. 3, p. 517–535, 2017. Citado na página 30.

BRAMER, M. *Principles of data mining*. [S.l.]: Springer, 2007. v. 180. Citado 7 vezes nas páginas 14, 19, 20, 30, 31, 32 e 33.

BRASIL. Constituição (1988). *Constituição da República Federativa do Brasil*. Brasília, DF: Senado, 1988. Citado na página 17.

CRAWFORD, S. L. Extensions to the cart algorithm. *International Journal of Man-Machine Studies*, Elsevier, v. 31, n. 2, p. 197–217, 1989. Citado na página 22.

DEKKER, G. W.; PECHENIZKIY, M.; VLEESHOUWERS, J. M. Predicting students drop out: A case study. *International Working Group on Educational Data Mining*, ERIC, 2009. Citado na página 14.

DIGIAMPIETRI, L. A.; NAKANO, F.; LAURETTO, M. de S. Mineraçao de dados para identificaçao de alunos com alto risco de evasao: Um estudo de caso. *Revista de Graduação USP*, v. 1, n. 1, p. 17–23, 2016. Citado na página 14.

DOMINGOS, P. A few useful things to know about machine learning. *Communications of the ACM*, ACM New York, NY, USA, v. 55, n. 10, p. 78–87, 2012. Citado na página 30.

DONG, G.; LIU, H. *Feature engineering for machine learning and data analytics*. [S.l.]: CRC Press, 2018. Citado na página 30.

DSA. *Deep Learning Book*. 2020. <http://deeplearningbook.com.br/algoritmo-backpropagation-parte1-grafos-computacionais-e-chain-rule/>. Citado 2 vezes nas páginas 21 e 22.

ENKE DAVID E THAWORNWONG, S. The use of data mining and neural networks for forecasting stock market returns. *Expert Systems with applications*, Elsevier, v. 29, n. 4, p. 927–940, 2005. Citado na página 14.

FAYYAD, U. M. et al. Knowledge discovery and data mining: Towards a unifying framework. In: *KDD*. [S.l.: s.n.], 1996. v. 96, p. 82–88. Citado na página 18.

FILHO, R. L. L. S. et al. A evasão no ensino superior brasileiro. *Cadernos de pesquisa*, SciELO Brasil, v. 37, n. 132, p. 641–659, 2007. Citado na página 17.

GARCIA, C.; LEITE, D.; SKRJANC, I. Incremental missing-data imputation for evolving fuzzy granular prediction. *IEEE Transactions on Fuzzy Systems*, IEEE, 2019. Citado 2 vezes nas páginas 19 e 38.

GOLDSCHMIDT, R.; BEZERRA, E.; PASSOS, E. Data mining: conceitos, técnicas, algoritmos, orientações e aplicações. *Rio de Janeiro-RJ: Elsevier*, p. 56–60, 2015. Citado na página 14.

GRUNSPAN, D.; WIGGINS, B.; GOODREAU, S. Understanding classrooms through social network analysis: A primer for social network analysis in education research. *CBE Life Sciences Education*, v. 13, p. 167–178, 01 2014. Citado na página 15.

GURNEY, K. *An introduction to neural networks*. [S.l.]: CRC press, 1997. Citado na página 21.

HAN, J.; PEI, J.; KAMBER, M. *Data mining: concepts and techniques*. [S.l.]: Elsevier, 2011. Citado 5 vezes nas páginas 14, 18, 19, 20 e 22.

INEP, c. Censo da educação superior 2019. *Brasília: MEC*, 2019. Citado na página 17.

JIN, C.; DE-LIN, L.; FEN-XIANG, M. An improved id3 decision tree algorithm. In: IEEE. *2009 4th International Conference on Computer Science & Education*. [S.l.], 2009. p. 127–130. Citado na página 22.

KOEDINGER, K. R. et al. Data mining and education. *Wiley Interdisciplinary Reviews: Cognitive Science*, Wiley Online Library, v. 6, n. 4, p. 333–353, 2015. Citado na página 15.

KORTING, T. S. C4. 5 algorithm and multivariate decision trees. *Image Processing Division, National Institute for Space Research–INPE Sao Jose dos Campos–SP, Brazil*, 2006. Citado na página 22.

LINOFF, G. S.; BERRY, M. J. *Data mining techniques: for marketing, sales, and customer relationship management*. [S.l.]: John Wiley & Sons, 2011. Citado na página 19.

LOBO, M. Panorama da evasão no ensino superior brasileiro: aspectos gerais das causas e soluções. *Associação Brasileira de Mantenedoras de Ensino Superior. Cadernos*, v. 25, 2012. Citado na página 14.

LOBO, R. A evasao no ensino superior brasileiro–novos dados. 2017. Citado na página 14.

