Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Uso de Aprendizado de Máquina para Classificação de Risco de Acidentes em Rodovias

Brunna de Sousa Pereira Amorim

Campina Grande, Paraíba, Brasil

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Uso de Aprendizado de Máquina para Classificação de Risco de Acidentes em Rodovias

Brunna de Sousa Pereira Amorim

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande - Campus I como parte dos requisitos necessários para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: Ciência da Computação
Linha de Pesquisa: Metodologia e Técnicas da Computação

Cláudio de Souza Baptista, Ph.D.
(Orientador)

Campina Grande, Paraíba, Brasil

A524u Amorim, Brunna de Sousa Pereira.

Uso de aprendizado de máquina para classificação de risco de acidentes em rodovias / Brunna de Sousa Pereira Amorim. – Campina Grande, 2019.

93 f. : il. color.

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2019.

"Orientação: Prof. Dr. Cláudio de Souza Baptista".

Referências.

1. Aprendizado de Máquina. 2. Aprendizado de Máquina Automatizado. 3. Seleção de Características. 4. Redução de Dimensionalidade. 5. Risco de Acidente em Rodovias. Classificação de Risco. I. Baptista, Cláudio de Souza. II. Título.

CDU 004.78(043)

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECÁRIA SEVERINA SUELI DA SILVA OLIVEIRA CRB-15/225

"USO DE APRENDIZADO DE MÁQUINA PARA CLASSIFICAÇÃO DE RISCO DE ACIDENTES EM RODOVIAS"

BRUNNA DE SOUSA PEREIRA AMORIM

DISSERTAÇÃO APROVADA EM 21/08/2019

CLÁUDIO DE SOUZA BAPTISTA, PhD., UFCG
Orientador(a)

HERMAN MARTINS GOMES, Ph.D, UFCG
Examinador(a)

GERALDO BRAZ JUNIOR, Dr., UFMA
Examinador(a)

CAMPINA GRANDE - PB

Resumo

Soluções para identificação dos fatores que influenciam o acontecimento de acidentes em rodovias e a identificação de trechos de risco estão sendo estudados e aplicados por pesquisadores e governos de todo o mundo, a fim de encontrar uma solução que possa diminuir o número de tais acidentes. No entanto, o estudo de acidentes em rodovias depende do local onde o mesmo acontece. Destarte, esta pesquisa faz uso de técnicas de aprendizado de máquina supervisionado e aprendizado de máquina automatizado com o uso de diferentes características para analisar seu impacto na tarefa de predição do risco de acidentes graves ou não-graves em trechos de rodovias brasileiras, a fim de otimizar o desempenho e a performance dos classificadores. Os dados de acidentes foram pré-processados, analisados e técnicas de seleção de atributos foram empregadas, resultando em uma base com informações sobre o dia da semana, o turno do dia em que o acidente aconteceu, o tipo da pista, o traçado da via, o sentido da rodovia, a condição meteorológica no momento do acidente e o tipo do acidente. Diferentes modelos de aprendizado de máquina foram treinados e avaliados em quatro cenários diferentes: o cenário A utiliza uma base de dados desbalanceada com o atributo "Frequência de Acidentes", enquanto o cenário B consiste na base de dados desbalanceada sem tal atributo; o cenário C faz uso da base de dados balanceada com o atributo "Frequência de Acidentes" e o cenário D utiliza a base de dados balanceada sem este atributo. A avaliação experimental ocorreu com o emprego das métricas acurácia, precisão, revocação e medida F. Os resultados dos cenários A e B não foram relevantes ao estudo, uma vez que os classificadores não convergiram, classificando os dados em apenas uma classe: não-grave. O melhor resultado para o cenário C foi a Rede Neural MLP, que obteve 85% de acurácia, 87% de precisão, 85% de revocação e 84% de medida F. Já para o cenário D, os melhores resultados foram combinações de dois modelos diferentes: Random Forest+BernoulliNB e Logistic Regression+ExtraTreesClassifier, ambos com 84,58% de acurácia, 88,14% de precisão, 84,58% de revocação e 84,06% medida F.

Abstract

In order to decrease the number of road accidents, solutions to identify influencing factors of road accidents and its risk areas are being researched throughout the world. However, road accident studies depend upon its location, hence this study uses supervised machine learning techniques and automated machine learning to classify accident risk sections of brazilian federal roads in severe or not-severe, using several features. The accident data was analized, pre-processed and its features were selected using different techniques, resulting in a set of information containing the week day and time the accident happened, the road type, the road route, the road orientation, the weather condition when the accident happened and the accident type. Machine learning models were trained and evaluated in four different scenarios: scenario A used a imbalanced database with the "accident frequency" feature, while scenario B used a imbalanced database without the "accident frequency" feature; scenario C used a balanced database with the "accident frequency" feature and scenario D used a balanced database without the "accident frequency" feature. To validate the model, the accuracy, precision, recall and F-measure metrics were used. Scenarios A and B results were disregarded since all models preticted only one class: not-severe. Scenario C best result was a MLP neural network model with 85% of accuracy, 87% of precision, 85% of recall and 84% of F-measure. The best results to scenario D were two combinations of classifiers: first, the combination of Random Forest and BernoulliNB; second, the combination of Logistic Regression and ExtraTreesClassifier, both resulting in 84,58% of accuracy, 88,14% of precision, 84,58% of recall and 84,06% of F-measure.

Agradecimentos

Primeiramente, gostaria de agradecer a Deus por me proporcionar esta oportunidade, provendo sabedoria e forças para concluir esta pesquisa.

Agradeço também aos meus pais, Ivanildo e Rosa, exemplos de amor e dedicação, por todo o esforço, força e apoio em todos os momentos da minha vida, sem os quais eu não teria conseguido alcançar tamanha conquista. Ao meu irmão, Brenno, por todo o incentivo, amizade e companhia durante todos esses anos.

À minha família.

Ao meu namorado, Hitalo, pelo carinho e apoio, por sempre torcer pelo meu sucesso e por toda compreensão, sobretudo nos momentos de maior dificuldade.

A todos os meus amigos, pelo apoio, pelos momentos de descontração e pela compreensão de minhas ausências.

Ao meu orientador, Cláudio Baptista, pelo incentivo, paciência e confiança dedicada, desde a graduação, à minha formação acadêmica. Obrigada por acreditar em meu potencial, por todo o conhecimento transmitido e por ser um exemplo de dedicação e motivação.

Aos professores Geraldo Braz Junior e Herman Martins Gomes, por aceitarem o convite para participar da banca examinadora de minha dissertação.

Aos integrantes do Laboratório de Sistemas de Informação (LSI) pelos momentos de aprendizado, pelo incentivo e por toda a ajuda prestada. Obrigada por sempre proporcionarem um ótimo ambiente de trabalho.

Agradeço à Universidade Federal de Campina Grande, ao Departamento de Sistemas e Computação e à Coordenação de Pós-Graduação em Ciência da Computação por minha formação acadêmica e por tornar possível a realização da minha pesquisa. À CAPES e ao CNPq pelo apoio financeiro.

A todos que participaram e ajudaram, direta ou indiretamente, dessa conquista.

Conteúdo

Lista de Siglas

OMS - *Organização Mundial de Saúde*

PIB - *Produto Interno Bruto*

PRF - *Polícia Rodoviária Federal*

ML - *Machine Learning*

AutoML - *Automated Machine Learning*

TPOT - *Tree-Based Pipeline Optimization Tool*

TF-IDF - *Term Frequency–Inverse Document Frequency*

PCA - *Principal Component Analysis*

SVM - *Support Vector Machine*

RBF - *Radial basis function*

DMLC - *Distributed Machine Learning Community*

LIME - *Local Interpretable Model-agnostic Explanations*

UK - *United Kingdom*

NB - *Naive Bayes*

SOM - *Self Organizing Map*

GUHA - *General Unary HypotesesAutomation*

WEKA - *Waikato Environment for Knowledge Analysis*

BHTrans - *Empresa de transportes e Trânsito de Belo Horizonte*

OLAP - *On-line analyticalprocessing)*

RP-MVPLN - *Poisson-lognormal*

ROC - *Receiver operating characteristic curve*

AUROC - *Area Under the Receiver Operating Characteristics*

RELU - *Rectified linear unit*

CNN - *Convolutional Neural Networks*

RNN - *Recurrent Neural Networks*

MLP - *Multilayer Perceptron*

Lista de Figuras

Lista de Tabelas

Lista de Códigos Fonte

Capítulo 1

Introdução

Com o crescimento populacional e a constante evolução da sociedade, os meios de transporte tornam-se cada vez mais importantes e essenciais para a vida humana. O transporte converteu ideias outrora impossíveis em realidade, hodiernamente já fazendo parte do dia-a-dia da sociedade como um todo e, consequentemente, tornando a população cada vez mais dependente de tais meios.

Para via de informação, essencial demonstrar que existem diversas modalidades de meios de transporte, separados por categorias específicas, com destaque para os meios: terrestres, onde se encontram o ferroviário, rodoviário e metroviário; aquáticos, com destaque para o marítimo, fluvial e lacustre; aéreos; e, por fim, dutoviários [10]. Destarte, o presente trabalho irá se ater ao estudo da primeira categoria citada.

No Brasil, nota-se uma considerável falta de equilíbrio quando analisados os meios de transporte mais utilizados, em comparação com as alguns aspectos geográficos e populacionais do país. Para ilustrar tal afirmação, basta realizar levantamento da malha ferroviária que estende-se pelo Brasil, que chega aproximadamente aos vinte e nove mil quilômetros, em relação à área total da nação, que ultrapassa os oito milhões de quilômetros quadrados. Por exemplo, as ferrovias da Alemanha[1], país de dimensões consideravelmente menores, chegam a quase igualar as ferrovias do Brasil[2].

Foi feita, no Brasil, a opção de priorizar o transporte por meio de veículos automotores, tornando indispensável o uso de rodovias. Como por exemplo, segundo o IBGE (Instituto

[1] https://w3.unece.org/CountriesInFigures/en/Home/Index?countryCode=276

[2] https://www.cnt.org.br/boletins

Brasileiro de Geografia e Estatística), o maior meio de transporte de carga brasileiro é feito usando o modal rodoviário, por onde cerca de 61,1% da carga é transportada[3].

Destarte, seja para viajar, para ir ao trabalho ou para transportar produtos, o uso de estradas nas tarefas cotidianas faz-se cada vez mais necessário. Toda essa dependência torna os acidentes nas estradas um problema real, que faz parte da realidade, não apenas do Brasil, mas de todo o mundo [5].

Atualmente, acidentes em rodovias são considerados um dos problemas mais importantes em todo o mundo, sendo um motivo de preocupação e objeto de estudo em muitos países, a exemplo da Turquia [9], Suíça [62], Índia [32, 34, 67], China [17, 68, 75], Bangladesh [64], Estados Unidos [19, 23], Reino Unido [73], Espanha [1, 47], Finlândia [74], Canadá [60], Arábia Saudita [2], Japão [63, 69], Holanda [76] e Líbano [24]. Isso se dá pois os acidentes em rodovias estão entre as principais causas de morte no mundo, possuindo números alarmantes de pessoas mortas e feridas, este último em diversos níveis. Além das incontáveis mortes, os acidentes também causam danos à propriedades e grandes perdas econômicas para os países.

De acordo com Organização Mundial de Saúde (OMS), os acidentes em rodovias são a oitava principal causa de morte no mundo[4]. Desde 2001, o número de vítimas fatais aumentou de forma constante, alcançando um total de 1,35 milhões de pessoas por ano. Esse tipo de acidente também é a primeira causa principal de morte de crianças e jovens entre 5 e 29 anos [51].

Além das preocupações humanitárias de ferimentos e mortes, estima-se que os custos econômicos mundiais causados pelo impacto dos acidentes de trânsito representam uma perda de aproximadamente 3% do PIB mundial [29, 51].

Portanto, para diminuir as perdas humanitárias e econômicas, os governos têm demonstrado cada vez mais interesse em estudos que analisam possíveis causas de acidentes em rodovias, visando a melhorar a segurança nas estradas e diminuir os danos causados. A melhor forma para aumentar a segurança nas estradas é entendendo o motivo do acidente ter acontecido. Porém, acidentes são incertos e imprevisíveis, definidos por diversas variáveis e

[3]https://www.ibge.gov.br/geociencias/cartas-e-mapas/redes-geograficas/15793-logistica-dos-transportes.html?=&t=acesso-ao-produto

[4]https://www.who.int/violence_injury_prevention/road_safety_status/2018/en/

características que nem sempre são levadas em consideração.

O Brasil, segundo a OMS, possui o quinto trânsito mais violento do mundo[5]. Em 2015, no Brasil, a Organização registrou cerca de 38.651 fatalidades decorridas de acidentes de trânsito que ocorreram nas vias urbanas e nas rodovias federais, das quais 82% foram vítimas do sexo masculino e 18% do sexo feminino. Dessas fatalidades, 31% das vítimas estavam em veículos de duas ou três rodas, enquanto 23% estavam em veículos de quatro rodas. 18% das fatalidades foram pedestres, 3% ciclistas e 2% foram passageiros ou motoristas de veículos pesados.

A Polícia Rodoviária Federal (PRF) brasileira coleta e divulga, desde 2007, dados de acidentes que aconteceram nas rodovias federais do país. Esses dados estão disponíveis no site da PRF[6] através de um conjunto de arquivos .csv contendo informações de cada acidente, separados por ano. De 2007 a 2017, mais de 1,6 milhão de acidentes foram registrados nas rodovias brasileiras, onde 83.498 pessoas morreram e mais de um milhão ficaram feridas. São, em média, 23 óbitos por dia.

De acordo com os dados disponibilizados pela PRF, em 2015 foram registrados 122.161 acidentes nas rodovias federais do Brasil, com 6.867 fatalidades e 90.251 feridos. Em 2016, foram registrados 96.363 acidentes, resultando em 6.398 fatalidades e 86.672 feridos. Já em 2017, 89.396 acidentes foram registrados, resultando em 6.243 mortos e 84.075 feridos. A 1.1 mostra a quantidade de acidentes, mortos e feridos para cada ano.

É possível perceber que, nos últimos anos, o número de acidentes nas estradas brasileiras tem diminuído. Isso vem acontecendo desde de 2013, que registrou o grave número de 186.748 acidentes. Ainda assim, são números alarmantes que causam uma grande quantidade de fatalidades e feridos, sendo um motivo de grande preocupação para o governo e sociedade como um todo.

Na economia do país, os acidentes de trânsito são responsáveis por grande impacto negativo. Segundo um levantamento feito pela Escola Nacional de Seguros[7], os acidentes graves nas ruas e estradas brasileiras causaram, no primeiro semestre de 2018, um impacto econômico de cerca de R$ 96,5 bilhões para o país. De acordo com o Observatório Nacional de

[5]https://apps.who.int/iris/bitstream/handle/10665/276462/9789241565684-eng.pdf?ua=1

[6]https://www.prf.gov.br/portal/dados-abertos/acidentes

[7]http://www.ens.edu.br/

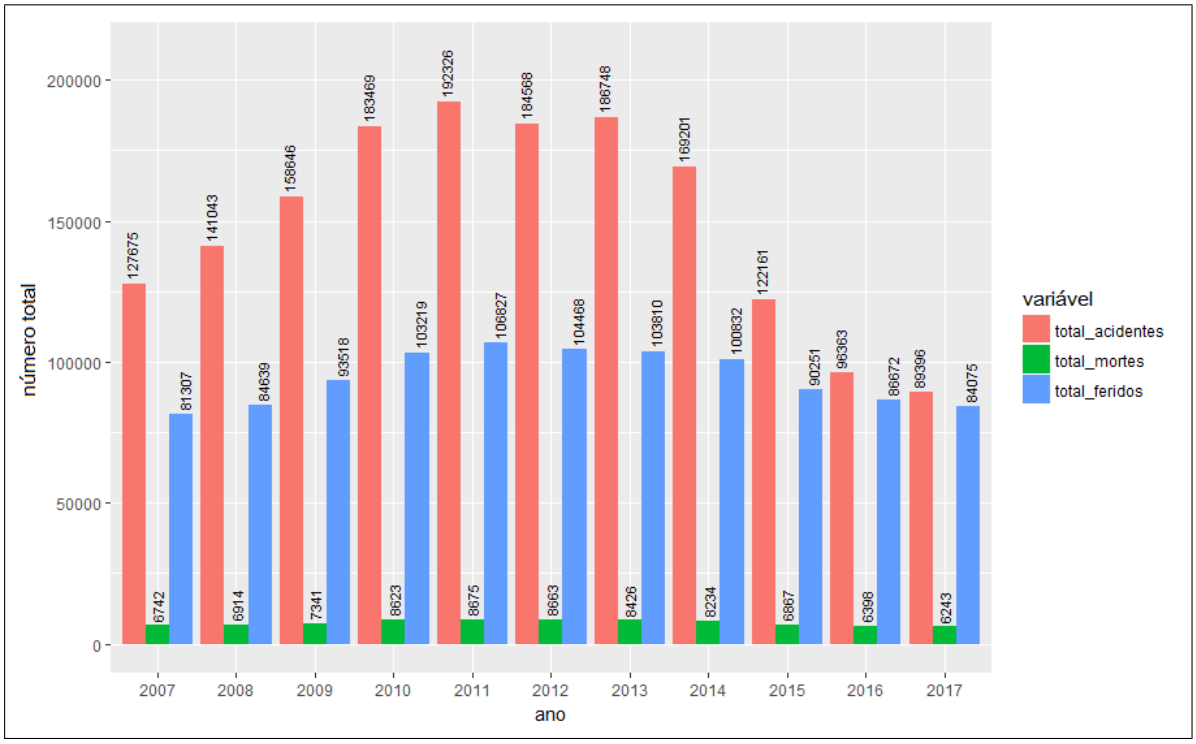

Figura 1.1: Quantidade de acidentes, mortos e feridos por ano.

Segurança Viária[8], os acidentes no trânsito resultam em custos anuais de cerca de R$ 52 bilhões para o país.

Esses valores são calculados com base no dinheiro que seria gerado pelo trabalho das vítimas caso elas não tivessem se envolvido nos acidentes graves. Além de contar com as fatalidades, esse cálculo leva em consideração as pessoas que vieram a ficar inválidas permanentemente por causa do acidente, os gastos com saúde, previdência e reparos das vias.

Por ser um problema que afeta não só o Brasil, mas o mundo inteiro, a área da análise de dados de acidentes tem se tornado cada vez mais popular e existem diversos estudos que tratam dessa problemática. Um dos principais objetivos destes estudos é o de entender os fatores de risco que contribuem para os acidentes acontecerem, tornando possível a criação de medidas para a diminuição de tais acidentes.

Muitos destes estudos utilizam métodos estatísticos [25, 27, 43, 55, 65] e técnicas de mineração de dados [11, 33–37, 71] para analisar os dados de acidentes e tentar estabelecer

[8]http://www.onsv.org.br/

relações entre as características dos acidentes e a severidade, ou seja, se foi um acidente com vítimas fatais ou não. Alguns outros estudos fazem uso de técnicas de aprendizado de máquina para a predição da gravidade dos acidentes [13, 23, 32, 60, 64], para prever o risco de acidentes em determinada área [28, 58, 61, 62], para prever a duração de acidentes [66] e para a detecção ou predição de acidentes, provendo informação para evitar que estes ocorram [3, 28, 67].

Entender os fatores de risco e prever características de acidentes graves é importante para ser possível tomar medidas preventivas quanto ao problema, justificando a importância dos estudos na área. Porém, também é verdade que fatores de risco de acidentes possuem impactos diferentes em localidades diferentes [32]. Por isso, analisar novos dados de acidentes produzirá novas informações acerca do problema em dada localidade.

Devido ao grande número de acidentes em rodovias no mundo e, principalmente, no Brasil, estudos referentes à identificação dos principais motivos para acidentes acontecerem e as áreas mais propícias a isso são cada vez mais importantes. Esses estudos permitem a criação de meios para alertar e informar os motoristas dos perigos e situações mais propícias a acidentes, bem como adverti-los de trechos perigosos nas rodovias. Essa conscientização e informação serve como um meio para evitar tais acidentes.

Portanto, esta pesquisa propõe-se a estudar algoritmos de classificação de Aprendizado de Máquina (ML) e Aprendizado de Máquina Automatizado (AutoML) em uma base de dados que contém as informações de uma década de acidentes que ocorreram nas rodovias do país, registrados pela PRF. O objetivo é comparar os resultados e identificar o algoritmo que obtém o maior sucesso na predição de trechos de rodovias brasileiras que podem ser considerados perigosos.

O perigo de um trecho da rodovia será previsto de acordo com acidentes que ocorreram no local e são considerados graves ou não-graves, considerando também outros diversos fatores: a condição climática, turno em que o acidente aconteceu, o dia da semana do acidente, o trecho da rodovia, o tipo da pista, o sentido da rodovia, o traçado da via e o tipo de acidente que aconteceu: uma colisão, um capotamento ou um atropelamento. Por exemplo, um trecho de rodovia que possui risco de acidentes graves (ou seja, é um trecho perigoso) em uma noite chuvosa de quarta-feira, pode ser considerado não-perigoso em um domingo de manhã com o céu-claro.

1.1 Objetivos

Nesta seção, serão apresentados os objetivos gerais e específicos deste trabalho.

1.1.1 Objetivo Geral

Esse estudo tem como objetivo geral classificar e prever, fazendo uso de algoritmos de aprendizado de máquina supervisionado, a potencialidade de acontecer acidentes graves em trechos das rodovias brasileiras.

1.1.2 Objetivos Específicos

A fim de que o objetivo geral desta pesquisa seja atingido, faz-se necessária sua fragmentação nos seguintes objetivos específicos:

- Analisar técnicas de redução de dimensionalidade dos dados a fim de selecionar os atributos mais importantes;
- Estudar e implementar diferentes técnicas de aprendizado de máquina supervisionado para realizar os experimentos;
- Avaliar o desempenho de cada técnica utilizada com diferentes métricas, comparando seus resultados;
- Encontrar o melhor algoritmo de aprendizado de máquina para identificar trechos das rodovias brasileiras que possuem risco de acidentes graves, de acordo com algumas características específicas.

1.2 Relevância

Atualmente, existe a necessidade de um maior entendimento relativo aos acidentes que acontecem nas rodovias brasileiras, bem como identificar o risco de um acidente grave ocorrer dado um trecho da rodovia. Este é um tópico de pesquisa muito estudado nas demais partes do mundo que poderia ser melhor explorado no Brasil, principalmente pelo fato de haver disponibilidade de uma base de dados oficiais, de mais de uma década, fornecida pela PRF.

A relevância desta pesquisa está na proposta de uma abordagem que utiliza Pré-processamento de dados, Seleção de Atributos, Aprendizado de Máquina Supervisionado e Aprendizado de Máquina Automatizado (AutoML) para avaliar o comportamento dos algoritmos ML e verificar qual a melhor técnica para identificar trechos de rodovias brasileiras consideradas perigosas.

Com este estudo, será possível desenvolver uma aplicação que faz uso dos algoritmos de classificação estudados para avisar ao motorista a potencialidade de acontecer um acidente grave em trechos da rodovia. Levando em consideração variáveis externas, coletadas em tempo real (como o dia da semana, turno, condição climática etc), o motorista será avisado dos trechos em seu percurso que possuem risco de acidentes graves.

Entre as propostas encontradas na literatura revisada, esta pesquisa traz ainda como contribuição o uso e comparação de diversos algoritmos de aprendizado de máquina. Além disso, são utilizadas diversas características dos acidentes que não são consideradas, em sua totalidade, nos outros estudos, tais como o dia da semana e o turno do dia em que o acidente aconteceu, o tipo da pista, o traçado da via, o sentido da rodovia, a condição meteorológica no momento do acidente e o tipo do acidente.

1.3 Organização do Trabalho

O restante desta dissertação está organizada da seguinte forma: no Capítulo 2 é apresentada a fundamentação teórica desta pesquisa, que aborda os temas Pré-processamento de dados, Seleção de Atributos, Aprendizado de Máquina Supervisionado, Aprendizado de Máquina Automatizado (AutoML) e métricas de avaliação para algoritmos de aprendizado de máquina. No Capítulo 3, é apresentada a revisão bibliográfica dos trabalhos que abordam o problema foco desta pesquisa. A metodologia utilizada neste estudo é discutida no Capítulo 4, enquanto os resultados obtidos ao longo desta pesquisa são apresentados no Capítulo 5. Finalmente, no Capítulo 6, estão descritas as conclusões e proposições para trabalhos futuros.

