UNIVERSIDADE FEDERAL DO ABC

ANÁLISE DE TÉCNICAS DE APRENDIZADO DE MÁQUINA NO RECONHECIMENTO DE PADRÕES EM DADOS DE ACIDENTES DE TRÂNSITO

FLAVIO ALVES DA SILVA

Santo André
2020

FLAVIO ALVES DA SILVA

ANÁLISE DE TÉCNICAS DE APRENDIZADO DE MÁQUINA NO RECONHECIMENTO DE PADRÕES EM DADOS DE ACIDENTES DE TRÂNSITO

Trabalho de graduação apresentado ao curso de Engenharia de Informação do Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas da Universidade Federal do ABC, como requisito para obtenção do título de Bacharel em Engenharia de Informação.

Orientador:

Prof. Dr. Luneque Del Rio de Souza e Silva Junior.

Santo André

2020

FLAVIO ALVES DA SILVA

ANÁLISE DE TÉCNICAS DE APRENDIZADO DE MÁQUINA NO RECONHECIMENTO DE PADRÕES EM DADOS DE ACIDENTES DE TRÂNSITO

Trabalho de graduação apresentado ao curso de Engenharia de Informação do Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas da Universidade Federal do ABC, como requisito para obtenção do título de Bacharel em Engenharia de Informação.

Santo André - SP, 03 de dezembro de 2020.

Banca Examinadora

Prof. Dr. Luneque Del Rio de Souza e Silva Junior
Orientador - UNIVERSIDADE FEDERAL DO ABC

Prof. Dr. André Kazuo Takahata
UNIVERSIDADE FEDERAL DO ABC

Prof. Dr. Ricardo Suyama
UNIVERSIDADE FEDERAL DO ABC

AGRADECIMENTOS

Agradeço especialmente à minha família e amigos pelo apoio durante toda a graduação, não medindo esforços para me auxiliar no que fosse necessário.

Ao meu orientador, Professor Doutor Luneque Del Rio de Souza e Silva Junior, que me direcionou durante o desenvolvimento deste estudo.

Aos membros da banca examinadora, Professor Doutor André Kazuo Takahata e Professor Doutor Ricardo Suyama, pela disposição em avaliar este trabalho.

À Universidade Federal do ABC, pelo seu programa de excelência, fornecendo aos seus alunos uma formação de altíssimo nível.

RESUMO

Acidentes de trânsito são hoje a oitava maior causa de mortes em todo o mundo, responsáveis por mais de 1,35 milhões de óbitos e 50 milhões de pessoas feridas ou incapacitadas a cada ano. Este estudo se propôs a utilizar técnicas de aprendizado de máquina para a identificação dos fatores mais relevantes na determinação da gravidade de lesões decorrentes de acidentes de trânsito, além da comparação dos resultados obtidos pelos algoritmos Floresta Aleatória, AdaBoost e Árvores de Decisão. Para este propósito foram utilizados dados de acidentes de trânsito ocorridos nos Estados Unidos no ano de 2015, disponibilizados pela Administração Nacional de Segurança de Tráfego Rodoviário (NHTSA). A seleção de atributos utilizada neste trabalho reduziu em 79% a quantidade de campos e indicou como mais relevantes os atributos *HOSPITAL*, *DEFORMED*, *TOWED*, *P_CRASH2* e *NUMOCCS*. Os resultados experimentais mostraram que o melhor desempenho foi obtido pelo modelo Floresta Aleatória treinado a partir da base completa do problema, com macro média *F-Measure* de 0,6719. Os modelos Floresta Aleatória e AdaBoost construídos com a base reduzida pelo processo de seleção de atributos obtiveram resultados ligeiramente inferiores, com macro média *F-Measure* de 0,6615 e 0,6499, respectivamente. O modelo Árvores de Decisão apresentou o menor resultado, com macro média *F-Measure* de 0,5900, evidenciando que a combinação de classificadores realizada pelos modelos *ensemble* é capaz de obter resultados significativamente melhores do que quando utilizados classificadores individuais. Destes modelos, o Floresta Aleatória obteve redução de 36% no tempo de treinamento em comparação ao observado para o modelo treinado com a base completa, que necessitou de 3 minutos e 54 segundos. O AdaBoost realizou este processo em 33 minutos e 37 segundos, o que representa um aumento de 862%. O desempenho dos modelos avaliados neste estudo revelou ainda uma possível indissociabilidade entre as classes Possivelmente Lesionado, Suspeita de Lesão Leve e Suspeita de Lesão Séria, tendo como consequência a redução da macro média *F-Measure* obtida para todos modelos utilizados.

Palavras-chave: Acidentes de Trânsito. Aprendizado de Máquina. Gravidade das Lesões.

ABSTRACT

Currently road traffic accidents are the eighth leading cause of death globally, they claim more than 1.35 million lives and 50 million people injured or disabled each year. This study aims to use machine learning techniques to identify the most relevant factors for determining the severity of injuries caused by road traffic accidents. In addition, the results obtained from the Random Forest, AdaBoost and Decision Trees algorithms will be compared. For this purpose, data from traffic accidents occurred in the United States in 2015, made available by the National Highway Traffic Safety Administration were used. The feature selection used in this paper reduced in 79% the number of fields and indicated as most relevant the features *HOSPITAL*, *DEFORMED*, *TOWED*, *P_CRASH2* and *NUMOCCS*. The experimental results displayed the best performance was achieved by the Random Forest model trained from the complete problem data, with macro-averaged F-Measure of 0.6719. The Random Forest and AdaBoost models built with a reduced data by the feature selection process obtained slightly minor results, with macro-averaged F-Measure of 0.6615 and 0.6499, respectively. The Decision Trees model showed the lowest result, with macro-averaged F-Measure of 0.5900, showing that the combination of classifiers performed by ensemble models is capable of obtaining significantly better results than when using individual classifiers. Among these models, Random Forest obtained a 36% reduction in training time compared to that observed for the model trained from the complete data, which required 3 minutes and 54 seconds. AdaBoost performed this process in 33 minutes and 37 seconds, which represents an 862% increase. The evaluated models' performance in this study also revealed a possible inseparability between the classes Possible Injury, Suspected Minor Injury and Suspected Serious Injury, resulting in a reduction in the macro-averaged F-Measure for all models used.

