Instituto Federal de Educação, Ciência e Tecnologia da Paraíba

Campus Campina Grande

Coordenação do Curso Superior de Tecnologia em Telemática

COMPUTANDO *ENSEMBLE METHODS* PARA PREDIZER EVASÕES ESTUDANTIS

RODOLFO BOLCONTE DONATO

Orientadora: Samara Martins Nascimento

Coorientador: Gustavo Wagner Diniz Mendes

Campina Grande, Junho de 2019

®Rodolfo Bolconte Donato

Instituto Federal de Educação, Ciência e Tecnologia da Paraíba

Campus Campina Grande

Coordenação do Curso Superior de Tecnologia em Telemática

COMPUTANDO *ENSEMBLE METHODS* PARA PREDIZER EVASÕES ESTUDANTIS

RODOLFO BOLCONTE DONATO

Monografia apresentada à Coordenação do Curso de Telemática do IFPB - Campus Campina Grande, como requisito parcial para conclusão do Curso Superior de Tecnologia em Telemática.

Orientadora: Samara Martins Nascimento

Coorientador: Gustavo Wagner Diniz Mendes

Campina Grande, Junho 2019

D677c Donato, Rodolfo Bolconte,
Computando ensemble methods para predizer evasões estudantis / Rodolfo Bolconte Donato. - Campina Grande, 2019.
53 f. : il.

Monografia (Curso Superior de Tecnologia em Telemática) - Instituto Federal da Paraíba, 2019.
Orientadora: Profa. Samara Martins Nascimento
Coorientador: Gustavo Wagner Diniz Mendes

1. Telemática – classificação de dados. 2. Ensemble Methods. 3. Educação - Evasão estudantil. I. Título.
CDU 004:37

COMPUTANDO *ENSEMBLE METHODS* PARA PREDIZER EVASÕES ESTUDANTIS

RODOLFO BOLCONTE DONATO

Dra. Samara Martins Nascimento
Orientadora

Me. Gustavo Wagner Diniz Mendes
Coorientador

Dr. Adriano Araujo Santos
Membro da Banca

Me. Thiago Pereira Rique
Membro da Banca

Campina Grande, Paraíba, Brasil
Junho 2019

"Nice boys don't play rock and roll
I'm not a nice boy"
Nice Boys - Guns N' Roses

Agradecimentos

O Primeiro, destino aos professores Gustavo Wagner e Samara Martins. Gustavo, coorientador do trabalho, que em 2017 me deu a oportunidade de iniciar minha trajetória no campo do Aprendizado de Máquina, em que trabalhamos na previsão de Evasões Estudantis no IFPB *campus* Campina Grande, o que me fez perceber o quão problemático é este termo, seja para os alunos como também para a própria instituição, me fazendo contribuir, embora que pouco, para tentar diminuir a quantidade de desistências nos cursos. E Samara, que embora prestes a sair da instituição se interessou pelo trabalho e aceitou ser a orientadora do mesmo, que contribuiu imensamente para que toda a pesquisa realizada se tornasse ainda maior do que já era.

Aos professores membros da Banca Avaliadora do trabalho, Adriano Santos e Thiago Rique, destino o Segundo, por todas as sugestões, ideias e também correções apresentadas, que contribuíram para que este trabalho aumentasse ainda mais o seu nível de conhecimento e também o meu interesse em especial pelas áreas de Análise de Dados e Aprendizado de Máquina, evidenciando a grandiosidade de cada uma delas.

Como o Terceiro, destino ao IFPB *campus* Campina Grande e todos os seus integrantes. Aos terceirizados, professores, coordenadores e diretores que contribuíram de alguma forma para o meu crescimento tanto pessoal como acadêmico, em especial o professor Jerônimo Rocha, uma das pessoas mais educadas, honestas, inteligentes e prestativas que tive o prazer de conhecer, me ajudando sempre que possível, principalmente com seus conselhos, experiências de vida e conversas descontraídas. Destino também ao Ramo Estudantil IEEE do IFPB *campus* Campina Grande bem como todos os seus voluntários, que me foi dada a honra de estar à frente do mesmo na Presidência onde pude vivenciar grandes experiências nesse mundo incrível e engajador que é o IEEE. Agradeço também aos membros da diretoria do Ramo Estudantil que me ajudaram a gerenciá-lo com bastante sensatez, Marlon Costa, Andréa Lima e Bryan Khelven.

O Quarto, destino à pessoa mais linda, amorosa e atenciosa que tive o prazer de conhecer na instituição, Anna Clara de Figueiredo, que embora longe, sempre esteve comigo me incentivando cada vez mais para a realização deste trabalho. Obrigado por existir e me suportar ao longo do tempo.

Como o Quinto, destino à todos os colegas que tive e os grandes amigos que fiz na instituição ao longo do curso, em especial ao grupo dos Mascotes de Telemática, juntos desde o primeiro período do curso que se ajudaram sempre em diversas questões. Obrigado

Maxsuel Medeiros, Miqueas Galdino, Nathalya Leite e Wemerson Vital.

O Sexto, destino para todos aqueles não citados, mas que estiveram envolvidos durante a minha trajetória no curso, contribuindo tanto com maior quanto menor intensidade. Obrigado pelo apoio e os bons momentos vividos.

O Sétimo e último, como o agradecimento mais considerável, destino às duas pessoas mais importantes da minha vida, Vilma Bolconte e Roberto Donato, que me ajudaram nos momentos mais difíceis que enfrentei. Sem meus pais, não sei onde estaria atualmente em meio aos desmaios, crises de ansiedade e demais problemas de saúde que vivenciei desde o meu nascimento, e muito menos não seria possível a minha conclusão no Curso Superior de Tecnologia em Telemática. Cada conselho dessas duas pessoas contribuiu para o meu crescimento, tanto pessoal, como acadêmico e profissional.

Resumo

A evasão estudantil ocorre em todas as esferas da educação, por diferentes fatores. Há estudos que apontam como causa fatores inerentes ao ensino, dentro da instituição, e outros apontam fatores sociais extra instituição, tais como condição financeira e distância da casa do estudante à instituição, causando problemas tanto sociais quanto econômicos. Do ponto de vista social, as expectativas dos alunos, são diminuídas quando os mesmos desistem dos estudos. Do ponto de vista econômico as instituições sofrem com a baixa expectativa criada pelos altos investimentos por parte dos governos e empresas. Tendo em vista os danos anteriores apresentados, este trabalho buscou analisar dois *Ensemble Methods* – técnicas computacionais de Aprendizado de Máquina – para obter e comparar resultados de predição. As estratégias computaram o mesmo algoritmo de classificação, como forma de encontrar aquele que obtenha o melhor desempenho na identificação de possíveis alunos evasores, do Curso Superior de Tecnologia em Telemática do Instituto Federal da Paraíba *campus* Campina Grande. Tal trabalho foi realizado a partir dos dados acadêmicos dos alunos utilizando atributos que sejam determinísticos para a previsão de evasão. O conjunto de dados totalizou 720 amostras de alunos, sendo 429 amostras evadidas e 291 amostras não evadidas. Os algoritmos foram testados de 3 formas distintas, com o conjunto de dados sem balanceamento, com balanceamento utilizando o método *Oversampling* e também com o método *Undersampling*, que correspondem a técnicas capazes de igualar a quantidade de amostras das classes utilizadas. Cada resultado dos testes foi comparado através dos valores de cinco métricas, sendo elas: Acurácia, Precisão, Sensibilidade, Taxa de Falsa Previsão Positiva e Tempo de Processamento. De forma geral, os testes experimentais mostraram maiores discrepâncias nos tempos de processamento dos algoritmos analisados.

Palavras-chave: Evasão Estudantil, Aprendizado de Máquina, Classificação de Dados.

Abstract

Student evasion occurs in all spheres of education, due to different factors. There are studies that point to factors inherent to teaching, within the institution, and others point out extra social institution factors, such as financial condition and distance from the student's home to the institution, causing both social and economic problems. From the social point of view, the expectations of students are diminished when they drop out of school. From the economic point of view institutions suffer from the low expectations created by high investments by governments and companies. Considering the previous damages presented, this work sought to analyze two *Ensemble Methods* - computational techniques of Machine Learning - to obtain and compare prediction results. The strategies use the same classification algorithm, as a way to find the one that obtains the best performance in the identification of possible evader students, of the Instituto Federal da Paraíba *campus* Campina Grande. This work was carried out from the academic data of the students using attributes that are deterministic for the prediction of evasion. The data set totaled 720 student samples, 429 samples being evaded and 291 samples not evaded. The algorithms were tested in 3 different ways, with the unbalanced data set, using the *Oversampling* method and also using the *Undersampling* method, which correspond to techniques capable of matching the sample quantity classes used. Each result of the tests was compared through the values of five metrics, being: Accuracy, Precision, Recall, False Positive Forecast Rate and Processing Time. In general, the experimental tests showed greater discrepancies in the processing times of the analyzed algorithms.

Keywords: Student Evasion, Machine Learning, Data Classification.

Sumário

Lista de Figuras

Lista de Abreviaturas

AA	*Academic Analytics*
ACM	Association for Computing Machinery
AM	Aprendizado de Máquina
CC	Ciência da Computação
CST	Curso Superior de Tecnologia
ENADE	Exame Nacional de Desempenho dos Estudantes
ETL	*Extract Transform Load*
FN	Falso Negativo
FP	Falso Positivo
GPS	*Global Positioning System*
IA	Inteligência Artificial
IBM	International Business Machines
IEEE	Institute of Electrical and Electronics Engineers
IFPB	Instituto Federal de Educação, Ciência e Tecnologia da Paraíba
LA	*Learning Analytics*
LASSO	*Least Absolute Shrinkage and Selection Operator*
ML	*Machine Learning*
RFE	*Recursive Feature Elimination*
SciELO	Scientific Electronic Library Online
SUAP	Sistema Unificado de Administração Pública
UFCG	Universidade Federal de Campina Grande
UNESP	Universidade Estadual Paulista
UNICAMP	Universidade Estadual de Campinas
USP	Universidade de São Paulo
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo
WEKA	*Waikato Environment for Knowledge Analysis*

Capítulo 1

Introdução

1.1 Contextualização

Para compreensão deste trabalho, é preciso uma contextualização de dois temas cruciais e como eles se conectam, sendo eles a Evasão Estudantil e como ela é problemática para os âmbitos social e financeiro, e o processo da Classificação Automática de Dados, investigando por que realmente é necessária a sua utilização.

1.1.1 Evasão Estudantil

A evasão estudantil, ou evasão escolar, é a desistência de um aluno de seu curso em uma instituição de qualquer categoria, seja ela pública ou privada, acarretando diversos problemas, tanto em termos sociais quanto econômicos. Do ponto de vista social, as expectativas dos alunos, como também dos familiares e os demais que os cercam, são diminuídas quando os mesmos desistem dos estudos. Do ponto de vista econômico, a evasão estudantil pode ser ainda mais problemática. Quando um aluno desiste dos estudos, as instituições sofrem com a baixa expectativa criada pelos altos investimentos por parte dos governos e também de empresas, em certos casos.

Com relação às causas que levam os estudantes a desistirem de seus cursos, existem inúmeras, que podem ser divididas em fatores inerentes ao ensino e fatores extra instituição. Fatores inerentes ao ensino ocorrem dentro da instituição que podem de fato levar um aluno a evadir. São fatores mais comuns como causas de evasão: notas baixas obtidas pelos alunos, aulas mal ministradas pelos professores que impossibilitam a compreensão do aluno e disciplinas com didáticas inapropriadas que fazem o aluno perder cada vez mais o interesse em seu curso [Gilioli 2016].

Os fatores extra instituição são aqueles encontrados no âmbito social do aluno, sendo também muito comuns como causas de evasões. Existem vários, dos quais é possível destacar: problemas familiares que atrapalham o desempenho do aluno nos estudos, o aluno estudar e trabalhar ao mesmo tempo, não conseguindo conciliar as duas atividades, a falta de aptidão na área, que muitas vezes é descoberta muito depois do aluno começar o curso, e um dos

principais fatores extra instituição, que é ingressar em um curso esperando a oportunidade de entrar em outro [Gilioli 2016].

Na próxima Seção é descrito o processo da classificação de dados de forma automática, adotado em diversos estudos para tentar combater a evasão estudantil nas instituições, através da análise de informações de alunos já matriculados nos cursos para prever a situação destes em um tempo futuro.

1.1.2 Classificação Automática de Dados

O processo de Classificação permite a descoberta de conhecimento a partir de um conjunto de dados, por meio da categorização de seus atributos. Este processo consiste na atribuição de rótulos em determinados objetos a partir da observação de similaridades com outros objetos já classificados, ou seja, uma classificação que é baseada em experiências anteriores. O conceito pode ser observado não somente na área da computação, mas também em diversas situações cotidianas, por exemplo, quando uma pessoa observa um indivíduo do outro lado da rua e, a partir de processos cerebrais, realiza deduções baseadas nas características físicas do indivíduo, podendo determiná-la como um homem ou mulher, um adulto ou criança, um conhecido ou desconhecido, entre outros. Tais deduções podem ser compreendidas como classificações [Ferreira 2016].

Embora a classificação manual seja suficiente para conjuntos de dados relativamente pequenos e simples, com apenas alguns atributos, para conjuntos de dados maiores e também mais complexos, com inúmeros atributos para determinar um possível resultado, é necessária uma solução automatizada [Tan *et al.* 2018]. Com o avanço tecnológico ao longo dos anos, sistemas computacionais já são capazes de classificar objetos com introspecção similar à dos seres humanos. Porém, para que a classificação computacional seja possível, podem ser utilizadas técnicas de Aprendizado de Máquina (AM), uma área idealizada por Arthur Lee Samuel em meados de 1952 [Sutton e Barto 1998], na qual sistemas são capazes de assimilar informações sozinhos, a partir do reconhecimento de padrões de dados, com o propósito de classificar objetos.

O Aprendizado de Máquina é, atualmente, dividido em três abordagens, sendo elas: 1) Aprendizado por Reforço, em que sistemas tentam aprender qual a ação adequada que deve ser seguida, dependendo das circunstâncias em que a mesma está presente [Honda, Facure e Yaohao 2017] 2) Aprendizado Não Supervisionado, no qual não há uma classificação conhecida para os objetos referente aos treinos introduzidos nos sistemas, cabendo a eles determinar se é possível uma classificação e qual devem receber [Rezende 2003] e 3) Aprendizado Supervisionado, quando os objetos utilizados para os treinos já têm uma marcação de classe prévia, com o propósito de classificar futuros objetos de acordo com os dados já assimilados [Müller e Guido 2016].

1.2 Formulação do Problema

Como apresentado, a evasão estudantil acarreta diversos problemas, sejam sociais ou financeiros, tanto para o aluno que desiste de seu curso, como para a instituição. Por meio de um levantamento informal feito à diretoria geral do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB) *campus* Campina Grande em dezembro de 2017, foi informado que um aluno custa em média 3,7 mil reais mensais para a instituição. Com uma evasão de 10 alunos no começo de um curso, acarreta um custo anual de 444 mil reais, e durante os 3 anos de um curso de informática, ocasiona em torno de 1,3 milhões de reais de gastos. Esse exemplo demonstra que a evasão tem um alto custo para as instituições, pois as mesmas devem arcar com diversas despesas financeiras, tais como energia, limpeza, manutenção e suporte, com o objetivo de manter uma boa estrutura para seus alunos.

Em se tratando de evasões estudantis no IFPB *campus* Campina Grande, e mais precisamente no Curso Superior de Tecnologia (CST) em Telemática da instituição, foi constatado que o número de evasões no curso de 2007 à 2016 representa mais da metade do número total de matrículas realizadas neste período, em que, das 839 matrículas já feitas, 439 vieram a evadir, representando 52% de evasões no curso. Estas informações foram obtidas a partir do acesso ao banco de dados do *QAcadêmico*, um antigo sistema de gestão acadêmica da instituição. Se considerarmos que desde o início do curso, datado em 2007, até 2016, um aluno custa para a instituição os mesmos 3,7 mil reais mensais estipulado no levantamento informal feito em 2017, 439 evasões representam um prejuízo de mais de 1,5 milhão de reais, isto se todos estes alunos evadiram do curso em até um mês de matrícula. Caso eles tenham permanecido na instituição por 6 meses completos, o "prejuízo" aumenta para mais de R$ 9,7 milhões.

A partir dos dados apresentados, é perceptível que a evasão estudantil representa um grande problema na parte financeira das instituições, e que é preciso a execução de medidas para prevenir ou até mesmo diminuir a ocorrência destas. Porém, métodos convencionais, como a análise do histórico dos alunos por parte das coordenações das instituições, podem ser complexos para os profissionais chegarem ao consenso da evasão e até mesmo demorados, com a possibilidade de uma evasão ser detectada já depois de sua ocorrência. O ideal é que a instituição, bem como seus professores, coordenadores e diretores, tenham um conhecimento prévio dos possíveis evasores, para que possam realizar medidas direcionadas aos principais "alvos" antes mesmo da desistência do curso. Sendo assim, uma solução prática é a utilização de métodos computacionais que analisam as informações acadêmicas do aluno e conseguem definir se ele virá a evadir ou não de seu curso, a partir do reconhecimento de padrões em tais dados.

1.3 Justificativa e Relevância

Em 2016 foi realizado um trabalho de Mestrado na Universidade Federal de Campina Grande (UFCG), onde foram testadas duas estratégias computacionais com o propósito de investigar qual a de melhor desempenho na previsão de evasões, utilizando apenas um algoritmo para impulsionar outro de classificação de dados. A primeira estratégia consiste na previsão da evasão por semestre independente de qual curso o aluno faz parte (Estratégia Global), e na segunda, as previsões são feitas divididas por cursos e seus períodos (Estratégia Específica). Como resultados, foi obtida acurácia média de 94,4% para a Estratégia Global e 89,8% de acurácia para a Estratégia Específica [Melo 2016].

Este trabalho serviu como ideia inicial para a realização de um projeto de pesquisa no IFPB *campus* Campina Grande em 2017 intitulado "Previsão Automática de Evasão Estudantil nos Cursos do IFPB", sendo também a base para o presente trabalho, onde foram analisadas evasões nos cursos da instituição até o primeiro período do ano de 2015. Como resultados do projeto, em [Bolconte e Mendes 2017] foram obtidas as acurácias de previsão dos modelos de classificação utilizados em duas propostas diferentes de testes: 1) na primeira proposta foram feitas previsões em todos os cursos da instituição, obtendo 84% de acurácia no melhor modelo, e 2) já na segunda, somente nos cursos da área de informática, a maior acurácia dos modelos atingiu 86,8%. Estendendo o projeto realizado em 2017, este trabalho apresenta uma terceira proposta, onde a previsão de evasões é feita somente no CST em Telemática da instituição.

Assim, a partir dos resultados obtidos nos trabalhos realizados nas instituições federais de Campina Grande, é perceptível que a utilização de técnicas computacionais de classificação para a previsão automática de evasões estudantis pode ajudar na construção de sistemas de informação tanto administrativa quanto estratégica. No contexto administrativo, é possível analisar os resultados obtidos a partir de relatórios detalhados que forneçam informações importantes para o setor administrativo das instituições, enquanto que no contexto estratégico, é possível definir mudanças relacionadas ao processo de ensino-aprendizagem, assim como acompanhamento no contexto social de um aluno, com o propósito de impedir que uma evasão ocorra.

Este trabalho utiliza e também compara dois *Ensemble Methods* em cima de um único algoritmo de classificação de dados, a fim de encontrar aquele que obtenha os melhores resultados métricos como Acurácia, Precisão, Tempo de Processamento e afins, quando utilizados em diversas estratégias de conjunto de dados, para que possa ser usufruído em pesquisas futuras relacionadas a previsão de evasões estudantis. É destacada também a utilização de uma estratégia pouco empregada na literatura para a organização de informações acadêmicas dos alunos, que é apenas uma linha de informações para definir todo o histórico acadêmico, ao invés de uma linha de informações por período/semestre do aluno.