Capítulo 2

Fundamentação Teórica

O objetivo deste capítulo é abordar os principais conceitos e tecnologias que fundamentam esta pesquisa. Para tanto, o mesmo está dividido em quatro seções: a Seção 2.1 discute o aprendizado de máquina e as métricas usadas para avaliar os modelos; a Seção 2.2 trata das técnicas de aprendizado de máquina supervisionado usadas neste estudo; a Seção 2.3 apresenta os algoritmos de aprendizado de máquina usados neste trabalho, enquanto a Seção 2.4 discute as técnicas de seleção de atributos utilizadas. Por fim, a Seção 2.5 apresenta as considerações finais.

2.1 Aprendizado de Máquina

O Aprendizado de Máquina é uma área de conhecimento de análise de dados que automatiza a construção de modelos analíticos [77]. É um ramo da Inteligência Artificial que estuda meios para que máquinas possam fazer tarefas que seriam executadas por pessoas. Para ser inteligente, o sistema deve possuir a habilidade de aprender com novas informações e adaptar-se a ambientes em mudança [4].

Formada por regras previamente definidas, o aprendizado de máquina permite que os computadores tomem decisões com base nos dados prévios e em dados usados pelo usuário, fazendo com que o computador tenha habilidade para tomar decisões que podem resolver problemas [45]. Essa habilidade faz o Aprendizado de Máquina ser capaz de ajudar na solução de diversos problemas, incluindo reconhecimento de fala, robótica, processamento e reconhecimento de imagens.

Segundo Alpaydin, um modelo de aprendizado de máquina pode ser configurado de acordo com parâmetros, e o aprendizado é feito por meio da execução de um programa de computador que otimizará os parâmetros do modelo usando dados de treinamento ou experiências passadas [4]. O modelo pode ser preditivo, para fazer predições futuras, descritivo, para aprender conhecimento com os dados, ou ambos.

O aprendizado de máquina pode ser usado em aplicações como: aprendizado por regras de associação (*association rule learning*), regressão e classificação. O aprendizado por regras de associação consiste na descoberta de relações entre variáveis de um grande conjunto de dados, com o objetivo de identificar regras fortes usando medidas de interesse [54]. A regressão é uma técnica estatística usada para prever quantidades. Dado um conjunto de atributos de entrada, a saída da regressão será dada por valores numéricos.

Na classificação, os algoritmos de aprendizado de máquina operam construindo um modelo a partir de entradas amostrais a fim de fazer previsões ou decisões guiadas pelos dados. A tarefa de classificação consiste em três etapas: a criação do modelo, que irá classificar a instância do problema de acordo com as classes definidas; o treinamento do modelo, onde o algoritmo cria uma função para descrever os dados em relação às classes do problema; e a validação, que consiste na utilização do modelo treinado para verificar a eficácia do modelo na predição de classes. Os dados usados na etapa da validação, idealmente, não são os mesmos utilizados na etapa de treinamento. Com isso, o algoritmo é considerado bom para o problema caso possua uma alta taxa de acerto para novas instâncias.

Existem dois tipos de aprendizado de máquina: o aprendizado supervisionado e o aprendizado não-supervisionado. O aprendizado não-supervisionado não provê um resultado específico, no entanto, o algoritmo busca identificar semelhanças entre os dados de entrada para que estes possam ser categorizados em conjunto. Nesse tipo de aprendizagem, os resultados mudam de acordo com as variáveis. A abordagem estatística da aprendizagem não supervisionada é conhecida como estimativa de densidade.

Já no aprendizado de máquina supervisionado, um conjunto de dados de treinamento com as respostas corretas (classe) é fornecido e, com base neste conjunto de treinamento, o algoritmo generaliza o que foi aprendido para responder corretamente a todas as entradas possíveis. Isso também é chamado de aprender com exemplos. Neste estudo apenas algoritmos de classificação de aprendizagem de máquina supervisionada foram utilizados.

2.1.1 Aprendizado de Máquina Supervisionado

No aprendizado de máquina supervisionado, tem-se variáveis de entrada, variáveis de saída e um algoritmo que irá aprender a função de mapeamento da entrada para a saída. O objetivo é aproximar a função de mapeamento tão bem que, quando colocados novos dados de entrada no modelo, será possível prever as variáveis de saída para esses dados.

A abordagem do aprendizado de máquina supervisionado assume que o modelo é definido por um conjunto de parâmetros $y = g(x|\theta)$, onde $g(\cdot)$ é o modelo e θ são seus parâmetros. y representa o código da classe (0 ou 1, à exemplo de uma classe binária). $g(\cdot)$ é a função de discriminação que separa as instâncias em classes diferentes. O aprendizado de máquina otimiza os parâmetros θ de forma que o erro aproximado é minimizado, ou seja, as estimativas chegam o mais próximo possível dos valores corretos providos pelo conjunto de treinamento.

Portanto, este aprendizado é chamado de supervisionado pois no processo de aprendizagem, o algoritmo aprende com o conjunto de dados de treinamento, uma vez que os dados de treinamento são rotulados com a classe ao qual pertencem. Após o treinamento, o modelo é validado com o uso de dados rotulados que não foram usados na etapa de treinamento.

O conjunto de dados de validação serve para testar a generalização do modelo, ou seja, verificar o quão bem o modelo treinado prevê a saída certa para novas instâncias. Com o modelo treinado e validado, dados de teste não-rotulados (ou seja, sem a classe ao qual pertencem) servem de entrada para esse modelo. A saída é a previsão das classes dos dados de teste.

Os resultados destas técnicas dependem da quantidade e qualidade das amostras (dados de entrada) que serão usadas para a construção dos modelos. Porém, alguns problemas podem acontecer durante a classificação. Caso o modelo não consiga boas taxas de classificação com novas instâncias, que não foram usadas no treinamento, pode ter acontecido um super ajustamento dos parâmetros para os dados de treinamento, chamado de *overfitting*.

O *overfitting* de um modelo se dá quando o modelo está muito ajustado a um conjunto limitado de dados e normalmente possui um número elevado de parâmetros, tornando-o excessivamente complexo. Tal modelo não é capaz de generalizar a classificação para novas instâncias pois aprendeu todos os ruídos e imprecisões dos dados de treinamento [45].

Um outro problema que pode ocorrer com a classificação é o *underfitting*. O *underfitting*,

caso contrário ao *overfitting*, acontece quando um modelo não consegue capturar a tendência subjacente dos dados, descrevendo um modelo generalizado demais que não se ajusta bem aos dados. O *underfitting* é frequentemente resultado de um modelo excessivamente simples, incapaz de atingir uma boa capacidade de classificação em dados de validação ou teste.

Para avaliar a qualidade do modelo e verificar se houve *overfitting* ou *underfitting*, diversas métricas podem ser utilizadas. Neste estudo, as métricas usadas para avaliar os modelos foram: a matriz de confusão, a acurácia, a precisão, a revocação e a medida F.

A matriz de confusão é uma matriz que indica quantas instâncias foram classificadas corretamente e quantas foram classificadas incorretamente para cada classe do problema. Isso permite uma análise mais detalhada da proporção de acerto do classificador. A Tabela 2.1 apresenta o arcabouço de uma matriz de confusão, que descreve a quantidade de falsos positivos, falsos negativos, verdadeiros positivos e verdadeiros negativos oriundos da classificação dos dados, onde:

- Verdadeiro positivo (*true positive — TP*): ocorre quando no conjunto real, a classe que estamos buscando prever foi prevista corretamente.
- Falso positivo (*false positive — FP*): ocorre quando no conjunto real, a classe que estamos buscando prever foi prevista incorretamente.
- Falso negativo (*false negative — FN*): ocorre quando no conjunto real, a classe que não estamos buscando prever foi prevista incorretamente.
- Verdadeiro negativo (*true negative — TN*): ocorre quando no conjunto real, a classe que não estamos buscando prever foi prevista corretamente.

		Classe Real	
		Classe Positiva	Classe Negativa
Classe Predita	Classe Positiva	Verdadeiro Positivo (VP)	Falso Positivo (FP)
	Classe Negativa	Falso Negativo (FN)	Verdadeiro Negativo (VN)

Tabela 2.1: Matriz de confusão para duas classes.

A Tabela 2.1 mostra uma matriz de confusão para duas classes. Assim, temos que instâncias da classe positiva que foram classificadas como Classe Positiva são os verdadeiros

positivos e as instâncias da classe negativa classificadas como Negativa são os verdadeiros negativos. Ou seja, o classificador acertou a classificação dessas duas classes de instâncias. As instâncias da Classe Positiva classificadas como Classe Negativa são os falsos negativos, enquanto as instâncias da Classe Negativa classificadas como Classe Positiva são os falsos positivos.

A partir da matriz de confusão, são definidas as outras métricas que poderão ser empregadas para avaliar os modelos de classificação. A acurácia é uma métrica usada para medição estatística do quão bem um classificador binário identifica ou exclui uma classe corretamente [77]. Ou seja, a acurácia é a soma dos acertos (verdadeiros e falsos) dividido pelo total de instâncias:

$$\text{acurácia} = \frac{VP + VN}{VP + FP + FN + VN} = \frac{\text{predições corretas}}{\text{todas as predições}}$$

Já a revocação (recall) mede a proporção de verdadeiros positivos que foram corretamente classificados [77]. Ou seja, a revocação é a divisão entre os verdadeiros positivos e a soma dos verdadeiros positivos com os falsos negativos:

$$\text{revocação} = \frac{VP}{VP + FN} \tag{2.1}$$

A precisão mede a proporção de resultados positivos, e é dada pela divisão entre os verdadeiros positivos e a soma dos verdadeiros positivos com os falsos positivos [77]:

$$\text{precisão} = \frac{VP}{VP + FP} \tag{2.2}$$

Já a medida F, ou a medida harmônica entre a precisão e a revocação, nos mostra o balanço entre a precisão e a revocação de nosso modelo, e é dada por [77]:

$$\text{Medida F} = 2 * \frac{\text{precisão} * \text{revocação}}{\text{precisão} + \text{revocação}} \tag{2.3}$$

As técnicas de aprendizado de máquina usadas neste estudo estão descritas na Seção 2.3.

2.2 Aprendizado de Máquina Automatizado

Atualmente, *frameworks* de AutoML (Automatic Machine Learning) vêm sendo cada vez mais utilizados por serem opções acessíveis, flexíveis e escaláveis. O AutoML é responsável por automatizar o processo de ponta a ponta da aplicação de aprendizado de máquina a problemas do mundo real [6]. Para usar algoritmos de aprendizado de máquina, os profissionais devem aplicar os métodos apropriados de pré-processamento de dados, engenharia de recursos, extração de recursos e seleção de atributos que tornam o conjunto de dados passível de aprendizado de máquina. Seguindo essas etapas de pré-processamento, os profissionais devem executar a seleção de algoritmo e a otimização do hiperparâmetro para maximizar o desempenho preditivo de seu modelo final de aprendizado de máquina.

Por ser um método complexo para não-especialistas, o AutoML foi proposto como uma solução baseada em inteligência artificial para uso de algoritmos de aprendizado de máquina. A automatização desse processo oferece as vantagens de produzir soluções mais simples, mais rápidas e modelos que geralmente superam os modelos projetados manualmente.

Por existirem várias áreas de foco para o aprendizado de máquina automático, um diverso conjunto de *frameworks* AutoML está disponível para uso. A grande maioria desses *frameworks* fazem uso de algoritmos de aprendizado de máquina open source, como o scikit-learn, que possui uma grande quantidade classificadores implementados e prontos para o uso [53]. No entanto, os métodos usados para automatizar a aplicação e avaliação dessas técnicas de aprendizado de máquina são diferentes.

Alguns dos *frameworks* de aprendizado de máquina automático são o H2Os AutoML, Auto_ml, o autosklearn e o TPOT. O H2O[1] é um *framework* que contém um conjunto de algoritmos de aprendizado de máquina disponíveis para uso em diversas interfaces e linguagens de programação. O módulo de aprendizado de máquina automático do H2O utiliza as próprias implementações dos algoritmos para gerar as configurações adequadas para os dados e parâmetros do modelo. Um ponto negativo desse *framework* é a coleta inadequada de "lixo", que acaba na falha de execução em processos longos.

Para extrair valor dos dados rapidamente, o Auto_ml[2] é um *framework* projetado para ser usado em empresas, e automatiza muitas partes do processo de aprendizado de máquina.

[1] http://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html

[2] https://github.com/ClimbsRocks/auto_ml

Primeiro, ele automatiza o processamento de atributos por meio do processamento de datas, categorização e transformação de escala de atributos numéricos e através do processamento TF-IDF (*Term Frequency–Inverse Document Frequency*), o qual determina a relevância que uma palavra possui em um determinado documento através do cálculo da frequência inversa [57].

O Auto_ml também executa redução de atributos quando existem mais de 100.000 colunas, usando métodos tais como o PCA (*Principal Component Analysis*), que encontra os principais componentes de um conjunto de dados multivariado e retorna combinações lineares de tais dados [78]. Ademais, este *framework* automatiza os processos de construção e de otimização dos algoritmos de aprendizado de máquina. Porém, além de não existir um recurso de limitação de tempo, sendo um ponto negativo para uso em cenários que precisam de uma janela pequena de tempo de execução, a performance do Auto_ml para problemas de classificação multi-classe não é boa [6].

O autosklearn é um *framework* que encapsula o *framework* scikit-learn, detentor de implementações de algoritmos de aprendizado de máquina, para gerar automaticamente um pipeline [16], sequência linear de transformações em dados que culmina em um processo de modelagem a ser avaliado. O autosklearn inclui métodos para processamento de atributos como *one-hot encoding*, padronização de atributos numéricos e PCA. Esse *framework* disponibiliza algoritmos de classificação e regressão, porém não possui a habilidade de processar entradas em linguagem natural nem de diferenciar entradas categóricas e numéricas.

O TPOT herda as implementações de algoritmos de aprendizado de máquina disponibilizados no scikit-learn [53] para criação da própria base de métodos de regressão e métodos de classificação. Restrições de tempo são aplicadas ao TPOT, alterando o tempo máximo de execução. O processo de otimização usado dá suporte à pausa na execução, que pode ser resumida depois. Ademais, a característica mais importante deste *framework* é a capacidade de exportar um modelo para código.

Um estudo comparativo de ferramentas AutoML foi feito por Balaji e Allen, onde os autores comparam o uso de quatro ferramentas: Auto_ml, autosklearn, TPOT e H2O AutoML, os melhores resultados foram obtidos pelo autosklearn e pelo TPOT [6]. Neste estudo, o TPOT foi a ferramenta escolhida por, diferente do autosklearn, possuir a habilidade de tratar dados esparsos e por permitir o uso de diversas *threads*, acelerando a execução e otimização

de *pipelines*.

2.2.1 TPOT

O TPOT (Tree-Based Pipeline Optimization Tool) é uma ferramenta de otimização baseada em programação genética que gera, de acordo com o banco de dados usado, *pipelines* de aprendizagem de máquina. O TPOT implementa seus próprios métodos de classificação com base no *framework* scikit-learn, pacote que implementa diversos classificadores de aprendizagem de máquina, e automatiza partes do processo de aprendizagem de máquina [50].

O TPOT é um *framework* que ajuda na escolha das configurações e classificadores mais adequados aos dados do usuário. Por ser uma ferramenta de AutoML, testes com diferentes configurações e classificadores são feitos com os dados fornecidos, retornando o melhor resultado possível. Uma das características mais importantes do *framework* é conseguir exportar os resultados em forma de código, que pode ser modificado à mão, permitir o uso de processos em paralelo durante a otimização de *pipelines* para tornar a execução mais rápida e dar suporte a bases de dados esparsas.

O TPOT é dividido em quatro pacotes de classificadores distintos: o TPOT Default, TPOT Sparse, TPOT Light e TPOT MDR. O TPOT Default vai procurar dentre um vasto número de pré-processadores, construtores de características, seletores de atributos, modelos de classificação e parâmetros para encontrar operadores que minimizem os erros de predição do modelo. Esse pacote também define os modelos de classificação que serão testados com os dados providos, são eles: GaussianNB, DecisionTreeClassifier, ExtraTreesClassifier, GradientBoostingClassifier, BernoulliNB, MultinomialNB, RandomForestClassifier, KNeighborsClassifier, LinearSVC, LogisticRegression, XGBClassifier.

O pacote TPOT Light procura operadores que minimizem os erros de predição do modelo dentre um número restrito de pré-processadores, construtores de características, seletores de atributos, modelos de classificação e parâmetros. Os modelos definidos neste pacote são mais simples e rápidos de serem executados, sendo eles: GaussianNB, BernoulliNB, MultinomialNB, DecisionTreeClassifier, KNeighborsClassifier, LogisticRegression.

O pacote TPOT MDR busca em uma série de seletores de características e modelos multifatores de redução de dimensionalidade, os melhores operadores que irão maximizar a acurácia da predição. Esta configuração é especializada em genome-wide association studies

(GWAS)[3], e possui utiliza apenas o modelo LogisticRegression para classificar os dados.

Já o pacote TPOT Sparse faz uso de todos os operadores disponíveis e de um dicionário de configuração, dando suporte ao uso de matrizes esparsas. Os classificadores disponíveis neste pacote são: BernoulliNB, MultinomialNB, RandomForestClassifier, KNeighborsClassifier, LinearSVC, LogisticRegression, XGBClassifier.

Entre os operadores de pré-processamento de características disponíveis no TPOT, estão: StandardScaler, RobustScaler, MinMaxScaler, MaxAbsScaler, RandomizedPCA, Binarizer e PolynomialFeatures. Já os operadores de seleção de características disponíveis são: VarianceThreshold, SelectKBest, SelectPercentile, SelectFwe e Recursive Feature Elimination (RFE).

Para o solução proposta neste estudo, os pacotes que mais se adequam às necessidades são o TPOT Default e o TPOT Sparse, uma vez que são pacotes mais completos e que permitem o uso de processos em paralelo e voltado à dados esparsos, respectivamente. Os experimentos feitos para a escolha desses algoritmos estão detalhados na Seção 4.

2.3 Algoritmos de Aprendizado de Máquina

Nesta seção, são apresentados os algoritmos de aprendizado de máquina utilizados neste estudo.

2.3.1 Support Vector Machine

Um dos classificadores mais utilizados atualmente e que também é muito popular em estudos relacionados a esta dissertação é o Support Vector Machine (SVM), também conhecido como Máquina de Vetores de Suporte. O SVM é um método de aprendizado supervisionado baseado em otimização, com um algoritmo que analisa dados tanto para Regressão quanto para Classificação [73].

O SVM usa planos de decisão para encontrar padrões nos dados e assim classificá-los. Para isso, o SVM mapeia os dados do plano original para um hiperplano n-dimensional, aumentando a complexidade do classificador, e define uma função que irá separar os dados de classes diferentes. Com isso, o SVM escolhe um vetor de suporte para cada classe, onde

[3]https://www.genome.gov/about-genomics/fact-sheets/Genome-Wide-Association-Studies-Fact-Sheet

sua medida será usada para diferenciar uma classe da outra. A função definida anteriormente é otimizada de forma a descrever melhor cada classe do problema, utilizando os vetores de suporte como base para aumentar essa distância [44].

A função usada pelo SVM para transformar o hiperplano dos dados é definida pelo kernel. O kernel do SVM é um conjunto de funções matemáticas que transformam os dados de entrada em uma outra dimensão na qual é possível encontrar divisões claras de margens entre as classes dos dados. Para este estudo, foram utilizados quatro tipos de kernel: o kernel linear, o kernel RBF, o kernel sigmoid e o kernel polinomial.

O SVM com kernel linear é o mais simples, e irá definir uma reta para separar o hiperplano. Esse método pode ser muito efetivo para diversos problemas, a depender dos dados utilizados. Porém, há casos em que uma reta não consegue separar os dados de maneira ótima. Por esse motivo, foram feitos testes outros kernels a fim de encontrar o método mais adequado ao problema.

O kernel polinomial representa a similaridade de vetores em um espaço de característica aplicando função polinomial ao vetor original, permitindo o aprendizado não-linear. Este kernel não olha apenas para as características dadas das amostras de entrada para determinar sua similaridade, mas também combinações destas. Já o kernel RBF, ou Gaussian Radial Basis function kernel, define uma função com base radial e é muito usado quando não há conhecimento prévio dos dados. O kernel Sigmoid vêm do campo de redes neurais, onde a função sigmoid bipolar é usada frequentemente como a função de ativação em neurônios artificiais.

O SVM tem a vantagem de ser um classificador efetivo para espaços com muitas dimensões e para casos em que o número de dimensões é maior que o tamanho da amostra. Também é versátil, possibilitando o uso de diversos kernels, incluindo o uso de kernels customizados. Uma desvantagem desse modelo é a dificuldade de encontrar os parâmetros ideais devido a sua sensibilidade à variação dos mesmos, sendo importante a otimização dessa escolha.

2.3.2 Redes Neurais

Segundo Nielsen, Redes Neurais Artificiais são técnicas computacionais capazes de adquirir conhecimento através da experiência, e apresentam um modelo matemático baseado na estru-

tura neural de organismos inteligentes [49]. Essas redes neurais são compostas por neurônios interconectados, organizados em camadas e representados por funções matemáticas que recebem uma ou mais entradas para gerar uma saída (função de ativação). Os neurônios são conectados por canais de comunicação que estão associados a um determinado peso. Uma rede neural é composta por uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída.

A função de ativação usada nos neurônios pode simplificar ou tornar a rede neural mais complexa. Desta forma, algumas funções de ativação comumente usadas são *step function*, *linear combination*, *sigmoid* e *rectifier (RELU)*. A step function é uma função binária em que, se a entrada u atingir um valor θ, a função de ativação retorna 1. Se não, retorna 0:

$$y = \begin{cases} 1 & \text{se} \quad u \geq \theta \\ 0 & \text{se} \quad u < \theta \end{cases} \tag{2.4}$$

A função *linear combination* retorna a soma ponderada da entrada com um *bias*. Já a função *sigmoid* é uma função matemática que possui uma curva no formato de "S", e irá transformar a entrada em um número que varia de 0 a 1. A função RELU é definida como a parte positiva do seu argumento, dada por:

$$f(x) = \max(0, x) \tag{2.5}$$

onde x é a entrada do neurônio.

Segundo Pradhan e Sameen, alguns tipos de redes neurais amplamente utilizadas são: as redes neurais convolucionais, compostas por uma ou mais camadas convolucionais totalmente conectadas (CNN), as redes neurais recorrentes (RNN), compostas por conexões que formam um grafo direcionado ao longo de uma sequência temporal, usa seu estado interno para processar sequências de entradas, e as redes neurais *feedforward*, composto por conexões não cíclicas, diferente das RNNs, as informações são passadas em apenas uma direção, começando pela camada de entrada, passando por suas camadas ocultas até chegar a camada de saída [56].

As redes neurais *feedforward* são muito utilizadas para classificação supervisionada, principalmente para casos onde os dados não são sequenciais nem dependentes do tempo [49]. Por esse motivo, este tipo de rede neural foi usada neste estudo.

2.3.3 Logistic Regression

A Regressão Logística (*Logistic Regression*) é um algoritmo muito usado em Aprendizado de Máquina, sendo o modelo de regressão mais apropriado para realização de análises onde a variável dependente é binária. As variáveis independentes podem ser categóricas ou não [21]. Como todas as análises de regressão, a regressão logística é uma análise preditiva, usada para descrever dados e explicar relacionamentos entre variáveis.

O modelo de regressão logística pode ser escrito da seguinte forma:

$$P(Y = 1) = \frac{1}{1 + e^{-g(x)}} \tag{2.6}$$

em que, $g(x) = B_0 + B_1X_1 + +... + B_pX_p$ e $x = X_1, X_2, ..., X_p$. Os coeficientes $B_0 + B_1 + ... + B_p$ são estimados a partir do conjunto dados pelo método da máxima verossimilhança, que encontra uma combinação de coeficientes que maximiza a probabilidade da amostra ter sido observada [39].

A Regressão Logística usa uma função logística que descreve uma curva em formato de S que pode receber qualquer número, e irá transformá-lo em um valor que varia de 0 a 1, mas nunca exatamente esses limites. O modelo usa esse valor para estimar a probabilidade de uma resposta Y, baseada em um conjunto de variáveis independentes de tamanho n, no qual $P(y|x)$ segue a distribuição de Bernoulli com uma probabilidade $P(x)$ de sucesso [30].