Keywords: Road Traffic Accidents. Machine Learning. Injury Severity.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AdaBoost	*Adaptive Boosting*
AUC	*Area Under the Curve*
CFS	*Correlation Based Feature Selection*
CR-T	*Classification and Regression Trees*
CS-CRT	*Cost-Sensitive Classification and Regression Tree*
CS-MC4	*Cost- Sensitive Classification using M-Estimate*
CSV	*Comma Separated Values*
FCBF	*Fast Correlation Based Filter*
FN	*False Negative*
FP	*False Positive*
FS	*Feature Selection*
ID3	*Iterative Dichotomiser* 3
KDD	*Knowledge Discovery from Data*
KNN	*K-Nearest Neighbors*
MIFS	*Mutual Information Feature Selector*
ML	*Machine Learning*
MODTree	*Multi valued Oblivious Decision Tree Filtering*
NHTSA	*National Highway Traffic Safety Administration*
ROC	*Receiver Operating Characteristic*
SMOTE	*Synthetic Minority Oversampling TEchnique*
SVM	*Support Vector Machine*
TN	*True Negative*
TP	*True Positive*
VIN	*Vehicle Identification Number*

SUMÁRIO

1 INTRODUÇÃO

1.1 O PROBLEMA DE PESQUISA

Atualmente mais de 1,35 milhões de pessoas morrem anualmente em decorrência de acidentes de trânsito no mundo todo. Esta é a oitava maior causa de morte entre pessoas de todas as idades e a principal em pessoas com idade entre 5 e 29 anos. Acidentes de trânsito também são responsáveis por cerca de 50 milhões de pessoas feridas ou incapacitadas todos os anos (ORGANIZAÇÃO MUNDIAL DA SAÚDE, 2018).

Os acidentes de trânsito têm potencial de gerar diversas consequências, tanto sociais quanto econômicas, como perdas de vidas e inestimável sofrimento para suas famílias, alterações de rotina devido à lesões permanentes, gastos com resgate e tratamento para as vítimas, além da perda de produtividade. A grandeza destes impactos torna necessária a criação de medidas para redução destes números, bem como da gravidade das lesões por eles ocasionadas, como o endurecimento das leis de trânsito, criação de agendas para as fabricantes definindo a obrigatoriedade da inclusão de itens de segurança em novos veículos, além da criação de uma base de dados com informações das ocorrências para monitoramento e definição de estratégias.

Diversos fatores podem estar ligados à gravidade das lesões ocasionadas por acidentes de trânsito, como tempo entre o acidente e resgate, disponibilidade de *airbags*, uso de cinto de segurança, entre outros. A investigação da contribuição destes fatores na determinação da gravidade das lesões pode revelar a existência de padrões, por este motivo a aplicação de técnicas de aprendizado de máquina representa uma promissora forma de analisar os dados e fornecer informações adicionais para a definição de ações efetivas para a redução da gravidade das lesões causadas por acidentes de trânsito.

1.2 OBJETIVO DA PESQUISA

1.2.1 Objetivo Geral

Identificar os fatores com maior contribuição na determinação da gravidade das lesões das vítimas de acidentes de trânsito e realizar a comparação dos resultados obtidos pelos diferentes modelos utilizados, buscando a melhor opção para o problema estudado.

1.2.2 Objetivos Específicos

Os objetivos específicos que conduzirão o desenvolvimento deste trabalho são:

- Realizar a análise e aplicação de técnicas para identificação de atributos relevantes (Seleção de Atributos);
- Criar modelos preditivos para classificação da gravidade das lesões de vítimas de acidentes de trânsito;
- Comparar o desempenho dos modelos desenvolvidos.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo compila os principais conceitos e técnicas relacionadas ao aprendizado de máquina, necessários e suficientes para a compreensão deste estudo.