Espera-se também que, com os possíveis evasores obtidos deste trabalho, as coordenações e diretorias possam se concentrar nestes para a realização de atividades mais engajadoras

visando a permanência dos alunos, para que não acabem se prejudicando e também prejudicando a instituição futuramente.

1.4 Objetivos

1.4.1 Objetivo Geral

O presente trabalho tem como principal objetivo comparar e definir o *Ensemble Method* mais adequado para a previsão automática de evasões estudantis no Curso Superior de Tecnologia em Telemática, através da comparação de valores métricos dos testes realizados, e então comunicar à coordenação do curso quais os possíveis evasores para que medidas preventivas sejam tomadas.

1.4.2 Objetivos Específicos

- Definir os atributos que mais descrevam a evasão ou não dos alunos no CST em Telemática do IFPB *campus* Campina Grande a partir dos atributos utilizados na literatura;
- Caracterizar o funcionamento do algoritmo de previsão de dados Árvore de Decisão e também dos *Ensemble Methods* Floresta Aleatória e Aumento de Gradiente;
- Testar e comparar o desempenho dos *Ensemble Methods* Floresta Aleatória e Aumento de Gradiente ao utilizarem o algoritmo de Árvore de Decisão;
- Definir qual o *Ensemble Method* mais adequado para a previsão de evasões estudantis do Curso Superior de Tecnologia em Telemática do IFPB *campus* Campina Grande;

1.5 Metodologia

Para a realização deste trabalho, as seguintes etapas metodológicas foram abordadas:

- Etapa 1: Levantamento atualizado de referências bibliográficas, tanto na área de Evasão Estudantil, como Aprendizado de Máquina;
- Etapa 2: Escrita da primeira parte do Trabalho de Conclusão de Curso;
- Etapa 3: Processo de *Extract Transform Load (ETL)* dos dados dos alunos do Curso Superior de Tecnologia em Telemática do IFPB *campus* Campina Grande;
- Etapa 4: Criação dos modelos de previsão de dados;
- Etapa 5: Realização dos primeiros testes de previsão automática de Evasão e comparação dos resultados através das métricas de análise estatística: Acurácia, Precisão,

Sensibilidade e Taxa de Falsa Previsão Positiva, assim como também o Tempo de Processamento;

- Etapa 6: Escrita da segunda parte do Trabalho de Conclusão de Curso;
- Etapa 7: Realização de melhorias nos modelos de aprendizado de máquina utilizados, a fim de melhorar o poder preditivo dos mesmos;
- Etapa 8: Escrita da parte final do Trabalho de Conclusão de Curso.

1.6 Organização do Documento

O restante deste documento está organizado da seguinte maneira: No Capítulo 2, é apresentada a fundamentação teórica do tema referente ao desenvolvimento do trabalho, sendo ele o Aprendizado de Máquina, evidenciando seu conceito, aplicações, desafios e suas abordagens; No Capítulo 3 é descrito todo o planejamento para a execução dos testes do trabalho, além de elucidar as técnicas que organizam melhor os dados trabalhados e também as métricas utilizadas na comparação dos resultados; No Capítulo 4 são apresentados os resultados obtidos bem como as melhorias elaboradas nos modelos de classificação para a apresentação de resultados mais expressivos; Já no Capítulo 5, são discutidas as considerações finais a partir dos resultados obtidos na seção anterior, como também os possíveis trabalhos futuros a serem elaborados.

Capítulo 2

Estado da Arte

Este capítulo apresenta a fundamentação teórica necessária para o entendimento deste trabalho. Nas seções, são descritos conceitos relacionados ao Aprendizado de Máquina, algoritmos e abordagens empregadas, além de trabalhos de análise de evasões estudantis nas instituições.

2.1 Aprendizado de Máquina

2.1.1 Contextualização Histórica

Em 1952, o cientista da computação da International Business Machines (IBM), Arthur Lee Samuel, propôs um jogo de damas para o IBM 701 e, três anos depois, em 1955, finalizou então seu primeiro jogo baseado no reconhecimento de padrões [Sutton e Barto 1998]. O propósito do desenvolvimento deste programa, que utilizava um reconhecimento computacional, era verificar o fato de que um computador pode ser programado para que ele aprenda a jogar um jogo de damas melhor do que aquele que pode ser jogado pela pessoa que escreveu o programa. Samuel afirma também que seu programa pode aprender a concluir seu objetivo em cerca de 8 a 10 horas, se forem apresentadas apenas as regras do jogo, um senso de direção das jogadas e uma lista redundante de parâmetros que descrevem o jogo, ainda que outras características sejam desconhecidas [Samuel 1959].

Conforme eram feitos ajustes no aprendizado de seu programa, a qualidade do mesmo melhorava de forma constante, porém a jogabilidade não era a desejada. Durante este período de ajustes, alguns jogadores amadores chegaram a testar o programa, chegando a conclusão de que era "complicado, mas superável", em que posteriormente chegou a ser descrito como "um jogador melhor que a média". A partir de análises detalhadas desses resultados, foi indicado que o procedimento de aprendizagem se mostrou satisfatório, apesar de bastante errático e não muito estável [Samuel 1959].

Uma versão posterior do programa de Samuel apresentada em 1966 incluiu melhorias em seu procedimento de busca, como o uso extensivo de um modo de aprendizado chamado "aprendizado de livros" e tabelas de consultas hierárquicas, embora ainda não alcançando o nível de jogadores profissionais [Sutton e Barto 1998]. Toda a pesquisa em torno do jogo de

damas criado por Samuel foi amplamente reconhecido como uma conquista significativa em Inteligência Artificial (IA) e também como um dos precursores de técnicas de Aprendizado de Máquina [Sutton e Barto 1998].

2.1.2 Definição

Segundo [Richert e Coelho 2013], o Aprendizado de Máquina (do inglês *Machine Learning (ML)*), é um ensinamento às máquinas de como elas devem executar tarefas sozinhas. Já [Müller e Guido 2016], define o AM de maneira mais formal, tratando-se de uma extração de conhecimento a partir de determinados dados, sendo um campo de pesquisa no encontro da Estatística, da Inteligência Artificial e da Ciência da Computação (CC), que também é conhecido como Análise Preditiva ou Aprendizagem Estatística.

Em termos mais técnicos, [Geitgey 2014] descreve o AM como a ideia de que existem algoritmos genéricos que podem apresentar algo interessante sobre um conjunto de dados, sem que seja necessário escrever qualquer código personalizado específico para o problema, apenas a escrita básica de um algoritmo genérico capaz de construir sua própria lógica de funcionamento a partir do recebimento de determinados dados. Já em [Carvalho e Silva 2014], pode-se considerar que um sistema tem a capacidade de "aprender", se o mesmo consegue melhorar sua performance em determinada tarefa, na medida em que é estimulado com dados de experiências passadas.

Exemplificando como o AM funciona na prática, tem-se uma adaptação da ideia de [Geitgey 2014], que propôs o problema de definir o preço de um determinado imóvel. Uma das soluções deste problema é contar com a ajuda de um corretor especialista na área, que para chegar num valor concreto do imóvel, analisa outros anteriormente vendidos. Para que a análise ocorra, são necessárias informações como: tipo de imóvel, área total, localização, quantidade de cômodos, entre outros. Com isso, o especialista pode definir um valor para o imóvel com base em experiências anteriores de imóveis similares que já foram vendidos. No entanto, no problema considerado, não é possível contar com a ajuda de um corretor especialista e são conhecidas apenas as informações de imóveis que já foram vendidos. Nesse caso, a forma principal de solucionar este problema é encontrar padrões dos dados coletados ou então definir quais informações são mais relevantes para a obtenção do preço do imóvel, com base no passado.

A solução ideal para o problema apresentado em [Geitgey 2014] deve utilizar um sistema baseado em AM, que tem o propósito de interligar as informações com o intuito de encontrar padrões que definam o preço mais real possível de forma automática. O sistema é treinado com base nas informações de outros imóveis já vendidos, que podem incluir o valor das vendas, para que descubra as possíveis lógicas que podem resultar num preço real. Porém, como o número de possibilidades para chegar no resultado satisfatório é enorme, o sistema deve ser capaz de determinar: *Quais informações predizem melhor o preço de um imóvel? Quais das informações utilizadas são as mais significativas? Quais as menos significativas? Se dois*

imóveis são iguais em informações de tamanho e cômodos, mas localizados em ambientes adversos, deve-se levar em conta estes ambientes?.

2.1.3 Aplicabilidades

Algumas aplicações usufruem de métodos de AM, a exemplo das primeiras aplicações "inteligentes" da Internet, que usavam regras de decisões elaboradas manualmente para processar informações de acordo com as preferências dos usuários [Müller e Guido 2016]. Dentro desse contexto, é possível citar os sistemas de *spam*, cuja função é mover um *e-mail* para uma pasta própria de *spams*, em que seja possível criar uma "lista negra" de palavras que resultariam em um *e-mail* marcado como *spam*. Essa aplicação usa regras de decisões elaboradas manualmente para obter um resultado, dado um processo de tomada de decisão [Müller e Guido 2016].

Projetar um sistema com regras de decisões é viável para determinadas aplicações. Particularmente, é necessário ter um bom entendimento de como uma decisão está sendo tomada, ou seja, as escolhas de cada regra a ser seguida devem ser previamente estabelecidas. Porém, sistemas que utilizam este aspecto de decisões têm duas desvantagens, sendo elas [Müller e Guido 2016]:

- A lógica necessária para tomar uma decisão é específica para um único domínio e/ou ação. Mudar esta ação, mesmo que de forma aparentemente irrelevante, pode exigir uma reescrita de todo o sistema [Müller e Guido 2016];
- Projetar regras requer um entendimento acentuado de como uma decisão deve ser tomada a partir de determinadas informações, tornando o sistema menos escalável [Müller e Guido 2016].

O reconhecimento facial de imagens é um exemplo de cenário onde um sistema de regras de decisões que foram elaboradas manualmente tende a falhar. Até 2001, este era um problema sem solução, visto que a maneira como os *pixels* de uma imagem são percebidos por computadores é diferente de como os humanos percebem. Esta diferença na percepção torna praticamente impossível para um humano criar um conjunto satisfatório de regras para descrever o que constitui um rosto em uma imagem. Porém, usar um sistema baseado em AM com uma coleção de imagens é suficiente para que seja possível determinar quais são as características necessárias para realizar a detecção [Müller e Guido 2016].

Como o AM é uma solução automática para problemas dos mais variados tipos, o investimento de empresas na construção destas soluções vem crescendo cada vez mais, possibilitando uma expansão no conjunto de áreas em que o AM pode ser aplicado. De acordo com [Columbus 2018], o registro de patentes que utilizam AM em sua composição cresceram a uma taxa anual de 34% entre 2013 e 2017, sendo a terceira categoria de crescimento mais

rápido entre as patentes concedidas, contando com a IBM[1], Microsoft[2] e Google[3] como os três maiores produtores de patentes na área de AM em 2017.

Das mais variadas áreas que adotam técnicas de AM para a solução de problemas, tem-se as seguintes:

- Reconhecimento de Imagens: cada vez mais comum nos dias atuais, o reconhecimento facial está presente desde a detecção de rostos em imagens do Facebook, até a detecção de sorrisos em tempo real presente em câmeras e *smartphones*. Contudo, a crescente utilização do AM para o reconhecimento de padrões em imagens é recente, tendo um impulso em sua popularização por volta de 2009, quando os cientistas da computação, Fei-Fei Li[4], Kai Li, entre outros, gratuitamente disponibilizaram seu projeto à comunidade, intitulado ImageNet, que corresponde a um banco de dados composto de amostras de imagens e descrições, com o propósito de ser utilizado em pesquisas que utilizam *softwares* de reconhecimento de padrões [Deng *et al.* 2009]. O principal motivo do não avanço das pesquisas nesta área foi justamente a falta de dados disponíveis em décadas passadas para que os sistemas fossem treinados com o intuito de realizarem os reconhecimentos digitais [Paladini 2016]. Porém o ImageNet se mostrou um divisor de águas para a área, ajudando na eficiência dos sistemas baseados em AM para o reconhecimento de objetos em imagens devido a seu acervo de conteúdo;

- Reconhecimento de Textos: assim como o reconhecimento de imagens, o reconhecimento de palavras e textos está presente desde sistemas com propósitos simples, tais como corretores ortográficos de *smartphones*, até em pesquisas mais avançadas para a caracterização de comportamentos humanos. Em [Medeiros 2004], é feita uma análise e interpretação de diversos textos para que técnicas de AM sejam empregadas para a classificação dos mesmos em categorias pré-estabelecidas. Já em [Kultzak 2016], é apresentada uma visão geral sobre a interpretação da linguagem em textos, a partir da utilização da plataforma *Waikato Environment for Knowledge Analysis (WEKA)*[5], sistema que integra diversos algoritmos de AM;

- Buscas Inteligentes: sugestões de mídias em plataformas de *streaming* como Netflix e Spotify, assim como rotas alternativas no Google Maps e Waze caracterizam o emprego de sistemas que utilizam AM em seus mecanismos de busca de informações. Os aprimoramentos das aplicações de *e-mail*, navegação por *Global Positioning System (GPS)* e até navegadores de internet oferecem melhorias adaptadas ao usuário, de acordo com seus costumes, tais como o modo de escrever e o histórico de navegações [Ribeiro 2018]. As inovações influenciam na criação de "filtros-bolha", com informações selecionadas

[1]Site da IBM: https://www.ibm.com/
[2]Site da Microsoft: https://www.microsoft.com/
[3]Site da Google: https://about.google/
[4]Fei-Fei Li em uma palestra explicando os avanços do Reconhecimento em Imagens a partir do ImageNet: https://youtu.be/40riCqvRoMs
[5]Site oficial do WEKA: www.cs.waikato.ac.nz/ml/weka

de acordo com gostos do usuário, que tendem a passar menos tempo na procura de informações de forma manual. Outros sistemas têm funcionalidades mais específicas, como o Google Play Music, com sugestões musicais de acordo com o clima ou a hora do dia, e também o Google Maps, que, além de reconhecer nomes de ruas e endereços a partir de imagens do Street View, leva em consideração as rotas de trânsito e a disponibilidade de estacionamento nas regiões, conforme o horário de uso atual [Ribeiro 2018];

- Diagnósticos de Saúde: o AM é uma tendência crescente na assistência médica devido ao surgimento de dispositivos de sensoriamento que permitem aos profissionais de saúde acessar dados de pacientes em tempo real, o que pode ajudar médicos especialistas na análise destes dados para identificar possíveis problemas de saúde em pacientes, aperfeiçoando cada vez mais diagnósticos e tratamentos. Na pesquisa [Subhani *et al.* 2017], métodos de AM foram utilizados para a detecção de estresse mental em múltiplos níveis, a partir da análise de sinais de eletroencefalogramas dos pacientes, e como resultados, obtiveram 94,6% de precisão para identificação em dois níveis de estresse e 83,4% de precisão para identificação em múltiplos níveis. Já em [Esteves, Lorena e Nascimento 2009], foram empregadas técnicas de AM como auxílio à análise de mamografias originadas da base de dados *Digital Database for Screening Mammography*, sendo um exemplo de aplicação de AM tanto em reconhecimento de imagens quanto em diagnósticos de saúde, atingindo 73,3% de taxa de acerto na detecção de câncer em alguns casos;

- Detecção de Fraudes: baseadas em engenharia social, cada vez mais as fraudes realizadas em diversos setores se apresentam mais desafiadoras para os métodos de detecção. Um ataque deste tipo pode se referir a qualquer operação em que a vítima é induzida a revelar detalhes financeiros confidenciais ou até mesmo transferir valores para fraudadores [Zoldi 2018]. Exemplos comuns de meios utilizados pelos fraudadores incluem *e-mails* falsos, conhecidos como *phishing*, e também mensagens de texto através de plataformas de comunicação. Dessa forma, os modelos baseados em AM conseguem detectar características e padrões genéricos ainda que só apareçam em certos tipos de fraudes [Zoldi 2018].

2.1.4 Desafios

Embora a utilização de AM para a solução de problemas esteja cada vez mais ampla, sua utilização enfrenta dificuldades e desafios de natureza organizacional, tecnológica e também filosófica. Dos mais variados problemas, é possível citar:

- Escassez de profissionais: as empresas precisam de cientistas de dados para que possam operar os sistemas com abordagens de AM. Os profissionais com tais habilidades se

tornaram mais procurados, porém a capacitação de um cientista de dados requer um considerável período de tempo [Academy 2018];

- Falta de cultura baseada em dados: embora as empresas compreendam os potenciais benefícios da tomada de decisão a partir de técnicas de AM, fazer com que toda uma entidade mude sua forma comportamental é um processo demorado e lento também [Academy 2018];

- Requisitos de infraestrutura: dependendo dos sistemas que utilizam técnicas de AM, é preciso um bom espaço de armazenamento de dados e capacidade de redes adequadas para a transação de dados [Academy 2018];

- Dilemas éticos: a IA está se tornando cada vez mais parecida com os seres humanos, quanto a forma de pensar e também quanto ao seu comportamento, porém falta o senso de moralidade em algumas utilizações. Por exemplo, o Tay, *bot* de mídia social que possui capacidades de AM, rapidamente aprendeu a disseminar mensagens inadequadas e inofensivas [Academy 2018];

- Aversão: embora pareça um assunto fictício, pessoas tem a ideia de que IA e também AM sejam perturbadores, de que sistemas automáticos possam assumir seus cargos, porém não estão totalmente errados. A Forrester previu que tecnologias cognitivas substituirão 7% dos empregos nos Estados Unidos até 2025 [Academy 2018].

Embora os problemas sociais e organizacionais citados possam de fato dificultar a execução dos sistemas baseados em AM, os dados com as informações com que estes sistemas podem trabalhar são os alvos principais para a ocorrência de erros, que podem impactar negativamente, de fato, em seus resultados. Após a extração e organização dos dados que serão analisados, é necessário um tratamento especial para eles, com o propósito de evitar os seguintes problemas:

- Dados Ruidosos: referem-se a modificações nos valores originais, que consistem em erros de medidas ou em valores consideravelmente diferentes da maioria dos demais no conjunto de dados. Alguns exemplos de dados ruidosos podem ser valores negativos que representam o ganho de uma renda, como também em atributos quantitativos, como quantidade de vendas [Silva, Peres e Boscarioli 2017];

- Dados Inconsistentes: observados quando para um mesmo atributo, ou atributos equivalentes, são encontrados valores diferentes em termos de tipo ou de domínio, por exemplo, em uma informação de data de nascimento de pessoas, uma delas apresenta somente números para dia, mês e ano, e outra apresenta números para dia e ano, porém nomes para os meses [Silva, Peres e Boscarioli 2017];

- Dados Ausentes: quando os atributos de um conjunto de dados não apresentam valores para determinados objetos, ou quando um conjunto de dados não possui valores para

um atributo de interesse, apresentando apenas valores agregados em relação àquele atributo [Silva, Peres e Boscarioli 2017]. Por exemplo, em um cadastro de clientes, alguns podem não informar o endereço residencial ou a ocupação profissional, em que tais dados poderiam ajudar em uma futura pesquisa para a criação de perfis dos clientes de uma empresa.

- Dados Redundantes: ocorre principalmente devido a três fatores: uso de nomenclaturas diferentes para atributos equivalentes, no caso, a repetição de informações; o armazenamento de atributos derivados, cujos valores são calculados a partir de valores em outros atributos; e por fim, a inserção de exemplares repetidos no conjunto de dados, por consequência de um erro de aquisição de dados [Silva, Peres e Boscarioli 2017].

- Dados Discrepantes (no inglês, *Outliers*): são dados que diferenciam drasticamente de todos os outros, conhecidos como "pontos fora da curva". É um valor que foge da normalidade e que pode causar anomalias nos resultados obtidos por meio de algoritmos e sistemas de análise, enviesando negativamente todo o resultado de uma análise [Aquarela 2017].