Este modelo tem como vantagens a facilidade em lidar com variáveis independentes categóricas, facilidade em classificar indivíduos em categorias e fornecer os resultados em termos de probabilidade.

2.3.4 Extra Trees Classifier

O ExtraTreesClassifier (Extremely Randomized Trees) é um método de aprendizado conjunto baseado fundamentalmente em árvores de decisão. Assim como o RandomForest, o ExtraTreesClassifier randomiza certas decisões e subconjuntos de dados para minimizar o excesso de aprendizado dos dados e o overfitting.

O ExtraTrees constrói um conjunto de árvores de decisão ou regressão não-ajustadas de acordo com o procedimento clássico de cima para baixo. Suas duas principais diferenças com outros métodos de agrupamento baseados em árvores são que os nós são divididos de

acordo com pontos de corte totalmente aleatórios e toda a amostra de aprendizado (em vez de uma réplica de bootstrap) é usada para treinar as árvores [18].

Segundo Geurts et al., do ponto de vista do viés de variância, a lógica por trás do método Extra-Trees é que a aleatoriedade do ponto de corte e do atributo combinados com a média do conjunto deve ser capaz de reduzir a variância mais fortemente do que os esquemas de randomização mais fracos usados por outros métodos [18]. Do ponto de vista computacional, a complexidade do procedimento de crescimento das árvores é, assumindo árvores balanceadas, na ordem de $nlogn$ em relação ao tamanho da amostra de aprendizagem, como a maioria dos outros procedimentos de cultivo de árvores. Contudo, dada a simplicidade do procedimento de divisão do nó, o fator constante é muito menor do que em outros métodos baseados em conjuntos que otimizam localmente os pontos de corte.

O uso de um valor randômico para dividir as árvores é uma das diferenças quando comparamos esse algoritmo com outros algoritmos baseados em árvores. Isso leva o algoritmo a ter uma boa performance e a criar árvores mais diversificadas, com menos divisões para serem avaliadas ao treinar o modelo.

2.3.5 XGBoost Classifier

O XGBoost Classifier é um algoritmo de aprendizado de máquina escalável e feito para melhorar algoritmos baseados em árvores [12]. Esse algoritmo vem sendo muito utilizado na área para criar soluções estado-da-arte em ciência de dados e em competições Kaggle para dados estruturados ou tabulares [46].

O XGBoost é uma implementação do Gradient Boosting Machines criado por Chen [12], que agora conta com a contribuição de vários desenvolvedores. Ele pertence a uma coleção de ferramentas que participam do Distributed Machine Learning Community (DMLC)[4], uma comunidade de algoritmos open-source.

Esse algoritmo foi criado com a intenção de melhorar a velocidade e performance de árvores de decisão, mas possui diversas vantagens que vão além disso: o XGBoost implementa processamento paralelo e é mais rápido do que outros algoritmos de árvores; é altamente flexível, permitindo usuários definirem objetivos de otimização e validação customizados; possui uma rotina feita para lidar com valores ausentes, onde o usuário também pode definir

[4]http://dmlc.ml/

o valor padrão para substituir pelo valor que falta. Além disso, o XGBoost possibilita ao usuário usar a validação cruzada a cada iteração do processo de otimização do algoritmo, facilitando a busca para o número ótimo de iterações em uma única execução.

2.3.6 Random Forest

Random Forest (ou Floresta Randômica) é um método de aprendizado de máquina flexível, muito popular na área por ser fácil de usar e produzir bons resultados que, na maioria das vezes, não precisa de ajuste nos hiperparâmetros [14, 22, 23, 32, 41, 58]. Ele pode ser usado para classificação ou regressão, além de corrigir problemas comuns em outras árvores de decisão: o overfitting do seu conjunto de treinamento [7].

O algoritmo Random Forest, como o nome já diz, é responsável por criar uma floresta de um modo aleatório. A floresta criada é uma combinação de árvores de decisão, na maioria dos casos treinados com o método de bagging. A idéia principal do método de *bagging* é que a combinação dos modelos de aprendizado aumenta o resultado geral. Ou seja, o algoritmo de florestas aleatórias cria várias árvores de decisão e as combina para obter uma predição com maior acurácia e mais estável [8].

O algoritmo ainda adiciona aleatoriedade extra ao modelo quando está criando as árvores: ao invés de procurar pela melhor característica ao fazer a partição de nós, o Random Forest busca a melhor característica em um subconjunto aleatório das características. Este processo cria uma grande diversidade nas árvores que são geradas, o que, geralmente, leva à construção de modelos melhores.

Portanto, ao criar uma árvore no método Random Forest, apenas um subconjunto aleatório das características é considerado na partição de um nó. É possível deixar a divisão das árvores ainda mais aleatórias utilizando limiares (thresholds) aleatórios para cada característica, ao invés de procurar pelo melhor limiar, como é feito pelas árvores de decisão.

Porém, uma das maiores limitações do Random Forest é que uma quantidade grande de árvores pode tornar o algoritmo lento e ineficiente para predições em tempo real. Geralmente, estes algoritmos são rápidos para treinar, mas lentos para fazer predições depois de treinados. Uma predição com mais acurácia requer mais árvores, o que faz o modelo ficar mais lento.

2.3.7 BernoulliNB

O algoritmo BernoulliNB, ou Bernoulli Naïve Bayes, faz parte da família de classificadores Naïve Bayes, que são classificadores probabilísticos comumente usados em problemas de classificação. A principal característica do Naïve Bayes, e também o motivo de receber "naive" (ingênuo) no nome, é que ele desconsidera completamente a correlação entre as variáveis. Ou seja, o algoritmo não vai levar em consideração a correlação entre esses fatores, tratando cada um de forma independente [48].

Por ser muito simples e rápido, o Naïve Bayes possui um desempenho relativamente maior do que outros classificadores. Além disso, ele só precisa de um pequeno número de dados de teste para concluir classificações com uma boa precisão. Entretanto, caso haja necessidade de correlacionar fatores, o Naïve Bayes tende a falhar na predição.

Por serem baseados em probabilidades, o que vai diferenciar as diferentes implementações da família Naïve Bayes é a forma com que ele modela as probabilidades. Nesse caso, o BernoulliNB é a implementação usada quando a classe dos dados é binária [44].

2.4 Seleção de Atributos

Para tornar possível o uso dos classificadores descritos acima, é importante que os dados estejam adequados de forma a maximizar o desempenho dos modelos. Por isso, é necessária a realização de diversos testes com diferentes variáveis e parâmetros, para assim ser possível a seleção dos atributos que são mais importantes.

Porém, quando um modelo de aprendizado de máquina é usado para classificar dados fornecidos como entrada, dificilmente sabemos quais características foram importantes para classificar um dado em uma determinada classe. A fim de selecionar os atributos mais importantes e significativos ao estudo, foi usado um pacote chamado LIME, que ajuda na seleção dos atributos mais significativos ao modelo.

2.4.1 LIME

Local Interpretable Model-agnostic Explanations (LIME) é uma técnica, disponibilizada no pacote LIME, que explica predições individuais para classificadores de texto ou classifica-

dores que têm como entrada tabelas (listas numéricas ou dados categóricos) ou imagens, utilizando classificadores lineares simples como base para a explicação. Baseado no trabalho de Ribeiro et al., o LIME tem o objetivo de trazer mais clareza no uso e funcionamento dos algoritmos de aprendizado de máquina, explicando o motivo para o qual uma instância foi classificada em uma determinada classe [59].

O LIME provê interpretabilidade local do modelo por meio da modificação de uma única amostra de dados através de pequenos ajustes nos valores das características para observar o resultado final e o impacto que isso tem na classificação. A saída do LIME é uma lista de explicações, refletindo a contribuição de cada recurso para a previsão de uma amostra de dados. Isso fornece interpretabilidade local e também permite determinar quais alterações de recurso terão mais impacto na previsão, ajudando assim na escolha das características mais importantes da base de dados.

2.5 Considerações Finais

Neste capítulo, apresentou-se a fundamentação teórica deste trabalho. Foram apresentadas as principais características e conceitos da área de Aprendizado de Máquina, englobando o Aprendizado de Máquina Supervisionado e o Aprendizado de Máquina Automatizado. As técnicas de seleção de atributos empregadas também foram apresentadas, assim como os algoritmos de Aprendizado de Máquina utilizados neste estudo.

Alguns trabalhos relacionados à pesquisa realizada nesta dissertação serão apresentados no próximo capítulo.

Capítulo 3

Trabalhos Relacionados

Atualmente, existem vários estudos sendo realizados utilizando dados de acidentes em rodovias. Alguns estudos abordam formas de tornar as rodovias um meio de transporte mais seguro, outros tentam identificar os principais motivos pelos quais acidentes acontecem. Vê-se, portanto, que trata-se de uma área que vem sendo muito estudada, possuindo diversas formas diferentes de tratar o problema.

Neste capítulo, são apresentados os principais trabalhos relacionados a esta pesquisa sobre acidentes ocorridos em rodovias. A fim de facilitar a descrição dos mesmos, os estudos foram divididos em seis seções, de acordo com os objetivos principais destes. Na primeira delas (Seção 3.1), são apresentados os trabalhos que visam estudar os impactos socio-econômicos e ambientais derivados de acidentes. Em seguida, na Seção 3.2, é apresentada uma análise dos trabalhos que buscam classificar os acidentes de acordo com as casualidades. A Seção 3.3 apresenta os trabalhos cujos objetivos eram analisar e buscar padrões nos dados de acidentes. Na quarta Seção (Seção 3.4), são apresentados os trabalhos que analisam e avaliam fatores que podem contribuir para ocorrência de acidentes. Em seguida, a Seção 3.5 detalha os estudos que visam a classificação e/ou predição da severidade de acidentes, enquanto a Seção 3.6 apresenta os trabalhos que visam detectar e/ou prever áreas que possuem risco de acidentes. Por fim, na Seção 3.7, são apresentadas as considerações finais.

3.1 Impactos Socio-econômicos e Ambientais

Com o objetivo de identificar precisamente os efeitos à longo prazo do clima e da economia na frequência de acidentes em Iowa, Estados Unidos, explorando também as correlações espaço-temporal dos acidentes, Liu e Sharma usaram um modelo Bayesiano espaço-temporal multivariado para analisar dados de acidentes que aconteceram em Iowa durante os anos de 2006 e 2015 [42]. Os dados usados pelos autores possuem informações do número de feridos graves e leves, tipo de colisão, severidade do acidente, a taxa de desemprego e da renda do local, quilometragem do veículo, o índice de chuvas e o índice de neve. Apesar do uso de dados de acidentes nesse estudo, o foco dos autores foi de identificar a influência dos atributos usados nos tipos de acidentes que ocorreram no local. Foi descoberto que a taxa de desemprego possui uma influência negativa na contagem de feridos, enquanto que a quilometragem do veículo possui influência positiva na identificação dos diferentes tipos de colisão: colisão fatal, colisão com feridos graves, colisão com feridos leves, possíveis feridos e dano à propriedade.

Com o intuito de investigar consequências negativas resultantes de acidentes de trânsito na saúde pública e na segurança, considerando diferentes regiões demográficas, Kocatepe et al. propuseram uma abordagem baseada no modelo estatístico *mixed-logit* e na análise espacial de Sistemas de Informações Geográficas para descobrir unidades suscetíveis ao risco imposto por acidentes [31]. O estudo foi conduzido na região de Tampa Bay, na Flórida, Estados Unidos, utilizando dados do censo de 2015 que possuem informações demográficas e socio-econômicas tais como: porcentagem da população negra, latina, jovem e idosa da região, população total, média de salário da região, o volume de caminhões que passam na região, o comprimento da rodovia, o número de interseções de rodovias e o identificador da rodovia. Para avaliar o modelo de estimativa proposto, os autores propuseram o conceito CIRS (*crash injury risk susceptibility*), que mede o quanto pessoas de uma região estão suscetíveis ao risco de ferimento em acidentes.

Na Tabela 3.1, pode ser conferido um comparativo dos estudos discutidos nesta seção.

Autores	Dados de Acidentes	Algoritmos	Métricas e Melhor Resultado
Liu e Sharma [42]	Iowa, Estados Unidos (2006 e 2015)	Modelo Bayesiano espaço-temporal multivariado (MBYM) e Modelo Besag-York-Mollie (BYM)	Melhor resultado foi o MBYM, com *DIC value* = 8,371.
Kocatepe et al. [31]	Tampa Bay, na Flórida, Estados Unidos (2015)	Modelo de estimativa *mixed-logit*.	Melhor resultado foi o modelo proposto *mixed-logit*, segundo a métrica CIRS.

Tabela 3.1: Comparativo de estudos com o objetivo de avaliar o impacto socio-econômico e ambiental dos acidentes.

3.2 Classificação de Casualidades

No tocante ao uso de dados de acidentes para classificação, alguns trabalhos possuem como objetivo principal a classificação dos acidentes de acordo com a casualidade, ou seja, visam classificar a categoria das pessoas vítimas de acidentes.

Tiwari et al. usaram diferentes técnicas de clusterização e classificação para classificar dados de acidentes de acordo com sua casualidade, essa podendo ser o motorista, passageiro ou pedestre [72]. Os dados usados possuem informações da data e hora em que o acidente ocorreu, a quantidade de veículos envolvidos no acidente, a condição climática no dia e o tipo do veículo. Inicialmente, eles usaram três técnicas de classificação para solucionar o problema: Decision Tree Classifier, Lazy Classifier e Multilayer Perceptron. Depois disso, os dados foram clusterizados usando Hierarchical Clustering a fim de obter melhores resultados na classificação dos dados. A acurácia de todos os classificadores melhorou após o uso da técnica de clusterização, porém o melhor resultado foi da técnica Lazy Classifier (IBK), que obteve 84,47% de acurácia. Contudo, a abordagem dos autores visa classificar, a partir dos dados de acidentes, qual o tipo de pessoa mais provável de ser afetada por tal acidente: o motorista, os passageiros ou os pedestres.

Com o objetivo de encontrar padrões de usuários envolvidos em acidentes, Tiwari et al. usaram técnicas de classificação e clusterização para analisar dados de acidentes que

aconteceram nas rodovias da cidade Leeds, UK [73]. A base de dados usada possui cerca de 13.062 acidentes ocorridos entre 2011 e 2015, contendo onze atributos relevantes para a análise: número de veículos, data e hora do acidente, condição climática, categoria da casualidade (motorista, passageiro ou pedestre), gênero e idade da casualidade e tipo do veículo. Para classificar os dados de acidentes de acordo com a categoria da casualidade, os autores fizeram uso do Support Vector Machine (SVM), Naive Bayes (NB) e Decision Tree (J48). Inicialmente, os dados foram classificados usando essas três técnicas, onde o classificador Decision Tree obteve a melhor acurácia: 70.7%. Com o intuito de melhorar os resultados, os autores clusterizaram os dados usando o Self Organizing Map (SOM) e o K-modes e classificaram os clusters usando as técnicas citadas. O melhor resultado obtido foi com o uso do cluster K-modes, que melhorou o resultado da classificação de todas as técnicas usadas. Porém, a melhor técnica foi a Decision Tree, com um resultado final de: 81% de acurácia, 73% de precisão e 70,6% de revocação. Contudo, apesar do uso dos dados de acidentes em rodovias, a abordagem dos autores, diferentemente do que é proposto nesta pesquisa, visa classificar a categoria dos usuários envolvidos e não o risco de acidentes acontecerem em um certo trecho da rodovia.

Na Tabela 3.2, pode ser conferido um comparativo dos estudos discutidos nesta seção.

Autores	Dados de Acidentes	Algoritmos	Métricas e Melhor Resultado
Tiwari et al. [72]	Leeds, UK (2011 a 2015)	Decision Tree Classifier, Lazy Classifier e Multilayer Perceptron	Melhor resultado foi o Lazy Classifier (IBK), com 84,47% de acurácia.
Tiwari et al. [73]	Leeds, UK (2011 a 2015)	Support Vector Machine (SVM), Naive Bayes (NB) e Decision Tree (J48)	Melhor resultado foi o Decision Tree, com 81% de acurácia, 73% de precisão e 70,6% de revocação.

Tabela 3.2: Comparativo de estudos que visam classificar a casualidade de acidentes.

3.3 Análise de Dados/Padrões de Acidentes

De acordo com a literatura pesquisada, alguns trabalhos fazem uso de dados de acidentes com o objetivo de analisar os dados em busca de padrões que possam ajudar na identificação de características importantes.

Turunen utilizou o método de mineração de dados GUHA (*General Unary Hypoteses Automation*) para analisar uma matriz de *big data* contendo informações de acidentes que ocorreram entre 2004 e 2008 na Finlândia [74]. A matriz contém mais de 80.000 ocorrências de acidentes e possui cerca de 100 atributos dos acidentes, tais como número de feridos e casualidades, dia e horário do acidente, condições climáticas, localidade do acidentes, condições da estrada e tipo da pista. A execução do GUHA encontrou mais de 10.000 associações e dependências entre os dados, o que tornou possível a conclusão de que esse método consegue extrair informações dos dados que outros métodos de mineração de dados não conseguem.

Com o intuito de descobrir padrões ocultos em dados de acidentes, Ali e Hamed propuseram uma abordagem que emprega algoritmos de mineração de dados de regras associadas e clusterização para a descoberta de novas informações dos acidentes [2]. Para isso, os autores coletaram dados de acidentes de quatro anos da Província de Al-Ghat, Arábia Saudita, contendo informações do ano do acidente, a localização, o tipo do acidente, a nacionalidade do motorista e o número de feridos. Usando da ferramenta WEKA, dois algoritmos de mineração se destacaram por seus bons resultados: Apriori e Cluster. Porém, o melhor algoritmo foi o Apriori, que obteve uma performance superior ao Cluster por encontrar uma maior quantidade de regras de forma mais rápida e eficiente.

A fim de analisar dados de acidentes de fontes oficiais e não-oficiais, verificando se as fontes não-oficiais são um complemento ou podem substituir as fontes oficiais, Dos Santos et al. exploram o potencial de integrar tais fontes de dados [15]. Para isso, os autores usaram dados de acidentes da cidade de Belo Horizonte, Brasil, entre setembro e novembro de 2014. A fonte de dados oficial utilizada foi fornecida pela companhia de trânsito municipal, a BHTrans, enquanto a fonte de dados não-oficial consistiu em dados coletados do aplicativo Waze, que ajuda motoristas a transitarem nas cidades. Para integrar e comparar os dados, os autores usaram uma série de critérios para comparar as fontes de dados e verificar a existência

de combinações de acidentes, isto é, se existe correspondência de um acidente da fonte não-oficial na fonte oficial. Com isso, os autores chegaram a conclusão que cerca de 7% dos dados reportados oficialmente também foram reportados nos dados não-oficiais.

Kumar e Toshniwal propuseram um framework para analisar padrões de acidentes em rodovias para diferentes tipos de acidentes [34]. O framework usa a técnica k-modes clustering como tarefa preliminar na segmentação de 11.574 acidentes em rodovias de Dehradun (Índia) entre os anos de 2009 e 2014, que possuem como atributos o dia e turno em que o acidente ocorreu, o número de feridos, a idade e o gênero da vítima, o tipo da rodovia, o traçado da pista, a severidade e o tipo do acidente. Os autores também fazem uso de mineração de regras de associação para identificar as várias circunstâncias associadas com a ocorrência de acidentes para toda a base de dados e também para os clusters identificados pelo algoritmo K-modes. Apesar do uso de uma base de dados detalhada, o foco do estudo foi provar que usar técnicas de clusterização antes de fazer a análise dos acidentes ajuda a identificar descobertas mais importantes sobre os dados.

A fim de identificar locais com uma grande frequência de acidentes de trânsito e investigar possíveis padrões e características que caracterizam os acidentes, Kumar e Toshniwal propuseram uma abordagem usando técnicas de mineração de dados para possibilitar a caracterização de locais de acidentes nas rodovias [36]. Para isso, foram considerados 9.640 dados de acidentes para um período de 6 anos (2009 à 2014) na cidade de Dehradun, Índia, contendo informações sobre: idade, gênero e número de vítimas, categoria do acidentes, data e hora, localização, luminosidade da pista, severidade do acidente e o tipo da rodovia. Os autores usaram o algoritmo k-means para agrupar os locais dos acidentes em três categorias diferentes: frequência alta, frequência moderada e frequência baixa de acidentes. Depois, foi usado o algoritmo de mineração de regras associadas em cada grupo para descobrir diferentes fatores associados com cada grupo. Apesar do uso dos dados de acidentes, o objetivo do estudo foi caracterizar os locais dos acidentes de acordo com características em comum dos acidentes ocorridos no local, diferentemente desta pesquisa, que visa prever o risco de acidentes graves em trechos de rodovias.

Na Tabela 3.3, pode ser conferido um comparativo dos estudos discutidos nesta seção.

Autores	Dados de Acidentes	Algoritmos	Métricas e Resultados
Turunen [74]	Finlândia (2004 a 2008)	GUHA	10.000 associações e dependências entre os dados
Ali e Hamed [2]	Al-Ghat, Arábia Saudita (2010, 2012, 2013, 2014)	Apriori e Cluster	Apriori obteve melhores regras e melhor performance.
Dos Santos et al. [15]	Belo Horizonte, Brasil (Setembro a Novembro de 2014)	Algoritmo de integração das bases oficial e não-oficial	7% dos dados reportados oficialmente também foram reportados nos dados não-oficiais.
Kumar e Toshniwal [34]	Dehradun, Índia (2009 a 2014)	*K-modes Clustering* e Apriori	Quantidade de clusters gerados (6) e regras fortes para cada cluster.
Kumar e Toshniwal [36]	Dehradun, Índia (2009 a 2014)	K-means e Apriori	Regras fortes de associação para três *clusters*: frequência alta, frequência moderada e frequência baixa de acidentes.

Tabela 3.3: Comparativo de estudos que possuem o objetivo de analisar dados ou encontrar padrões em acidentes.

3.4 Análise e Avaliação de Fatores

No que se refere à análise de dados de acidentes, existem estudos que possuem como objetivo a análise e avaliação de fatores que podem ser importantes na determinação da severidade do acidente, podendo oferecer risco à segurança nas rodovias.

A fim de avaliar fatores que podem influenciar o acontecimento de acidentes na via expressa de Xangai, China, Gao et al. usaram regras de associação para descobrir associações entre estes fatores, que podem ser usadas para reduzir as ocorrências de acidentes [17]. Os dados usados foram coletados entre Abril e Junho de 2014, contendo inforamções sobre o tipo do acidente, a data em que o acidente ocorreu, presença de placa com limite de velocidade no local do acidente e as condições climáticas no momento do acidente. Os autores propuseram dois métodos, o primeiro é um método de triagem automática de regras fortes baseada em clusterização, e o outro é um método de filtragem de regras fracas baseado em experiência. Com isso, os autores conseguiram bons resultados na identificação de regras fortes associadas aos dados de acidentes.

Com o intuito de analisar os diversos fatores que podem influenciar na ocorrência de um acidente, Li et al. propuseram sistema para análise de dados de acidentes de tráfego, composto por sete partes: a análise das informações básicas dos acidentes, a análise do motorista, a análise do veículo, a análise da estrada onde o acidente ocorreu, análise da razão do acidente, análise de acidentes multi-dimensional e geração de relatório da análise geral dos acidentes [40]. Para isso, os autores utilizaram a tecnologia OLAP (*on-line analytical processing*) para análise e visualização dos dados e Redes Bayesianas para analisar os dados multidimensionais. O *framework* proposto pode ser usado e configurado para quaisquer bases de acidentes.

Wang e Ohsawa propuseram um modelo de avaliação para risco de acidentes de tráfego, onde definiram o relacionamento entre dados urbanos e dados de acidentes de tráfego [75]. Para isso, os autores usaram análise fatorial, modelagem de equações estruturais e mineração de dados para construir um quadro teórico para a análise da taxa de acidentes de tráfego, usando os dados urbanos de Beijing, China. Os dados de acidentes usados pelos autores são o resultado do uso de equações a partir de outras informações, que somadas resultam na taxa de acidentes. A taxa de acidente é a soma da taxa de fatalidade, a taxa de feridos e a taxa

de casualidades, onde essas taxas são calculadas a partir da divisão do número de mortes pela quantidade de acidentes, número de feridos pela quantidade de acidentes e número de casualidades por número de acidentes, respectivamente. Os autores dividiram os dados urbanos de acordo com suas categorias e com a taxa de acidente, e usaram essa combinação para analisar o risco de acidentes de tráfego. Eles concluíram que a taxa de acidentes de trânsito pode ser descrita pela combinação da estrutura da população, o caráter da estrada, o sistema de tráfego público e as instalações públicas.