2.1 APRENDIZADO DE MÁQUINA

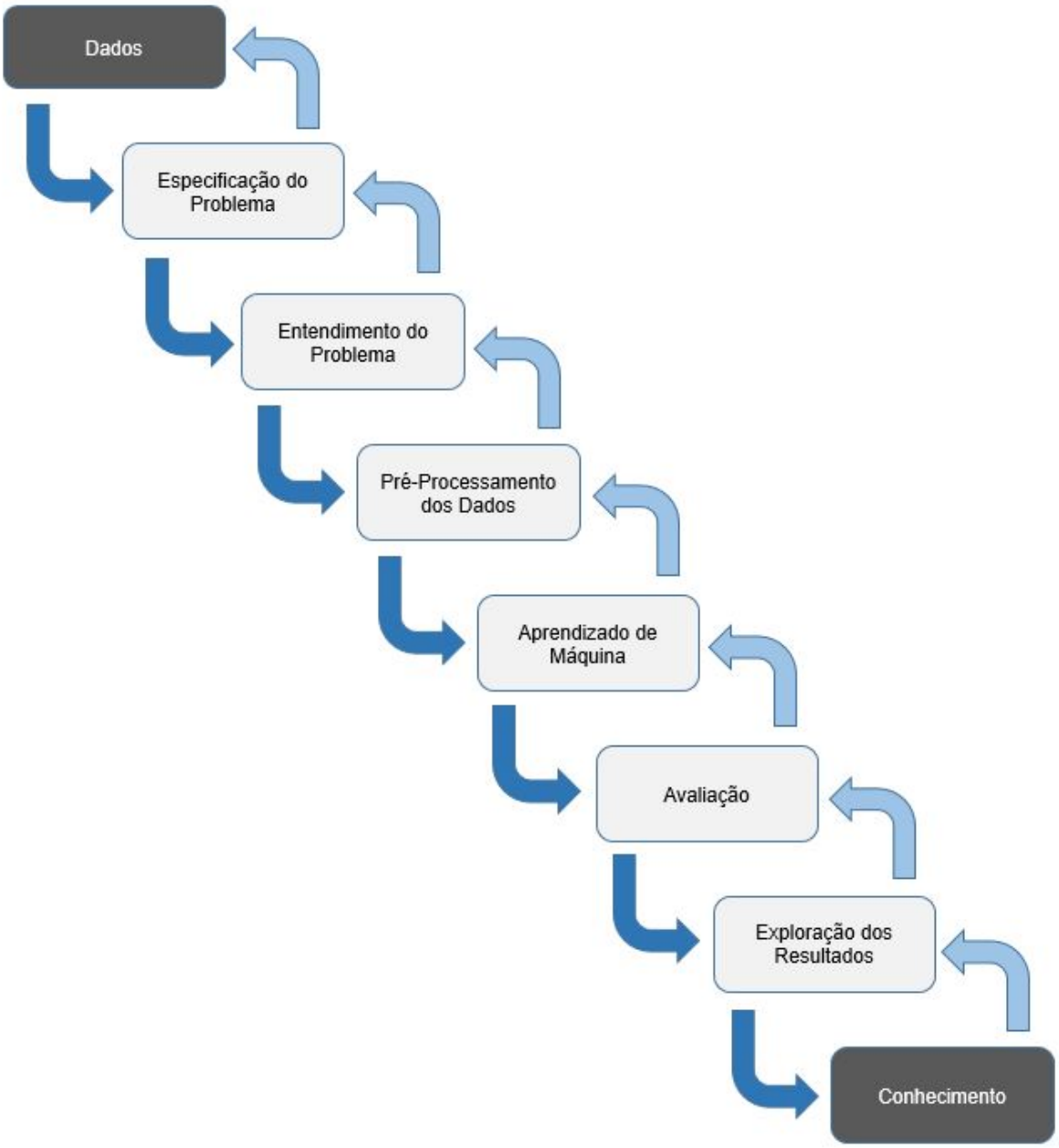
O termo aprendizado de máquina refere-se à programação de computadores para que eles sejam capazes de “aprender” a partir de dados que lhe são fornecidos, através da detecção da existência de padrões (SHALEV-SHWARTZ e BEN-DAVID, 2014). Aprendizado de máquina é um dos ramos da inteligência artificial (AI - *Artificial Intelligence*), ciência originada a partir dos trabalhos de Warren McCulloch e Walter Pitts em 1943, que propuseram um modelo de neurônios arficiais que podem estar ativos ou inativos, sendo ativados a partir de um estímulo realizado por uma quantidade suficiente de neurônios vizinhos. Foi apresentado que qualquer função computável poderia ser computada por uma rede de neurônios conectados, indicando que uma rede deste tipo seria capaz de “aprender” (RUSSEL e NORVIG, 1995).

O processo de extração de conhecimento a partir de dados, KDD (*Knowledge Discovery from Data*), contempla todas as etapas da investigação de um fenômeno, desde a formulação do problema a ser investigado e aquisição de dados até a análise e interpretação dos resultados (GARCÍA et al., 2016), conforme exibido na Figura 1. No KDD há grande interação entre as etapas do processo, pois a dinâmica permite avançar e regredir buscando refinar os resultados. Dessas etapas, algumas necessitam de análise humana, como a especificação e entendimento do problema, e exploração dos resultados. Por outro lado, há etapas que podem ser desempenhadas sem nenhuma ou com pouca intervenção humana, como o pré-processamento dos dados, aprendizado de máquina e avaliação dos resultados.

A capacidade de identificar padrões é uma das principais características da inteligência humana, e portanto um dos grandes motivadores das ideias precursoras de aprendizado de máquina, logo a comparação com o processo de aprendizado humano é inevitável. Sobre o aprendizado, entende-se como um “[...] processo de conversão de experiência em conhecimento” (SHALEV-SHWARTZ e BEN-DAVID, 2014, p.19, tradução própria).

Para os algoritmos de aprendizado de máquina, a experiência é simulada por uma representação do conhecimento, os dados do problema a ser estudado. Utilizando-

Figura 1 – Etapas do processo de extração de conhecimento - KDD



Fonte: Adaptado de García et al (2016, p. 3)

se estes dados, realiza-se o treinamento do modelo de aprendizado de máquina, que consiste no ajuste de parâmetros de funções matemáticas para que seja possível identificar quais parâmetros permitem obter, a partir dos dados de entrada, um resultado útil sobre determinado tópico.

Na prática, espera-se que os dados de treinamento representem apenas uma pequena fração das possíveis entradas para o problema em questão. Após o processo de treinamento o modelo deve ser capaz de utilizar o conhecimento adquirido para categorizar dados que não estavam presentes nos dados de treinamento. À esta capacidade dá-se o nome de generalização e é um dos principais objetivos da identificação de padrões em dados (BISHOP, 2006).