Como soluções para os problemas relacionados aos dados a serem utilizados, tem-se algumas formas:

- Preenchimento de Valores: caso haja a presença de um especialista no conjunto de dados, é possível encontrar o valor correto para determinado atributo de um exemplar e então realizar o preenchimento do mesmo, seja manual ou automaticamente [Han, Kamber e Pei 2012].

- Correção Manual e Automática: dependendo do tamanho do conjunto de dados e também da quantidade de dados ruidosos, é possível que o analista do sistema corrija estes dados adequando-os em faixas de valores de acordo com médias e medianas do conjunto [Han, Kamber e Pei 2012]. Caso o tamanho impossibilite uma correção manual, é possível a utilização de algoritmos de identificação e limpeza automática, para suavizar ou até anular certas informações;

- Remoção de Informações: em alguns casos que seja impossível a correção, deve-se então ser feita a remoção das informações, que pode ser realizada tanto na remoção de objetos, quanto na remoção de atributos, a depender do sistema que trabalhará com o conjunto, porém esta técnica não é aconselhada para conjuntos de dados pequenos [Han, Kamber e Pei 2012].

2.1.5 Abordagens

Embora o AM seja um subcampo da IA que, por sua vez, é um subcampo da Ciência da Computação, existem três subcampos ou abordagens do AM, sendo eles o Aprendizado

Por Reforço, Não Supervisionado – que utiliza técnicas indutivas – e o Supervisionado – utilizando técnicas preditivas. Na literatura, é comum a explanação dos aprendizados Não Supervisionado e Supervisionado, porém o Aprendizado Por Reforço não se identifica com nenhuma das outras duas abordagens, embora os princípios sejam parecidos. A Figura 2.1 apresenta as três abordagens do AM e suas principais vertentes.

Figura 2.1: *Hierarquia com as abordagens e suas vertentes do Aprendizado de Máquina.*

Aprendizado Por Reforço

Nesta abordagem, o sistema tenta aprender qual é a melhor ação a ser tomada dependendo das circunstâncias na qual essa ação será executada. Como não se sabe o que irá acontecer, é desejável que o Aprendizado Por Reforço leve em consideração uma incerteza futura, e consiga incorporar as eventuais mudanças no ambiente do processo de tomada da melhor decisão [Honda, Facure e Yaohao 2017].

A ideia de levar em consideração acontecimentos futuros deriva do conceito de "aprendizagem por reforço" da psicologia, no qual uma recompensa ou punição é dada a um agente dependendo da ação que ele venha a executar. Dado um determinado tempo e repetição de tarefas, espera-se que o agente consiga associar as ações que geram maior recompensa para cada situação que o ambiente apresente, passando a evitar as ações que possam gerar punições ou recompensas menores [Honda, Facure e Yaohao 2017].

Ilustrando um cenário para melhor entendimento do Aprendizado Por Reforço, tem-se um cachorro que deve ser adestrado para obedecer comandos de seu dono. Dificilmente o animal realizará o comando se escutar na primeira vez, então é necessário um reforço negativo (punição), como uma repreensão verbal e facial para o animal. Na medida em que o cachorro se aproxima do que realmente deve fazer, então é necessário um reforço positivo (recompensa), como uma comida, por exemplo. Realizando várias repetições com o animal, é esperado, com o passar do tempo, que o cachorro comece a associar a relação de "causa-efeito"

entre o comando e a possível recompensa recebida, fazendo com que aprenda a obedecer ao comando [Honda, Facure e Yaohao 2017].

Aprendizado Não Supervisionado

Também conhecido como Aprendizado Descritivo, no qual não há uma saída conhecida para as entradas. O sistema que trabalha a partir de tal abordagem deve analisar os dados fornecidos e tentar determinar se os mesmos podem ser agrupados de alguma maneira e, após o agrupamento, é necessária uma outra análise para identificar o que cada agrupamento significa no contexto do problema proposto [Rezende 2003].

Para exemplificar a utilização do Aprendizado Não Supervisionado, tem-se o problema de gestores de um supermercado que querem conhecer o perfil de seus consumidores. Pode haver um perfil de consumidor que rotineiramente compra vinho e queijo, ou que compra carne e carvão, ou ainda leite em pó e fraldas. Se for esse o caso, colocar tais produtos em setores distantes do supermercado pode aumentar o número de vendas, uma vez que aumentará o tempo de percurso dos consumidores, dando a eles a oportunidade de avaliar mais produtos para possíveis compras. Neste problema, não se sabe quantos e nem quais perfis de consumidores existem, cabendo ao sistema baseado no Aprendizado Não Supervisionado descobrir tais perfis. Uma opção para distinguir os perfis dos consumidores é analisar os registros de compras, verificando se existem padrões repetidos e que permitam a dedução de um perfil de consumidor ou até mesmo analisar diretamente quais produtos frequentemente são comprados em conjunto e, então, aprender uma regra associativa entre eles [Honda, Facure e Yaohao 2017].

O Aprendizado Não Supervisionado possui três tipos principais de vertentes para a solução de tarefas, sendo elas o Agrupamento, a Associação e a Sumarização, todas elas descritas a seguir.

Agrupamento Também conhecido no inglês como *Clustering*, é a técnica mais importante do Aprendizado Não Supervisionado, que consiste em encontrar grupos *(clusters)* no conjunto de dados. Durante o processo, é esperado que os itens que compõem um grupo sejam os mais similares possíveis entre si, enquanto que os itens em diferentes grupos serão os mais distintos. A semelhança ou diferença dos dados é dada de acordo com alguma função de similaridade adotada [Filho 2017]. Existe uma grande variedade de métodos e algoritmos que executam tarefas de agrupamento, e algumas das abordagens incluem o método baseado na distância, bem como encontrar um ponto central para cada grupo ou usar técnicas estatísticas baseadas em distribuições [Julian 2016].

Associação Consiste em identificar quais atributos estão relacionados. O foco de aplicações consiste em problemas de análise de "carrinhos de compras", gerando modelos descritivos que permitem descobrir regras do gênero, por exemplo, consumidores que adquirem pizzas tem uma probabilidade três vezes maior de adquirirem também queijo. Nesta vertente, a

compra de um conjunto de produtos é manipulada como uma única transação, que tem o objetivo de encontrar tendências nas várias transações analisadas que possam ser utilizadas para entender e explorar padrões de compra natural [Marreiros e Oliveira 2000]. Tais resultados podem ser utilizados no ajuste de estoques, para a modificação da disposição dos produtos no ambiente, como também a realização de campanhas promocionais.

Sumarização Nesta vertente, o objetivo é encontrar uma descrição simples e compacta de um conjunto de dados para então indicar qualquer similaridade entre seus registros. O formato de sumarização mais adotado é o sumário estatístico, que apresenta o número de opiniões positivas e negativas para cada aspecto encontrado, facilitando a visualização e o entendimento de todo o resultado ou de um aspecto específico [Aragão e Wilpert 2018]. Bastante utilizado em textos e conjunto de palavras, visto que a sumarização tem a ideia de criar uma espécie de resumo a partir das entradas fornecidas [Crispim 2014].

Aprendizado Supervisionado

Também conhecido como Aprendizado Preditivo, é utilizado para predizer um determinado resultado de uma dada entrada, a partir de exemplos de pares de entrada-saída, sendo indicado para tarefas em que seja necessário prever um comportamento ou resultado. São construídos modelos de AM a partir destes pares de entrada-saída que compõem o conjunto de treinamento dos mesmos, tendo como objetivo fazer previsões para dados de entrada nunca vistos antes pelos modelos. Esta abordagem requer um esforço por parte de especialistas para a construção do conjunto de treinamento, posteriormente sendo automatizados, acelerando muitas vezes uma tarefa incansável ou inviável [Müller e Guido 2016].

Exemplificando o uso da Aprendizagem Supervisionada temos um cenário em que é preciso definir o preço de veículos para suas vendas. Em um sistema com tal abordagem, é necessário a informação de carros que já foram vendidos bem como o preço final de cada um deles. Essas informações são chamadas de dados de treino de modelo, que tem o propósito de o modelo baseado em AM reconhecer padrões – entre as informações passadas – que possam determinar a razão de tal preço. A partir do modelo já treinado com as informações de vendas antigas, é possível inserir as informações dos veículos para que seus preços possam ser predizíveis.

Englobando o Aprendizado Supervisionado, existem duas vertentes principais para a predição de informações, sendo elas a Classificação e a Regressão.

Classificação Nesta vertente, o objetivo é prever um rótulo de classe, a partir de uma lista predefinida de possibilidades [Müller e Guido 2016], ou seja, predizer a qual categoria (ou classe) uma determinada entrada pertence. De acordo com o propósito do sistema, a Classificação pode ser dividida em:

- Binária: caso especial para a distinção de apenas duas classes. A classificação de *e-mails*

em *spam* ou não é um exemplo de problema de classificação binária, em que a pergunta do tipo "sim" ou "não" (duas classes) seria "Este *e-mail* é um *spam*?". É comum, na classificação binária, uma classe ser chamada de positiva e outra de negativa, porém nem sempre uma classe positiva representa algo benéfico e de valor, mas sim o objeto de estudo principal [Müller e Guido 2016];

- Várias Classes: quando há três ou mais opções para uma classificação. Para exemplificar uma classificação de várias classes, temos a meta de prever em qual idioma um site está escrito. No exemplo, as classes podem ser predefinidas em uma lista, visto a grande quantidade de idiomas existentes [Müller e Guido 2016].

Regressão Nesta vertente, o objetivo é predizer um número contínuo (número real, em termos matemáticos). Uma maneira de distinguir se um problema é de natureza de Classificação ou Regressão é saber se existe algum tipo de ordenação ou uma continuidade na saída. Caso exista, então é um problema de Regressão. Prever a renda anual de uma pessoa a partir de suas informações quanto a educação, idade e local de moradia é um exemplo de regressão, uma vez que o valor previsto pode ser qualquer número em um determinado intervalo [Müller e Guido 2016].

2.2 *Learning Analytics* e *Academic Analytics* para identificar Evasões Estudantis

Além do Aprendizado de Máquina, diversos são os trabalhos que utilizam métodos de análise, como as pesquisas acerca de *Learning Analytics (LA)* e *Academic Analytics (AA)*, para gerar informações que possam ser úteis aos sistemas escolares e aqueles que os cercam, como professores, coordenadores, diretores e os próprios alunos [Oliveira 2018; Junior e Oliveira 2016; Almeida 2016; Souza 2016; Ferreira e Andrade 2013; Nogueira *et al.* 2018].

Os estudos relacionados a *LA* se referem a medições, coletas, análises e relatos de dados sobre alunos e professores, com o objetivo de compreender e otimizar o aprendizado e os ambientes em que estão situados – geralmente em ambientes virtuais de aprendizado –, sendo um campo relacionado à mineração de dados educacionais [Oliveira 2018]. As pesquisas relacionadas a esses estudos têm foco principal no processo de ensino e aprendizagem.

Uma revisão de literatura foi feita em [Junior e Oliveira 2016], com o objetivo de compreender e apresentar os principais trabalhos e ferramentas relacionados a Análise de Aprendizado, visto que o interesse cresce cada vez mais a respeito da aplicação de tecnologias da informação e comunicação na educação, quanto à análise dos dados gerados a partir das interações dos alunos nos ambientes. O trabalho realizou uma revisão sistemática da área de AA disponível em diversas bases científicas, tais como Association for Computing Machinery (ACM), Google Scholar, Institute of Electrical and Electronics Engineers (IEEE), Scientific

Electronic Library Online (SciELO), Universidade Estadual Paulista (UNESP), Universidade Estadual de Campinas (UNICAMP) e Universidade de São Paulo (USP), entre 2010 e 2016, selecionando os trabalhos que demonstraram maior relevância para o objetivo do mesmo. Os pesquisadores do trabalho apresentam o total de trabalhos analisados, a distribuição temporal e geográfica das publicações, os conceitos principais referentes a *LA* e áreas correspondentes, além de uma exposição das ferramentas desenvolvidas em cada uma das pesquisas e suas principais características.

[Souza 2016] propôs uma ferramenta de avaliação das ações comportamentais de professores tutores, que atuam em disciplinas de ensino à distância. O objetivo da ferramenta é possibilitar a avaliação dos comportamentos tanto dos tutores como das turmas e como um pode influenciar os demais. As técnicas de análise de correlação de variáveis utilizadas nos dados adquiridos revelaram uma forte conexão entre os tutores e alunos, que serviu para chegar à conclusão de que os tutores à distância de fato influenciam na participação efetiva dos alunos das turmas do ensino a distância.

Os estudos acerca de *AA* estão relacionados com o nível institucional e visa obter informações a partir dos dados administrativos obtidos a partir da aplicação de *LA*. Dentro desses conceitos, é possível identificar informações acerca de alunos com riscos de evasão, predizendo possíveis motivações para essa ocorrência. Além disso, pode-se identificar problemas relacionados à gestão de uma instituição de ensino e, com isso, definir metas para implementar mudanças de melhorias [Almeida 2016; Ferreira e Andrade 2013].

Em [Nogueira *et al.* 2018], foi proposta uma arquitetura que visa o armazenamento de dados estruturados e não estruturados, espalhados em diferentes bases de conteúdo de ambientes virtuais de aprendizagem e sistemas de informações educacionais do ensino superior, para que estes mesmos dados concentrados num único *Big Data* sejam utilizados em técnicas de Análise de Aprendizado com o propósito de gerar informações úteis para estudantes, professores e também para as gestões acadêmicas.

2.3 Conclusão do Capítulo

Neste capítulo, foi explanado o conceito relacionado com Aprendizado de Máquina, uma subárea da Inteligência Artificial capaz de reconhecer padrões a partir de informações passadas e realizar classificações, além dos campos em que é utilizado, tais como reconhecimento de imagens e textos, os desafios que são encontrados quanto aos dados, que geralmente devem ser tratados antes de sua utilização, e também todas as suas abordagens e divisões conhecidas atualmente. Ademais, foram abordados trabalhos relacionados a *Learning Analytics* e *Academic Analytics*, identificando tanto análises acerca do processo de ensino e aprendizagem, como também descobertas de conhecimento para intervenção de melhorias na gestão de instituições de ensino. Neste presente trabalho, será utilizada a Classificação do Aprendizado de Máquina Supervisionado, buscando identificar, dado o conhecimento prévio dos

dados dos alunos, se os mesmos evadiram ou não do curso.

Capítulo 3

Uso de *Ensemble Methods* para Predizer Evasões Estudantis

Este capítulo descreve o uso de *Ensemble Methods* para obter resultados de predição. As análises realizadas consideraram os algoritmos Floresta Aleatória e Aumento de Gradiente, para realização de diferentes tipos de testes. Assim, as próximas Seções descrevem o ambiente construído para realização dos experimentos; os tipos dos experimentos realizados; o processo de coleta dos dados; os tipos de técnicas para ajuste dos conjuntos de dados; e as análises estatísticas que foram usadas para validar os algoritmos estudados.

3.1 Descrição do Trabalho

A presente pesquisa consiste na comparação de dois *Ensemble Methods* (Floresta Aleatória e Aumento de Gradiente) utilizando um único Algoritmo de Classificação (Árvore de Decisão), quando explorados em determinadas estratégias de divisão de Conjunto de Dados para a previsão de Evasões Estudantis. O conjunto de dados utilizado para testes se refere ao Curso Superior de Tecnologia em Telemática do IFPB *campus* Campina Grande, desde o período 2007.1 até o período 2015.1, incluindo períodos de greves ocorridas na instituição. A comparação se dá pela análise de métricas estatísticas a partir dos valores obtidos pelos dois algoritmos, sendo elas Acurácia, Precisão, Sensibilidade, Taxa de Falsa Previsão Positiva e Tempo de Processamento.

Com base no trabalho em que foi utilizado apenas o algoritmo Floresta Aleatória [Melo 2016], o presente trabalho evidencia e coloca em comparação este algoritmo com o algoritmo Aumento de Gradiente, ambos utilizados em estratégias de classificação. A diferença entre estes dois *Ensemble Methods* é que o Floresta Aleatória trabalha com aleatoriedade do Conjunto de Dados – particionamento em subconjuntos distintos para as árvores criadas – e também das combinações dos atributos descritivos utilizados em seus nós, enquanto o Aumento de Gradiente não trabalha com estas aleatoriedades, a diferença dos modelos criados se dá através de distribuições ponderadas – diversas variáveis com pesos distintos

– adicionadas a criação de cada árvore com o propósito de um modelo criado ter melhor desempenho que seu anterior.

Para os dois *Ensemble Methods*, é utilizado o algoritmo Árvore de Decisão, o mesmo Algoritmo de Classificação de Dados utilizado no trabalho [Melo 2016], que cria estruturas do tipo árvores de busca binária, para a classificação de dados a partir das informações que são passadas. Estes três algoritmos são melhores descritos na Seção 3.4.

Como forma de execução dos algoritmos de AM no trabalho, o conjunto de dados é dividido em dois subconjuntos de forma aleatória, sendo um para treino e outro para teste dos algoritmos. Este particionamento do conjunto de dados é conhecido como Validação Cruzada, técnica para avaliar a capacidade de generalização de um algoritmo. A Validação Cruzada possui três métodos de divisão de conjuntos, sendo eles:

- Método *Holdout*: indicado para grandes conjuntos de dados que pressupõe a criação de dois subconjuntos de dados distintos, a partir do conjunto de dados disponível para uso dos algoritmos. Um dos subconjuntos será usado para treinamento, e o segundo para validação após o término do treinamento. Neste método, o conjunto é dividido de forma que 70% dos dados sejam referentes ao subconjunto de treino, enquanto os 30% restantes dos dados são destinados para o subconjunto de validação [Silva, Peres e Boscarioli 2017];

- Método *K-Fold*: também indicado para grandes conjuntos de dados. Este método divide a população dos dados em um número K de subconjuntos (geralmente são divididos em 5 ou 10 conjuntos), onde cada um deles é usado para teste em algum momento. Por exemplo, um conjunto de dados é dividido em cinco subconjuntos, em que na primeira execução o primeiro subconjunto é utilizado para teste e o restante para treino, na próxima execução o segundo subconjunto para teste e o restante para treino, e assim sucessivamente [Hewa 2018].

- Método *Bootstrap*: indicado para pequenos conjuntos de dados. Este método é similar ao *Holdout*, que também utiliza dois subconjuntos de dados, sendo um para treino e outro para teste. Porém, a diferença é que no método *Bootstrap* durante o particionamento do conjunto de dados, uma amostra já escolhida para fazer parte de um dos subconjuntos pode ser escolhida novamente, com igual probabilidade, no caso a reposição de amostras é permitida neste método. Geralmente, 63,2% do conjunto de dados é destinado para treino, e os 36,8% restantes, destinado para validação [Silva, Peres e Boscarioli 2017].

O método de Validação Cruzada escolhido para o trabalho foi o *Bootstrap*, dado que o Conjunto de Dados utilizado para análise experimental é formado por apenas 720 amostras, que correspondem aos dados dos alunos do CST em Telemática do IFPB *campus* Campina Grande. Os testes realizados estendem a estratégia do algoritmo *K-Fold*, com K=10. Nesse caso, o particionamento da base de dados ocorre para treino e teste, possibilitando a reposição

de uma amostra, e os experimentos se repetem em 10 diferentes análises. Ao final destas execuções, é calculada a média dos valores das métricas estatísticas obtidas.

3.2 Ambiente de Testes

Todos os algoritmos foram implementados pela linguagem de programação *Python*, devido a sua crescente utilização em Análise de Dados e Aprendizado de Máquina [Matos 2015]. Em se tratando de bibliotecas da linguagem, foram utilizadas: 1) *scikit-Learn*[1], para os algoritmos de AM, modelos de seleção de dados e métricas estatísticas, por se tratar de uma biblioteca de fácil usabilidade, melhor qualidade na construção dos códigos, colaboração, documentação e também desempenho [Brownlee 2014] 2) *imbalanced-Learn*[2], biblioteca derivada da *scikit-learn* somente para a realização de técnicas de balanceamento em conjuntos de dados, e 3) *Pandas*[3], que oferece estruturas de dados e operações compatíveis com a biblioteca *scikit-learn*.