Já Bülbül e Kaya fizeram um estudo com o objetivo de encontrar as melhores técnicas de classificação por aprendizagem de máquina para analisar dados de acidentes que aconteceram em Istambul, na Turquia [9], visando estimar o número de acidentes para prevenir futuras ocorrências. Os métodos de classificação foram usados para analisar os fatores dos acidentes que aconteceram. Contudo, diferentemente desta dissertação, os autores levaram em consideração apenas o tipo do veículo envolvido no acidente, a hora e localização em que o acidente ocorreu e se no momento do acidente estava chovendo ou não. Usando a ferramenta WEKA, os autores concluíram que os melhores algoritmos para solucionar esse problema foram: CART, IBK, C4,5 e Naive Bayes, pois obtiveram a melhor acurácia, Kappa statistic e F-criterion. Os resultados da acurácia foram: 81,5%, 81,3%, 81%, 80,2%, respectivamente.

Guo et al. fizeram um estudo com o objetivo de avaliar o impacto de vários fatores de risco nos acidentes de trânsito que apresentam diferentes tipos de colisões em áreas de rodovias [19]. Foram coletados dados da rodovia 367 da Flórida, Estados Unidos, em um período de três anos, resultando numa base com 3.315 acidentes com três diferentes formas de colisão: traseira, regular e angular. Os autores desenvolveram um modelo de parâmetro multivariado chamado Poisson-lognormal (RP-MVPLN) para correlacionar acidentes através do tipo da colisão com a heterogeneidade não observada através de observações. Para mostrar a importância do modelo desenvolvido, os autores o compararam com um MVPLN construído com base no algoritmo Naive Bayes, onde o modelo proposto obteve melhores resultados.

Abellán et al. usaram a técnica de classificação Decision Tree para analisar a gravidade de acidentes usando dados coletados de rodovias rurais da cidade de Granada, Espanha [1]. O objetivo principal do estudo é encontrar uma forma mais eficaz de extrair as regras geradas

pela Decision Tree quando aplicada a dados de acidentes. Cerca de 1.801 dados de acidentes foram coletados no período de 2003 à 2009 com informações sobre o tipo do acidente (colisão, saída da pista, outros), a causa, o dia e a hora do acidente, condições climáticas, características da rodovia, número de feridos e de pessoas envolvidas, tipo do veículo e severidade do acidente. A fim de extrair mais conhecimento dos dados, os autores usaram Decision Tree para cada atributo disponível da base de dados. Isso resultou em mais de setenta regras válidas e significativas, que foram usadas como métricas para comprovar a efetividade do método proposto.

Kwon et al. usaram algoritmos de classificação para analisar possíveis fatores de risco à segurança nas rodovias através de dados de acidentes [38]. Para isso, foram utilizados relatórios de acidentes que aconteceram nas rodovias da Califórnia, coletados pela California Highway Patrol (CHP) e acumulados desde 1973. Contudo, os autores escolheram trabalhar apenas com dados dos anos 2004 à 2010, que possuem como atributos as características do veículos envolvidos no acidente, o tipo da rodovia, a data e hora em que o acidente ocorreu, a condição climática e o tipo de acidente. Usando o Naïve Bayes Classifier e o Decision Tree Classifier para classificar seus dados de acordo com fatores de risco às rodovias, os autores compararam seus resultados usando um modelo de regressão logística. O melhor classificador para o problema foi o Decision Tree que considera a dependências entre fatores, obtendo melhores resultados para todos os valores de threshold da curva ROC. Os autores ranquearam os fatores de risco mais significativos dos dados, sendo eles: tipo de colisão, população, rodovia estadual e o movimento precedente à colisão.

Na Tabela 3.4, pode ser conferido um comparativo dos estudos discutidos nesta seção.

Autores	Dados de Acidentes	Algoritmos	Métricas e Resultados
Gao et al. [17]	Xangai, China (Abril a Junho de 2014)	Triagem Automática de Regras Fortes baseada em Clusterização e Filtragem de Regras Fracas baseado em experiência	Número de regras associadas: 33040.
Li et al. [40]	Beijing, China	OLAP (on-line analytical processing) e Redes Bayesianas	Análise da correlação entre fatores de acidentes e probabilidade dada pela rede bayesiana.
Wang e Ohsawa [75]	Beijing, China (2010 a 2014)	Keygraph e análise de fatores	Modelo de avaliação de risco de acidentes de trânsito.
Bülbül e Kaya [9]	Istambul, Turquia (2013)	CART, IBK, C4,5 e Naive Bayes	Melhor algoritmo foi o CART, com 81,5% de acurácia, 81,2% de precisão, 81% de revocação, Kappa statistic = 0.620 e F-criterion = 0.810.
Guo et al. [19]	Flórida, Estados Unidos (2004 a 2006)	Poisson-lognormal (RP-MVPLN) e MV-PLN	Melhor resultado foi do modelo RP-MVPLN, com *DIC measure* = 3298.65.
Abellán et al. [1]	Granada, Espanha (2003 a 2009)	Decision Tree	Número de regras associadas: mais de 70.
Kwon et al. [38]	Califórnia, Estados Unidos (2004 a 2010)	Naïve Bayes Classifier, Decision Tree Classifier e Regressão Logística	Decision Tree obteve os melhores valores de threshold da curva ROC.

Tabela 3.4: Comparativo de estudos que visam analisar e avaliar fatores importantes dos acidentes.

3.5 Classificação/Predição da Severidade do Acidente

No que concerne ao uso de dados de acidentes para classificação e/ou predição, alguns estudos visam determinar a severidade do acidente usando diferentes abordagens, tais como técnicas de clusterização, métodos estatísticos e técnicas de aprendizagem de máquina.

Tambouratzis et. al. usaram uma combinação de redes neurais artificiais e árvores de decisão para prever a severidade (leve, sério ou fatal) de acidentes [70]. Os dados usados no estudo são referentes a acidentes ocorridos em Chipre, durante o ano de 2005, e foram coletados e disponibilizados pela polícia local. Cada acidente possui informações do dia e horário em que aconteceu, características da pista (como limite de velocidade, largura da pista, tipo da pista), condições climáticas, informações do motorista (idade, tipo da carteira de motorista) e características do carro. A rede neural utilizada é baseada em probabilidades e possui quatro camadas no total: uma de entrada, duas camadas ocultas e uma de saída. A árvore de decisão foi utilizada em conjunto com a rede neural a fim de maximizar a acurácia da classificação. A combinação destes dois algoritmos de aprendizado empregado para a classificação da severidade de acidentes obteve uma acurácia de 70%.

Richard e Ray usaram o modelo de classificação Random Forest para prever se um acidente tem casualidades ou não [60]. Para isso, os autores usaram dados de acidentes públicos que aconteceram nas cidades canadenses de Fredericton, entre 2007 e 2016, e Laval, entre 2011 e 2016. Os dados possuem informações de número de feridos e casualidades, data do acidente, número de veículos envolvidos, tipo do acidente, estação do ano, condição climática e a visibilidade da pista. Uma análise de dados usando frameworks de dados espaciais e sistemas de *big data* foi feita a fim de comparar a importância dos fatores para cada cidade, concluindo que há diferenças: na cidade de Fredericton, os fatores mais importantes foram a condição climática e o número de veículos, enquanto em Laval os mais importantes foram o limite de velocidade e o dia da semana. Os fatores mais importantes de cada cidade foram considerados na classificação, que foi avaliada pela métrica AUROC. Quanto maior o valor do AUROC, melhor o classificador. O valor da métrica AUROC para a classificação usando dados de Fredericton foi de 0.716, enquanto o valor AUROC para a classificação usando dados de Laval foi de 0.702.

Kumar et al. fizeram um estudo comparando o uso de diferentes técnicas de clusterização

para dados de acidentes em rodovias de um distrito Indiano, com o objetivo de melhorar a classificação da severidade dos acidentes [32]. Os dados usados no estudo possuem como atributos o dia e turno em que o acidente ocorreu, o número de vítimas, o gênero da vítima, o tipo da rodovia, o traçado da pista e a severidade do acidente. Os autores fizeram uso de três técnicas de clusterização: Latent Class Clustering (LCC), K-modes Clustering e BIRCH Clustering. Para classificar os dados clusterizados, foram usadas as técnicas Naïve Bayes, Random Forest e Support Vector Machine, disponibilizadas na ferramenta WEKA. Como resultado, os autores chegaram à conclusão que o melhor cluster para esse problema é o LCC, um vez que aumentou a acurácia dos três classificadores usados. O classificador que obteve o melhor resultado foi o Random Forest, com 81% de acurácia.

Usando dados de acidentes que aconteceram em uma das rodovias mais movimentadas de Bangladesh, Satu et al. analisaram os dados e propuseram uma abordagem para predição da severidade dos acidentes de trânsito que ocorrem nessa rodovia usando árvores de decisão [64]. Com dados coletados durante 5 anos, os autores conseguiram acumular informações de 892 acidentes, que possuem atributos como: local do acidente, número de veículos envolvidos no acidente, data, número de casualidades, tipo da colisão, condição climática, entre outros. Os atributos mais significativos da base de dados foram extraídos e usados na classificação de doze diferentes implementações do algoritmo de árvores de decisão, que obteve como melhor resultado a árvore J48 (pruned), com 78,9% de acurácia e 62,6% de precisão.

Para prever a severidade de acidentes de trânsito, Iranitalaba e Aemal Khattakb fizeram um estudo comparativo usando quatro métodos estatísticos e de aprendizagem de máquina, usando dados de acidentes coletados entre 2012 e 2015 no estado de Nebraska, Estados Unidos [23]. Os autores consideraram apenas acidentes que envolveram dois carros, resultando em uma base de dados com cerca de 68.448 entradas e informações sobre as características da pista, condição climática, luminosidade e características do acidente. Para a classificação dos dados, os autores escolheram Multinomial Logit (MNL), Nearest Neighbor Classification (NNC), Support Vector Machines (SVM) e Random Forests (RF) como os métodos de aprendizagem de máquina a serem usados para prever a severidade de acidentes, usando os dados de 2012 à 2014 como treino e os dados de 2015 como teste. Eles também investigaram o impacto do uso de dois métodos de clusterização, K-means Clustering (KC) e Latent

Class Clustering (LCC), na performance dos modelos de predição. Os resultados do estudo mostraram que a combinação que resultou na melhor performance foi o uso do método de classificação NNC com o cluster KC, apresentando uma acurácia de 73,95%.

Na Tabela 3.5, pode ser conferido um comparativo dos estudos discutidos nesta seção.

Autores	Dados de Acidentes	Algoritmos	Métricas e Resultados
Tambouratzis et. al. [70]	Chipre (2005)	Redes neurais artificiais e árvores de decisão	Modelo proposto obteve uma acurácia de 70%
Richard e Ray [60]	Fredericton, Canadá (2007 a 2016) e Laval,Canadá (2011 a 2016)	*Frameworks* de dados espaciais e sistemas de *big data*	Valor AUROC (Fredericton)=0.716; Valor AUROC (Laval)=0.702.
Kumar et al. [32]	Muzzafarnagar district, Índia	Naïve Bayes, Random Forest e Support Vector Machine (weka)	Melhor classificador foi o Random Forest, com 81% de acurácia
Satu et al. [64]	Bangladesh, (2007 a 2011)	Árvores de Decisão	Melhor classificador foi a J48 (pruned), com 78,9% de acurácia e 62,6% de precisão
Iranitalaba e Aemal Khattakb [23]	Nebraska, Estados Unidos (2012 a 2015)	Multinomial Logit (MNL), Nearest Neighbor Classification (NNC), Support Vector Machines (SVM), Random Forests (RF), K-means Clustering (KC) e Latent Class Clustering (LCC)	Melhor resultado obtido com classificador NNC e cluster KC. Acurácia = 73,95%

Tabela 3.5: Comparativo de estudos com o objetivo de classificar ou prever a severidade dos acidentes.

3.6 Detecção/Predição de Áreas de Risco de Acidentes

No tocante ao uso de dados de acidentes em rodovias, existem estudos que visam detectar e/ou prever áreas que possuem risco de acidentes a fim de ajudar na prevenção destes. Nesse contexto, diferentes técnicas são usadas, tais como algoritmos de aprendizado de máquina e de clusterização.

Katsoukis et al. usaram mineração de dados e ferramentas algébricas *fuzzy* (*Fuzzy Algebraic Tools*) para classificar áreas de risco de acidentes na Grécia, de acordo com o número de ocorrências de acidentes em determinada localidade [28]. Diferentemente do proposto nesta dissertação, os autores utilizaram apenas a quantidade de acidentes e a quantidade de fatalidades por município da Grécia para inferir as áreas de risco.

Ryder and Wortmann propuseram uma abordagem que visa detectar e classificar locais propícios à acidentes [62]. O objetivo dos autores é de alertar o motorista, em tempo real e por meio de um aplicativo mobile, de perigos iminentes na estrada e dos locais em que acidentes podem acontecer. Para isso, os autores fizeram uso de dados de acidentes nas ruas da Suíça e de eventos detectados e identificados dentro do veículo, como a frenagem brusca do carro e a localização na qual esse evento ocorreu. Com os dados de acidentes, foi investigada a ligação da frequência do tráfego no local com o número de acidentes ocorridos e o fluxo do tráfego na rodovia. Esses dados foram filtrados para levar em consideração um período de cinco anos e agrupados para representar os locais suscetíveis a acidentes na Suíça. Apesar da proposta dos autores de identificar locais suscetíveis a acidentes, o foco da pesquisa foi investigar a efetividade de combinar técnicas para a identificação automática de possíveis causas do motorista apresentar um comportamento evasivo na direção ou de frear bruscamente enquanto dirige, divergindo da ideia de usar o dados de acidentes para classificar a gravidade de trechos da rodovia.

Ryder et al. fizeram uso de dados de acidentes de trânsito para desenvolver um sistema que usa Decision Support Systems para ajudar a prevenir acidentes [61]. Esse sistema é responsável por mandar avisos aos motoristas quando os mesmos estão próximos de áreas consideradas de “risco”. Para identificar tais áreas, os autores usaram uma base de dados de acidentes da Suíça composta por 266.000 acidentes que ocorreram entre 2011 e 2015, com atributos detalhados sobre cada acidente que foram separados em três categorias: “o quê”,

"por quê" e "onde". O atributo "o quê" identifica os tipos de veículos envolvidos no acidente. O atributo "por quê" descreve a causa predominante do acidente, enquanto o atributo "onde" guarda a informação de onde o acidente aconteceu. O algoritmo DBSCAN foi usado para clusterizar os dados e assim encontrar os locais com risco de acidentes. A validação do sistema foi feita com a ajuda de 57 motoristas em um teste de campo que cobriu mais de 170.000 km, e os autores concluíram que o uso do sistema de avisos ajuda na melhora do comportamento do motorista de acordo com o uso.

Para prever áreas com risco de acidentes trânsito na cidade de Beijing, Ren et al. propuseram um modelo de Deep Learning para a predição do risco de acidentes baseada na correlação espaço-temporal dos dados [58]. Para isso, os autores coletaram dados de acidentes entre 2016 e 2017, que possuíam informações sobre o horário e a coordenada geográfica na qual o acidente aconteceu. Por ser difícil prever se um acidente vai acontecer ou não, os autores tentaram prever a frequência de acidentes de trânsito, a qual chamaram de risco, que foi calculada por meio da média de contagem de acidentes que aconteceram no mesmo horário em um período de tempo de três dias. Por exemplo, se cinco acidentes aconteceram no intervalo de 8 a 9 horas da manhã nos últimos três dias, a frequência é de aproximadamente 1.67 acidentes/hora. Com os dados em função do espaço e do tempo, foi construído um modelo de deep learning baseado no Long short-term memory (LSTM), com duas camadas de entrada, sete camadas escondidas, das quais quatro são LSTM e três são camadas densas que usam RELU como função de ativação e uma camada de saída. Para avaliar o modelo de classificação, foram usadas métricas que medem a acurácia de variáveis contínuas: o Erro Absoluto Médio (Mean Absolute Error), o Erro Quadrático Médio (Mean Squared Error) e a Raiz do Erro Quadrático Médio (Root Mean Squared Error), e os resultados foram comparados com os resultados usando os algoritmos: Lasso, Ridge, Support Vector Regression (SVR), Decision Tree Regression (DTR), Random Forest Regression (RFR), Multilayer Perceptron(MLP) e Autoregressive Moving Average Model (ARMA). Dentre todos os algoritmos, o modelo proposto obteve o menor valor em todas as métricas analisadas. Porém, esse estudo leva em consideração apenas a quantidade de acidentes de trânsitos na classificação do risco, sem considerar fatores que podem ser relevantes para classificação, como o tipo do acidente, a condição climática, o tipo da pista e o sentido da via.

Na Tabela 3.6, pode ser conferido um comparativo dos estudos discutidos nesta seção.

Autores	Dados de Acidentes	Algoritmos	Métricas e Resultados
Katsoukis et al. [28]	Grécia (2016)	*Fuzzy Algebraic Tools*	Coeficiente de similaridade do total de acidentes e Coeficiente de similaridade do total de acidentes fatais.
Ryder and Wortmann [62]	Suíça (2011 a 2015)	*Inception Neural Network*	Proposta de modelo que detecta eventos associados a acidentes em tempo real.
Ryder et al. [61]	Suíça (2011 a 2015)	Decision Support Systems, DBSCAN.	Validação feita em teste de campo com 57 motoristas.
Ren et al. [58]	Beijing (2016 e 2017)	Deep Learning (LSTM), Lasso, Ridge, SVR, DTR, RFR, MLP e ARMA.	Melhor Resultado foi o Deep Learning, com um Erro Absoluto Médio = 0.014, Erro Quadrático Médio = 0.001 e a Raiz do Erro Quadrático Médio = 0.034.

Tabela 3.6: Comparativo de estudos que possuem como objetivo a detecção ou predição de áreas de risco de acidentes.

3.7 Considerações Finais

Neste capítulo foram apresentados alguns trabalhos relacionados à pesquisa realizada nesta dissertação, incluindo estudos sobre análise e avaliação de fatores de risco a acidentes, classificação e predição da severidade de acidentes, detecção e predição de áreas em estradas que possuem risco de acidentes, impactos socio-econômicos e ambientais derivados de acidentes e busca por padrões nos dados.

No próximo capítulo será mostrada a metodologia e os experimentos realizados neste trabalho.

Capítulo 4

Metodologia e Experimentos

Neste capítulo, é apresentada a metodologia utilizada nesta dissertação, que faz uso de técnicas de aprendizagem de máquina para identificar e prever trechos das rodovias brasileiras com risco de acidentes graves. Também são apresentados os experimentos realizados neste estudo. Inicialmente, será apresentada a metodologia adotada neste estudo (Seção 4.1). Em seguida, na Seção 4.2, será discutida a análise dos dados de acidentes da PRF, bem como o pré-processamento ao qual foram submetidos. A Seção 4.4 apresenta os modelos de classificação usados no experimento, bem como a modelagem do experimento e o ambiente de execução utilizado. Por fim, na Seção 4.4, são apresentadas as considerações finais.

4.1 Metodologia

A solução desenvolvida neste estudo tem como objetivo identificar e prever, nas rodovias brasileiras, trechos que possuem risco de acidentes graves de acordo com os acidentes que já aconteceram e foram registrados pela Polícia Rodoviária Federal (PRF). A cada ano, a PRF disponibiliza informações sobre acidentes que aconteceram nas rodovias brasileiras. Portanto, dez anos de dados de acidentes podem ser usados, de 2007 a 2017.

A identificação das áreas de risco de acidentes graves abre inúmeras oportunidades de desenvolvimento de soluções para melhorar a segurança nas estradas, sendo possível também identificar as condições que tornam a rodovia mais propícia a acidentes. A fim de desenvolver uma solução utilizando os dados abertos disponíveis, este estudo incorpora técnicas de pré-processamento, análise de dados, aprendizagem de máquina supervisionada e aprendizado

de máquina automatizado para prever, de acordo com informações coletadas, trechos de rodovias com potencialidade de ocorrência de acidentes graves ou não graves, de acordo com as características especificadas.

Portanto, para este estudo, foram utilizados os dados de acidentes disponibilizados pela PRF que descrevem acidentes que ocorreram nas rodovias do país. Esse dados possuem informações da data em que o acidente ocorreu, a condição climática no momento do acidente, as pessoas envolvidas, se teve feridos leves ou graves, o tipo do acidente, o dia da semana em que o acidente ocorreu, entre outras características. Um exemplo de acidente contido na base de dados pode ser visto na Tabela 4.1. A análise detalhada dos dados está descrita na Seção 4.2.

Na Figura 4.1 é apresentado o fluxo metodológico adotado neste estudo. A primeira etapa do fluxo consiste na coleta dos dados de acidentes. Após a coleta dos dados, a segunda etapa é responsável pelo pré-processamento dos dados coletados e pela seleção dos atributos relevantes ao estudo. Os dados, apesar de informarem o número de feridos ou mortos nos acidentes registrados, não informam se o acidente foi grave ou não. Portanto, para categorizar um acidente em grave ou não grave, foi utilizado um atributo derivado chamado GRAVIDADE, que descreve, de acordo com a quantidade de feridos ou mortos, a gravidade do acidente. Detalhes sobre o pré-processamento dos dados, seleção dos atributos e o critério usado na categorização da gravidade dos acidentes são apresentados na Seção 4.2.

Com os dados estruturados, a terceira etapa da metodologia adotada consiste na criação de duas bases de dados: a base de dados balanceada e a base de dados desbalanceada. Existe uma grande desproporção entre a quantidade de acidentes considerados “graves” e a quantidade de acidentes considerados “não-graves”, a maioria sendo “não-grave” (Seção 4.2). Isto posto, o uso de dados desbalanceados pode comprometer a performance de algoritmos de aprendizado [20]. Desta forma, a base de dados balanceada foi gerada por meio da técnica *random undersampling*, que remove instâncias randômicas da classe majoritária [20].

Com as bases de dados definidas, foi necessário selecionar a melhor maneira de usar os dados de forma a prever trechos das rodovias com risco de acidentes graves. De acordo com o estudo da literatura, muitos trabalhos abordam problemáticas similares utilizando técnicas de mineração e clusterização para encontrar padrões nos dados, e assim identificar áreas ou informações importantes.

Atributo	Valor
data	2017-01-01
dia_semana	Domingo
horario	01:00:00
uf	PB
br	104
km	3
município	Nova Floresta
causa_acidente	Ingestão de Álcool
tipo_acidente	Tombamento
classificação_acidente	Com Vítimas Fatais
fase_dia	Plena noite
sentido_via	Decrescente
condição_meteorológica	Céu claro
tipo_pista	Simples
tracado_via	Curva
uso_solo	Sim
pessoas	1
mortos	1
feridos_leves	0
feridos_graves	0
feridos	0
Ilesos	0
Ignorados	0
veiculos	1

Tabela 4.1: Exemplo da estrutura de um dado de acidente.