Um exemplo da utilização da generalização por computadores pode ser visto em alguns jogos digitais. No início, a dificuldade para ganhar da interligência artificial é pequena, mas quanto mais jogamos, maior a dificuldade para derrotar a máquina, até que não consigamos ganhar novamente. Isso ocorre porque a máquina foi capaz de aprender as estratégias do jogo a partir das jogadas. Uma vez adquirido este conhecimento, a máquina é capaz de utilizar contra novos oponentes e também aprender novas estratégias (MARSLAND, 2014).

Existem diferentes tipos de algoritmos de aprendizado de máquina, cada um com suas particularidades e aplicações específicas, podendo fornecer diferentes tipos de respostas para um dado problema. No aprendizado não-supervisionado os algoritmos são capazes de identificar similaridades entre os dados de entrada e separá-los em grupos de afinidades. Por outro lado, no aprendizado supervisionado é necessário informar as respostas corretas do problema para que o algoritmo identifique as regras que transformam os dados de entrada nas respostas fornecidas. A partir deste conhecimento adquirido o algoritmo se torna capaz de produzir novas respostas a partir de novos dados de entrada fornecidos (generalização). Este é o tipo mais comum de aprendizado de máquina (MARSLAND, 2014) e será abordado em mais detalhes a seguir.

O aprendizado supervisionado pode ser dividido em dois tipos, classificação e regressão. Em ambos os tipos o objetivo é criar um mapeamento dos dados de entrada às saídas, ou respostas do problema. A diferença básica entre os dois tipos de algoritmos está no tipo de dado da saída. De acordo com Murphy (2012), quando a saída é categórica ou nominal se trata de um problema de classificação, e quando é um dado numérico contínuo se trata de um problema de regressão.

2.2 PRÉ-PROCESSAMENTO

Atualmente grandes quantidades de dados são geradas a todo instante por diversas fontes como sites, redes sociais, aplicativos dos mais variados segmentos, sistemas de monitoramento, eletrodomésticos conectados à rede, entre muitos outros. O crescimento exponencial na geração de dados pôde ser alcançado graças à evolução das tecnologias de armazenamento e de conexão, e impôs maior dificuldade para entendimento dos dados e obtenção de conhecimento a partir deles, uma vez que os dados não são mais passíveis de análise humana nem manual devido à sua quantidade, requerindo sistemas de processamento de alta performance (GARCÍA et al., 2016).

O processo de KDD não depende somente da performance dos métodos utilizados, mas também em grande parte da qualidade dos dados analisados. Nos problemas reais é muito comum encontrar dados com características que influenciam negativamente na qualidade do conhecimento gerado, como ruído, valores faltantes, inconsistentes e supérfluos.

O pré-processamento possibilita analisar e tratar estes aspectos indesejados presentes nos dados combinando técnicas de preparação e redução de dados. De acordo com García et al, "O pré-processamento de dados é capaz de adaptar os dados aos requerimentos impostos por cada algoritmo de aprendizado de máquina, permitindo processar dados que seriam inviáveis de outra forma."(GARCÍA et al, 2016, p. 3, tradução própria). Devido à alta necessidade de processamento computacional, esta etapa consome uma vasta quantidade de horas, sendo responsável por grande parte do tempo necessário no processo de geração de conhecimento.

Portanto, o pré-processamento de dados é um estágio desafiador e essencial para aumentar a qualidade dos dados, fornecendo uma base que possa ser considerada correta e útil para a aplicação de algoritmos de aprendizado de máquina, possibilitando a extração de conhecimento de alta qualidade.

Nesta seção serão abordadas as principais técnicas de pré-processamento utilizadas em estudos de aprendizado de máquina.

2.2.1 Limpeza de Dados

A limpeza de dados é usualmente empregada na eliminação de ruídos e tratamento de valores faltantes. De acordo com Abdallah et al, “a limpeza de dados é definida como o processo de detecção e correção (ou remoção) de dados corrompidos ou imprecisos de um banco de dados.”(ABDALLAH et al, 2017, p. 4, tradução própria).

A aplicação das técnicas de aprendizado de máquina assume que o conjunto de dados utilizado representa fidedignamente o problema analisado, considerando que não há distorções presentes nos dados. Porém, em problemas do mundo real, a aquisição de dados é sempre um processo sujeito à falhas, fazendo com que o aparecimento de valores imprecisos ou equivocados seja um problema comum em diversos conjuntos de dados (García et al, 2016).

Outro problema frequentemente identificado e altamente prejudicial à qualidade do conhecimento extraído é a ausência de dados. A utilização de registros com dados faltantes em análises de dados não é recomendada pois representam um grande risco para o resultado do estudo, dado que o mal gerenciamento deste problema pode conduzir à extração de conhecimento de baixa qualidade, além de direcionar para conclusões equivocadas (GARCÍA et al., 2016).

Este problema pode ser enfrentado de diversas formas, uma delas se dá pela eliminação dos registros com dados ausentes, porém apesar da simplicidade e facilidade de aplicação este método raramente é utilizado, pois com a exclusão do registro inteiro perdem-se diversos dados que podem ser úteis, podendo inclusive causar a introdução de vieses, o que influencia no resultado da análise.

As melhores práticas para a correção deste problema surgem de métodos estatísticos, cuja aplicação é capaz de modelar uma função de probabilidade e, através da análise de máxima verossimilhança, preencher os valores ausentes.

2.2.2 Transformação de Dados

A transformação de dados é uma técnica muito utilizada no pré-processamento de dados, através dela é possível criar novos atributos a partir da aplicação de fórmulas matemáticas a uma combinação de atributos originais (GARCÍA et al., 2014).