Os testes realizados foram oriundos de uma máquina com processador Intel Core i5-6200U com frequência de até 2,40 GHz, 8 GB de Memória Principal e 1 TB de Memória Secundária.

3.2.1 Testes Elaborados

Neste trabalho de pesquisa, os testes experimentais contemplaram seis diferentes tipos de execuções, as quais foram elaboradas e divididas de acordo com técnicas de Balanceamento de Dados (explicadas na Seção 3.3.2). A seguir, cada teste experimental é listado.

- Teste 1: Execução do Floresta Aleatória sem balanceamento de dados;
- Teste 2: Execução do Aumento de Gradiente sem balanceamento de dados;
- Teste 3: Execução do Floresta Aleatória com balanceamento do tipo *Oversampling*;
- Teste 4: Execução do Aumento de Gradiente com balanceamento do tipo *Oversampling*;
- Teste 5: Execução do Floresta Aleatória com balanceamento do tipo *Undersampling*;
- Teste 6: Execução do Aumento de Gradiente com balanceamento do tipo *Undersampling*;

Além das análises individuais, as previsões realizadas de cada algoritmo são comparadas através de métricas estatísticas, descritas na Seção 3.5.

[1]Site da biblioteca *scikit-Learn*: https://scikit-learn.org/
[2]Site da biblioteca *imbalanced-Learn*: https://imbalanced-learn.readthedocs.io/en/stable/index.html
[3]Site da biblioteca *Pandas*: https://github.com/pandas-dev/pandas

3.3 Preparação dos Dados

Os dados utilizados no presente trabalho foram fornecidos em forma de *backup* do banco de dados do *QAcadêmico* (antigo sistema de gestão acadêmica), por parte do setor administrativo da instituição. A partir do período de 2016.1, houve uma migração deste sistema, quando o *QAcadêmico* deixou de ser utilizado para dar lugar ao Sistema Unificado de Administração Pública (SUAP).

As informações contidas no *QAcadêmico* que foram disponibilizadas são referentes aos cursos ofertados e alunos matriculados de todos os *campi* do IFPB, desde suas fundações até o ano de desativação do sistema em 2015. Logo, foi necessária uma filtragem destas informações para que somente dados dos alunos do CST em Telemática do IFPB *campus* Campina Grande fossem organizados em um único documento.

A filtragem realizada resultou num conjunto de dados em que cada tupla de informações representa uma matrícula de aluno no curso, ou seja, se um aluno tiver mais de uma matrícula no curso, cada matrícula será representada por uma tupla do conjunto, ao invés de uma tupla representar um aluno, independente de quantas matrículas ele tiver. Das 720 matrículas obtidas para realização das análises, 429 são matrículas evadidas e 291 são matrículas não evadidas.

3.3.1 Atributos Descritivos

Para que os algoritmos de AM possam realizar a classificação das amostras inseridas, é necessário que estas contenham atributos que as caracterizem, com o propósito dos algoritmos criarem possíveis padrões que justifiquem as suas classificações.

Existem dois tipos gerais de atributos, sendo eles: qualitativos e quantitativos. Os atributos qualitativos representam uma característica de qualidade, que não pode ser medida, por exemplo, a cor dos olhos de uma pessoa, cujo valor esperado pode ser castanho, azul ou preto. Já os atributos quantitativos assumem valores numéricos, que podem ser discretos (a partir de um grupo com possíveis valores), ou contínuos (a partir de um intervalo ao qual os valores podem pertencer), por exemplo, a quantidade de pessoas atendidas num consultório que possuem diabetes [Action 2019].

Com base nos atributos descritivos utilizados em pesquisas com resultados satisfatórios quanto a previsão de evasões [Manhães, Cruz e Zimbrão 2014; Melo 2016], alguns destes atributos disponíveis no *QAcadêmico* foram utilizados no trabalho. No total, foram considerados dez atributos descritivos, dos quais nove deles são quantitativos, sendo utilizados pelos algoritmos para a classificação, enquanto o último é utilizado como marcação para os dados, sendo do tipo qualitativo. Abaixo seguem os atributos descritivos considerados na pesquisa, sendo suas siglas e o que representam:

- por_curso: a porcentagem que o aluno realizou desde a sua matrícula, que no âmbito do IFPB *campus* Campina Grande, é calculada com base na quantidade de disciplinas

já aprovadas em relação a quantidade de todas as disciplinas do curso;

- cre: o Coeficiente de Rendimento Escolar (CRE), que corresponde a uma média ponderada das notas finais obtidas em cada disciplina cursada, seja com aprovação ou não, determinado pela Equação 3.1, onde N_k representa a nota obtida na disciplina e H_k representa a carga horária da disciplina [IFPB 2016]:

$$CRE = \frac{(N_1 * H_1) + (N_2 * H_2) + ... + (N_k * H_k)}{(H_1 + H_2 + ... + H_k)} \tag{3.1}$$

- qtd_p_curs: a quantidade de períodos letivos cursados desde a realização da matrícula do aluno até a conclusão do curso ou até o período 2015.1, o último semestre considerado na pesquisa;

- qtd_d_curs: a quantidade total de disciplinas obrigatórias do curso que os alunos precisam ser aprovados para conseguir a formação;

- qtd_d_ap: a quantidade de disciplinas em que o aluno obteve aprovação no decorrer do curso;

- qtd_d_rep_n: a quantidade de disciplinas em que o aluno foi reprovado por nota no decorrer do curso;

- qtd_d_rep_f: a quantidade de disciplinas em que o aluno foi reprovado por falta no decorrer do curso;

- qtd_d_c: a quantidade de disciplinas em que o aluno se matriculou, mas de alguma forma foi cancelada pela instituição;

- qtd_d_t: a quantidade de disciplinas em que o aluno veio a realizar o trancamento, para ser cursada posteriormente;

- evadiu: atributo responsável por mostrar se o aluno evadiu ou não do curso, com apenas dois valores possíveis, 0 para representar a não evasão e 1 para representar a evasão do curso. Este atributo serve como marcação das amostras do conjunto para que seja possível gerar os resultados das métricas utilizadas.

Vale salientar que, para o atributo "evadiu" no conjunto de dados utilizado, as matrículas marcadas pelo *QAcadêmico* como jubilada, transferida internamente ou externamente, afastada, evasão, cancelada e não concluída, foram consideradas como matrículas evadidas, enquanto que para as matrículas marcadas como não evadidas, foram consideradas as situações: matriculada, trancada, concludente, concluída, falecida, aguardando colação de grau, formada e aguardando Exame Nacional de Desempenho de Estudantes (ENADE).

3.3.2 Balanceamento de Dados

Muitos aspectos podem influenciar o desempenho de um modelo de classificação criado por um sistema de aprendizado supervisionado. Um desses aspectos está relacionado com a diferença entre o número de amostras de cada uma das classes, e esta diferença é conhecida como Desbalanceamento de Dados. Trazendo esta definição para o contexto da Evasão Estudantil, seria o caso de ter a maioria de alunos não evadindo enquanto poucos evadem de seus cursos, ou vice-versa. Quando o desbalanceamento está presente nos conjuntos de dados, os sistemas de aprendizado encontram dificuldades em assimilar informações que justifiquem a presença da classe minoritária. Nestas condições, os modelos de classificação que recebem as bases de dados desbalanceadas com o propósito de classificar os dados, são induzidos a predizerem a classe majoritária com bastante frequência, deixando-os "viciados" em um único resultado [Prati, Batista e Monard 2003].

Uma solução para este problema, é o balanceamento artificial do conjunto de dados. Existem vários métodos capazes de realizar o balanceamento, classificados na literatura de duas formas, seja ou acrescentando artificialmente amostras da classe minoritária (técnicas de *Oversampling*), ou retirando amostras da classe majoritária (técnicas de *Undersampling*). Para o presente trabalho, as duas formas de balanceamento são utilizadas, sendo de forma aleatória devido a sua simples implementação, no caso sem levar em consideração se as amostras do conjunto são ou não relativamente parecidas ou qualquer outra forma de seleção sofisticada de dados.

3.4 Algoritmos de AM Utilizados

Como explicado na Seção 3.1, este trabalho compara dois Algoritmos que utilizam a estratégia de Classificação (Floresta Aleatória e Aumento de Gradiente), os quais são utilizados para implementar o mesmo Algoritmo de Classificação de Dados (Árvore de Decisão), a fim de encontrar aquele que obtenha os melhores resultados na previsão de evasões estudantis. Os algoritmos utilizados são melhores apresentados e descritos nas próximas seções.

3.4.1 Algoritmos de Classificação

Como já explicado na Seção 2.1.5, a Classificação é considerada uma vertente do AM Supervisionado, que com ela, é possível identificar a qual categoria já assimilada uma amostra pode pertencer. Diversos são os algoritmos capazes de realizar a classificação de dados, porém não é possível afirmar que um algoritmo seja melhor para todos os problemas existentes, sem levar em consideração o tamanho e o tipo do conjunto de dados a ser utilizado, além também da classificação que é requerida. Abaixo seguem breves descrições de alguns dos algoritmos mais utilizados em pesquisas de classificação de dados [Le 2018]:

- *K-Nearest Neighbor*: algoritmo baseado na premissa de descobrir o vizinho mais pró-

ximo de uma amostra. Neste caso, classificando uma amostra de acordo com as respectivas amostras k vizinhas mais próximas, pertencentes ao conjunto de dados para treino. O algoritmo calcula a distância da amostra em evidência, para cada amostra do conjunto de treinamento e então ordena todas elas da mais próxima para a de maior distância [Marques 2015];

- *Naive Bayes*: algoritmo baseado no teorema de Bayes – teoria probabilística para descrever a ocorrência de um evento –, em que o classificador entende que a presença de um atributo determinístico de uma amostra não está relacionado com a presença de outro atributo. Por exemplo, um fruto pode ser considerado como uma maçã se for vermelho, redondo, e tiver cerca de 3 polegadas de diâmetro. Mesmo que estes atributos dependam uns dos outros ou da existência de outras características, todos eles contribuem de forma independente para a probabilidade de que este fruto é uma maçã. Por isso que é conhecido como "naive", ou, ingênuo [Ray 2017].

- *Support Vector Macchine*: algoritmo que utiliza abordagens geométricas para a classificação de dados. Em um determinado conjunto de dados, cada amostra pode ser vista como um ponto qualquer em um espaço delimitado, em que o aprendizado deste algoritmo consiste em "dividir" as amostras positivas das negativas neste espaço através de vetores [Oguri 2007].

- Árvore de Decisão: algoritmos que trabalham em torno de uma hierarquia de questões do tipo se e senão (*if-else*), através da criação de estruturas de dados do tipo árvore, que levam a uma decisão no final de sua utilização. Uma das principais vantagens deste algoritmo é que são naturalmente fáceis de visualizar e conceituar, permitindo uma inspeção detalhada e não apenas a resposta final [Müller e Guido 2016].

O Algoritmo de Classificação utilizado nesta pesquisa é o Árvore de Decisão, que funciona a partir da criação de uma estrutura de dados do tipo árvore de busca binária. Uma árvore possui um conjunto de elementos capazes de armazenar informações, conhecidos como nós. Toda árvore possui um nó raiz, como sendo o maior nível hierárquico da estrutura, que serve como ligação para outros elementos, denominados nós filhos, armazenando um conjunto de regras que os dados devem seguir para a sua classificação, que por sua vez podem possuir elementos filhos também, e assim por diante. O nó que não possui filho é conhecido como nó folha, que representa a decisão a ser tomada, no caso as classificações possíveis [Campos 2017].

Para entender a classificação de uma Árvore de Decisão, tem-se o seguinte exemplo: imagine que seja necessário distinguir quatro animais (urso, falcão, pinguim e golfinho), e o objetivo é chegar na resposta certa fazendo o menor número possível de perguntas. A primeira pergunta a ser feita pode ser se o animal tem penas, e em caso afirmativo resta somente dois animais possíveis. Se a resposta for sim, então a próxima pergunta deverá ser capaz de distinguir um falcão de um pinguim, por exemplo, se o animal pode voar ou não.

Porém, se o animal não tiver penas, então é necessário uma pergunta que possa diferenciar um urso de um golfinho, como saber se o animal tem patas, ou até mesmo se vive na água [Müller e Guido 2016].

Na Figura 3.1 é possível observar a árvore de decisão para o exemplo anterior, onde cada nó da árvore representa uma questão ou um nó terminal que contém a resposta da pergunta anterior, ou a própria classificação.

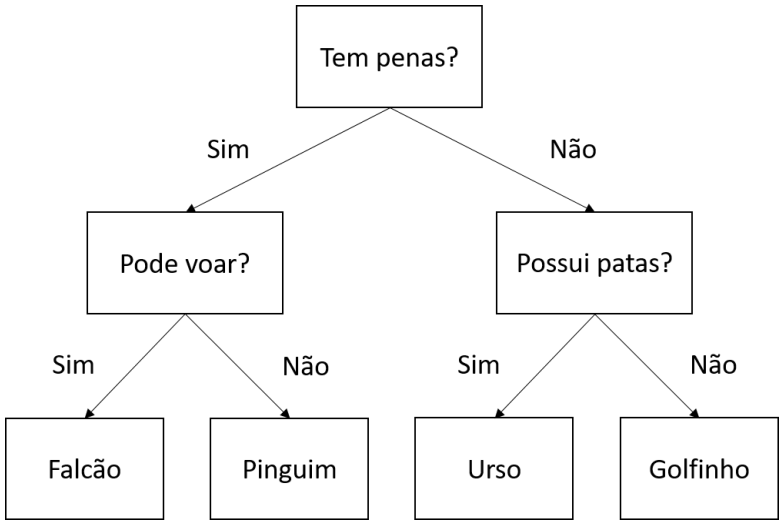

Figura 3.1: *Árvore de Decisão para a distinção de animais (urso, falcão, pinguim ou golfinho).*

A ilustração anterior apresenta uma estratégia de árvore binária, em que cada nó tem, no máximo, dois filhos. Nesse exemplo, a distinção em quatro animais é relativamente simples de ser feita. A previsão de evasões estudantis também pode seguir a mesma estratégia de árvore, no entanto, o volume de dados utilizado pode gerar centenas ou até milhares de nós de decisão. Dessa forma, é impossível realizar uma análise detalhada da mesma, assim como construir uma possível ilustração.

3.4.2 *Ensemble Methods*

Os *Ensemble Methods* são técnicas que criam vários modelos a partir de um algoritmo base de classificação, e os combinam para produzir resultados aprimorados, produzindo soluções mais precisas que apenas um algoritmo de classificação. Estas técnicas são amplamente utilizadas por vencedores de diversas competições, uma vez que estas são capazes de criar modelos menos dependentes dos Conjuntos de Dados de treinamento e consequentemente, modelos com viés baixo quanto as classificações [Demir 2016].

Essencialmente, a criação e utilização dos algoritmos que utilizam estes métodos devem considerar, pelo menos, três importantes questões: 1) como particionar os dados utilizados 2) como selecionar os modelos criados pelos algoritmos, e 3) quais métodos utilizar para combinar os resultados obtidos [Julian 2016]. Simplificando estas questões, é possível compreender essas propostas em duas categorias, de acordo com seu método de funcionamento, sendo elas [Julian 2016]:

- Algoritmos de Agregação: em que vários modelos são criados igualmente, onde sua variância se dá pela divisão do conjunto de treinamento em subconjuntos distintos para cada modelo criado;

- Algoritmos de Impulso: onde os modelos são criados de forma distinta, em que o próximo modelo a ser criado tem o intuito de obter melhor desempenho que o modelo anterior, utilizando distribuições ponderadas com base em taxas de erro e criação de modelos.

Algoritmo Floresta Aleatória

O Algoritmo Floresta Aleatória, empregado neste trabalho, utiliza o método de Agregação. Seu funcionamento ocorre a partir da criação de várias Árvores de Decisão em que recebem subconjuntos distintos a partir do conjunto de dados original, além de utilizarem combinações diferentes de atributos descritivos em seus nós para realizarem a classificação. Depois que uma determinada quantidade – indicada na construção do algoritmo – de árvores são geradas, cada uma apresenta um voto para uma classificação do problema, considerando uma amostra de entrada. Então, a classificação com maior quantidade de votos será escolhida na previsão da amostra do algoritmo [Silva, Almeida e Yamakami 2012].

A ideia de uma Floresta Aleatória é que cada árvore tenha um bom desempenho nas previsões a partir da aleatoriedade possibilitada pelo algoritmo, porém geralmente ocorre um sobre-ajuste, situação quando um algoritmo tem bom desempenho na classificação apenas nas amostras do conjunto utilizado em seu treinamento, tornando-os incapazes de prever com eficiência novos dados. Ao construir diversas Árvores de Decisão distintas numa Floresta Aleatória, quando se adaptam de maneiras diferentes para a classificação das amostras, é possível reduzir a quantidade de sobre-ajuste pela média dos resultados de cada árvore.

A Floresta Aleatória não é somente a execução de várias Árvores de Decisão para que seja feita uma média dos resultados. Diferentemente deste tipo de execução, o algoritmo é capaz de dividir, de forma aleatória, o conjunto de dados em subconjuntos diferentes para cada Árvore de Decisão criada, ou seja, cada árvore do algoritmo tem o seu próprio subconjunto de dados para a realização de treino e teste. Além de que para cada nó de decisão das árvores construídas, o algoritmo determina, também de forma aleatória, uma parte dos atributos descritivos do conjunto como tomadas de decisão de cada nó. Ou seja, a aleatoriedade do algoritmo ocorre tanto no conjunto de dados quanto nos atributos utilizados na classificação, fazendo com que as árvores sejam distintas umas das outras [Müller e Guido 2016].

Na Figura 3.2 consta um exemplo em que é possível entender o conceito de uma Floresta Aleatória. O algoritmo cria três Árvores de Decisão, onde cada uma delas recebe um subconjunto aleatório a partir do Conjunto de Dados informado ao algoritmo, utilizado para o treino das árvores. Cada nó de decisão das árvores criadas utiliza uma combinação aleatória dos atributos descritivos, utilizando desde apenas um atributo em um nó até o total de atributos possíveis, para então classificar as amostras passadas. No exemplo, ao inserir

uma amostra nas três árvores criadas, as Árvores 1 e 3 classificam a amostra como do Tipo 1, enquanto a Árvore 2 classifica como do Tipo 2. Sendo assim, fazendo uma média dos resultados das três árvores, a Floresta Aleatória entende que a classificação para a amostra inserida seja do Tipo 1.

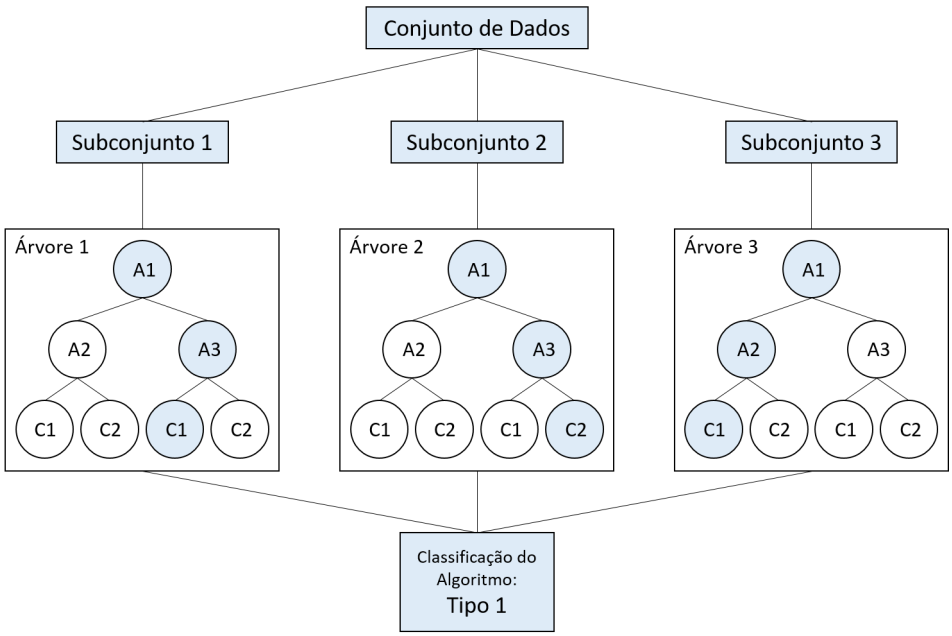

Figura 3.2: *Representação do funcionamento do algoritmo Floresta Aleatória.*

É importante entender que embora o termo "média" seja utilizado na definição do funcionamento do algoritmo de Floresta Aleatória, trata-se do valor de classificação possível que mais ocorre, o que seria a moda das classificações previstas.