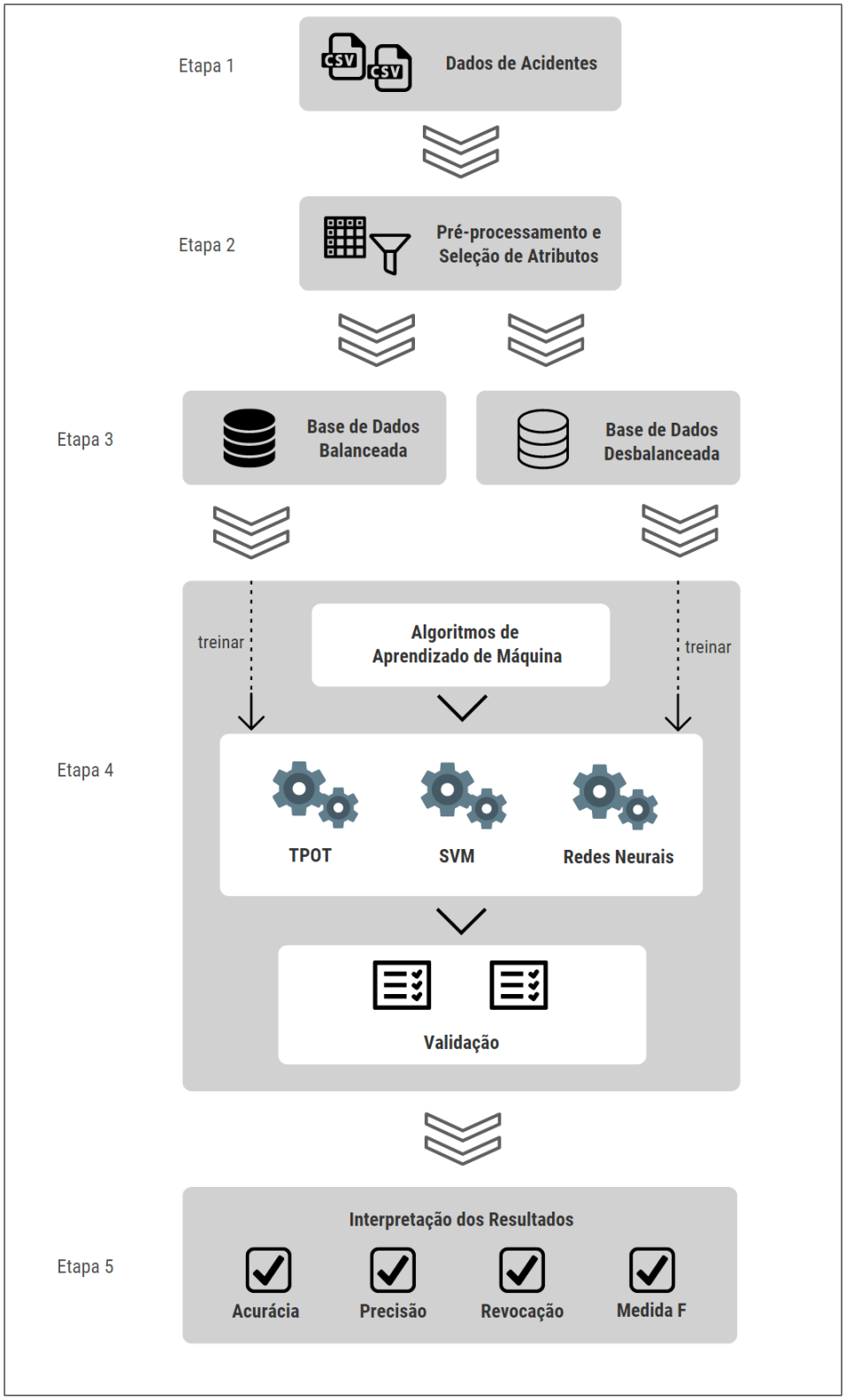

Figura 4.1: Fluxo metodológico do estudo.

Porém, o objetivo deste estudo é ir um pouco mais além da identificação de padrões utilizando clusters, partindo assim para a previsão de áreas de risco de acordo com as informações de acidentes disponíveis. Como dito, o estudo de dados de acidentes é popular em diversos lugares, mas não é comum no Brasil, principalmente fazendo uso dos dados da PRF.

Como os dados usados neste estudo são oriundos de acidentes, não é todo trecho de rodovia que possui essa informação. Esse é um dos motivos que torna a previsão de trechos com risco acidentes graves importante. Uma outra motivação, é fazer uso da previsão de trechos com potencialidade de acidentes graves para criar uma aplicação que avisa ao motorista, de acordo com informações captadas no momento, se a rodovia possui trechos com risco de acidentes graves. Por causa desses motivos, foi feito o uso de algoritmos de aprendizagem de máquina para prever tais informações, fazendo um comparativo entre alguns classificadores a fim de encontrar o melhor resultado para esse problema.

A quarta etapa da metodologia adotada consiste em quatro fases: na primeira, é feito um estudo dos algoritmos de aprendizado de máquina mais adequados a esta pesquisa; na segunda fase, os algoritmos escolhidos são implementados; a terceira fase faz uso da base de dados balanceada e da base de dados desbalanceada para treinar os modelos implementados; e a quarta fase consiste na validação dos modelos treinados.

Um dos classificadores mais utilizados atualmente é o SVM (Support Vector Machine), muito popular em diversas áreas incluindo estudos relacionados a esta dissertação [13, 23]. Devido ao seu grande uso e resultados [13], para este estudo, utilizamos a implementação do SVM disponível na biblioteca do scikit-learn [53], que permite o uso de diferentes kernels: linear, RBF, sigmoid e polinomial.

Para ajudar na escolha dos melhores algoritmos e configurações para os dados usados neste estudo, foi utilizado o framework de AutoML TPOT, que vem sendo cada vez mais utilizado por ser uma opção acessível, flexível e escalável. Os algoritmos disponíveis no TPOT estão divididos em pacotes, onde os mais adequados para a solução proposta neste estudo estão no pacote TPOT Default e no pacote TPOT Sparse. Esses pacotes são mais completos e permitem o uso de processos em paralelo, sendo adequados para dados esparsos.

Além do SVM e dos algoritmos que, segundo o TPOT, são adequados a este estudo, foi utilizado um outro método muito utilizado na literatura, a rede neural artificial. O uso desse modelo pode gerar melhores resultados uma vez que ele consegue aprender uma represen-

tação mais efetiva das características disponíveis, o que pode ser muito útil no contexto das características dos dados de acidentes.

Com a implementação dos métodos discutidos e o uso das bases de dados para treinamento, a quinta etapa da metodologia adotada consiste na interpretação dos resultados das métricas acurácia, precisão, revocação e medida F. Essa métricas foram avaliadas para cada experimento realizado e, ao final do estudo, as melhores técnicas e configurações para o objetivo pretendido foram identificadas e indicadas.

4.2 Dados de Acidentes

Por meio da política de dados abertos do governo, a Polícia Rodoviária Federal (PRF) disponibiliza, todo ano, os dados que coletaram referentes à acidentes e infrações de trânsito que ocorreram em todo o Brasil. Neste estudo, a base de dados utilizada é composta por todos os dados de acidentes em rodovias que foram disponibilizados no site da PRF, do ano 2007 à 2017 e possui aproximadamente 1,6 milhões de entradas de acidentes.

Os dados de 2007 a 2016 possuem 24 atributos, descritos na Tabela 4.2. Já os dados do ano 2017 possuem os 24 atributos dos anos anteriores, com a adição de uma nova característica: a latitude e a longitude do ponto onde o acidente aconteceu.

Os dados de acidentes possuem informações do quilômetro da rodovia no qual o acidente aconteceu, característica muito importante uma vez que o objetivo deste trabalho é classificar o risco de acidentes em trechos das rodovias. Porém, mesmo com a grande quantidade de dados de acidentes disponíveis, existem trechos de rodovias que não possuem essa informação.

A Figura 4.2 mostra as dez rodovias com o maior número de acidentes registrados pela PRF entre 2007 a 2017. É possível verificar que a rodovia com o maior número de acidentes é a SP-116. Ainda assim, nesta rodovia, existe um grande trecho que não possui nenhuma informação de acidente: do quilômetro 232 ao quilômetro 267. Isto pode ser visto na Figura 4.3, que mostra a quantidade de acidentes por quilômetro da rodovia SP-116. Esse é um dos motivos que torna a previsão de trechos com risco acidentes graves importante.

Algumas rodovias possuem apenas um acidente registrado, outras possuem dois. Não se sabe se a pequena quantidade de informações se dá pela pouca incidência de acidentes na

Atributo	Descrição
data_inversa	Data da ocorrência no formato aaaa/mm/dd. Ex.: 2017/12/01
dia_semana	Dia da semana da ocorrência. Ex.: Segunda, Terça, etc.
horario	Horário da ocorrência no formato hh:mm:ss. Ex.: 08:30:45
uf	Unidade da Federação. Ex.: MG, PE, DF, etc.
br	Variável com valores numéricos representando o identificador da BR do acidente. Ex.: 101, 230, 116, etc.
km	Identificação do quilômetro onde ocorreu o acidente, com valor mínimo de 0,1km. Ex.: 10, 50, 114, etc.
município	Nome do município de ocorrência do acidente. Ex.: Campina Grande, São Paulo, Salvador, etc.
causa_acidente	Identificação da causa presumível do acidente. Ex.: falta de atenção, Velocidade incompatível, etc.
tipo_acidente	Identificação do tipo de acidente. Ex.: colisão frontal, saída de pista, etc.
classificação_acidente	Classificação quanto à gravidade do acidente: sem vítimas, com vítimas feridas, com vítimas fatais e ignorado.
fase_dia	Fase do dia no momento do acidente. Ex.: amanhecer, pleno dia, plena noite e anoitecer.
sentido_via	Sentido da via considerando o ponto de colisão. Ex.: crescente e decrescente.
condição_meteorológica	Condição meteorológica no momento do acidente. Ex.: céu claro, chuva, sol, granizo, vento, nublado e neve.

Atributo	Descrição
tipo_pista	Tipo da pista considerando a quantidade de faixas. Ex.: simples, dupla ou múltipla.
tracado_via	Descrição do traçado da via. Ex.: reta, curva, ponte, cruzamento e túnel.
uso_solo	Descrição sobre as características do local do acidente. Ex.: urbano ou rural.
pessoas	Total de pessoas envolvidas na ocorrência. Ex.: 1, 2, etc.
mortos	Total de pessoas mortas envolvidas na ocorrência. Ex.: 0, 2, etc.
feridos_leves	Total de pessoas com ferimentos leves envolvidas na ocorrência. Ex.: 0, 1, etc.
feridos_graves	Total de pessoas com ferimentos graves envolvidas na ocorrência. Ex.: 1, 2, etc.
feridos	Total de pessoas feridas envolvidas na ocorrência (é a soma dos feridos leves com os graves). Ex.: 1, 2, 3, etc.
Ilesos	Total de pessoas ilesas envolvidas na ocorrência. Ex.: 0, 1, etc.
Ignorados	Total de pessoas envolvidas na ocorrência e que não se soube o estado físico. Ex.: 1, 2, etc.
veiculos	Total de veículos envolvidos na ocorrência. Ex.: 1, 2, 3, etc.

Tabela 4.2: Atributos dos dados de acidentes da PRF.

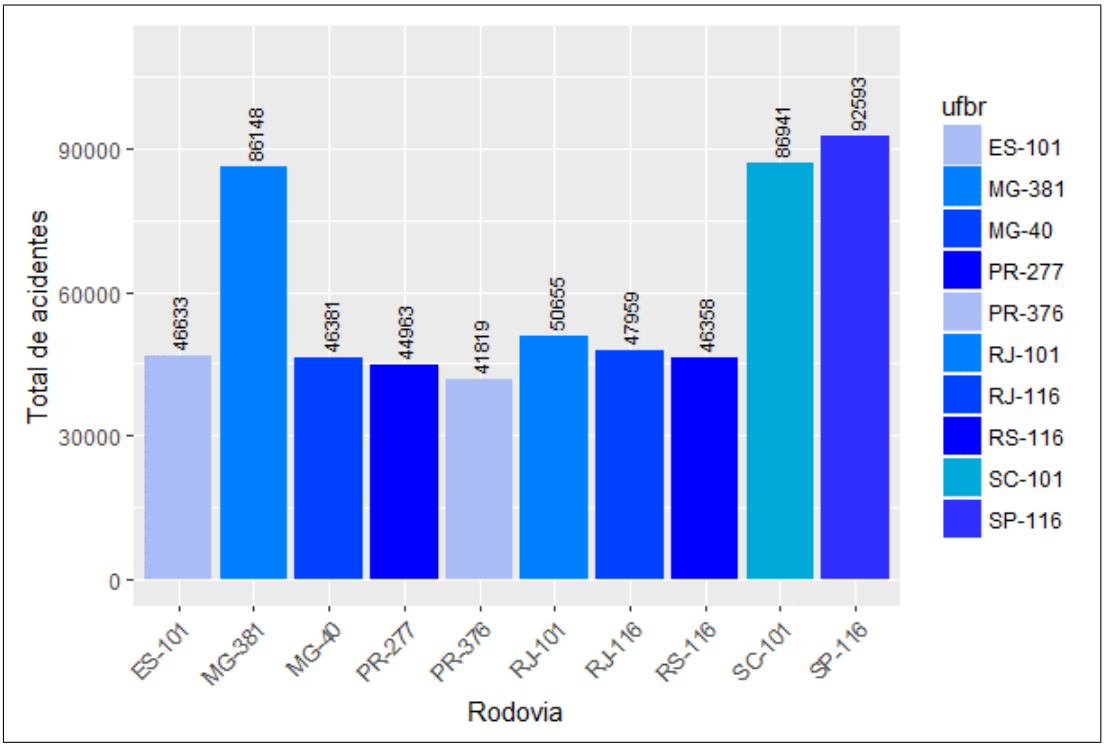

Figura 4.2: As dez rodovias brasileiras com maior número de acidentes.

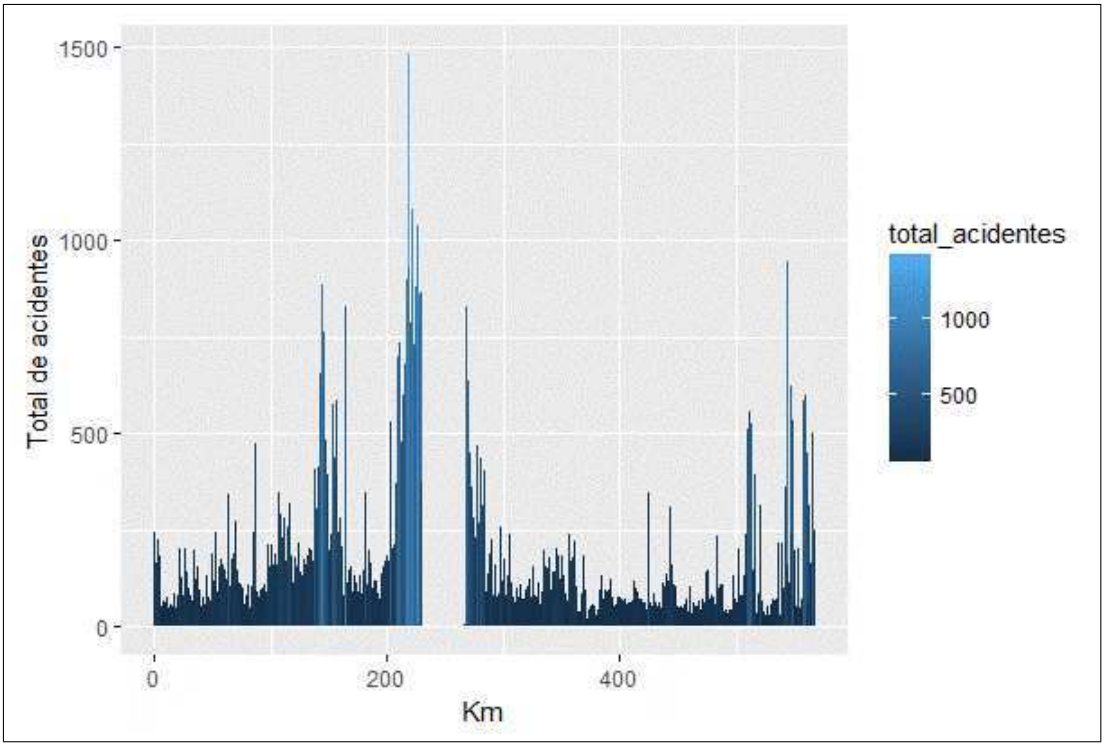

Figura 4.3: Acidentes por quilômetro na rodovia SP-116.

região, ou se falta o registro e denúncia de acidentes no local.

4.2.1 Pré-processamento

Como visto na Tabela 4.2, grande parte dos atributos dos dados de acidentes coletados pela PRF são textuais, sendo eles: a causa do acidente, o tipo do acidente, a classificação do acidente, a fase do dia no qual o mesmo ocorreu, o sentido da via, a condição meteorológica no momento do acidente, o tipo da pista, o traçado da via e o uso do solo. Esses atributos possuem um número limitado de possíveis valores, porém ainda assim possuíam inconsistência na grafia e valores similares, que poderiam ser considerados o mesmo. A exemplo temos o atributo tipo do acidente, inicialmente com 12 valores diferentes. Destes valores, a maioria apresentava diferença de grafia, tais como "queda de ocupante" e "queda de ocupante de veículo", "saída do leito" e "saída da estrada".

O pré-processamento de valores foi realizado para todos os atributos da base de dados, simplificando repetições e informações redundantes, tornando possível a categorização dos dados: cada valor de cada atributo foi substituído por um número para tornar a classificação dos dados mais eficiente.

Como o objetivo deste trabalho é propor uma abordagem para a identificação de áreas de risco de acidentes nas rodovias brasileiras, foi criado um atributo derivado chamado "gravidade". Esse atributo derivado indica se o acidente é considerado grave ou não-grave de acordo com o número de feridos leves, graves e vítimas de cada acidente, informados pelos atributos: mortos, feridos_leves, feridos_graves, feridos e ilesos. Um acidente é considerado grave caso tenha vítimas e/ou feridos graves, enquanto é considerado não-grave caso tenha feridos leves ou nenhum ferido. A próxima seção descreve os processos usados para selecionar os atributos mais importantes para este estudo.

4.2.2 Análise e Seleção das Características

Como descrito na seção anterior, os dados utilizados neste trabalho possuem muitos atributos. No caso deste estudo, essa grande quantidade de atributos e a complexidade dos seus valores tornou complexo a classificação, fazendo com que fosse necessário selecionar os atributos mais importantes e significativos ao estudo. Portanto, além do pré-processamento

e da simplificação dos valores das características descritos na seção anterior, também foi necessário avaliar quais os atributos não são relevantes ao estudo.

Inicialmente, o atributo "causa_acidente" foi retirado por descrever uma característica pós-acidente, que não necessariamente caracteriza a rodovia em si, mas um possível comportamento do motorista que causou o acidente. Possíveis valores da causa do acidente são: falta de atenção do motorista, direção sob efeito do álcool, ultrapassagem indevida, entre outras. Além da causa descrever qualquer trecho da rodovia, o atributo "tipo_acidente" é muito parecido com a causa do acidente e se adequa melhor à este estudo, possuindo como possíveis valores: colisão, capotamento e atropelamento.

Como foi criado um atributo derivado chamado "gravidade" que descreve a gravidade do acidente de acordo com a quantidade de vítimas, feridos graves, feridos leves e ilesos, os atributos pessoas, mortos, feridos leves, feridos graves, feridos, ilesos, ignorados e classificação do acidente foram retirados por já serem contemplados neste novo atributo criado.

Para ajudar na identificação dos demais atributos não relevantes ao estudo, foi utilizada a ferramenta LIME, responsável por identificar o peso e a importância que cada atributo da base de dados possui na classificação dos dados [59]. Para utilizar o LIME, é necessário escolher um modelo de classificação que servirá de base para o Explainer, classe provida pela ferramenta que permite a identificação do atributo. Para este teste, utilizamos o RandomForest por ser um classificador capaz de lidar bem com dados desbalanceados [52].

Inicialmente, o modelo foi treinado com os dados de acidente, de acordo com a classe "gravidade". Com isso, o LIME foi utilizado para explicar, para uma instância aleatória, os atributos que mais contribuíram para sua classificação, juntamente com a probabilidade daquela instância ser um verdadeiro positivo.

Dada uma instância aleatória classificada como GRAVE, de acordo com a Figura 4.4, ela possui 95% de chance de ser um verdadeiro positivo, ou seja, de ser um acidente realmente grave. A Figura 4.5 mostra quais os atributos influenciam mais para considerar a instância como grave (destacados em azul) e quais os atributos que têm mais peso para considerar essa instância como não-grave (destacados em laranja). Para essa instância analisada, vemos que o tipo do acidente tem mais influência para classificar a instância como grave, e que os atributos sentido da via, tipo da pista e traçado da via possuem mais influência para classificar a instância como não-grave.

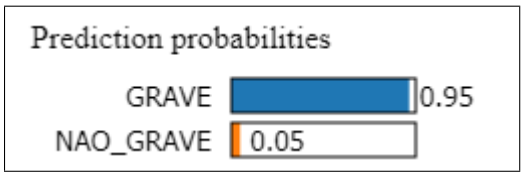

Figura 4.4: Probabilidade da instância ser grave.

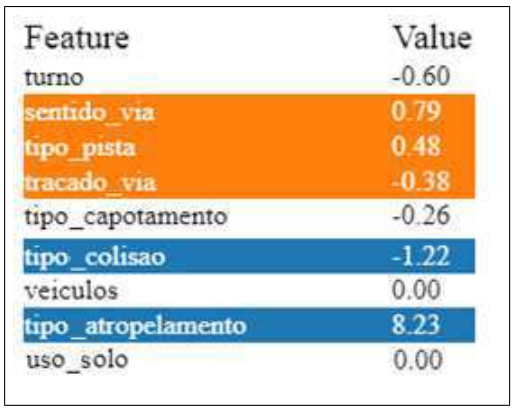

Figura 4.5: Influência dos atributos na classificação.

De forma análoga, temos um exemplo de instância classificada como não-grave. O LIME explica que a instância tem 80% de chance de ser não grave (Figura 4.6), que o atributo tipo_acidente tem influência na classificação da instância como grave (destacado em azul na Figura 4.7) e que os atributos turno, sentido via e tipo da pista ajudaram na classificação da instância como não-grave (destacados em laranja na Figura 4.7).

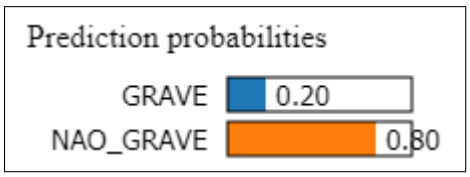

Figura 4.6: Probabilidade da instância ser grave.

Essa análise foi feita para diversas instâncias e tornou possível chegar à conclusão que os atributos número de veículos, uso do solo, data do acidente e município não contribuem na classificação dos dados. A Tabela 4.3 descreve o conjunto de atributos finais.

Com a definição dos atributos a serem usados, foi criado um novo atributo derivado chamado “frequência”. Segundo Ren et al., é difícil prever se um acidente de trânsito vai

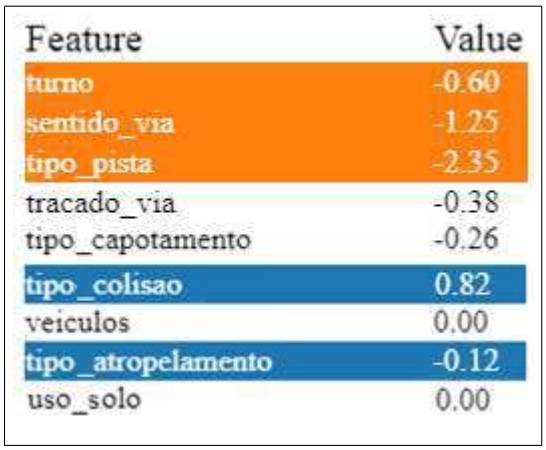

Figura 4.7: Influência dos atributos na classificação.

acontecer ou não, motivo que os levaram a criar o atributo "risco" em seu trabalho, que permitiu melhorar a classificação de risco de acidentes em Beijing [58]. Esse atributo "risco" representa a frequência de acidentes que aconteceram na mesma janela de tempo, para uma determinada quantidade de dias.

Atributo	Descrição
ufbr	Unidade da federação e o identificador da BR do acidente. Ex.: PB-230; MG-116.
km	Identificação do quilômetro onde ocorreu o acidente.
dia_semana	Dia da semana da ocorrência, representado por números. Ex.: 1 (Domingo), 2 (Segunda-feira), etc.
turno	Turno do dia no momento do acidente: manhã ou noite.
tipo_pista	Tipo da pista considerando a quantidade de faixas: simples ou múltipla.
sentido_via	Sentido da via considerando o ponto de colisão: Crescente ou decrescente.
traçado_via	Descrição do traçado da via: reta, curva ou cruzamento.

condição_meteorológica	Condição meteorológica no momento do acidente: boa (céu claro, sol, nublado) ou ruim (chuva, granizo, nevoeiro/neblina).
tipo_acidente	Identificação do tipo de acidente: colisão, capotamento ou atropelamento.
gravidade	Indicação da gravidade do acidente de acordo com as vítimas e feridos: grave ou não grave.

Tabela 4.3: Atributos dos dados de acidentes da PRF.