Um dos métodos mais simples de transformação, a transformação linear, se baseia em transformações algébricas como soma, média, translação, entre outras, e pode ser descrito pela Equação 1.

$$Z = r_1 B_1 + r_2 B_2 + ... + r_M B_M \tag{1}$$

Nesta equação, Z é um novo atributo derivado da combinação linear dos atributos B.

Outro tipo de transformação utilizada é a transformação quadrática, que aplica

a transformação dada pela Equação 2.

$$Z = r_{1,1}B_1^2 + r_{1,2}B_1B_2 + ... + r_{m-1,m}B_{m-1}B_m + r_{m,m}B_m^2 \quad (2)$$

Além das transformações acima apresentadas, podemos citar também as transformações por aproximação polinomial e não polinomial, as quais não serão abordadas neste trabalho.

2.2.3 Balanceamento de Classes

Diversas aplicações dos algoritmos de aprendizado de máquina lidam com problemas que possuem uma diferença significativa entre a quantidade de registros para cada uma das classes, situação conhecida como o problema do desbalanceamento de classes (GARCÍA et al., 2016). De acordo com Chawla et al “Um conjunto de dados é desbalanceado se as classes não são representadas de maneira aproximadamente igual” (CHAWLA et al., 2002, p. 321, tradução própria).

O principal problema da utilização de dados desbalanceados está no enviesamento do classificador para a classe majoritária, ocasionando altas taxas de erro de classificação para a classe minoritária. Em problemas deste tipo, o custo de um erro de classificação da classe minoritária pode ser muito maior do que para a classe majoritária (CHAWLA et al., 2002), por isso a acurácia de predição, métrica comumente utilizada para avaliação da performance de algoritmos de aprendizado de máquina, não é apropriada quando os dados estão desbalanceados.

O enviesamento causado pelo desbalanceamento dos dados pode ser mitigado através de técnicas de reamostragem. Basicamente há duas opções, *undersample* e *oversample*.

O *undersample* consiste em criar um subconjunto de dados através da eliminação de registros pertencentes à classe majoritária. Apesar da vantagem de utilizar apenas dados originais do problema, esta técnica tem como grande desvantagem a necessidade de realizar o descarte de dados, podendo perder informações importantes (GARCÍA et al., 2016).

A segunda opção, *oversample*, cria um superconjunto de dados a partir da replicação de registros pertencentes à classe minoritária. A principal desvantagem desta técnica é o alto risco de indução ao sobreajuste (GARCÍA et al., 2016).

Além destas duas técnicas há opções mais sofisticadas como o SMOTE

(*Synthetic Minority Oversampling TEchnique* - Técnica de Super Amostragem Sintética da Minoria). Esta técnica realiza a interpolação de diversos pontos da classe minoritária para a criação de novos pontos, também pertencentes à classe minoritária (GARCÍA et al., 2016). Os novos pontos são gerados a partir do cálculo da diferença entre a amostra e seus vizinhos mais próximos, o resultado é então multiplicado por um valor aleatório entre 0 e 1 (CHAWLA et al., 2002).

2.2.4 Redução de Dimensionalidade

O problema de alta dimensionalidade de dados pode ser resolvido através da utilização de técnicas de redução de dimensionalidade como a Seleção de Atributos (FS - *Feature Selection*).

A utilização da FS permite obter um conjunto de dados reduzido, o que pode ser útil para diversos propósitos, como aumentar a velocidade e poder preditivo, melhorar a visualização dos dados e remover ruído (GARCÍA et al., 2014). García et al define a FS como “um processo que escolhe um subconjunto ótimo de atributos de acordo com um certo critério” (GARCÍA et al, 2014, p. 163, tradução própria).

De acordo com Venkatesh e Anuradha,“Na FS um subconjunto de variáveis é selecionado a partir do conjunto original de variáveis baseado na redundância e relevância dos atributos.” (VENKATESH e ANURADHA, 2019, p.3, tradução própria). Neste novo subconjunto de dados estão presentes apenas variáveis que possuam relação estatística com as demais variáveis.

Chandrashekar e Sahin definem que “o foco da FS é selecionar um subconjunto de atributos a partir da entrada que possa eficientemente descrever os dados da entrada reduzindo o ruído e os dados irrelevantes e fornecer bons resultados de predição.” (CHANDRASHEKAR e SAHIN, 2014, p.16, tradução própria)