Algoritmo Aumento de Gradiente

O algoritmo comparado com o Floresta Aleatória que utiliza o método de Impulso, é o Aumento de Gradiente, em que na literatura faz uso frequentemente de modelos de Árvores de Decisão, na qual a classificação dos dados ocorre através da frequência de previsões referentes a maior classe, assim como o Floresta Aleatória. Porém, diferente do Floresta Aleatória, o algoritmo não utiliza técnicas de aleatoriedade no conjunto de dados para a construção de subconjuntos, nem uma aleatoriedade na utilização dos atributos descritivos para a classificação. Em vez disso, o Aumento de Gradiente utiliza distribuições ponderadas – variáveis de diferentes pesos – na criação das Árvores de Decisão, onde cada árvore criada tenta obter um melhor desempenho que a anterior a partir da adição destas variáveis. Além destas variáveis, é possível atribuir ao algoritmo um valor para definir até que nível de profundidade as árvores podem chegar, cujo valor geralmente varia de 1 a 5, para que a criação das árvores se deem de forma mais rápida. Estas técnicas contribuem para que as árvores do algoritmo realizem as classificações de forma mais eficaz em termos de processamento [Julian 2016].

O conceito principal do Aumento de Gradiente é a utilização de vários modelos simples (conhecidos como aprendizes fracos) para a criação de um modelo com bom desempenho.

Por exemplo, uma Árvore de Decisão pode fornecer boas previsões para uma parte de um conjunto de dados. Desta forma, o Aumento de Gradiente cria cada vez mais árvores para melhorar o desempenho da anterior, de forma iterativa. O Aumento de Gradiente utilizando Árvores de Decisão é frequentemente usufruído em pesquisas e competições de AM, por serem mais sensíveis a configurações de parâmetros do que o Floresta Aleatória, fornecendo melhores resultados se ajustado corretamente para os problemas propostos. Entretanto, a literatura não recomenda a utilização do algoritmo para grandes conjuntos de dados devido seu funcionamento de forma sequencial, o que pode aumentar o tempo de processamento para classificação das amostras [Müller e Guido 2016].

Na Figura 3.3 é exemplificado o funcionamento do Aumento de Gradiente utilizando Árvores de Decisão. Nesse exemplo, percebe-se que todas as árvores criadas trabalham com distribuições ponderadas, tendo o propósito de que a próxima árvore criada obtenha um melhor desempenho em relação a árvore anterior criada, de modo a corrigir os "erros" do modelo antecessor. Estas distribuições realizam ajustes quanto ao processamento do algoritmo, não realizando diversas requisições ao conjunto de dados, diminuindo a velocidade de tempo de processamento do mesmo. Ao inserir uma amostra nas três árvores criadas no exemplo, as Árvores 1 e 2 classificam a amostra como do Tipo 2, enquanto a Árvore 3 classifica como do Tipo 1, sendo assim, a classificação do Aumento de Gradiente para a amostra será do Tipo 2, pela classificação de maior presença nas árvores.

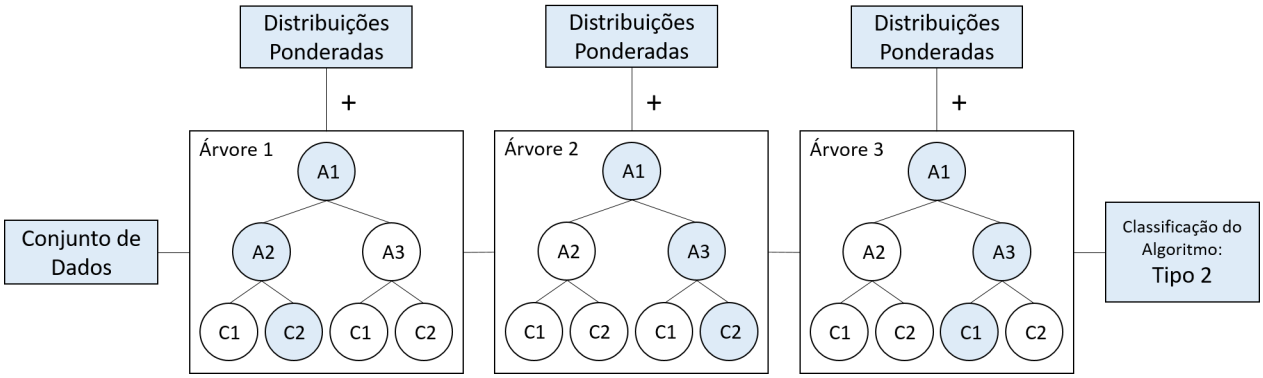

Figura 3.3: *Representação do funcionamento do algoritmo Aumento de Gradiente.*

Para os dois *Ensemble Methods* foram utilizadas as mesmas configurações básicas de construção, sendo elas: 1) os dois algoritmos criam 100 Árvores de Decisão para a realização das classificações 2) a profundidade máxima de cada árvore criada é de até 3 níveis 3) como cada árvore tem até 3 níveis de profundidade, 8 é o valor máximo de construção dos nós de classificação, e 4) as Árvores de Decisão podem utilizar combinações com até todos os 9 atributos descritivos em um único nó de decisão. Adicionalmente, o Aumento de Gradiente possibilita a configuração de distribuições ponderadas, em que para o presente de trabalho foram utilizadas as seguintes: 1) função de perda por desvio binomial para classificação binária 2) 0,2 como taxa de aprendizado em cada árvore criada, e 3) 1 como valor de subamostragem, ou seja, utilizando todo o conjunto de dados em que foi passado para o

modelo como treino.

3.5 Métricas Estatísticas

A fim de comparar os resultados das previsões oriundas dos algoritmos utilizados, são utilizadas cinco métricas estatísticas amplamente empregadas em pesquisas que envolvem algoritmos de classificação em Aprendizado de Máquina. Quatro das cinco métricas utilizadas são baseadas no conceito da Matriz de Confusão, que corresponde a uma tabela com as possibilidades de classificação para as previsões dos algoritmos.

A Matriz de Confusão é uma estratégia computacional de avaliação que facilita o entendimento das previsões dos algoritmos de classificação, permitindo determinar e analisar o desempenho dos algoritmos com diferentes métricas estatísticas [Souza 2019]. Como mostrado na Figura 3.4, as duas linhas representam os valores reais, oriundos do conjunto de dados, enquanto que as duas colunas da matriz representam os valores previstos pelos algoritmos de classificação, resultando em quatro possíveis previsões, sendo elas:

- Verdadeiro Positivo (VP): quando a amostra positiva é prevista de forma correta. Por exemplo, quando um aluno que evadiu é previsto pelos algoritmos.
- Falso Positivo (FP): quando a amostra negativa é prevista de forma positiva. Por exemplo, quando um aluno que não evadiu é previsto que evadiu pelos algoritmos;
- Falso Negativo (FN): quando a amostra positiva é prevista de forma negativa. Por exemplo, quando um aluno que evadiu é previsto que não evadiu pelos algoritmos;
- Verdadeiro Negativo (VN): quando a amostra negativa é prevista corretamente. Por exemplo, quando um aluno que não evadiu é previsto como não evadido;

		VALOR PREVISTO	
		POSITIVO	NEGATIVO
VALOR REAL	POSITIVO	(VP) VERDADEIRO POSITIVO	(FP) FALSO POSITIVO
	NEGATIVO	(FN) FALSO NEGATIVO	(VN) VERDADEIRO NEGATIVO

Figura 3.4: *Representação da uma Matriz de Confusão.*

Os resultados previstos pelos algoritmos serão comparados com as marcações do conjunto de dados, para que os valores de cada possibilidade da Matriz de Confusão sejam calculados, bem como as métricas estatísticas também.

3.5.1 Acurácia

O termo "acurária" foi criado a partir de um neologismo com a palavra da língua inglesa, *accuracy* [Cintra 2018]. Mikhail e Ackermann apresentam a acurácia como sendo o grau de proximidade de uma estimativa com seu parâmetro (ou valor verdadeiro). Esse valor reflete a proximidade de uma grandeza estatística ao valor do parâmetro para o qual ela foi estimada, ou seja, é a proporção de previsões corretas, sem levar em consideração o que é positivo e negativo [Mikhail e Ackerman 1976]. Sua fórmula é representada na Equação 3.2:

$$Acurácia = \frac{(VN + VP)}{(VN + FP + FN + VP)} \tag{3.2}$$

3.5.2 Precisão

Conhecida na língua inglesa como *precision*, representa o grau de consistência da grandeza medida em sua média, que está diretamente ligada com a dispersão da distribuição das observações. Nesse caso, a precisão é a proporção de previsões positivas verdadeiras para o total de previsões positivas [Mikhail e Ackerman 1976]. Sua fórmula é representada na Equação 3.3:

$$Precisão = \frac{VP}{(VP + FP)} \tag{3.3}$$

3.5.3 Sensibilidade

É a proporção de previsões positivas para o total de amostras positivas do conjunto de dados, ou seja, a capacidade do sistema em predizer corretamente a condição para casos que realmente a têm [Zhu, Zeng e Wang 2010]. A Sensibilidade é conhecida na língua inglesa como *Recall*, sua fórmula é representada na Equação 3.4:

$$Sensibilidade = \frac{VP}{(VP + FN)} \tag{3.4}$$

3.5.4 Taxa de Falsa Previsão Positiva

Uma métrica que não é comumente utilizada na literatura, tanto que o nome definido no presente trabalho é de elaboração própria dos autores. A Taxa de Falsa Previsão Positiva trata-se de uma métrica que compara a quantidade de Falsos Positivos com a quantidade de Falsos Negativos previstos. O valor ideal para a métrica é de 50%, indicando que os modelos estão prevendo os dados de forma imparcial, visto que a quantidade de Falsos Positivos será próxima à quantidade de Falsos Negativos. Um valor distante, seja acima ou abaixo, de 50% representa uma inclinação dos modelos para preverem uma maior quantidade de uma

previsão falsa. Sua fórmula é representada na Equação 3.5:

$$Taxa\ de\ Falsa\ Previsão\ Positiva = \frac{FP}{(FP + FN)} \tag{3.5}$$

3.5.5 Tempo de Processamento

Sendo a única métrica do trabalho que não é baseada nas possibilidades da Matriz de Confusão, o tempo de processamento serve para calcular o desempenho do algoritmo. Neste trabalho o tempo de processamento foi computado a partir dos valores do tempo final do processamento (quando o algoritmo termina seus cálculos) e tempo inicial (quando o algoritmo inicia seus cálculos), gerando o resultado da métrica. Sua fórmula é representada na Equação 3.6, em que T_f representa o tempo final e T_i representa o tempo inicial do processamento, onde o resultado é contabilizado na unidade de milissegundos:

$$Tempo\ de\ Processamento = T_f - T_i \tag{3.6}$$

3.6 Conclusão do Capítulo

Este capítulo apresentou, detalhadamente, os Algoritmos utilizados nos testes experimentais e as métricas, que serão usadas para comparar as duas estratégias. Ademais, foram apresentados os dados a serem utilizados pelos algoritmos, no caso referente a alunos do CST em Telemática do IFPB *campus* Campina Grande, e os atributos que descrevem a situação dos mesmos no curso. No próximo capítulo, serão apresentados os resultados deste trabalho de pesquisa, indicando as comparações entre os algoritmos de acordo com o balanceamento ou não do Conjunto de Dados.

Capítulo 4

Resultados

Os resultados do trabalho são divididos e analisados por testes experimentais, os quais foram apresentados na Seção 3.2.1. Os Testes realizados são divididos nesta Seção entre Testes com o Conjunto de Dados Desbalanceado, Testes com o Conjunto de Dados Balanceado com *Oversampling* e Testes com o Conjunto de Dados Balanceado com *Undersampling*, conforme descrito na Seção 3.1. Cada Teste apresentado estende a estratégia do Algoritmo *K-Fold*, com K=10, resultando em 10 diferentes análises. Os valores apurados nas métricas estatísticas são computados a cada execução e sua análise ocorre a partir da média de cada métrica. Todos os resultados são apresentados graficamente e construídos nas mesmas escalas, tanto percentual como de tempo.

4.1 Testes com o Conjunto de Dados Desbalanceado

A utilização do Conjunto de Dados Desbalanceado significa que nenhuma técnica para igualar o número de amostras das duas classes trabalhadas (não evadida e evadida) foi utilizada. Sendo assim, os modelos foram treinados de forma desbalanceada.

4.1.1 Teste 1: Floresta Aleatória sem Balanceamento

O Algoritmo Floresta Aleatória foi utilizado no Teste 1, considerando o conjunto de dados desbalanceado. Como mostrado no gráfico da Figura 4.1, os valores das métricas estatísticas, baseadas na Matriz de Confusão (Acurácia, Precisão, Sensibilidade e Taxa de Falsa Previsão Positiva), são organizados numa escala percentual (%) na parte esquerda do gráfico, enquanto o Tempo de Processamento em milissegundos (ms) é mostrado na escala de tempo na parte direita do mesmo. A divisão em duas escalas numéricas se dá por conta dos valores das cinco métricas estatísticas utilizadas não serem apresentadas na mesma unidade métrica. Esta amostragem dos gráficos é utilizada para todos os Testes do trabalho.

Para o primeiro teste executado – ainda com base na Figura 4.1 –, foram obtidos valores relativamente altos em algumas métricas, elucidando que o Floresta Aleatória pode ser um

bom algoritmo para a previsão de evasões estudantis no conjunto de dados, desde que utilizados os atributos descritivos da Seção 3.3.1. De Acurácia foram computadas 83,46% previsões corretas, seja não evasões ou evasões previstas. Em se tratando da Precisão, a métrica obteve o segundo maior valor dentre as demais, com 84,38%, mostrando que o algoritmo é um forte preditor de valores VP, enquanto que valores FP são pouco abordados. A Sensibilidade obteve o maior valor dentre as métricas, sendo 89,01%, o que reforça a ideia apresentada com o valor da Precisão, de que os valores VP são previstos com maior frequência em relação os valores FP e agora também em relação aos valores FN.

Quanto à Taxa de Falsa Previsão Positiva, a média se apresentou em torno de 59,64%, mostrando que das previsões erradas do algoritmo, a quantidade de FP foi maior que a quantidade de FN. E por fim, o Tempo de Processamento se mostrou demasiadamente alto, próximo do limite escalar especificado atingindo 14,06 milissegundos de tempo médio, sendo esta uma desvantagem de uso desse algoritmo. Nesse caso, o Algoritmo analisado não se mostra escalável, dado o aumento do volume de dados (isto é, quanto maior o volume de dados, pior será o tempo de processamento da solução).

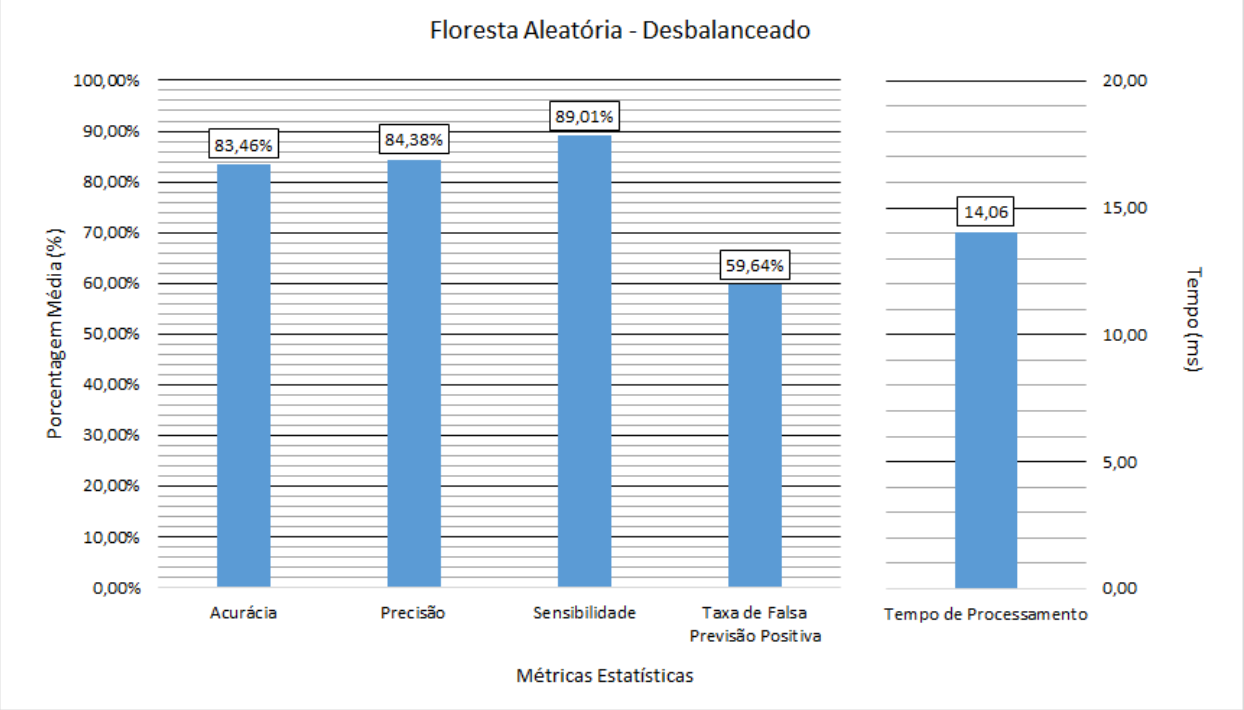

Figura 4.1: *Valores das Métricas Estatísticas dos Resultados do Teste 1.*

4.1.2 Teste 2: Aumento de Gradiente sem Balanceamento de Dados

Tanto para o Teste 1 como o Teste 2, a quantidade de amostras do conjunto de dados é a mesma, visto que não foi realizado nenhum Balanceamento de Dados para que houvesse a replicação ou retirada de determinadas informações. Logo, a quantidade se manteve como 720 registros presentes no conjunto. Como explicado na Seção 3.1, foi realizada a divisão do conjunto de dados em dois subconjuntos utilizando o método *Bootstrap*: um de treinamento,

com 63,2% dos dados e outro para teste, com 36,8% dos dados. Dessa forma, a quantidade de registros ficou 455 e 265 respectivamente, para cada uma das dez execuções dos seis Testes desta Seção.

No Teste 2, o Algoritmo utilizado foi o Aumento de Gradiente que, para as métricas estatísticas, obteve valores acima dos apresentados no Teste 1 utilizando o Algoritmo Floresta Aleatória. De acordo com a Figura 4.2, onde são mostrados os valores da análise realizada usando o Aumento de Gradiente, a Acurácia média das execuções se mostrou em torno de 89%, pouco mais de 230 registros previstos corretamente em média, dos 265 possíveis para teste, evidenciando um baixo número de previsões falsas, cerca de 28 em média. A Precisão e a Sensibilidade se mostraram ainda maiores, 90,24% e 91,64% respectivamente, onde a quantidade média de VP em cada execução ficou em torno de 142 evasões previstas pelo Algoritmo e cerca de 15 não evasões previstas como evasões (FP) utilizadas no cálculo da Precisão e em média 13 evasões não previstas (FN) utilizada na Sensibilidade.