Adaptando o atributo risco para este estudo, propusemos a frequência do acidente, que é dada pela soma de acidentes que aconteceram em um trecho de um quilômetro de uma rodovia brasileira, dividida pela quantidade de acidentes totais da base de dados, dada por:

$$f = \frac{\sum a(r,k)}{n} \tag{4.1}$$

em que f é a frequência, $a(r,k)$ são os acidentes que aconteceram na rodovia r e no quilômetro k e n é o número total de acidentes registrados.

A base de dados final possui 1.650.400 instâncias de acidentes, dos quais 1.390.423 são considerados não-graves e 259.977 são considerados graves, o que a caracteriza como uma base de dados desbalanceada. É possível ver esse desbalanceamento na Figura 4.8, que mostra a quantidade de acidentes por estado, juntamente com sua classe. Os testes realizados neste estudo levam em consideração a base desbalanceada, com todos os dados de acidentes, e uma base de dados balanceada, gerada pela exclusão de instâncias aleatórias consideradas não-grave. A base balanceada possui 259.977 acidentes considerados graves e 260.000 acidentes não-grave (Figura 4.9).

Também foi feita a correlação entre as características finais para cada base de dados, onde podemos visualizar quais atributos podem influenciar mais na classificação dos trechos das rodovias. A Figura 4.10 mostra a correlação dos atributos para a base de dados desbalanceada, enquanto a Figura 4.11 mostra a correlação para a base de dados balanceada. Comparando as duas figuras, é possível notar que, para a base de dados desbalanceada, existe

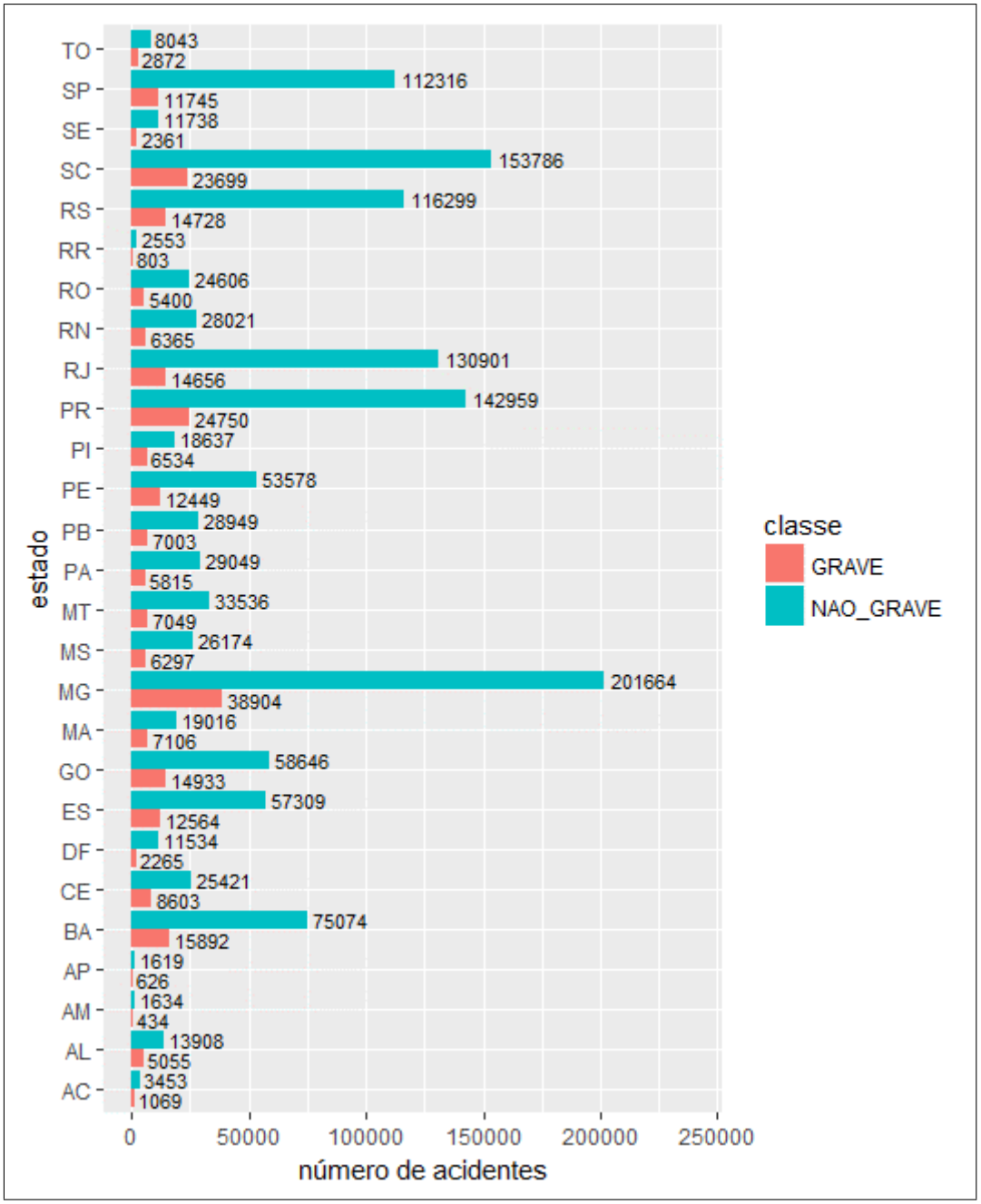

Figura 4.8: Número de acidentes por estado na base desbalanceada.

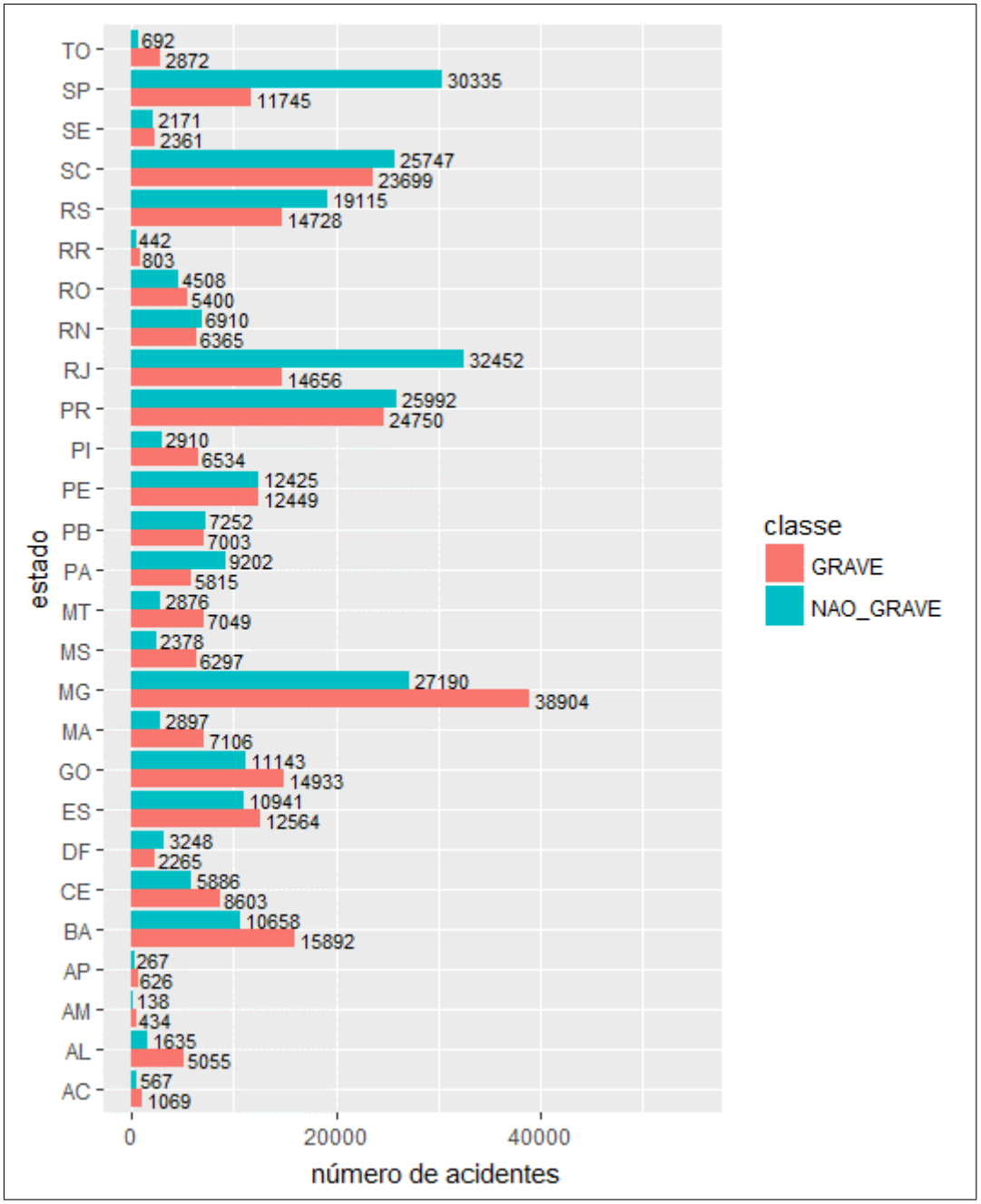

Figura 4.9: Número de acidentes por estado na base balanceada.

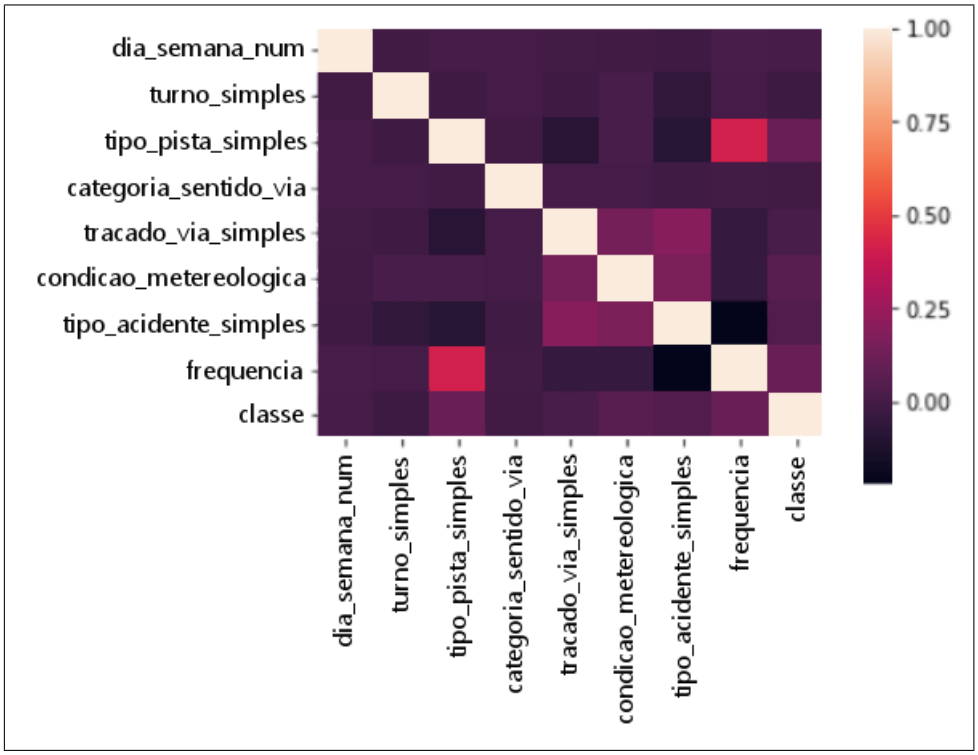

Figura 4.10: Correlação entre atributos da base de dados desbalanceada.

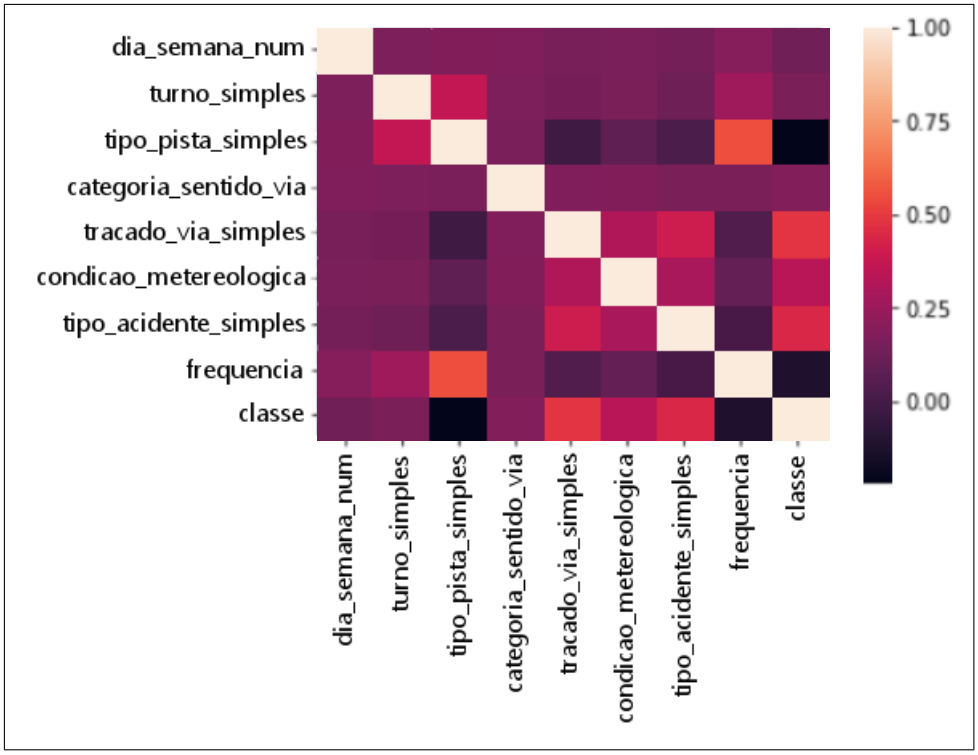

Figura 4.11: Correlação entre atributos da base de dados balanceada.

pouca correlação entre os atributos. Já na base de dados balanceada, existe correlação entre a classe do acidente com o traçado da via e com o tipo do acidente. A frequência e o tipo da pista também possuem correlação, assim como o tipo do acidente e o traçado da via e o tipo da pista com o turno no qual o acidente aconteceu.

Como forma de possibilitar a visualização de possíveis padrões nos dados de acidentes, foram criados dois gráficos de dispersão: um para a base de dados desbalanceada, mostrado na Figura 4.12, e outro para a base de dados balanceada, detalhado na Figura 4.13. Desta forma, a técnica PCA foi utilizada para tornar possível a representação dos dados de acidentes em um plano bidimensional.

A Figura 4.12 mostra a dispersão dos dados desbalanceados, onde a quantidade de instâncias consideradas "Não-Grave" é bem maior que a quantidade de instâncias consideradas "Grave". É possível ver que a maior parte dos dados considerados "Grave" estão próximos um do outro, no intervalo de 1 a 4 do eixo X (*Principal Component 1*) e -1 a 2 do eixo Y (*Principal Component 2*).

Já a Figura 4.13 mostra a dispersão dos dados balanceados. É possível ver que, com a redução da dimensionalidade e balanceamento da base, as instâncias consideradas "Não-Grave" ficaram agrupadas no intervalo de -2 a 2 do eixo X (*Principal Component 1*) e -1 a 2 do eixo Y (*Principal Component 2*), enquanto as instâncias "Grave" ficaram mais dispersas. Ainda assim, com o balanceamento da base, é possível ver que há uma distribuição linear dos dados.

4.3 Experimentos

Para a execução dos experimentos, foi necessário escolher modelos de classificação adequados ao problema apresentado neste trabalho e aos dados disponíveis. Além disso, foram feitos testes sem o uso do atributo "frequência" (Seção 4.2) e com o uso do atributo, a fim de avaliar se o mesmo é importante para melhorar a performance dos classificadores usados.

Inicialmente, foram feitos testes usando a implementação do algoritmo SVM (Support Vector Machine) disponível na biblioteca scikit-learn [53], para a base de dados balanceada e para a desbalanceada, com e sem o atributo "frequência". A fim de comparar a melhor implementação do SVM para a classificação dos dados utilizados neste estudo, foi feita uma

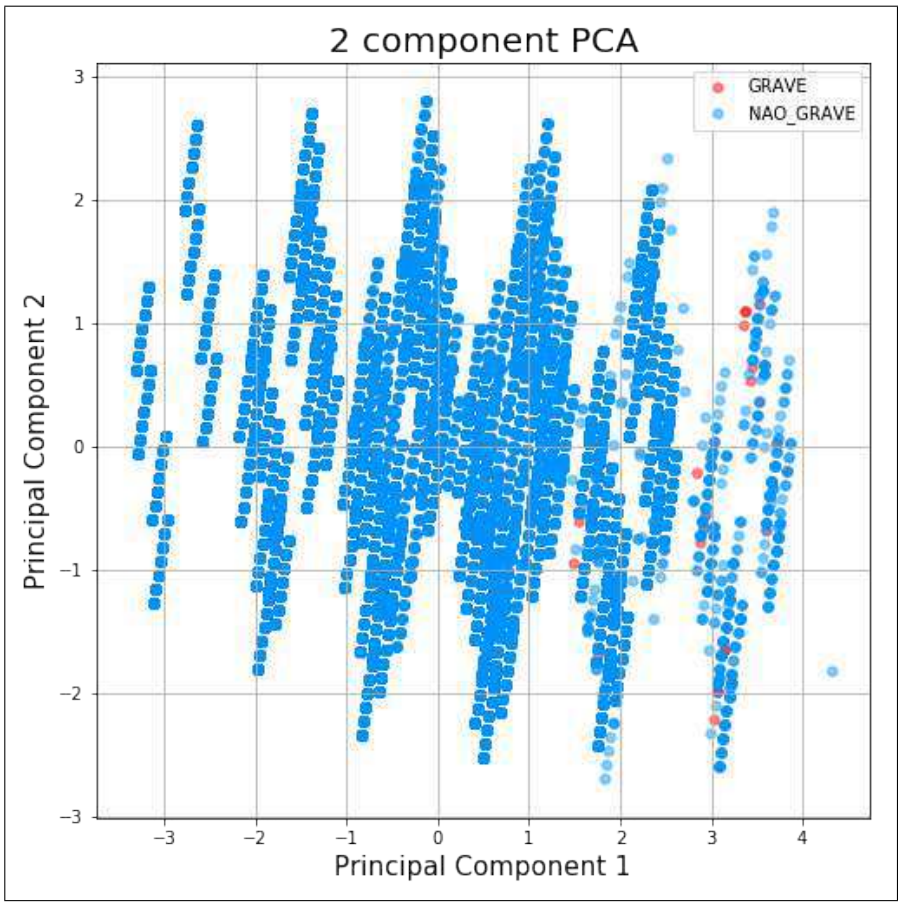

Figura 4.12: Gráfico de dispersão da base de dados desbalanceada.

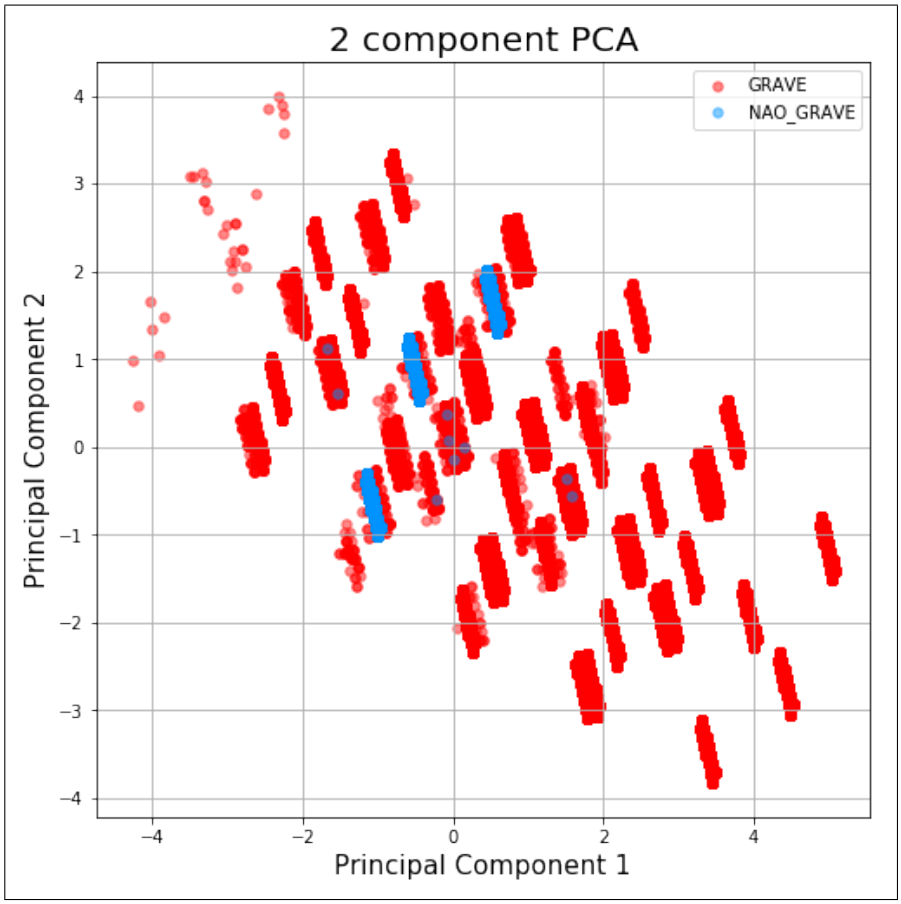

Figura 4.13: Gráfico de dispersão da base de dados balanceada.

análise comparativa entre os quatro tipos de kernels disponíveis nessa implementação do SVM: linear, RBF, sigmoid e polinomial. Como visto na Seção 2, o kernel do SVM é um conjunto de funções matemáticas que transformam os dados de entrada em uma outra dimensão na qual é possível encontrar divisões claras de margens entre as classes dos dados.

Para cada classificador SVM com um determinado kernel, foram testados diversos valores para o parâmetro C, responsável por estimar o quanto o algoritmo deve evitar classificar instâncias erroneamente, por meio de um *GridSearch*. Considerando os dados de acidentes coletados neste estudo, a melhor configuração para o SVM com kernel linear foi com o parâmetro C = 10, enquanto que para o kernel RBF o melhor valor foi C = 10. Para o SVM com kernel sigmoid, o melhor valor foi C = 1 e para o kernel polinomial foi C = 15. O código fonte dos algoritmos SVM usados pode ser visto no Código Fonte 4.1. O parâmetro de probabilidade presente na construção dos modelos faz com quê o SVM estime a probabilidade das classes.

Código Fonte 4.1: Configuração dos algoritmos SVM utilizados

```
1 model_linear = svm.SVC(kernel='linear', C=10, probability=True)
2 model_rbf = svm.SVC(kernel='rbf', C=10, gamma=1, probability=True)
3 model_sigmoid = svm.SVC(kernel='sigmoid', C=1, probability=True)
4 model_poly = svm.SVC(kernel='poly', C=15, probability=True)
```

Os experimentos usando SVM descritos acima foram realizados para a base de dados balanceada e para a base de dados desbalanceada usando a técnica de validação cruzada 10-fold. A validação cruzada permite que o conjunto de dados seja particionado em n subconjuntos mutuamente exclusivos, onde n-1 subconjuntos serão usados para treinar o modelo, e o subconjunto restante será usado para teste. Neste estudo, foi usado o método de validação cruzada k-fold, com k=10. Os resultados foram comparados e podem ser vistos na Seção 5.1.

Além do SVM, foi utilizada a ferramenta TPOT para auxiliar na busca pelo classificador mais adequado e mais eficiente para este trabalho. Para cada pacote do TPOT, foram feitos testes com a base balanceada e com a base desbalanceada, com e sem o atributo derivado "frequência", a fim de encontrar modelos eficientes para os dados de acidente. Em todos os experimentos feitos, o TPOT retornou que os melhores resultados foram obtidos usando a técnica de validação cruzada com 10-folds.

O primeiro experimento feito foi usando os pacotes Default e Sparse com a base de dados desbalanceada e sem o atributo “frequência”. Os dois pacotes retornaram como melhor classificador o XGBClassifier. O Código Fonte 4.2 mostra, para o XGBClassifier, os melhores valores para os parâmetros learning_rate, max_depth, min_child_weight, nthread e subsample.