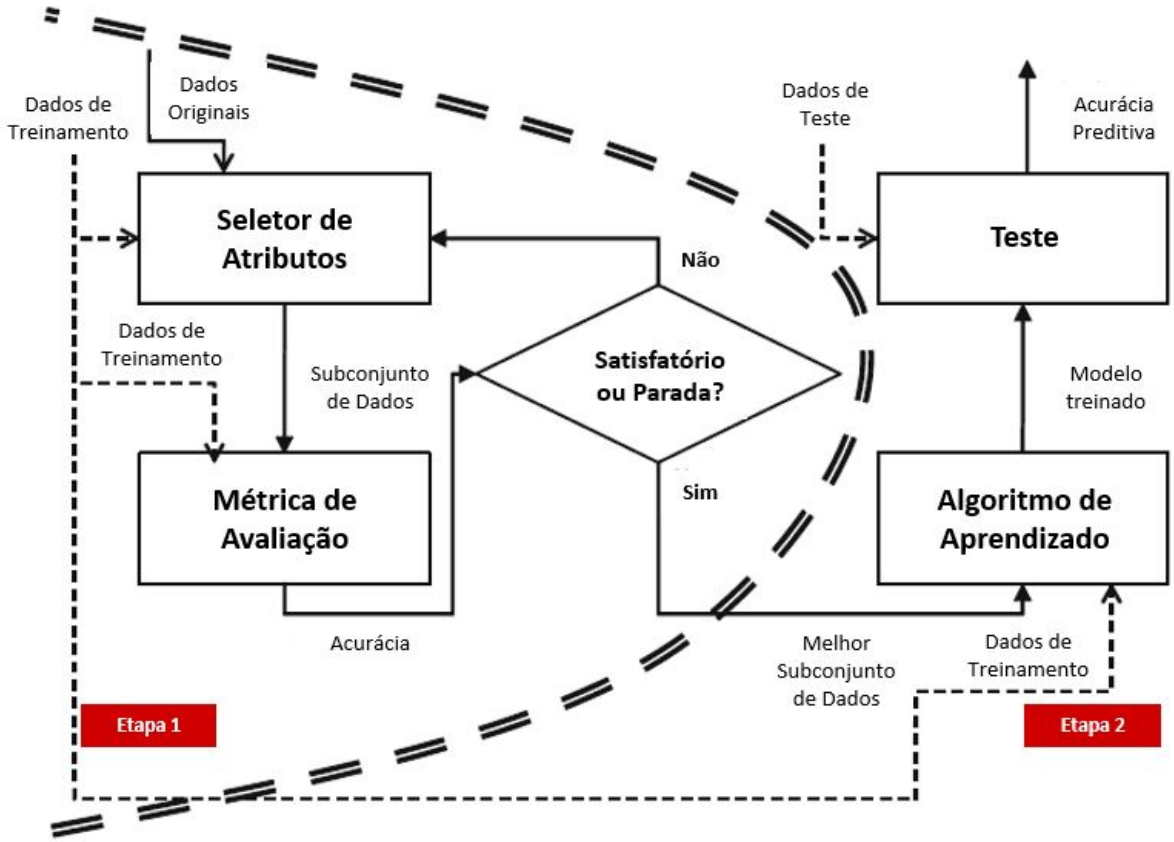
As técnicas de FS podem ser classificadas em três tipos: Filtros, *Wrappers* e *Embeddeds*.

2.2.4.1 Filtro

Os filtros realizam a seleção de atributos a partir da construção de *ranking* utilizando técnicas estatísticas para determinação do *score* de cada um dos atributos. A utilização deste método em aplicações práticas têm obtido bons resultados, mesmo se tratando de um modelo mais simples, o que requer menor poder computacional.

Basicamente estes métodos visam avaliar a relação de cada um dos atributos

Figura 2 – Etapas da seleção de atributos de um filtro



Fonte: García et al (2014, p. 175, tradução própria)

com suas classes, buscando identificar quais possuem maior relevância. De acordo com Chandrashekar e Sahin “um atributo que não possui influência com as classes pode ser descartado” (CHANDRASHEKAR e SAHIN, 2014, p.17, tradução própria).

Ainda de acordo com Chandrashekar e Sahin, “os métodos para construção de *ranking* são filtros desde que eles sejam aplicados antes da classificação para realizar a remoção dos atributos menos relevantes” (CHANDRASHEKAR e SAHIN, 2014, p.17, tradução própria), esta é uma das principais diferenças entre filtros, *wrappers* e *embeddeds*.

O funcionamento de um filtro pode ser dividido em duas etapas, a seleção dos atributos de acordo com as métricas utilizadas, independente do algoritmo de aprendizado de máquina escolhido, e realização do treinamento e validação do modelo utilizando o novo subconjunto de dados. A Figura 2 ilustra o processo de seleção de atributos com um filtro.

Diversas métricas podem ser utilizadas para construção do ranking, como

Correlação de Pearson, Informação Mútua e *Chi-Square*, sendo a Correlação de Pearson uma das técnicas mais utilizadas (BACHU e ANURADHA, 2019).

2.2.4.2 *Wrapper*

Os *wrappers* utilizam um classificador como método de avaliação para decidir sobre a inclusão ou remoção de determinado atributo ao subconjunto de dados. O objetivo deste método é selecionar os atributos que permitam ao classificador obter a maior acurácia (GARCÍA et al., 2014). Em outras palavras, os “algoritmos podem ser usados para obter o subconjunto de variáveis que maximize a função objetiva, que é o resultado da classificação” (CHANDRASHEKAR e SAHIN, 2014, p.18, tradução própria).

Este método possui duas etapas, inicialmente são selecionados os atributos que maximizam a acurácia do classificador. Neste processo pode-se iniciar o subconjunto de atributos vazio ou totalmente preenchido, a depender da direção de busca selecionada, Sequencial para frente (*Sequential Forward*) ou Sequencial para trás (*Sequential Backward*), respectivamente, e a cada iteração uma nova variável é adicionada ou removida do subconjunto e o novo subconjunto é então avaliado. Este movimento, inclusão ou remoção da variável, é mantido até obter a máxima função objetiva (CHANDRASHEKAR e SAHIN, 2014). Após a seleção dos atributos é realizado o treinamento e validação do modelo utilizando o novo subconjunto de dados. A Figura 3 ilustra as etapas do processo de seleção de atributos utilizando um *wrapper*.