MACHADO, R. D. et al. Estudo bibliométrico em mineração de dados e evasão escolar. In: *Congresso Nacional de Excelencia em Gestao*. [S.l.: s.n.], 2015. v. 8. Citado 2 vezes nas páginas 14 e 15.

MANHÃES, L. M. B. et al. Previsão de estudantes com risco de evasão utilizando técnicas de mineração de dados. In: *Brazilian symposium on computers in education (simpósio brasileiro de informática na educação-sbie)*. [S.l.: s.n.], 2012. v. 1, n. 1. Citado na página 14.

MANOUSELIS, N. et al. Recommender systems in technology enhanced learning. In: *Recommender systems handbook*. [S.l.]: Springer, 2011. p. 387–415. Citado na página 15.

MEC. *Plataforma Nilo Peçanha - Rede Federal de Educação Profissional, Científica e Tecnológica SETEC/MEC - Ano base 2019*. 2020. <encurtador.com.br/ekK18>. Citado na página 15.

MONARD, M. C.; BARANAUSKAS, J. A. Indução de regras e árvores de decisão. *Sistemas Inteligentes-Fundamentos e Aplicações*, v. 1, p. 115–139, 2003. Citado na página 23.

NERI, M. C. Motivos da evasão escolar. 2009. Citado 2 vezes nas páginas 14 e 18.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Citado na página 32.

PEDRO, M. O. et al. Predicting college enrollment from student interaction with an intelligent tutoring system in middle school. In: *Educational Data Mining 2013*. [S.l.: s.n.], 2013. Citado na página 15.

PENG, W.; CHEN, J.; ZHOU, H. An implementation of id3-decision tree learning algorithm. *From web. arch. usyd. edu. au/wpeng/DecisionTree2. pdf Retrieved date: May*, Citeseer, v. 13, 2009. Citado na página 22.

PERON, V. D.; BEZERRA, R. C.; PEREIRA, E. N. Causas e monitoramento da evasão universitária no contexto brasileiro: uma revisão sistemática. *Revista de Estudos e Pesquisas sobre Ensino Tecnológico (EDUCITEC)*, v. 5, n. 11, 2019. Citado na página 17.

POTDAR, K.; PARDAWALA, T. S.; PAI, C. D. A comparative study of categorical variable encoding techniques for neural network classifiers. *International journal of computer applications*, v. 175, n. 4, p. 7–9, 2017. Citado 2 vezes nas páginas 28 e 29.

RAJESH, D. Application of spatial data mining for agriculture. *International Journal of Computer Applications*, Citeseer, v. 15, n. 2, p. 7–9, 2011. Citado na página 14.

RAJU, D.; SCHUMACKER, R. Exploring student characteristics of retention that lead to graduation in higher education using data mining models. *Journal of College Student Retention: Research, Theory & Practice*, SAGE Publications Sage CA: Los Angeles, CA, v. 16, n. 4, p. 563–591, 2015. Citado na página 14.

RIGO, S. J. et al. Aplicações de mineração de dados educacionais e learning analytics com foco na evasão escolar: oportunidades e desafios. *Revista Brasileira de Informática na Educação*, v. 22, n. 01, p. 132, 2014. Citado na página 14.

ROIGER, R. J. *Data mining: a tutorial-based primer*. [S.l.]: CRC press, 2017. Citado 2 vezes nas páginas 14 e 19.

ROMERO CRISTOBAL E VENTURA, S. Data mining in education. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Wiley Online Library, v. 3, n. 1, p. 12–27, 2013. Citado na página 14.

ROPODI, A.; PANAGOU EZ E NYCHAS, G.-J. Data mining derived from food analyses using non-invasive/non-destructive analytical techniques; determination of food authenticity, quality & safety in tandem with computer science disciplines. *Trends in Food Science & Technology*, Elsevier, v. 50, p. 11–25, 2016. Citado na página 14.

SESU, B. M. da Educação. Secretaria da E. retenção e evasão nos cursos de graduação em instituições de ensino superior públicas. *Avaliação: Revista de rede de avaliação institucional da educação superior. Campinas*, v. 1, n. 2, p. 55–65, 1996. Citado na página 18.

TSUMOTO SHUSAKU E HIRANO, S. Risk mining in medicine: Application of data mining to medical risk management. *Fundamenta Informaticae*, IOS Press, v. 98, n. 1, p. 107–121, 2010. Citado na página 14.

YUKSELTURK, E.; OZEKES, S.; TÜREL, Y. K. Predicting dropout student: an application of data mining methods in an online education program. *European Journal of Open, Distance and e-learning*, Sciendo, v. 17, n. 1, p. 118–133, 2014. Citado na página 14.