Assim como o Teste 1, a Taxa de Falsa Previsão Positiva ultrapassou o percentual médio, atingindo 55,27%, ou seja, uma quantidade maior de FP (cerca de 15 matrículas) foi indicada pelo Aumento de Gradiente em relação a quantidade de FN (cerca de 13 matrículas). E a última métrica, o Tempo de Processamento, atingiu um valor médio de 0,90 milissegundos em cada execução do Teste, sendo apresentado próximo do limite mínimo da escala gráfica, confirmando seu bom desempenho com relação ao conjunto de dados, alcançando ótimos valores devido aos ajustes ponderados na criação de cada árvore do algoritmo.

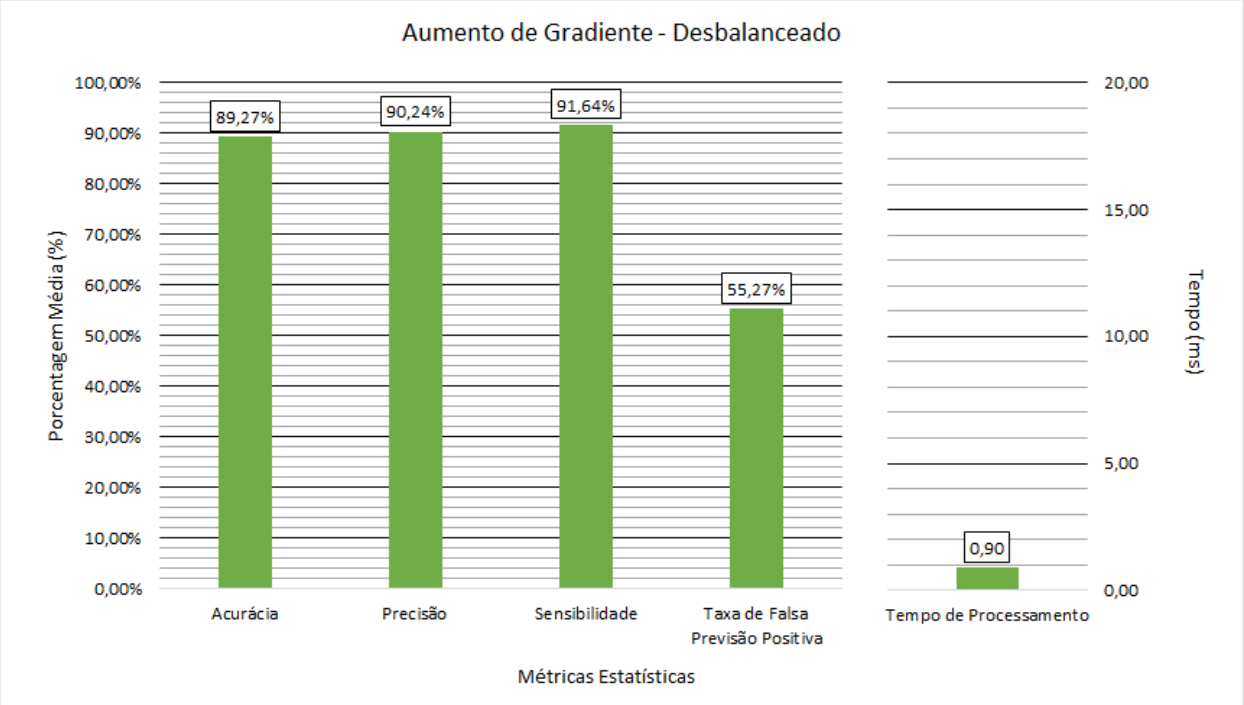

Figura 4.2: *Valores das Métricas Estatísticas dos Resultados do Teste 2.*

4.1.3 Comparação dos Testes 1 e 2

Ao ser feita uma comparação dos Testes 1 e 2, é perceptível um desempenho maior do Aumento de Gradiente em relação ao Floresta Aleatória, visto que obteve valores maiores para as métricas baseadas na matriz de confusão, apesar da diferença para elas entre os dois testes não ultrapassar 6,5 pontos percentuais. A maior diferença entre os algoritmos, em se tratando das métricas baseadas na matriz de confusão, é apresentada na Precisão, com o Aumento de Gradiente do Teste 2 sendo 6,02% maior que o Floresta Aleatória do Teste 1. A segunda maior diferença ocorre na Acurácia, sendo o Floresta Aleatória menor cerca de 5,81% que o Aumento de Gradiente.

Apesar de um algoritmo ter valores maiores nas métricas anteriormente apresentadas em relação ao outro, estes valores são próximos, o que torna a métrica Tempo de Processamento como decisiva para definir o melhor Algoritmo, uma vez que os valores dos dois algoritmos para a métrica são bem distintos, onde o Aumento de Gradiente realiza a classificação das matrículas num tempo médio de 0,9 milissegundos, enquanto o Floresta Aleatória realiza o mesmo processo em um tempo médio de 14,06 milissegundos, sendo cerca de 15 vezes mais lento.

A intuição por trás do bom desempenho quanto ao Tempo de Processamento do Algoritmo Aumento de Gradiente, está relacionada com a criação das Árvores de Decisão. Nessa estratégia, cada Árvore recebe um valor de taxa de aprendizado, fazendo com que a próxima árvore criada tenha um melhor desempenho em relação a anterior, além de trazer os dados trabalhados para a memória, não realizando várias requisições no conjunto de dados para realizar particionamentos como o Floresta Aleatória.

A Figura 4.3 consiste nos gráficos dos Testes 1 e 2 mesclados em apenas um, sendo melhor perceptível a comparação dos resultados onde o Aumento de Gradiente (barras na cor verde) tem melhores resultados que o Floresta Aleatória (barras na cor azul):

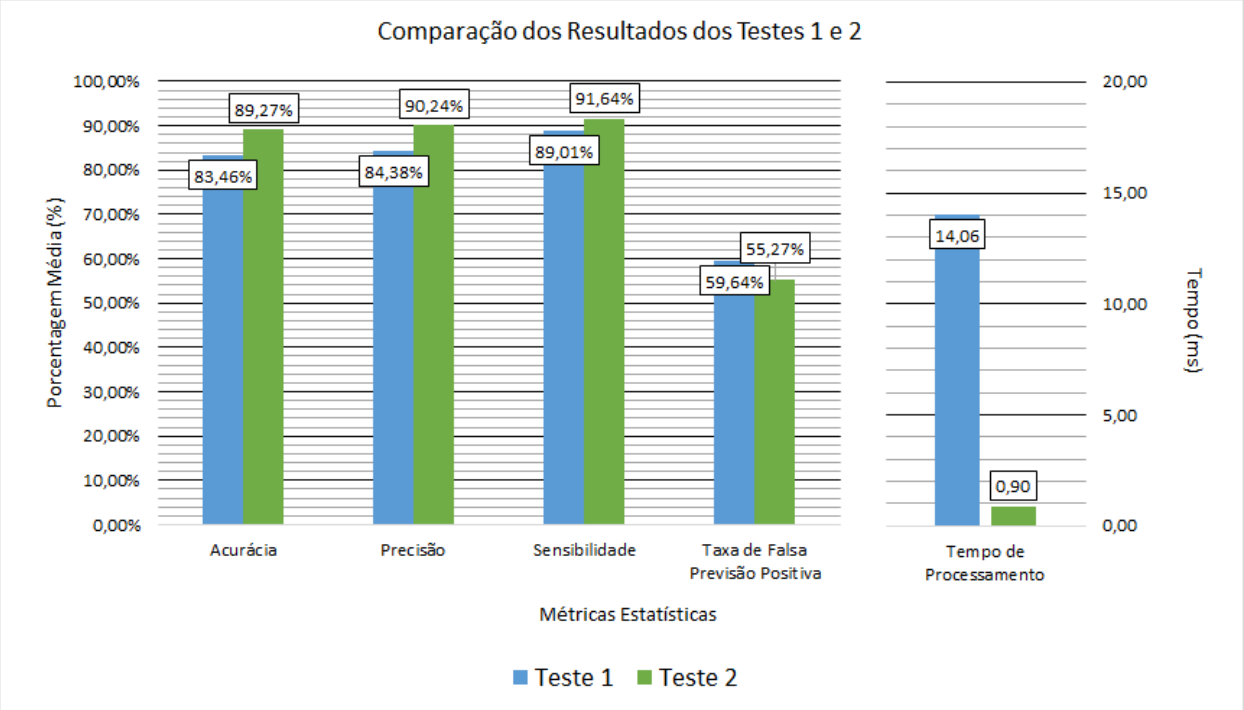

Figura 4.3: *Comparação dos Resultados dos Testes 1 e 2.*

4.2 Testes com o Conjunto de Dados Balanceado com *Oversampling*

Com o Conjunto de Dados balanceado utilizando o método *Oversampling*, ocorre uma replicação de amostras durante o treino dos algoritmos, onde as amostras da classe minoritária são replicadas para que o seu número se iguale com a quantidade de amostras da classe majoritária. Como o Conjunto de Dados em cada uma das 10 execuções de cada Teste é dividido aleatoriamente e com a reposição dos dados, não é feita uma contabilização de quantos dados de cada classe são utilizados para treino, uma vez que este número pode variar para cada execução de acordo com a divisão do conjunto, em que os treinos podem conter uma ou todas as amostras de uma classe, sendo seu número igual a quantidade de amostras da outra classe.

4.2.1 Teste 3: Floresta Aleatória com *Oversampling*

Os valores das métricas estatísticas baseadas na matriz de confusão se mostraram bem distintos no Teste do Floresta Aleatória com o Conjunto de Dados balanceado com *Oversampling*. A Acurácia obteve o segundo maior valor, com média de 81,54% de previsões realizadas corretamente. A Precisão atingiu o maior valor dentre as métricas, com cerca de 96% de previsões positivas corretas em relação ao total de previsões positivas do modelo. Distintamente da Acurácia e principalmente da Precisão, a Sensibilidade atingiu um valor médio de 71,30%, evidenciando um grande número de previsões negativas falsas.

O baixo valor notado na Sensibilidade, reflete na Taxa de Falsa Previsão Positiva que atingiu uma média de 7,77%, ou seja, a quantidade de FN ficou em média de 45 previsões por execução, muito maior que a quantidade de FP, por volta de 4 previsões a cada execução. O Tempo de Processamento diminuiu em comparação com o Teste 1 utilizando o mesmo algoritmo, porém um pouco acima do percentual gráfico médio, atingindo uma média de 12,77 milissegundos a cada execução. Na Figura 4.4 são mostrados os valores obtidos para as métricas estatísticas:

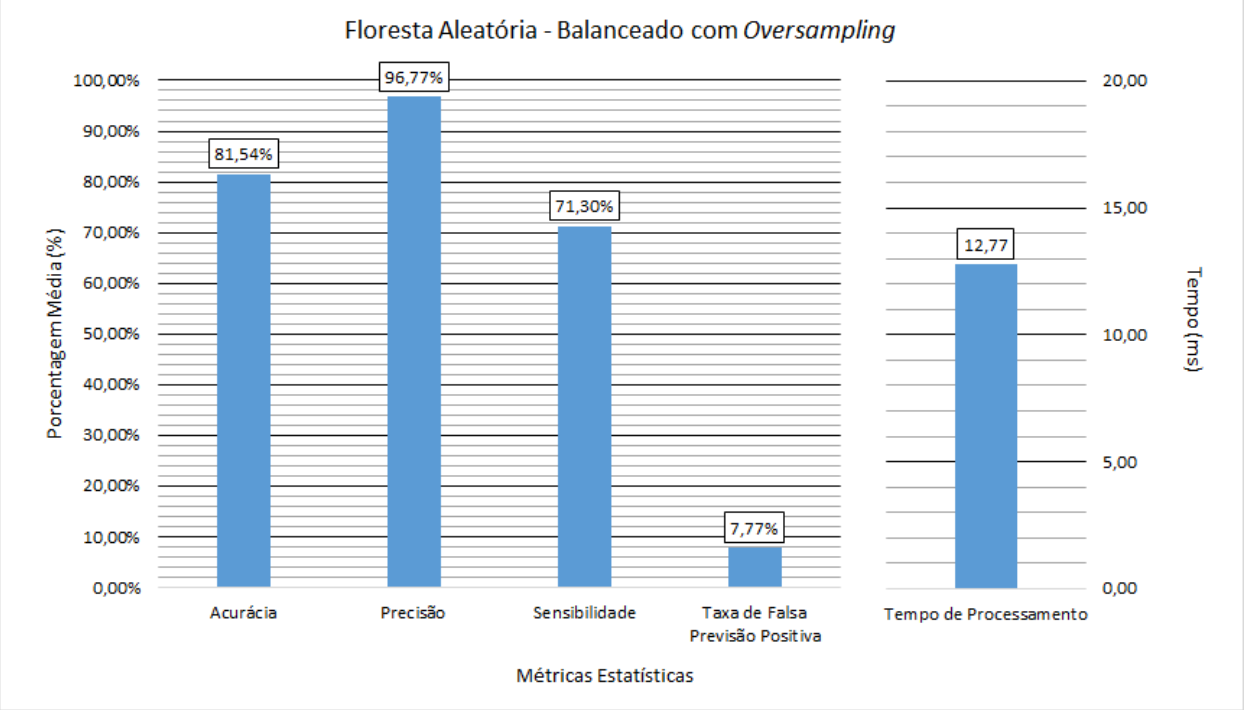

Figura 4.4: *Valores das Métricas Estatísticas dos Resultados do Teste 3.*

4.2.2 Teste 4: Aumento de Gradiente com *Oversampling*

Como mostrado na Figura 4.5, tanto Acurácia, Precisão e Sensibilidade, atingem valores acima dos 90% porém com pouca distinção de uma métrica para outra. O maior valor obtido foi o da Precisão (92,29%), refletindo a baixa quantidade de falsas previsões positivas em relação a quantidade de previsões positivas verdadeiras. O segundo maior valor obtido foi o da Sensibilidade, com média de 91,07% enquanto a Acurácia se mostrou com o terceiro maior valor, 90,16%.

Com relação a Taxa de Falsa Previsão Positiva, seu valor diminuiu em relação ao Teste 2 executando o mesmo algoritmo, atingindo 48,60%, mostrando que a quantidade de Falsos Negativos é maior que a quantidade de Falsos Positivos nos testes, porém sem uma grande distinção de quantidade, por este valor estar bem próximo do percentual médio do gráfico. Já o Tempo de Processamento aumentou em comparação com o Teste 2, atingindo uma média de 1,20 milissegundos a cada execução das previsões.

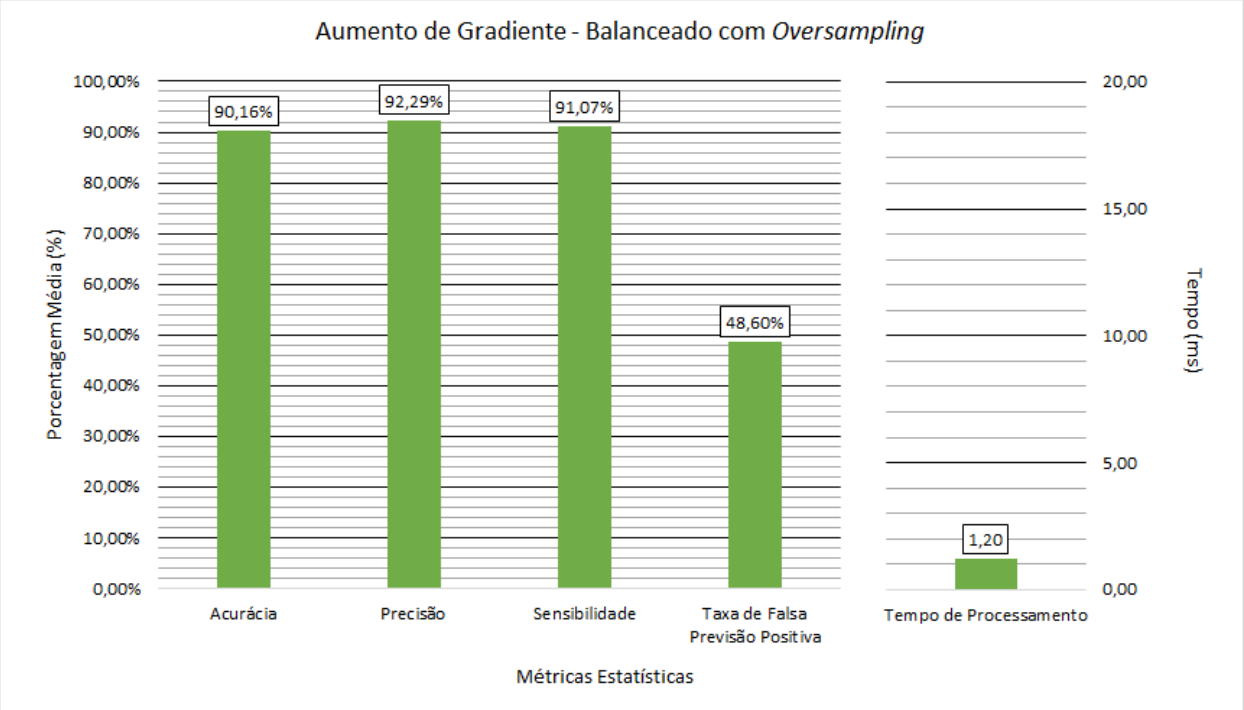

Figura 4.5: *Valores das Métricas Estatísticas dos Resultados do Teste 4.*

4.2.3 Comparação dos Testes 3 e 4

Ao comparar os dois Testes na Figura 4.6, é perceptível principalmente a diferença dos valores da Taxa de Falsa Previsão Positiva, uma vez que o Floresta Aleatória possui um valor menor cerca de 40 pontos percentuais em relação ao valor do Aumento de Gradiente. Porém, esse baixo valor para a Taxa representa que o Floresta Aleatória está inclinado em prever as amostras como Falso Negativo, sendo o motivo possível a alta quantidade de amostras não evadidas utilizadas em seu treino, o que resultou em média 45 Falsos Negativos e 4 Falsos Positivos a cada execução do algoritmo.

A Sensibilidade é outra métrica que representa o baixo desempenho do modelo no Teste 3, uma vez que compara a quantidade de Verdadeiros Positivos com a quantidade de Falsos Negativos, atingindo valor médio por volta de 71%, quase 20 pontos percentuais abaixo que a Sensibilidade do Aumento de Gradiente. Sendo assim, é notável que o Floresta Aleatória não oferece um bom desempenho nas previsões para o Conjunto de Dados balanceado com *Oversampling*, ao contrário do Aumento de Gradiente, que realiza uma quantidade de previsões próximas entre as possíveis Previsões Falsas e Verdadeiras.

E assim como nos Testes 1 e 2, com o Conjunto de Dados Desbalanceado, o Tempo de Processamento se mostrou a métrica mais diferenciativa, evidenciando o melhor desempenho do Aumento de Gradiente em comparação com o Floresta Aleatória, cerca de 10 vezes mais rápido, atingindo em média 1,20 milissegundos a cada execução, evidenciando mais uma vez que a utilização das distribuições ponderadas e também os dados trabalhados em memória ao invés de várias requisições ao conjunto são fundamentais para o rápido desempenho do Aumento de Gradiente.