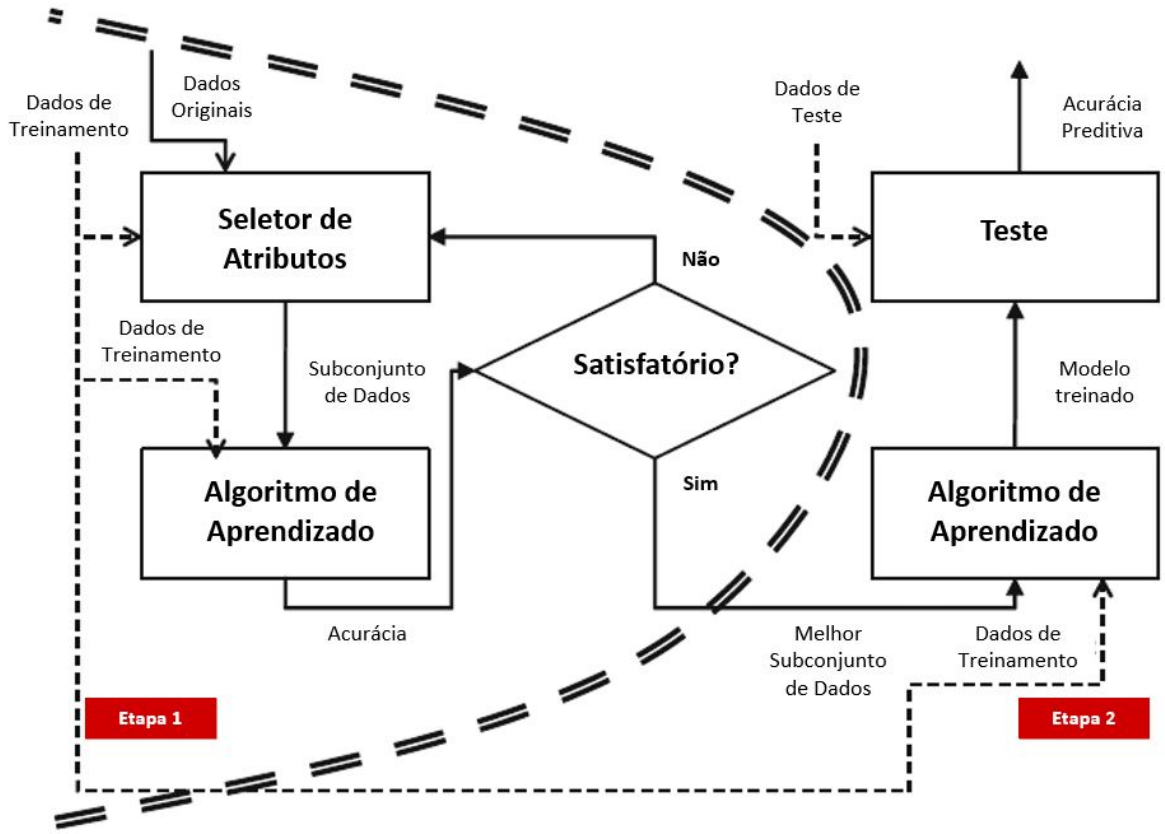
Os *wrappers* são opções simples e universais para a realização da seleção de atributos, porém apresentam elevado custo computacional. De acordo com Urbanowicz et al, na utilização de um *wrapper* "[...] um novo modelo necessita ser treinado para testar qualquer subconjunto de atributos, portanto os *wrappers* são tipicamente iterativos e computacionamente intensos, porém conseguem identificar as variáveis que resultam em melhor performance para o modelo."(URBANOWICZ et al, 2019, p.190, tradução própria) .

2.2.4.3 *Embedded*

Diferentemente dos filtros e *wrappers*, os *embeddeds* incorporam a seleção de variáveis como parte do processo de treinamento dos modelos (GUYON e ELISSEEFF, 2003), obtendo maior eficiência em diversos aspectos.

Uma das vantagens da utilização deste método está na desnecessidade de realizar a separação dos dados em bases de treino e validação, uma vez que o próprio modelo fica encarregado de gerar aleatoriedade na seleção de dados, evitando assim

Figura 3 – Etapas da seleção de atributos de um *wrapper*



Fonte: García et al (2014, p. 174, tradução própria)

o superajuste (*overfitting*). Outra vantagem está na velocidade do processo de seleção de variáveis, pois este modelo dispensa a repetição do processo de treinamento para cada um dos subconjuntos de dados gerados (GUYON e ELISSEEFF, 2003).

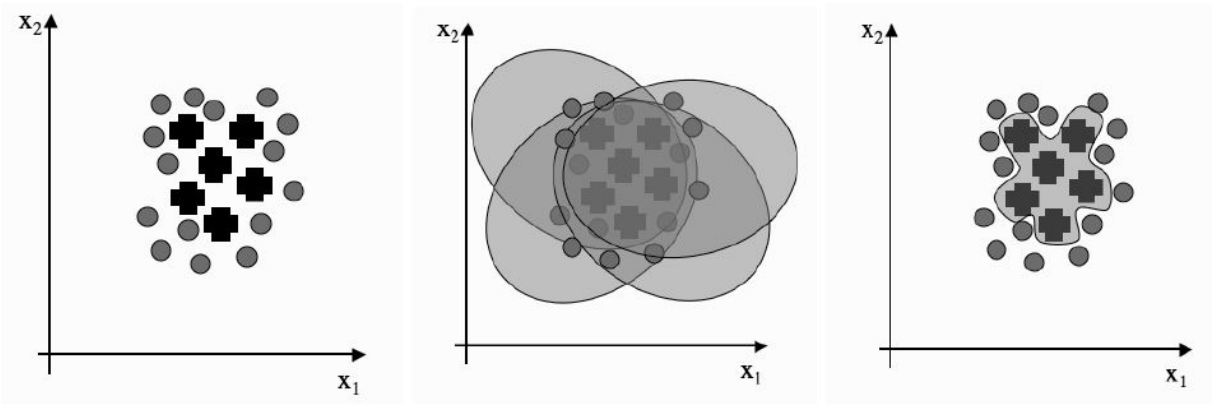
2.3 MODELOS DE CLASSIFICADORES

Atualmente há diversos modelos disponíveis para utilização, cada um com suas peculiaridades. Uma das formas de escolher o modelo adequado para um estudo é avaliar a taxa de erro de classificação para cada um dos modelos candidatos e então escolher o modelo com menor taxa de erro (MURPHY, 2012).