Código Fonte 4.2: Configuração para o XGBClassifier

```
1 XGBClassifier(learning_rate=0.5, max_depth=4, min_child_weight=8, nthread
    =1, subsample=0.8)
```

O parâmetro learning_rate é usado para prevenir o overfitting do modelo, seu valor padrão é 0,3. Já o max_depth define a profundidade máxima da árvore, onde o valor padrão é 6. O grande aumento deste parâmetro pode tornar o modelo mais complexo e causar overfitting. O min_child_weight corresponde ao número mínimo de instâncias necessárias em cada nó da árvore, e quanto maior seu valor, mais conservador o algoritmos será. O nthread representa o número máximo de threads que serão usadas pelo classificador, enquanto o subsample define a porcentagem dos dados que serao usados para treinar o modelo.

O segundo experimento também foi feito com a base de dados desbalanceada, porém com a adição do atributo “frequência”, cada pacote do TPOT retornou uma configuração diferente. O pacote Default definiu que o melhor modelo a ser usado seria o BernoulliNB, enquanto o pacote Sparse retornou a técnica LogisticRegression como a melhor. O Código Fonte 4.3 e o Código Fonte 4.4 mostram as melhores configurações dos parâmetros para o BernoulliNB e LogisticRegression, respectivamente.

Código Fonte 4.3: Configuração para o BernoulliNB

```
1 BernoulliNB(alpha=0.001)
```

Código Fonte 4.4: Configuração para o LogisticRegression

```
1 LogisticRegression(C=0.1, penalty="l2")
```

O parâmetro utilizado no BernoulliNB (Código Fonte 4.3) foi o alpha, um hiperparâmetro de adição de suavização Laplace, que soma seu valor à probabilidade dada pelo classificador, impedindo que a mesma seja zero. Já para o o LogisticRegression (Código Fonte 4.4), os parâmetros utilizados foram: o parâmero C, que define o inverso da força de

regularização do algoritmo e é similar ao parâmetro C do SVM, onde valores menores especificam uma regularização mais forte; e o parâmetro penalty, que especifica a norma utilizada na penalização do algoritmo, com valor padrão l2.

O terceiro experimento foi feito usando a base de dados balanceada, sem o atributo "frequência" e cada pacote do TPOT retornou uma configuração diferente. O pacote Sparse definiu que o melhor modelo a ser usado com os dados de acidentes balanceados é a combinação do RandomForest e do BernoulliNB (Código Fonte 4.5), enquanto o pacote Default retornou a combinação do classificador LogisticRegression com o ExtraTreesClassifier (Código Fonte 4.6).

Código Fonte 4.5: Configuração para RandomForest + BernoulliNB

```
1 StackingEstimator(estimator=RandomForestClassifier(bootstrap=False,
    max_features=0.5, min_samples_leaf=13, min_samples_split=7,
    n_estimators=100)), BernoulliNB(alpha=0.01)
```

O Código Fonte 4.5 permite a visualização da melhor configuração para a combinação do RandomForest e do BernoulliNB. O parâmetro usado no BernoulliNB foi o mesmo do experimento anterior. Já os parâmetros usados para o RandomForest foram: o bootstrap, que quando falso determina que todos os dados serão usados para construir cada árvore; o max_features, que define a quantidade de atributos que serão considerados nas divisões das árvores; o min_samples_leaf, que determina o número de amostras necessárias às folhas das árvores; o min_samples_split define o número mínimo de amostras necessárias para dividir um nó interno da árvore; e o n_estimators, que define o número de árvores da floresta. O StackingEstimator é o responsável por combinar as duas técnicas.

O Código Fonte 4.6 mostra a melhor configuração para a combinação do LogisticRegression com o ExtraTreesClassifier. Os parâmetros usados no LogisticRegression foram os mesmos do experimento anterior. Os parâmetros usados no ExtraTreesClassifier foram o bootstrap, o max_features, min_samples_leaf, min_samples_split e n_estimators, todos eles possuem o mesmo propósito que os parâmetros definidos para o RandomForest.

Código Fonte 4.6: Configuração para LogisticRegression + ExtraTreesClassifier

```
1 StackingEstimator(estimator=LogisticRegression(C=0.5, penalty="l2")),
    ExtraTreesClassifier(bootstrap=False, max_features=0.8,
    min_samples_leaf=6, min_samples_split=11, n_estimators=100)
```

O quarto experimento foi feito usando a base de dados balanceada, com o atributo "frequência". Cada pacote do TPOT retornou uma configuração diferente, na qual o Default definiu que o melhor modelo é o ExtraTreesClassifier (Código Fonte 4.7), enquanto o pacote Sparse retornou o classificador XGBClassifier (Código Fonte 4.8). Os resultados da execução dos experimentos com os classificadores resultantes do TPOT são discutidos no próximo capítulo.

Código Fonte 4.7: Configuração para ExtraTreesClassifier

```
1 ExtraTreesClassifier(bootstrap=False, criterion="gini", max_features
    =0.65, min_samples_leaf=14, min_samples_split=15, n_estimators=100)
```

Código Fonte 4.8: Configuração para XGBClassifier

```
1 XGBClassifier(learning_rate=0.5, max_depth=8, min_child_weight=18,
    n_estimators=100, nthread=1, subsample=0.25)
```

O outro modelo utilizado neste estudo foi uma rede neural artificial. Conhecida também pelo seu poder de descobrir estruturas complexas em dados com altas dimensões, é muito utilizada na literatura para reconhecimento de imagem, reconhecimento de voz, processamento de linguagem natural, etc. Também é possível encontrar alguns estudos na área de predição de acidentes de trânsito que usam esse modelo [26, 70].

Para esse experimento, foi construído um modelo de rede neural artificial *feed foward* com múltiplas camadas para classificar e prever o risco de acidentes em trechos das rodovias brasileiras. Esse modelo consiste em uma camada de entrada, onde a entrada usada são os dados de acidentes tratados, que é diretamente ligada com duas camadas ocultas sequenciais e totalmente conectadas, cada uma com quatro nós. A última camada é a camada de saída, que retorna o risco de acidente de um trecho da rodovia, dado o acidente (Figura 4.14).

Cada camada oculta contou com a função de ativação *Rectified Linear Units* (RELU), que pode ser matematicamente definida como max(0,x). A camada de saída utiliza a função de ativação *Sigmoid*. O Código Fonte 4.9 mostra o código usado para a construção do modelo. Com esse modelo, foram feitos quatro experimentos com entradas diferentes: o primeiro foi feito usando a base de dados desbalanceada sem o atributo frequência; o segundo foi feito com a base de dados balanceada sem o atributo frequência; o terceiro, usou a base de dados desbalanceada com o atributo frequência; finalmente, o quarto experimento foi feito com a

base de dados balanceada com o atributo frequência. Todos os experimentos fazem uso dos hiper-parâmetros *epoch* = 5, *batch_size* = 64 e *learning_rate* = 0,1, calculados manualmente para a otimização do modelo.

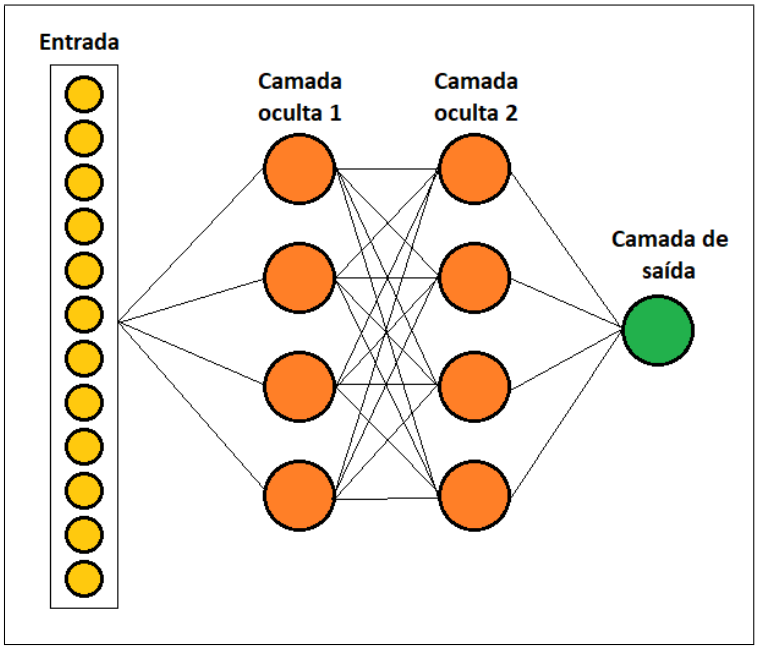

Figura 4.14: Modelo Rede Neural Artificial.

Código Fonte 4.9: Configuração da rede neural

```
1 classifier = Sequential()
2 classifier.add(Dense(4, activation='relu', kernel_initializer='
    random_normal', input_dim=24))
3 classifier.add(Dense(4, activation='relu', kernel_initializer='
    random_normal'))
4 classifier.add(Dense(1, activation='sigmoid', kernel_initializer='
    random_normal'))
```

4.3.1 Ambiente de Execução e Performance

Para as análises feitas na Seção 4.2, foram utilizados *scripts* em Python e R, executados em uma máquina com 16GB de memória RAM, processador Intel Core i7 3.4 GHz e 1 Terabyte de armazenamento em disco, com o sistema operacional Windows 10. Nesta mesma máquina, também foram executados os experimentos usando algoritmos de classificação.

Por serem mais custosos, os experimentos feitos utilizando o TPOT foram realizados utilizando o Google Colaboratory[1], ferramenta do Google destinada à pesquisa e educação de aprendizado de máquina, que disponibiliza uma máquina virtual para execução do código com 12,6GB de RAM, 320GB de armazenamento em disco, processador Intel Xeon 2.3Hz com 45MB de cache e placa de vídeo Tesla K80, 2496 CUDA cores, 12GB GDDR5 VRAM.

O banco de dados utilizado para armazenar os dados estruturados foi o PostgreSQL[2]. Para a execução dos algoritmos de classificação, foram utilizadas as implementações disponíveis no pacote scikit-learn para Python [53], utilizado no JupyterNotebook[3].

O tempo de execução dos experimentos feitos usando o SVM com a base de dados desbalanceada foi de aproximadamente 30 horas. Já usando o SVM com a base balanceada, o tempo de execução diminuiu cerca de 6 horas, resultando em um total de 24 horas de execução.

Os demais experimentos feitos utilizando a base de dados desbalanceada e os classificadores XGBClassifier, BernoulliNB e Logistic Regression levaram cerca de 5 horas para executar, enquanto a Rede Neural Artificial levou cerca de 6 horas. Nos experimentos feitos com a base de dados balanceada, o XGBClassifier, o RandomForest + BernoulliNB, o LogisticRegression + ExtraTreesClassifier e o ExtraTreesClassifier levaram cerca de 4 horas para terminar a execução, ao passo que a Rede Neural Artificial levou aproximadamente 5 horas.

Nos experimentos feitos com o TPOT e a base de dados desbalanceada, os testes feitos com o pacote Sparse levaram aproximadamente 6 horas para completar a execução, enquanto que os testes feitos com o pacote Default levaram cerca de 4 horas. Já nos experimentos feitos com o TPOT e a base de dados balanceada, os testes feitos com o pacote Sparse levaram de 3,4 a 3,7 horas para completar a execução, ao passo que os teste do pacote Default levaram entre 3,1 e 3,6 horas de execução.

[1] https://colab.research.google.com/

[2] https://www.postgresql.org/

[3] https://jupyter.org/

4.4 Considerações Finais

Neste capítulo foi mostrado em maiores detalhes a metodologia utilizada nesta pesquisa, explicitando todo o fluxo metodológico adotado. Os processos pelos quais os dados foram submetidos foram detalhados, bem como a implementação e execução dos algoritmos de classificação e o ambiente de execução utilizado para os experimentos.

No capítulo a seguir serão mostrados os resultados dos experimentos descritos nesta seção e uma análise comparativa entre os mesmos.

Capítulo 5

Resultados

Neste capítulo, são discutidos os resultados dos experimentos realizados nesta dissertação. A Seção 5.1 apresenta os resultados dos experimentos feitos com o algoritmo de classificação SVM, enquanto a Seção 5.2 apresenta os resultados do XGBClassifier. A Seção 5.3 detalha os resultados para a combinação de duas técnicas: Random Forest e BernoulliNB. Os resultados da combinação entre Logistic Regression e Extra Trees Classifier são apresentados na Seção 5.4. Na Seção 5.5 são mostrados os resultados do classificador Extra Trees Classifier, enquanto a Seção 5.6 mostra os resultados para o BernoulliNB. Os resultados do experimento feito com o Logistic Regression são apresentados na Seção 5.7 e os resultados dos experimentos usando a Rede Neural Artificial são detalhados na Seção 5.8. Em seguida, na Seção 5.9, é apresentada uma análise e comparação dos resultados obtidos. Por fim, na Seção 5.10 são apresentadas as considerações finais.

5.1 SVM

Em cada experimento feito com o SVM, foi utilizada a técnica de validação cruzada com 10-folds e a base de dados balanceada, com e sem o atributo derivado "frequência". No teste feito sem considerar o atributo "frequência", o SVM Linear obteve uma acurácia de 57,3%, com precisão de 60%, revocação de 57% e medida F de 54%. Já o SVM RBF obteve 57% de acurácia, 60% de precisão, 57% de revocação e 54% de medida F, resultado similar ao SVM Linear.

O SVM Sigmoid obteve 55,3% de acurácia, 58% de precisão, 55% de revocação e 51%

de medida F, enquanto os resultados para o SVM polinomial foram 55,3% de acurácia, 59% de precisão, 55% de revocação e 50% de medida F. A Tabela 5.1 mostra todos os resultados dos testes com o SVM.

Modelo	Acurácia	Precisão	Revocação	Medida F
SVM Linear	57,3%	60%	57%	54%
SVM RBF	57%	60%	57%	54%
SVM Sigmoid	55,3%	58%	55%	51%
SVM Polinomial	55,3%	59%	55%	50%

Tabela 5.1: Resultados do SVM sem o atributo frequência, usando a base balanceada.

Os testes feitos considerando o atributo "frequência" não obtiveram uma diferença significativa quando comparados com os resultados acima. Já os resultados dos experimentos usando o SVM com a base desbalanceada não foram bons, uma vez que o modelo desconsiderou a classe "Grave" e classificou todas as instâncias como "Não-grave". A Tabela 5.2 mostra os resultados dos testes usando o SVM com a base desbalanceada.

Modelo	Acurácia	Precisão	Revocação	Medida F
SVM Linear	66,8%	38%	66,5%	48,4%
SVM RBF	65,7%	37,2%	65,6%	47,5%
SVM Sigmoid	64,6%	35%	64,4%	45,4%
SVM Polinomial	64,2%	34,7%	64%	45%

Tabela 5.2: Resultados do SVM sem o atributo frequência, usando a base desbalanceada.

Portanto, dentre os resultados obtidos usando a base balanceada e os diferentes kernels do SVM, o melhor foi o SVM Linear que apresentou o maior valor para a medida F, para a revocação, a precisão e a acurácia.

5.2 XGBClassifier

Para o experimento feito com o TPOT usando os dados desbalanceados sem o atributo "frequência", o melhor classificador foi o XGBClassifier, executado usando a validação cru-

zada com 10-folds. A média de acurácia para esse classificador foi 84,24% , com uma precisão de 42,12%, revocação de 49,99% e medida F de 45,72%.

A Tabela 5.3 mostra a matriz de confusão resultante deste experimento, calculada por meio da média aritmética das matrizes de confusão de cada *fold*. Com isso, podemos concluir que o uso da base de dados desbalanceada não é a melhor escolha, pois existe uma quantidade muito maior de acidentes não-graves. O classificador ignorou a classe grave e classificou todas as instâncias como não-grave a fim de atingir uma acurácia e uma precisão maior.

	Não-Grave	Grave
Não-grave	138.999	43
Grave	25.997	0

Tabela 5.3: Matriz de confusão do XGBClassifier, para base de dados desbalanceada sem o atributo “frequência”.

O XGBClassifier também foi o classificador escolhido pelo pacote Sparse do TPOT como o mais adequado para a base de dados balanceada com o atributo frequência (Seção 4.2.2). A média dos resultados desse experimento para a validação cruzada 10-fold foram: 64,9% de acurácia, 65,13% de precisão, 64,9% de revocação e 64,8% de medida F. Comparando com o resultado anterior, o uso da base balanceada resolveu a não-classificação das instâncias graves, como pode ser visto na Tabela 5.4, que mostra a matriz de confusão deste novo teste. Porém, o uso do da base de dados contendo o atributo “frequência” não foi muito satisfatório pelo seu baixo percentual de acerto.

	Não-Grave	Grave
Não-grave	16.520	9.478
Grave	8.902	17.096

Tabela 5.4: Matriz de confusão do XGBClassifier, para base de dados balanceada com o atributo “frequência”.

5.3 RandomForest + BernoulliNB

O experimento feito com o TPOT usando os dados balanceados, sem o atributo "frequência", com o pacote de classificadores Sparse retornou como melhor classificador a combinação entre o RandomForest e o BernoulliNB, executado usando a validação cruzada com 10-folds. A média de acurácia para esse classificador foi 84,58% , com uma precisão de 88,14%, revocação de 84,58% e medida F de 84,06%. Comparando com o resultado do XGBClassifier descrito na seção anterior (Seção 5.3.2), é possível verificar o quanto a base balanceada influenciou positivamente no resultado do classificador, melhorando significativamente as métricas precisão, revocação e medida F. A Tabela 5.5 mostra a matriz de confusão resultante, calculada por meio da média aritmética das matrizes de confusão de cada *fold*.

	Não-Grave	Grave
Não-grave	25.689	308
Grave	5.419	20.579

Tabela 5.5: Matriz de confusão do RandomForest + BernoulliNB.

5.4 LogisticRegression + ExtraTreesClassifier

Para o experimento feito com o TPOT usando os dados balanceados e o pacote de classificadores Default, o melhor classificador foi a combinação entre o LogisticRegression e o ExtraTreesClassifier, executado usando a validação cruzada com 10-folds. A média de acurácia para esse classificador foi 84,58%, com uma precisão de 88,14%, revocação de 84,58% e medida F de 84,06%, resultados iguais a combinação dos classificadores RandomForest e o BernoulliNB descritos na Seção 5.3.3. A Tabela 5.6 mostra a matriz de confusão resultante.

	Não-Grave	Grave
Não-grave	25.689	308
Grave	5.419	20.579

Tabela 5.6: Matriz de confusão do LogisticRegression + ExtraTreesClassifier.

5.5 ExtraTreesClassifier

No experimento feito com o pacote Default do TPOT usando os dados balanceados com a adição do atributo "frequência", foi selecionado como melhor classificador o ExtraTrees-Classifier, executado com validação cruzada com 10-folds. A média da acurácia, precisão, revocação e medida F para esse classificador foi 61,2%, 61,23%, 61,2% e 61,15%, respectivamente. A Tabela 5.7 mostra a matriz de confusão resultante, calculada por meio da média aritmética das matrizes de confusão de cada *fold*.

	Não-Grave	Grave
Não-grave	14.904	11.094
Grave	9.301	16.697

Tabela 5.7: Matriz de confusão do ExtraTreesClassifier.

É possível concluir que o uso do atributo "frequência" não ajudou na melhora da performance do classificador, uma vez que os resultados foram inferiores aos resultados descritos na Seção 5.4, que também faz uso da base de dados balanceada, porém sem a presença do atributo "frequência".

5.6 BernoulliNB

O resultado do experimento feito com o pacote Default do TPOT usando os dados desbalanceados com a adição do atributo "frequência" foi o classificador o BernoulliNB. Esse classificador foi executado com validação cruzada com 10-folds, também escolhido pelo TPOT como a configuração mais adequada para esse teste. A média da acurácia, precisão, revocação e medida F para esse classificador foi 84,25%, 42,13%, 50% e 45,7%, respectivamente. A Tabela 5.8 mostra a matriz de confusão resultante. Apesar do bom resultado da métrica acurácia, vemos que com a precisão, a revocação e a medida F baixos, o resultado do experimento não foi bom. A matriz de confusão também mostra que o classificador ignorou a classe "Grave", considerando todas as instâncias como "Não-grave", não sendo a escolha ideal para o objetivo deste trabalho.

	Não-Grave	Grave
Não-grave	139.043	0
Grave	25.997	0

Tabela 5.8: Matriz de confusão do BernoulliNB.

5.7 Logistic Regression

No experimento feito com o pacote Sparse do TPOT usando os dados desbalanceados com a adição do atributo "frequência", foi selecionado como melhor técnica a Regressão Logística (Logistic Regression), executada com validação cruzada com 10-folds. A média da acurácia, precisão, revocação e medida F para esse classificador foi 84,25%, 42,13%, 50% e 45,7%, respectivamente. Da mesma forma que o experimento anterior, vemos que a precisão, revocação e medida F deste experimento foram baixos. A matriz de confusão mostrada na Tabela 5.9 também mostra que o classificador ignorou a classe "Grave" e considerou todas as instâncias como "Não-grave", não sendo a escolha ideal para o objetivo deste trabalho.

	Não-Grave	Grave
Não-grave	139.043	0
Grave	25.997	0

Tabela 5.9: Matriz de confusão do Logistic Regression.

5.8 Rede Neural Artificial

Usando uma Rede Neural Artificial, descrita na Seção 4.4, foram feitos dois experimentos: o primeiro foi feito sem considerar o atributo derivado "frequência", para a base de dados desbalanceada e para a base de dados balanceada. O segundo foi feito usando o atributo "frequência", também para as duas bases de dados: desbalanceada e balanceada.

No primeiro experimento, feito sem o uso do atributo frequência e usando a base de dados desbalanceada, a acurácia desse modelo chegou a 84%, enquanto a precisão foi de 71%, a revocação foi 84% e a medida F foi 77%. Porém, por existir uma diferença muito grande na

quantidade de dados da classe Grave e da classe Não-Grave, o modelo classificou todas as instâncias de teste como Não-Grave, como pode ser visto na matriz de confusão detalhada na Tabela 5.10.

	Não-Grave	Grave
Não-grave	417.185	0
Grave	77.935	0

Tabela 5.10: Matriz de confusão para primeiro experimento usando a Rede Neural e dados desbalanceados.

Usando a base de dados balanceada sem o atributo "frequência", a acurácia do modelo chegou a 83%, enquanto a precisão, revocação e a medida F também foram 83%. A Tabela 5.11 detalha a matriz de confusão resultante deste experimento, sendo possível perceber que o problema do experimento anterior, usando a base de dados desbalanceada, foi resolvido. Com a base de dados balanceada, o classificador conseguiu classificar instâncias como Grave e como Não-Grave.

	Não-Grave	Grave
Não-grave	59.924	18.068
Grave	9.039	68.956

Tabela 5.11: Matriz de confusão para primeiro experimento usando a Rede Neural e dados balanceados.

O segundo experimento, feito considerando o atributo "frequência"e usando a base de dados desbalanceada, resultou em uma acurácia de 84%, precisão de 71%, revocação de 84% e medida F de 77%. Da mesma forma do teste anterior usando a base de dados desbalanceada, nesse segundo experimento o modelo também desconsiderou a classe Grave, classificando tudo como Não-Grave (Tabela 5.12).

Utilizando o atributo "frequência" e a base de dados balanceada, foi possível evitar o problema de classificar todas as instâncias com a mesma classe. A Tabela 5.13 mostra a matriz de confusão para esse novo experimento, que obteve um total de 85% de acurácia, 87% de precisão, 85% de revocação e 84% de medida F.

	Não-Grave	Grave
Não-grave	417.041	0
Grave	78.079	0

Tabela 5.12: Matriz de confusão para segundo experimento usando a Rede Neural e dados desbalanceados.

	Não-Grave	Grave
Não-grave	55.495	22.815
Grave	1.264	76.413

Tabela 5.13: Matriz de confusão para segundo experimento usando a Rede Neural e dados balanceados.

Na Tabela 5.14 é possível ver os resultados de todos os experimentos feitos usando a Rede Neural Artificial. Sabemos que os testes feitos com a base desbalanceada, independente do uso do atributo "frequência" ou não, não são ideais ao objetivo deste trabalho por desconsiderar a classe "Grave" de acidente. Uma vez que queremos saber o risco que um trecho de uma rodovia têm de acidentes graves, essa é a classe mais importante na classificação.

Experimento	Acurácia	Precisão	Revocação	Medida F
RN sem frequência e base desbalanceada	84%	71%	84%	77%
RN sem frequência e base balanceada	83%	83%	83%	83%
RN com frequência e base desbalanceada	84%	71%	84%	77%
RN com frequência e base balanceada	85%	87%	85%	84%

Tabela 5.14: Comparação dos experimentos feitos com a Rede Neural Artificial.