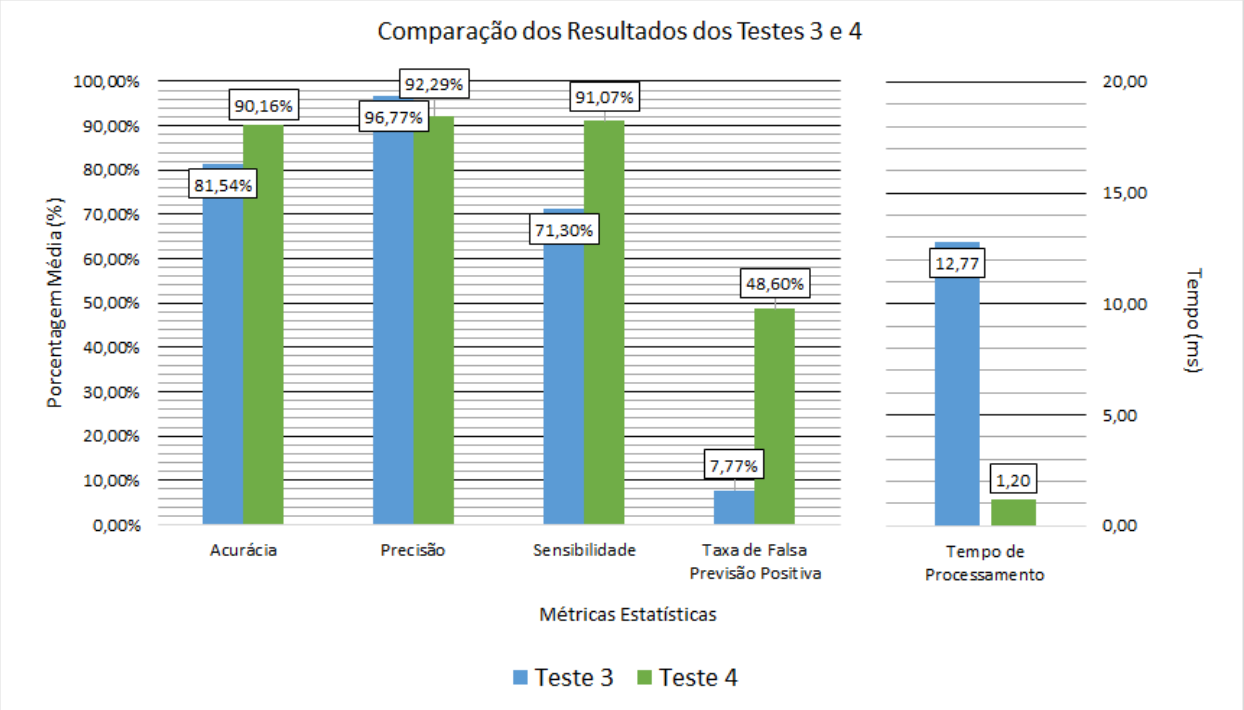

Figura 4.6: *Comparação dos Resultados dos Testes 3 e 4.*

4.3 Testes com o Conjunto de Dados Balanceado com *Undersampling*

Já para os Testes 5 e 6, o Conjunto de Dados é balanceado aleatoriamente utilizando o método *Undersampling*, ou seja, a quantidade de amostras da classe majoritária é diminuída por exclusão aleatória de amostras para que seja igual a quantidade de amostras da classe minoritária. E, assim como os Testes utilizando *Oversampling*, a quantidade de amostras de cada classe não é computada devido a aleatoriedade da divisão do Conjunto de Dados utilizando o método de *Bootstrap*, e também por causa da aleatoriedade do balanceamento.

4.3.1 Teste 5: Floresta Aleatória com *Undersampling*

Ao executar o Floresta Aleatória com o Conjunto de Dados balanceado com *Undersampling*, o algoritmo apresentou valores distintos entre as métricas baseadas na matriz de confusão, assim como no Teste 3. A Precisão atingiu o maior valor, com cerca de 94,20% de média na relação de previsões positivas verdadeiras com falsas previsões positivas. O segundo maior valor apresentado foi obtido pela Acurácia, cerca de 82,60% das amostras sendo previstas corretamente, enquanto que a Sensibilidade atingiu uma média de 75,25%, mostrando uma alta quantidade de Falsos Negativos previstos.

Justificando o valor médio relativamente baixo na Sensibilidade, a Taxa de Falsa Previsão Positiva mostra que há uma grande distinção nas falsas previsões, em que a quantidade de Falsos Negativos é bem maior que a quantidade de Falsos Positivos previstos, cerca de 38

e 8 amostras a cada execução do Teste, respectivamente, fazendo com que a métrica atinja um valor médio de 18,19%. O Tempo de Processamento do algoritmo constou a realização das classificações em cerca de 12,97 milissegundos, acima do valor médio do gráfico, de 10 milissegundos.

Os resultados das métricas do Teste 5 são apresentados na Figura 4.7:

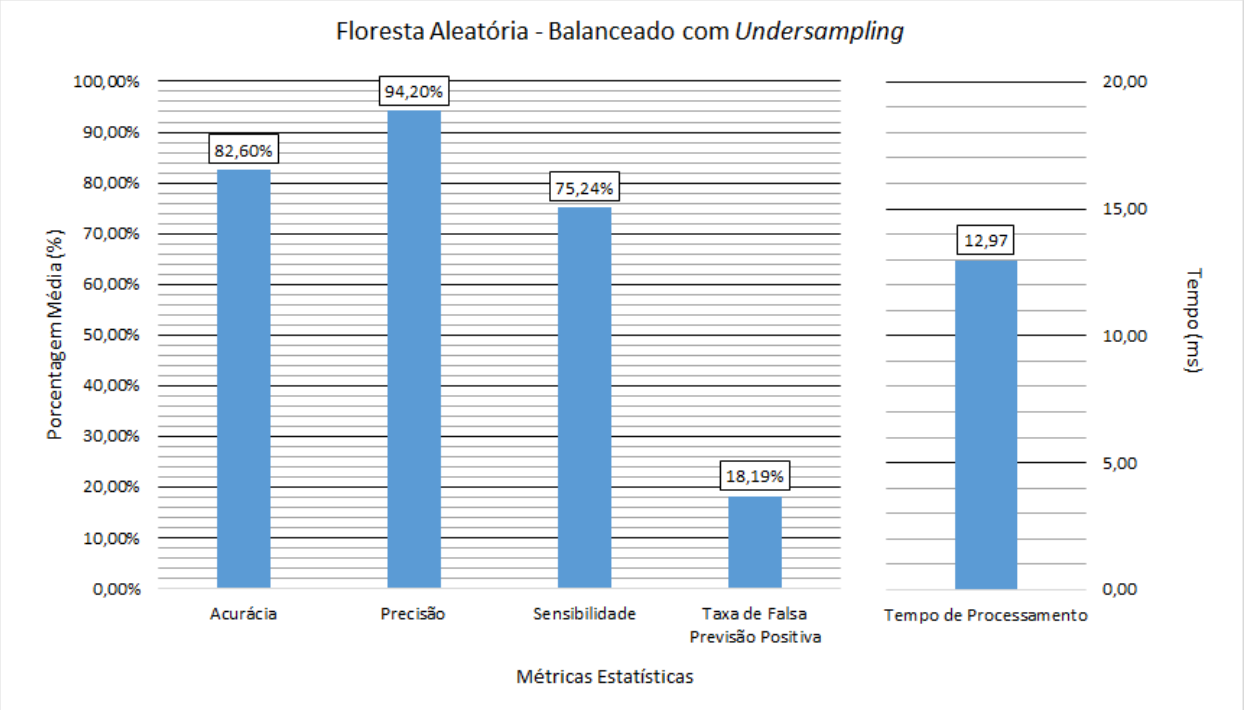

Figura 4.7: *Valores das Métricas Estatísticas dos Resultados do Teste 5.*

4.3.2 Teste 6: Aumento de Gradiente com *Undersampling*

Como o último Teste do trabalho, o Aumento de Gradiente realizou as classificações a partir de um treinamento com o Conjunto de Dados balanceado com o método *Undersampling*. Os valores médios de Acurácia, Precisão e Sensibilidade se mostraram bem próximos, sendo todos eles acima dos 88%, como mostra a Figura 4.8. A Precisão apresentou o maior valor, com média de 92,29%, já o segundo maior valor foi apresentado pela Acurácia, com cerca de 88,79% das previsões sendo realizadas corretamente, enquanto que a Sensibilidade atingiu o menor valor das três métricas, com 88,32%.

O valor da Taxa de Falsa Previsão Positiva diminuiu drasticamente em comparação com as demais métricas, quase 10 pontos percentuais menor, atingindo um valor médio de 38,86%, sendo o menor valor dos três Testes realizados com o algoritmo para a Taxa, o que mostra uma maior distinção na quantidade de Falsos Positivos com Falsos Negativos, sendo este último uma quantidade maior. O Tempo de Processamento, assim como nos outros testes com o algoritmo, se mostrou baixo, próximo do limite inferior do gráfico, realizando as classificações em cerca de 1,19% milissegundos, valor bem próximo do Teste 5.

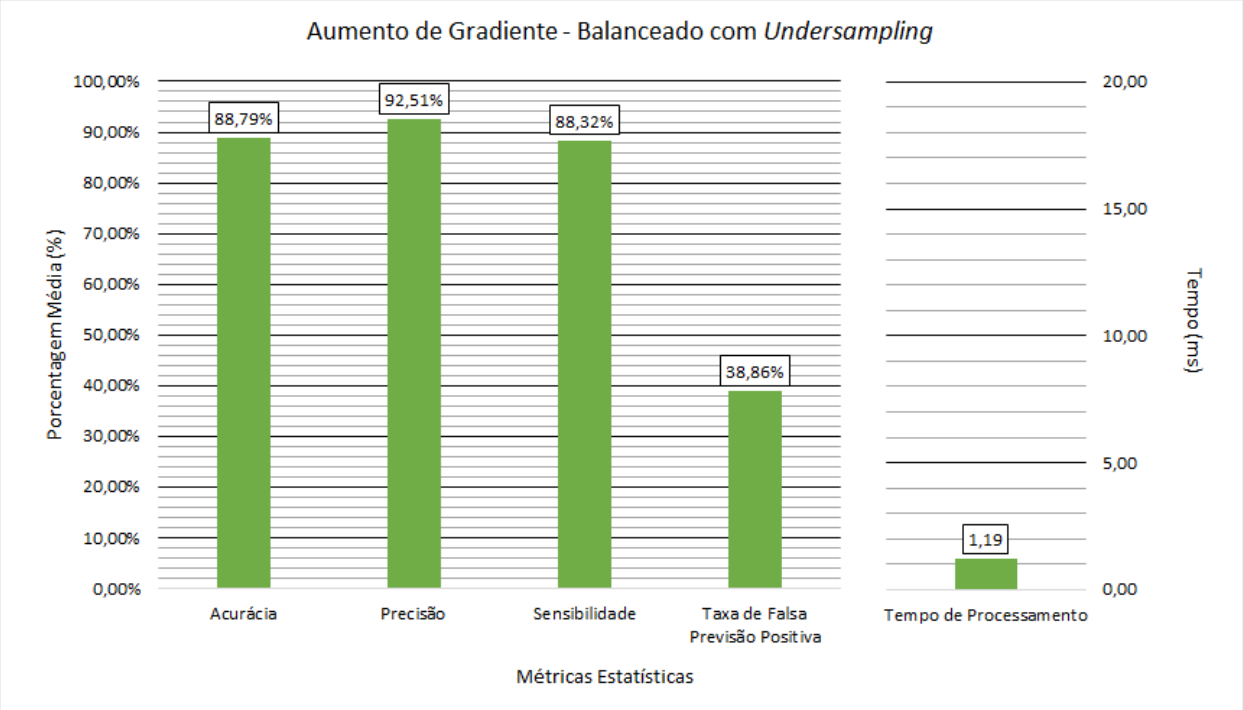

Figura 4.8: *Valores das Métricas Estatísticas dos Resultados do Teste 6.*

4.3.3 Comparação dos Testes 5 e 6

Na comparação realizada com os Testes 5 e 6, apresentados na Figura 4.9, é perceptível mais uma vez o bom desempenho do Aumento de Gradiente em relação ao Floresta Aleatória. Porém, existe uma leve queda, principalmente na Taxa de Falsa Previsão Positiva, uma vez que o valor de 38,86% mostra que o algoritmo está inclinado a prever uma maior quantidade de Falsos Negativos, em que o ideal seria um valor próximo de 50% representando uma imparcialidade nas previsões.

A única métrica em que o Floresta Aleatória teve resultados melhores foi a Precisão, atingindo 94,20%, mostrando que a quantidade de amostras positivas previstas corretamente é bem maior que a quantidade de falsas previsões positivas. E, como esperado, o Tempo de Processamento do Aumento de Gradiente se mostrou melhor que o tempo do Floresta Aleatória, como apresentado nos Testes anteriores, uma vez que o algoritmo computa todas as informações dos dados em memória, não realizando requisições diretamente ao Conjunto de Dados para realizar as análises e construções necessárias. Quanto ao desempenho elevado das métricas baseadas na matriz de confusão, a construção das árvores com valores ponderados em relação as árvores anteriores é capaz de fazer com que o algoritmo realize as classificações de forma bastante eficaz.

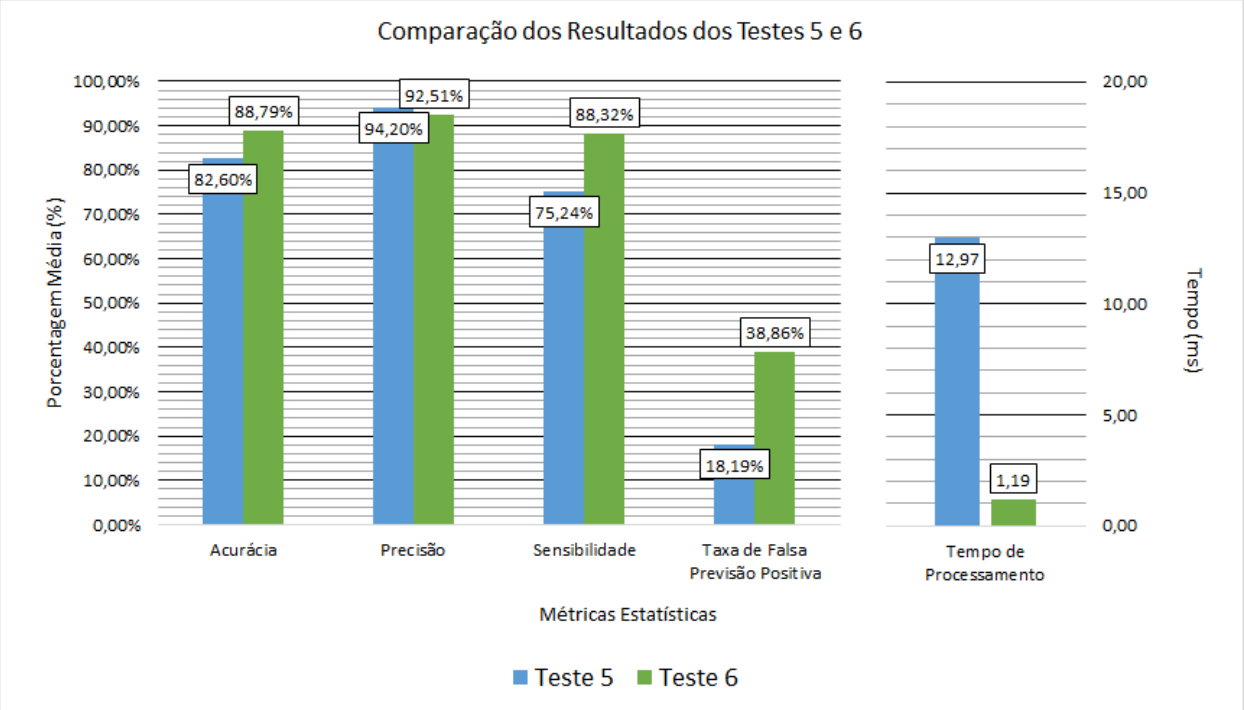

Figura 4.9: *Comparação dos Resultados dos Testes 5 e 6.*

4.4 Conclusão do Capítulo

Neste Capítulo foram apresentados os Resultados obtidos dos seis Testes propostos no trabalho. Os resultados, mostrados graficamente, tiveram o objetivo de realizar análises acerca da Acurácia, Precisão, Sensibilidade e Taxa de Falsa Previsão Positiva. Ademais, foi investigado também o Tempo de Processamento das soluções, indicando o Algoritmo de Aumento de Gradiente com o melhor desempenho, quando comparado com o Floresta Aleatória, seja com o Conjunto de Dados desbalanceado ou não.

Capítulo 5

Considerações Finais e Trabalhos Futuros

Neste último capítulo são revistas as motivações e objetivos do trabalho. Uma breve revisão do que foi proposto e desenvolvido bem como os resultados alcançados são apresentados. As limitações encontradas, como também os outros caminhos que podem ser seguidos são discutidos nos trabalhos futuros.

5.1 Visão Geral

A evasão estudantil é uma ação cada vez mais recorrente nos diversos níveis de ensino, sejam eles do fundamental até o superior, uma vez que o fato de um aluno desistir de seu curso acarreta em diversos problemas sociais e econômicos, seja para o próprio aluno que se sente desmotivado em continuar o seu curso ou ingressar em um outro, como também para a instituição, que sofre com a perda destes futuros profissionais, ocasionando um desperdício de investimentos por parte dos governos e também de algumas empresas.

Tendo em vista estes problemas, pesquisas que envolvam tentativas de predizer quando um aluno pode evadir de seu curso se tornaram frequentes no âmbito acadêmico, uma vez que o problema da evasão estudantil é basicamente um problema de classificação: assimilar se um aluno irá pertencer a classe de alunos evadidos ou não evadidos. Sendo assim, este problema é utilizado no presente trabalho como forma de definir a técnica computacional com o melhor desempenho para realizar estas classificações.

No geral, foram comparados dois *Ensemble Methods*, que correspondem a técnicas de agrupamento capazes de obter resultados de predição. As duas técnicas analisadas são Floresta Aleatória e Aumento de Gradiente, as quais foram construídas a partir do mesmo algoritmo de classificação de dados: Árvore de Decisão. O conjunto de dados utilizado na classificação é formado por registros de alunos do Curso Superior de Tecnologia em Telemática do IFPB *campus* Campina Grande, do período 2007.1 até 2015.1. No total, foram utilizadas 720 matrículas de alunos, das quais 429 são matrículas evadidas e 291 não evadi-

das.

Os algoritmos foram testados de três formas: 1) com o Conjunto de Dados original informado 2) com o Conjunto de Dados balanceado com método de *Oversampling*, igualando a quantidade de amostras das classes com a replicação das mesmas, e 3) com o Conjunto de Dados balanceado com método de *Undersampling*, desta vez igualando a quantidade de amostras das classes com a exclusão delas, de forma aleatória. Estas três formas de comparação dos algoritmos serviu para a conclusão quanto o algoritmo de melhor desempenho para o cenário utilizado, no caso a previsão de evasões estudantis.

5.2 Discussão dos Resultados

A partir dos três cenários de comparações dos algoritmos, considerando a forma que o Conjunto de Dados é utilizado, é perceptível ao analisar os resultados a partir de métricas estatísticas, um desempenho melhor do Aumento de Gradiente em relação ao Floresta Aleatória. Com o conjunto de dados desbalanceado o Aumento de Gradiente obteve os melhores valores para as métricas, embora os algoritmos se mostrarem bem próximos nos resultados, em que suas previsões geram resultados satisfatórios nas métricas baseadas na matriz de confusão. O valores próximos de 50% na Taxa de Falsa Previsão Positiva, mostra que os algoritmos não estão totalmente induzidos a preverem erroneamente as amostras como pertencentes a uma única classe, ou seja, ambos os modelos criados trabalham de forma equiparada. Na comparação, o Tempo de Processamento se mostra a métrica determinante para a conclusão do melhor modelo, uma vez que o Aumento de Gradiente realiza as classificações em um tempo médio de 0,90 milissegundos, valor bem próximo do limite inferior proposto na comparação, trabalhando cerca de 15 vezes mais rápido que o Floresta Aleatória, que atinge uma média de 14,06 milissegundos a cada execução.

Na segunda comparação, os algoritmos computaram testes experimentais utilizando o conjunto de dados balanceado, com o método *Oversampling*. Nesse caso, as amostras da classe minoritária foram replicadas para que a quantidade destas fosse igual a quantidade de amostras da classe majoritária, sendo a replicação das amostras realizadas de forma aleatória. Mais uma vez, o Aumento de Gradiente teve um bom desempenho em relação ao Floresta Aleatória, em que este último se mostrou bastante inclinado em prever valores Falsos Negativos, ou seja, evasões previstas como não evasões. O inclinamento do Floresta Aleatória é perceptível na Taxa de Falsa Previsão Positiva, em que apresentou um valor médio e 7,77%, mostrando que a quantidade de Falsos Negativos é bem maior que a de Falsos Positivos, em média, a cada execução eram previstos 45 FN e 4 FP. Já o Aumento de Gradiente, apresentou valor próximo dos 50%, mostrando que suas falsas previsões foram realizadas de forma igualitária, evidenciando uma não inclinação do algoritmo. O Tempo de Processamento dos dois algoritmos mais uma vez se mostrou bem distinto, com o Aumento de Gradiente tendo vantagem ao realizar as classificações em cerca de 1,20 milissegundos a cada

execução, e o Floresta Aleatória realizando as execuções por volta de 12,77 milissegundos, sendo 10 vezes mais lento.

Na terceira e última comparação dos algoritmos, o conjunto de dados foi balanceado aleatoriamente com o método *Undersampling*, excluindo amostras da classe majoritária para que sua quantidade seja igual à quantidade de amostras da classe minoritária. O Floresta Aleatória, assim como executado com o conjunto de dados balanceado com *Oversampling*, apresentou uma inclinação em prever uma maior quantidade de Falsos Negativos, porém teve um leve aumento, onde a Taxa de Falsa Previsão Positiva alcançou cerca de 18,19% em cada execução, ao contrário do Aumento de Gradiente que em relação a comparação anterior, o valor da métrica diminuiu para 38,86%, uma diminuição de quase 10 pontos percentuais, colocando em evidência uma inclinação nas previsões para o algoritmo. Esta inclinação refletiu na Sensibilidade dos dois algoritmos, resultando em um valor médio baixo para o Floresta Aleatória, por volta dos 75%, e cerca de 88,32% para o Aumento de Gradiente. Já a Acurácia, obteve um valor médio acima dos 80% para ambos os Testes, sendo o Aumento de Gradiente com um valor médio maior, por volta de 88,79%, apresentando que o modelo tem resultados satisfatórios para a previsão de amostras sejam elas evadidas ou não. E, mais uma vez, o Aumento de Gradiente se mostrou o algoritmo de melhor desempenho quanto a agilidade nas previsões, obtendo cerca de 1,19 milissegundos no Tempo e Processamento, contra os 12,97 do Floresta Aleatória.

Baseado nestes resultados, fica claro que o Aumento de Gradiente se mostra superior ao Floresta Aleatória para a previsão de evasões, como um algoritmo de viés baixo na classificação das amostras, ou seja, sem uma inclinação nas previsões que torne o algoritmo "viciado", apresentando valores métricos satisfatórios que evidenciam seu alto poder de classificação. Além da previsão das amostras, o algoritmo alcançou um bom desempenho quanto a agilidade na execução dos testes, proporcionando valores médios abaixo de 2 milissegundos, sendo mais rápido que o Floresta Aleatória em até 15 vezes.

5.3 Pesquisas Futuras

A pesquisa realizada neste trabalho – hospedada em um repositório no GitHub[1] – ainda se encontra em fase inicial, pois embora os resultados adquiridos tenham sido satisfatórios para determinar o melhor algoritmo na previsão de evasões estudantis, os algoritmos precisam ser ajustados para que possam realizar classificações de forma mais genérica, não somente aos dados utilizados.

Saindo da área de comparação do desempenho de algoritmos, é interessante que as futuras pesquisas utilizem o Aumento de Gradiente para a solução de novos problemas como forma de extensão do presente trabalho, tal como um estudo para determinar quais os atributos que mais descrevem a ocorrência de uma evasão ou não, utilizando técnicas de seleção de

[1]Repositório da pesquisa no GitHub: https://github.com/rodolfobolconte/evasao-estudantil-telematica

atributos sofisticadas, tais como *Least Absolute Shrinkage and Selection Operator (LASSO)*, que realiza uma análise regressiva dos atributos para definir os mais importantes e também *Recursive Feature Elimination (RFE)*, que testa os atributos de forma recursiva, sempre diminuindo sua quantidade utilizada com base na eliminação dos menos importantes. Além disso, uma pesquisa utilizando mais atributos descritivos pode ocasionar em resultados melhores para o algoritmo, como a utilização de atributos socioeconômicos que descrevem a situação econômica dos alunos, que são fatores determinantes para a ocorrência de evasões, uma vez que alunos de baixa renda podem morar longe de suas instituições, que sem apoio financeiro são incapazes de concluir os cursos, por exemplo. Quanto aos dados, é necessário realizar uma etapa de pré-processamento para a realização de limpezas e correções dos mesmos, caso tenham valores inconsistentes, ausentes ou discrepantes.

Outra fator que pode contribuir para um bom desempenho nas previsões, é um Conjunto de Dados com amostras que contenham informações de apenas um período cursado por um aluno, uma vez que estariam presentes também informações de quais disciplinas foram cursadas pelo aluno, como também um índice de desempenho por período, capaz de representar ainda melhor a situação em que o aluno se encontra dentro do curso.

Por fim, um outro trabalho futuro proposto é assimilar quais são os possíveis evasores dos cursos, para que possam ser apresentados à instituição com o propósito de auxiliar o processo de tomadas de decisões, através de uma ferramenta de gerenciamento de evasões. Tal ferramente pode até ser integrada com o SUAP, facilitando ainda mais a percepção de evasões para professores e coordenadores dos cursos. Dessa forma, os setores responsáveis pelo apoio aos estudantes podem tomar medidas preventivas para que a evasão estudantil não ocorra (ou diminua).

5.4 Conclusão do Capítulo

Neste Capítulo foi abordada uma visão geral de todo o trabalho bem como a discussão de seus resultados, mostrando que o Aumento de Gradiente é o *Ensemble Method* que obtém o melhor desempenho para a previsão de evasões estudantis do Curso Superior de Tecnologia em Telemática do IFPB *campus* Campina Grande, se comparado com o outro *Ensemble Method* utilizado, o Floresta Aleatória. Por fim, foram apresentadas as ideias futuras para a continuação do trabalho, mostrando que o mesmo ainda se encontra em fase inicial, onde o Aumento de Gradiente necessita de diversas outras técnicas e testes para se tornar um exemplo a ser seguido por demais pesquisas sobre evasões estudantis.

Referências Bibliográficas

[Academy 2018] ACADEMY, D. S. *17 Casos de Uso de Machine Learning*. 2018. Disponível em: <http://datascienceacademy.com.br/blog/17-casos-de-uso-de-machine-learning/>. 12

[Action 2019] ACTION, P. *1.3 Exposição dos Dados*. 2019. Disponível em: <http://www.portalaction.com.br/estatistica-basica/13-exposicao-dos-dados>. 23

[Almeida 2016] ALMEIDA, L. H. G. de. Learning analytics em ambiente virtual de aprendizagem moodle: um estudo de caso em componentes curriculares para cursos semipresenciais. *Gestão & Aprendizagem*, v. 4, n. 2, p. 76–93, 2016. 17, 18

[Aquarela 2017] AQUARELA. *Outliers, o que são e como tratá-los em uma análise de dados?* 2017. Disponível em: <https://www.aquare.la/o-que-sao-outliers-e-como-trata-los-em-uma-analise-de-dados/>. 13

[Aragão e Wilpert 2018] ARAGãO, I. L. J.; WILPERT, N. *Solução computacional para classificação e sumarização de polaridade de comentários em português*. Monografia (Graduação) — Universidade Federal de Santa Catarina, FLorianópolis, SC, 2018. 16

[Bolconte e Mendes 2017] Bolconte, R.; Mendes, G. W. D. Previsão automática de evasão estudantil nos cursos do ifpb. *Simpósio de Pesquisa, Pós-Graduação e Inovação do IFPB*, p. 33–34, November 2017. 4

[Brownlee 2014] BROWNLEE, J. *A Gentle Introduction to Scikit-Learn: A Python Machine Learning Library*. 2014. Disponível em: <https://machinelearningmastery.com/a-gentle-introduction-to-scikit-learn-a-python-machine-learning-library/>. 22

[Campos 2017] CAMPOS, R. *Árvores de Decisão*. 2017. Disponível em: <https://medium.com/@raphaelrcampos>. 26

[Carvalho e Silva 2014] CARVALHO, H. M.; SILVA, N. C. *Aprendizado de Máquina voltado para Mineração de Dados: Árvores de Decisão*. Monografia (Graduação) — Universidade de Brasília, Brasília, DF, 2014. Disponível em: <http://bdm.unb.br/bitstream/10483/9487/1/2014_HialoMunizCarvalho.pdf>. 8

[Cintra 2018] CINTRA, S. . *Qual a diferença entre precisão e acurácia?* 2018. Disponível em: <https://www.santiagoecintra.com.br/blog/geo-tecnologias/qual-a-diferenca-entre-precisao-e-acuraciay>. 32

[Columbus 2018] COLUMBUS, L. *Roundup Of Machine Learning Forecasts And Market Estimates, 2018*. 2018. Disponível em: <https://www.forbes.com/sites/louiscolumbus/2018/02/18/roundup-of-machine-learning-forecasts-and-market-estimates-2018/#a9dd63d2225c>. 9

[Crispim 2014] CRISPIM, J. *Sumarização automática de textos na prática: Extração baseada em grafos é o que há!* 2014. Disponível em: <https://jucacrispim.wordpress.com/2014/11/10/sumarizacao-automatica-de-textos-na-pratica-extracao-baseada-em-grafos-e-o-que-ha/>. 16

[Demir 2016] DEMIR, N. *Ensemble Methods: Elegant Techniques to Produce Improved Machine Learning Results*. 2016. Disponível em: <https://www.toptal.com/machine-learning/ensemble-methods-machine-learning>. 27

[Deng *et al.* 2009] Deng, J. *et al.* Imagenet: A large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2009. p. 248–255. ISSN 1063-6919. 10

[Esteves, Lorena e Nascimento 2009] ESTEVES, R. S.; LORENA, A. C.; NASCIMENTO, M. Z. Aplicação de técnicas aprendizado de máquina na classificação de imagens mamográficas. 2009. 11

[Ferreira 2016] FERREIRA, L. D. *Técnicas de aprendizado de máquina aplicadas à identificação de perfis de aprendizado em um ambiente real de ensino*. Monografia (Mestrado) — Universidade de São Paulo - São Carlos, São Carlos, SP, 2016. 2

[Ferreira e Andrade 2013] FERREIRA, S. A.; ANDRADE, A. Desenhar e implementar um sistema de learning analytics no ensino superior. 2013. 17, 18

[Filho 2017] FILHO, C. H. P. *Técnicas de Aprendizado Não Supervisionado baseadas no algoritmo da caminhada do turista*. Monografia (Pós-Graduação) — Universidade de São Paulo, São Carlos, SP, 2017. 15

[Geitgey 2014] GEITGEY, A. *Machine Learning is Fun!* 2014. Disponível em: <https://medium.com/@ageitgey/machine-learning-is-fun-80ea3ec3c471>. 8

[Gilioli 2016] GILIOLI, R. S. P. Evasão em instituições federais de ensino superior no brasil: Expansão da rede, sisu e desafios. *Consultoria Legislativa: Câmara dos Deputados*, Maio 2016. Disponível em: <https://www2.camara.leg.br/atividade-legislativa/estudos-e-notas-tecnicas/publicacoes-da-consultoria-legislativa/areas-da-conle/tema11/2016_7371_evasao-em-instituicoes-de-ensino-superior_renato-gilioli>. 1, 2

[Han, Kamber e Pei 2012] HAN, J.; KAMBER, M.; PEI, J. *Data Mining: Concepts and Techniques*. [S.l.]: Morgan Kaufmann, 2012. 13

[Hewa 2018] HEWA, K. *K-Fold Cross Validation*. 2018. Disponível em: <https://medium.com/datadriveninvestor/k-fold-cross-validation-6b8518070833>. 21

[Honda, Facure e Yaohao 2017] HONDA, H.; FACURE, M.; YAOHAO, Y. *Os Três Tipos de Aprendizado de Máquina*. 2017. Disponível em: <https://lamfo-unb.github.io/2017/07/27/tres-tipos-am/>. 2, 14, 15

[IFPB 2016] IFPB. *Cálculo CRE*. 2016. Disponível em: <https://www.ifpb.edu.br/pre/controle-academico/arquivos/calculo-cre.pdf>. 24

[Julian 2016] JULIAN, D. *Designing Machine Learning Systems with Python*. [S.l.]: Packt Publishing, 2016. 15, 27, 29

[Junior e Oliveira 2016] JUNIOR, C. B. S.; OLIVEIRA, I. C. A. Learning analytics: Revisão da literatura e o estado da arte. *Congresso Internacional ABED de Educação a Distância*, 2016. 17

[Kultzak 2016] KULTZAK, A. F. *Categorização de Textos Utilizando Algoritmos de Aprendizagem de Máquina com WEKA*. Monografia (Graduação) — Universidade Tecnológica do Paraná, Ponta Grossa, PR, 2016. Disponível em: <http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/7419/1/PG_COADS_2016_1_04.pdf>. 10

[Le 2018] LE, J. K. *A Tour Of The Top 10 Algorithms For Machine Learning Newbies*. 2018. Disponível em: <https://jameskle.com/writes/tour-10-machine-learning-algorithms>. 25

[Manhães, Cruz e Zimbrão 2014] MANHãES, L. M. B.; CRUZ, S. M. S.; ZIMBRãO, G. Evaluating performance and dropouts of undergraduates using educational data mining. 2014. Disponível em: <https://www.aspiringminds.com/pages/assess/2014/camera_ready/poster/manhaes_etal.pdf>. 23

[Marques 2015] MARQUES, L. *Algoritmos de Aprendizagem de Máquina*. 2015. Disponível em: <https://www.wattpad.com/user/LucasMarques3>. 26

[Marreiros e Oliveira 2000] MARREIROS, G.; OLIVEIRA, P. *Inovação e Tecnologia - Data Mining*. Monografia (Pós-Graduação) — Faculdade de Engenharia da Universidade do Porto, Porto, Portugal, 2000. 16

[Matos 2015] MATOS, D. *Por que Cientistas de Dados escolhem Python?* 2015. Disponível em: <http://www.cienciaedados.com/por-que-cientistas-de-dados-escolhem-python/>. 22

[Medeiros 2004] MEDEIROS, E. A. *Técnica de Aprendizagem de Máquina para Categorização de Textos*. Monografia (Graduação) — Escola Politécnica de Pernambuco, Recife, PE, 2004. Disponível em: <https://tcc.ecomp.poli.br/20061/EriclesMedeiros.pdf>. 10

[Melo 2016] MELO, A. S. C. *Previsão automática de evasão estudantil: um estudo de caso na UFCG*. Monografia (Mestrado) — Universidade Federal de Campina Grande, Campina Grande, PB, 2016. 4, 20, 21, 23

[Mikhail e Ackerman 1976] MIKHAIL, E.; ACKERMAN, F. Observations and least squares. 1976. 32

[Müller e Guido 2016] MÜLLER, A.; GUIDO, S. *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media, 2016. ISBN 9781449369897. Disponível em: <https://books.google.com.br/books?id=vbQlDQAAQBAJ>. 2, 8, 9, 16, 17, 26, 27, 28, 30

[Nogueira *et al.* 2018] NOGUEIRA, S. P. *et al.* Big data com learning analytics para apoiar o planejamento pedagógico acadêmico. *7º DesafIE!*, 2018. 17, 18

[Oguri 2007] OGURI, P. Aprendizado de máquina para o problema de sentiment classification. 2007. Disponível em: <https://www.maxwell.vrac.puc-rio.br/Busca_etds.php?strSecao=resultado&nrSeq=9947@1>. 26

[Oliveira 2018] OLIVEIRA, F. *Afinal o que é Learning Analytics?* 2018. Disponível em: <https://medium.com/@limaolflavio/a-importância-do-learning-analytics-na-avaliaç˜ao-do-aprendizado-on-line-b43d67685848>. 17

[Paladini 2016] PALADINI, F. *Reconhecimento de imagens: o fim da infância prolongada*. 2016. Disponível em: <https://universoracionalista.org/reconhecimento-de-imagens-o-fim-da-infancia-prolongada/>. 10

[Prati, Batista e Monard 2003] PRATI, R. C.; BATISTA, G. E. A. P. A.; MONARD, M. C. Uma experiência no balanceamento artificial de conjuntos de dados para aprendizado com classes desbalanceadas utilizando análise roc. 2003. 25

[Ray 2017] RAY, S. *6 Easy Steps to Learn Naive Bayes Algorithm*. 2017. Disponível em: <https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/>. 26

[Rezende 2003] REZENDE, S. O. *Sistemas inteligentes: fundamentos e aplicações*. [S.l.]: Manole, 2003. 2, 15

[Ribeiro 2018] RIBEIRO, D. *O que é Machine Learning? Tecnologia permite 'adivinhar' o que você quer*. 2018. Disponível em: <https://www.techtudo.com.br/noticias/2018/05/o-que-e-machine-learning-tecnologia-permite-adivinhar-o-que-voce-quer.ghtml>. 10, 11

[Richert e Coelho 2013] RICHERT, W.; COELHO, L. P. *Building Machine Learning Systems with Python*. [S.l.]: Packt Publishing, 2013. (Community experience distilled). ISBN 9781782161417. 8

[Samuel 1959] SAMUEL, A. L. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, v. 3, n. 3, p. 210–229, July 1959. ISSN 0018-8646. Disponível em: <https://ieeexplore.ieee.org/document/5392560>. 7

[Silva, Peres e Boscarioli 2017] SILVA, L. A.; PERES, S. M.; BOSCARIOLI, C. *Introdução à mineração de dados: Com Aplicações em R*. [S.l.]: Elsevier Academic, 2017. 12, 13, 21

[Silva, Almeida e Yamakami 2012] SILVA, R. M.; ALMEIDA, T. A.; YAMAKAMI, A. Análise de métodos de aprendizagem de máquina para detecç ao automática de spam hosts. 2012. Disponível em: <http://www.dt.fee.unicamp.br/~tiago/papers/SBSEG12.pdf>. 28

[Souza 2019] SOUZA, E. G. *Entendendo o que é Matriz de Confusão com Python*. 2019. Disponível em: <https://medium.com/data-hackers/>. 31

[Souza 2016] SOUZA, R. C. *Aplicação de Learning Analytics para Avaliação do Desempenha de Tutores a Distância*. Monografia (Mestrado) — Universidade Federal Rural do Semi-Árido, Mossoró, PB, 2016. 17, 18

[Subhani *et al.* 2017] Subhani, A. R. *et al.* Machine learning framework for the detection of mental stress at multiple levels. *IEEE Access*, v. 5, p. 13545–13556, 2017. 11

[Sutton e Barto 1998] SUTTON, R. S.; BARTO, A. G. *Reinforcement Learning: An Introduction*. MIT Press, 1998. Disponível em: <http://www.incompleteideas.net/book/ebook/>. 2, 7, 8

[Tan *et al.* 2018] TAN, P. P. *et al. Introduction to Data Mining*. [S.l.]: Pearson, 2018. 2

[Zhu, Zeng e Wang 2010] ZHU, W. X.; ZENG, N. F.; WANG, N. Sensitivity, specificity, accuracy, associated confidence interval and roc analysis with practical sas implementations. *NESUG - Health Care and Life Sciences*, 2010. 32

[Zoldi 2018] ZOLDI, S. *Machine learning contra fraudes de engenharia social*. 2018. Disponível em: <https://computerworld.com.br/2018/10/17/machine-learning-contra-fraudes-de-engenharia-social/>. 11