Portanto, considerando os experimentos feitos com a base de dados balanceada, é possível ver que, diferente dos outros classificadores, a rede neural conseguiu melhores resultados com o uso do atributo "frequência", obtendo um valor superior em todas as métricas consideradas neste estudo.

5.9 Discussão

De modo geral, os resultados obtidos neste estudo foram bastante satisfatórios. Comparando com a literatura, foi possível melhorar a classificação dos dados de acidentes de acordo com sua gravidade, tornando viável o uso do classificador na identificação de trechos com risco de acidentes graves nas rodovias do Brasil.

Como foi possível concluir após a análise dos resultados, os experimentos feitos com a base de dados desbalanceada não são interessantes para o objetivo desta pesquisa. Esses experimentos obtiveram um bom nível de acurácia, mas não quer dizer que foram resultados bons, já que os valores das métricas de precisão, revocação e medida F foram baixos.

Portanto, a comparação entre os resultados considerou apenas os experimentos feitos com a base de dados balanceada, com ou sem o uso do atributo "frequência". A Tabela 5.15 mostra as métricas de acurácia, precisão, revocação e medida F obtidos nos experimentos, detalhando se o experimento foi feito com a presença ou ausência do atributo frequência.

Experimento	Frequência?	Acurácia	Precisão	Revocação	Medida F
SVM Linear	Não	57,3%	60%	57%	54%
XGBClassifier	Sim	64,9%	65,13%	64,9%	64,8%
RandomForest + BernoulliNB	**Não**	**84,58%**	**88,14%**	**84,58%**	**84,06%**
Logistic Regression + ExtraTreesClassifier	**Não**	**84,58%**	**88,14%**	**84,58%**	**84,06%**
ExtraTreesClassifier	Sim	61,2%	61,23%	61,2%	61,15%
Rede Neural Artificial	Não	83%	83%	83%	83%
Rede Neural Artificial	**Sim**	**85%**	**87%**	**85%**	**84%**

Tabela 5.15: Resultados dos experimentos.

Como pode ser visto na comparação entre resultados, SVM foi o classificador com o pior resultado para este trabalho. Os outros piores resultados foram de modelos treinados com a base de dados balanceada com a adição do atributo frequência, que indica a frequência de acidentes ocorridos por quilômetro da rodovia: XGBClassifier e Extra Trees Classifier.

Nos modelos de Rede Neural Artificial, a combinação entre Random Forest e Bernoul-

liNB e a combinação entre Logistic Regression e Extra Trees Classifier retornaram os melhores valores para as métricas avaliadas. O Random Forest + BernoulliNB e o Logistic Regression + Extra Trees Classifier obtiveram os mesmos valores para todas as métricas e ambos foram treinados com a base de dados balanceada sem a adição do atributo frequência.

Já a Rede Neural Artificial obteve dois bons resultados: com e sem o uso do atributo frequência. Ainda assim, quando comparados, o teste feito com o uso do atributo frequência obteve uma revocação 2% mais alta, uma precisão 4% maior e uma acurácia 2% maior. Isso mostra que o uso de tal atributo ajudou o classificador a melhorar a previsão dos riscos de acidentes em trechos de rodovias. Além disso, a rede neural foi a única que, com o uso de tal atributo, conseguiu melhores resultados. Todos os outros classificadores não conseguiram lidar bem com essa característica.

Comparando o uso da Rede Neural Artificial com o atributo frequência e com o Random Forest + BernoulliNB e o Logistic Regression + Extra Trees Classifier sem o uso do atributo, podemos ver que a diferença entre os resultados das métricas é pequena, onde a medida F da rede neural é apenas 0,06% mais baixa e a precisão apenas 1,14%. A acurácia e a revocação da rede neural foram melhores em 0,42%.

Considerando os estudos da literatura citados na Seção 3 que fazem uso das métricas avaliadas nesta pesquisa, a Tabela 5.16 apresenta um comparativo entre os resultados de tais estudos e os resultados obtidos nesta pesquisa, mostrando que foi possível melhorar os resultados dos classificadores.

5.10 Considerações Finais

Neste capítulo, foram discutidos em maiores detalhes os resultados dos experimentos realizados com as diferentes técnicas de aprendizado de máquina abordadas neste estudo. De modo geral, os resultados foram bastante satisfatórios, mostrando que foi possível melhorar os resultados obtidos pelos classificadores, tornando possível identificação de trechos com risco de acidentes graves nas rodovias do Brasil.

No próximo capítulo, serão apresentadas as considerações finais sobre o trabalho desenvolvido nesta pesquisa, suas contribuições e os trabalhos futuros.

Autor(es)	Algoritmos	Resultados
Nossa abordagem	Rede Neural Artificial	Acurácia: 85%; Precisão: 87%; Revocação: 85%. Medida F: 84%.
Nossa abordagem	RandomForest + BernoulliNB e Logistic Regression + ExtraTreesClassifier	Acurácia: 84,58%; Precisão: 88,14%; Revocação: 84,58%. Medida F: 84,06%.
Tiwari et al. [72]	Lazy Classifier (IBK)	Acurácia: 84,47%.
Bülbül e Kaya [9]	CART	Acurácia: 81,5%; Precisão: 81,2%; Revocação: 81%.
Tiwari et al. [73]	Decision Tree	Acurácia: 81%; Precisão: 73%; Revocação: 70,6%.
Kumar et al. [32]	Random Forest	Acurácia: 81%.
Satu et al. [64]	J48 (pruned)	Acurácia: 78,9%; Precisão: 62,6%.
Iranitalaba e Aemal Khattakb [23]	Nearest Neighbor Classification (NNC) e K-means Clustering	Acurácia: 73,95%
Tambouratzis et. al. [70]	Redes Neurais Artificiais e Árvores de Decisão	Acurácia: 70%.

Tabela 5.16: Comparação de resultados entre trabalhos.

Capítulo 6

Conclusão

Recentemente, a comunidade científica tem trabalhado no sentido de propor metodologias para a identificação de áreas ou trechos, em ruas e rodovias, que possuem risco de acidentes. Tal linha de estudo é motivada pela necessidade de soluções que ajudam na diminuição do número de acidentes, que segundo a OMS, é uma das maiores causas de morte no mundo.

Apesar da grande quantidade de estudos para identificação de áreas de risco de acidentes, a análise de dados de acidentes e a classificação de trechos depende do local onde o mesmo acontece, uma vez que cada local possui particularidades.

Estudos utilizam técnicas de mineração de dados e técnicas estatísticas para a identificação de áreas de risco de acidentes, enquanto alguns estudos utilizam técnicas de aprendizado de máquina para prever essas áreas, treinando os modelos com dados históricos. Dos estudos que englobam aprendizado de máquina para identificar áreas de risco, alguns utilizaram apenas o número de ocorrências de acidentes agrupados por local para o treinamento do modelo.

Esta pesquisa teve por objetivo classificar, considerando diversos fatores, trechos de rodovias federais brasileiras de acordo com seu risco de acidente, este podendo ser grave ou não-grave. Um trecho de rodovia classificado como grave indica que, dado uma série de fatores, este trecho é propenso a ocorrência de acidentes graves. De forma análoga, um trecho de rodovia classificado como não-grave indica que, dado uma série de fatores, o trecho não possui propensão a acidentes.

Para isso, este estudo fez uso de uma base de dados com informações de dez anos de acidentes, de 2007 a 2017, disponibilizados pela Polícia Rodoviária Federal (PRF). Estes

dados possuem diversas informações sobre acidentes que aconteceram em rodovias de todo o Brasil.

Os dados coletados foram pré-processados para a retirada de valores repetidos, e técnicas de seleção de atributos foram aplicadas com o intuito de reduzir a dimensionalidade dos dados, usando apenas características consideradas importantes para a classificação. O conjunto final de atributos dos dados consiste em informações sobre a data do acidente, o trecho da rodovia na qual o acidente ocorreu, o dia da semana, o turno, o tipo da rodovia, o sentido da pista, o traçado da via, a condição meteorológica no momento do acidente, o tipo do acidente e a gravidade. A gravidade do acidente foi calculada a partir da quantidade de casualidades e feridos, onde um acidente foi considerado grave quando houve mortos ou feridos graves, e foi considerado não-grave quando não houveram feridos ou os ferimentos dos envolvidos foram leves.

Diferentes modelos de aprendizado de máquina foram usados para classificar os dados e, ao final do estudo, comparar os resultados. Além do SVM e da Rede Neural, modelos muito utilizados na literatura, foi utilizada uma ferramenta de aprendizado de máquina automatizado chamada TPOT, com o intuito de encontrar os melhores classificadores e configurações para o dados utilizados.

Os resultados obtidos nos experimentos mostraram que algumas técnicas de aprendizado de máquina supervisionado produzem uma ótima classificação dependendo dos atributos utilizados e da base de dados. Os testes feitos com a base de dados desbalanceada foram ruins, uma vez que a classe “grave” foi ignorada e todos os trechos foram classificados como não-graves.

Já os testes com a base de dados balanceada foram feitos de duas formas: com o uso do atributo frequência e sem o uso do atributo. Para a base de dados balanceada com a adição do atributo frequência, o melhor modelo foi a rede neural, que obteve 85% de acurácia, 87% de precisão, 85% de revocação e 84% de medida F.

Para o teste feito com a base de dados balanceada sem a adição do atributo frequência, dois classificadores obtiveram ótimos resultados: a combinação dos classificadores RandomForest + BernoulliNB e a combinação dos classificadores LogisticRegression + ExtraTreesClassifier. Ambas as combinações resultaram em 84,58% de acurácia, 88,14% de precisão, 84,58% de revocação e 84,06% medida F.

6.1 Contribuições

As principais contribuições deste trabalho são:

- Implementação de modelos capazes de classificar e prever trechos de rodovias brasileiras que possuem risco de acidentes graves ou não-graves;
- Análise comparativa de diversos modelos de aprendizado de máquina supervisionado;
- Modelo de aprendizado de máquina com resultados de métricas superiores a outras soluções na literatura;
- Uso de boas técnicas de pré-processamento e seleção de características para redução da dimensionalidade dos dados;
- Uso de características de acidentes que, em sua totalidade, não são consideradas em outros estudos na literatura.

6.2 Trabalhos Futuros

Como trabalhos futuros, são sugeridos alguns tópicos para dar prosseguimento à pesquisa:

- Implementação de um aplicativo para *smartphone* que faz uso dos algoritmos de classificação estudados para avisar ao motorista a potencialidade de acontecer um acidente grave em trechos da rodovia, levando em consideração variáveis externas, coletadas em tempo real (como o dia da semana, turno, condição climática etc). Esses dados serão usados como entrada para o classificador, que retornará os trechos passíveis de acidentes graves. Com isso, o motorista será avisado dos trechos em seu percurso que possuem risco de acidentes graves;
- Utilização de outras soluções para detecção e predição de trechos de risco em rodovias, a exemplo do Deep Learning;
- Atualização da base de dados utilizada com dados referentes aos anos de 2018 e 2019, disponibilizados pela PRF após os experimentos feitos neste estudo;

- Utilizar fontes de dados de acidentes não-oficias como forma de incrementar a base de dados, a exemplo de informações coletadas pelo Twitter e Waze;
- Estimar a probabilidade de um trecho de rodovia ser grave ou não grave, onde tal probabilidade irá representar o risco de acidente do trecho; e
- Replicar estudo em um contexto menor, utilizando dados de acidentes de uma cidade.

Revisão Bibliográfica

[1] Joaquín Abellán, Griselda López, and Juan De OñA. Analysis of traffic accident severity using decision rules via decision trees. *Expert Systems with Applications*, 40(15):6047–6054, 2013.

[2] Faisal Mohammed Nafie Ali and Abdelmoneim Ali Mohamed Hamed. Usage apriori and clustering algorithms in weka tools to mining dataset of traffic accidents. *Journal of Information and Telecommunication*, 2(3):231–245, 2018.

[3] Fadi Aloul, Imran Zualkernan, Ruba Abu-Salma, Humaid Al-Ali, and May Al-Merri. ibump: Smartphone application to detect car accidents. *Computers & Electrical Engineering*, 43:66–75, 2015.

[4] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2009.

[5] Shanthi Ameratunga, Martha Hijar, and Robyn Norton. Road-traffic injuries: confronting disparities to address a global-health problem. *The Lancet*, 367(9521):1533–1540, 2006.

[6] Adithya Balaji and Alexander Allen. Benchmarking automatic machine learning frameworks. *arXiv preprint arXiv:1808.06492*, 2018.

[7] Mariana Belgiu and Lucian Drăguţ. Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*, 114:24–31, 2016.

[8] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[9] Halil İbrahim Bülbül, Tarık Kaya, and Yusuf Tulgar. Analysis for status of the road accident occurance and determination of the risk of accident by machine learning in

istanbul. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 426–430. IEEE, 2016.

[10] Kenneth Button and Werner Rothengatter. Global environmental degradation: The role of transport. *Transport, the environment and sustainable development*, pages 19–52, 1993.

[11] Li-Yen Chang and Wen-Chieh Chen. Data mining of tree-based models to analyze freeway accident frequency. *Journal of safety research*, 36(4):365–375, 2005.

[12] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

[13] Miao Chong, Ajith Abraham, and Marcin Paprzycki. Traffic accident analysis using machine learning paradigms. *Informatica*, 29(1), 2005.

[14] N. Dogru and A. Subasi. Traffic accident detection using random forest classifier. In *2018 15th Learning and Technology Conference (L T)*, pages 40–45, Feb 2018.

[15] Salatiel Ribeiro dos Santos, Clodoveu Augusto Davis Junior, and Rodrigo Smarzaro. Analyzing traffic accidents based on the integration of official and crowdsourced data. *JIDM*, 8(1):67–82, 2017.

[16] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2962–2970. Curran Associates, Inc., 2015.

[17] Z. Gao, R. Pan, R. Yu, and X. Wang. Research on automated modeling algorithm using association rules for traffic accidents. In *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 127–132, Jan 2018.

[18] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

[19] Yanyong Guo, Zhibin Li, Pan Liu, and Yao Wu. Exploring risk factors with crashes by collision type at freeway diverge areas: accounting for unobserved heterogeneity. *IEEE Access*, 7:11809–11819, 2019.

[20] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge & Data Engineering*, (9):1263–1284, 2008.

[21] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.

[22] Joel Huting, Joey Reid, Uchechukwu Nwoke, Elizabeth Bacarella, and Kim Eng Ky. Identifying factors that increase bus accident risk by using random forests and trip-level data. *Transportation Research Record*, 2539(1):149–158, 2016.

[23] Amirfarrokh Iranitalab and Aemal Khattak. Comparison of four statistical and machine learning methods for crash severity prediction. *Accident Analysis & Prevention*, 108:27–36, 2017.

[24] S. Jamal, H. Zeid, M. Malli, and E. Yaacoub. Safe driving: A mobile application for detecting traffic accidents. In *2018 IEEE Middle East and North Africa Communications Conference (MENACOMM)*, pages 1–6, April 2018.

[25] Bryan Jones, Lester Janssen, and Fred Mannering. Analysis of the frequency and duration of freeway accidents in seattle. *Accident Analysis & Prevention*, 23(4):239–255, 1991.

[26] Yong Gyu Jung and Jong Han Lim. Automobile traffic accidents prediction model using by artificial neural networks. In Geuk Lee, Daniel Howard, Dominik Ślęzak, and You Sik Hong, editors, *Convergence and Hybrid Information Technology*, pages 713–719, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[27] Matthew G Karlaftis and Andrzej P Tarko. Heterogeneity considerations in accident modeling. *Accident Analysis & Prevention*, 30(4):425–433, 1998.

[28] Anastasios Katsoukis, Lazaros Iliadis, Avrilia Konguetsof, and Basil Papadopoulos. Classification of road accidents using fuzzy techniques. In *2018 Innovations in Intelligent Systems and Applications (INISTA)*, pages 1–5. IEEE, 2018.

[29] Devashish Khulbe and Soumya Sourav. Modeling severe traffic accidents with spatial and temporal features. *ArXiv*, abs/1906.10317, 2019.

[30] David G Kleinbaum, K Dietz, M Gail, Mitchel Klein, and Mitchell Klein. *Logistic regression*. Springer, 2002.

[31] Ayberk Kocatepe, Mehmet Baran Ulak, Eren Erman Ozguven, and Mark W Horner. Who might be affected by crashes? identifying areas susceptible to crash injury risk and their major contributing factors. *Transportmetrica A: transport science*, 15(2):1278–1305, 2019.

[32] Sachin Kumar, Prayag Tiwari, and Kalitin Vladimirovich Denis. Augmenting classifiers performance through clustering: A comparative study on road accident data. *International Journal of Information Retrieval Research (IJIRR)*, 8(1):57–68, 2018.

[33] Sachin Kumar and Durga Toshniwal. Analysing road accident data using association rule mining. In *2015 International Conference on Computing, Communication and Security (ICCCS)*, pages 1–6. IEEE, 2015.

[34] Sachin Kumar and Durga Toshniwal. A data mining framework to analyze road accident data. *Journal of Big Data*, 2(1):26, 2015.

[35] Sachin Kumar and Durga Toshniwal. Analysis of hourly road accident counts using hierarchical clustering and cophenetic correlation coefficient (cpcc). *Journal of Big Data*, 3(1):13, 2016.

[36] Sachin Kumar and Durga Toshniwal. A data mining approach to characterize road accident locations. *Journal of Modern Transportation*, 24(1):62–72, 2016.

[37] Sachin Kumar and Durga Toshniwal. A novel framework to analyze road accident time series data. *Journal of Big Data*, 3(1):8, 2016.

[38] Oh Hoon Kwon, Wonjong Rhee, and Yoonjin Yoon. Application of classification algorithms for analysis of road safety risk factor dependencies. *Accident Analysis & Prevention*, 75:1–15, 2015.

[39] Stanley Lemeshow and David W Hosmer Jr. A review of goodness of fit statistics for use in the development of logistic regression models. *American journal of epidemiology*, 115(1):92–106, 1982.

[40] Zhuan Li, Xin Guo, and Jiadong Sun. Analysis and research on the temporal and spatial correlation of traffic accidents and illegal activities. In *International Conference on Cloud Computing and Security*, pages 418–428. Springer, 2018.

[41] Lei Lin, Qian Wang, and Adel W Sadek. Real-time traffic accident risk prediction based on frequent pattern tree. *arXiv preprint arXiv:1701.05691*, 2017.

[42] Chenhui Liu and Anuj Sharma. Using the multivariate spatio-temporal bayesian model to analyze traffic crashes by severity. *Analytic methods in accident research*, 17:14–31, 2018.

[43] Michael J Maher and Ian Summersgill. A comprehensive methodology for the fitting of predictive accident models. *Accident Analysis & Prevention*, 28(3):281–296, 1996.

[44] Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103, 2010.

[45] Stephen Marsland. *Machine learning: an algorithmic perspective*. Chapman and Hall/-CRC, 2014.

[46] Jindřich Matoušek and Daniel Tihelka. Using extreme gradient boosting to detect glottal closure instants in speech signal. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6515–6519. IEEE, 2019.

[47] Serafín Moral-García, Francisco Castellano, Carlos J. Mantas, Alfonso Montella, and Joaquín Abellán. Decision tree ensemble method for analyzing traffic accidents of novice drivers in urban areas. *Entropy*, 21:360, 04 2019.

[48] Kevin P Murphy et al. Naive bayes classifiers. *University of British Columbia*, 18:60, 2006.

[49] Michael A Nielsen. *Neural networks and deep learning*, volume 25. Determination press San Francisco, CA, USA:, 2015.

[50] Randal S Olson, Nathan Bartley, Ryan J Urbanowicz, and Jason H Moore. Evaluation of a tree-based pipeline optimization tool for automating data science. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, pages 485–492. ACM, 2016.

[51] World Health Organization. Global status report on road safety, 2015.

[52] Mahesh Pal. Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, 26(1):217–222, 2005.

[53] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[54] Gregory Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. *Knowledge discovery in databases*, pages 229–238, 1991.

[55] Mark Poch and Fred Mannering. Negative binomial analysis of intersection-accident frequencies. *Journal of transportation engineering*, 122(2):105–113, 1996.

[56] Biswajeet Pradhan and Maher Ibrahim Sameen. Review of traffic accident predictions with neural networks. In *Laser Scanning Systems in Highway and Safety Assessment*, pages 97–109. Springer, 2020.

[57] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 133–142. Piscataway, NJ, 2003.

[58] Honglei Ren, You Song, Jingwen Wang, Yucheng Hu, and Jinzhi Lei. A deep learning approach to the citywide traffic accident risk prediction. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3346–3351. IEEE, 2018.

[59] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM, 2016.

[60] R. Richard and S. Ray. A tale of two cities: Analyzing road accidents with big spatial data. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 3461–3470, Dec 2017.

[61] Benjamin Ryder, Bernhard Gahr, Philipp Egolf, Andre Dahlinger, and Felix Wortmann. Preventing traffic accidents with in-vehicle decision support systems-the impact of accident hotspot warnings on driver behaviour. *Decision support systems*, 99:64–74, 2017.

[62] Benjamin Ryder and Felix Wortmann. Autonomously detecting and classifying traffic accident hotspots. In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*, pages 365–370. ACM, 2017.

[63] Keitaro Sato and Wonseok Yang. Design development of the support tool to prevent secondary accidents on highway. In *International Conference on Human-Computer Interaction*, pages 381–388. Springer, 2019.

[64] Md Shahriare Satu, Sharif Ahamed, Faruk Hossain, Tania Akter, and Dewan Md Farid. Mining traffic accident data of n5 national highway in bangladesh employing decision trees. In *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, pages 722–725. IEEE, 2017.

[65] Peter T Savolainen, Fred L Mannering, Dominique Lord, and Mohammed A Quddus. The statistical analysis of highway crash-injury severities: a review and assessment of methodological alternatives. *Accident Analysis & Prevention*, 43(5):1666–1676, 2011.

[66] Lina Shan, Zikun Yang, Huan Zhang, Ruyi Shi, and Li Kuang. Predicting duration of traffic accidents based on ensemble learning. In *International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pages 252–266. Springer, 2018.

[67] D. Singh and C. K. Mohan. Deep spatio-temporal representation for detection of road accidents using stacked autoencoder. *IEEE Transactions on Intelligent Transportation Systems*, 20(3):879–887, March 2019.

[68] Tomoyuki Suzuki, Hirokatsu Kataoka, Yoshimitsu Aoki, and Yutaka Satoh. Anticipating traffic accidents with adaptive loss and large-scale incident db. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3521–3529, 2018.

[69] Ryo Takeno, Yosuke Seki, Masahiko Sano, Kenji Matsuura, Kenji Ohira, and Tetsushi Ueta. A route navigation system for reducing risk of traffic accidents. In *2016 IEEE 5th Global Conference on Consumer Electronics*, pages 1–5. IEEE, 2016.

[70] T. Tambouratzis, D. Souliou, M. Chalikias, and A. Gregoriades. Combining probabilistic neural networks and decision trees for maximally accurate and efficient accident prediction. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, July 2010.

[71] Ali Tavakoli Kashani, Afshin Shariat-Mohaymany, and Andishe Ranjbari. A data mining approach to identify key factors of traffic injury severity. *PROMET-Traffic&Transportation*, 23(1):11–17, 2011.

[72] Prayag Tiwari, Huy Dao, and Gia Nhu Nguyen. Performance evaluation of lazy, decision tree classifier and multilayer perceptron on traffic accident analysis. *Informatica*, 41(1), 2017.

[73] Prayag Tiwari, Sachin Kumar, and Denis Kalitin. Road-user specific analysis of traffic accident using data mining techniques. In *International Conference on Computational Intelligence, Communications, and Business Analytics*, pages 398–410. Springer, 2017.

[74] Esko Turunen. Using guha data mining method in analyzing road traffic accidents occurred in the years 2004–2008 in finland. *Data Science and Engineering*, 2(3):224–231, Sep 2017.

[75] Jianshi Wang and Yukio Ohsawa. Evaluating model of traffic accident rate on urban

data. In *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 181–186. IEEE, 2016.

[76] Hans Christian Augustijn Wienen, Faiza Allah Bukhsh, E Vriezekolk, and Roelf J Wieringa. Accident analysis methods and models—a systematic literature review. In *Centre for Telematics and Information Technology (CTIT)*, 2017.

[77] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[78] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.