Em aplicações cujos recursos computacionais são limitados, um fator importante a ser levado em consideração na escolha do modelo é o custo associado (i.e. custo computacional dada a complexidade do algoritmo implementado e problema analisado).

Um dos métodos bastante utilizados é chamado de *ensemble* e consiste na cria-

Figura 4 – Etapas do processo de aprendizado com *ensemble*



Fonte: MARSLAND (2014, p. 268)

ção de um conjunto de classificadores que tem seus resultados individuais combinados para gerar um resultado conjunto (DIETTERICH, 2000). De acordo com Marsland, “A ideia básica é que, tendo muitos modelos, cada um pode obter resultados ligeiramente diferentes sobre um conjunto de dados [...] e então juntando-os, os resultados gerados serão significativamente melhores do que os resultados individuais” (MARSLAND, 2014, p. 267, tradução própria). A Figura 4 ilustra a ideia básica do aprendizado com *ensemble* aplicado a um problema de classificação binária, onde cada um dos classificadores insere uma elipse em torno de seu subconjunto de dados, e então, combinando os resultados, obtém-se um resultado mais preciso.

Os resultados dos classificadores individuais podem ser combinados de diversas formas, das mais simples como voto majoritário, até combinações mais complexas, que podem envolver pesos ou penalidades nas decisões (MARSLAND, 2014).

Uma das vantagens da utilização dos *ensemble* é que os resultados são sempre mais precisos do que os obtidos com a utilização dos classificadores individuais (DIETTERICH, 2000), independentemente da quantidade de dados disponível para treinar o modelo (MARSLAND, 2014).

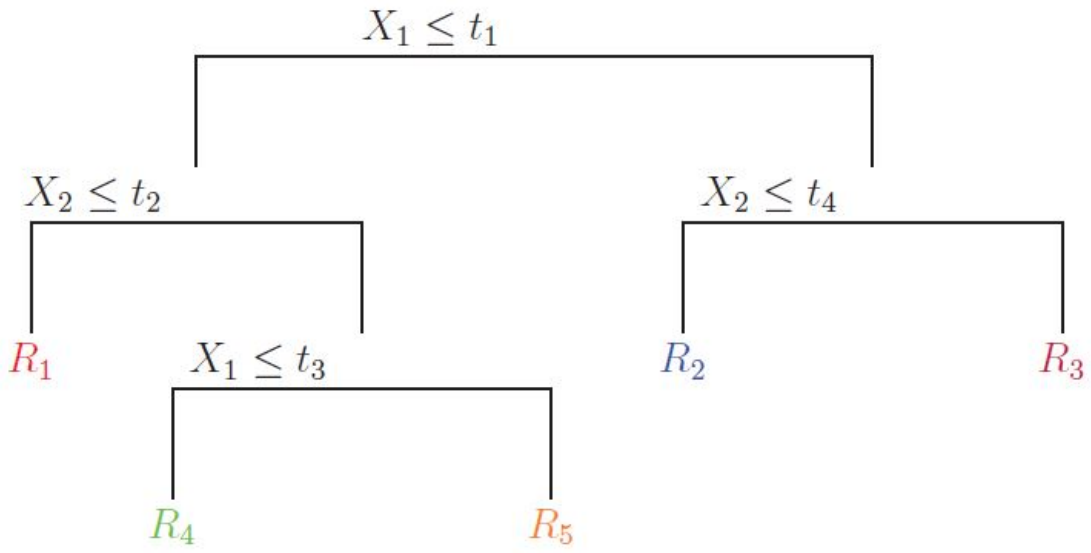
A seguir serão apresentados alguns dos algoritmos que utilizam esta abordagem.

2.3.1 Floresta Aleatória (*Random Forest*)

Este modelo ganhou muita popularidade nos últimos anos e consiste da utilização de um conjunto de árvores de decisão (*decision trees*).

As árvores de decisão são um dos mais conhecidos modelos para aprendizado

Figura 5 – Árvore de decisão para um problema com duas entradas



Fonte: Murphy (2012, p. 545)

de máquina e possuem diversas vantagens como suporte à dados heterogêneos (ordenados e categóricos, ou ainda misturados), robustez à *outliers* e erros nos rótulos de classe, facilidade de interpretação (podendo ser intuitivamente entendido como um conjunto de regras condicionais), além da implementação intrínseca da seleção de atributos (LOUPPE, 2014). De acordo com Marsland, “A ideia de uma árvore de decisão é dividir a classificação em um conjunto de escolhas, um atributo por vez, iniciando na raiz (base) da árvore e progredindo para as folhas, onde recebemos a decisão da classificação.” (MARSLAND, 2014, p. 249, tradução própria).

Murphy explica o funcionamento do modelo da seguinte forma: considerando a Figura 5, "O primeiro nó pergunta se x_1 é menor que um limiar t_1. Se sim, nós então perguntamos se x_2 é menor que outro limiar t_2. Se sim, nós estamos no quadrante esquerdo do espaço, R_1. Se não, nós perguntamos se x_1 é menor que t_3. E assim por diante."(MURPHY, 2012, p.544, tradução própria).

A cada etapa do processo de construção da árvore é selecionado o atributo mais informativo, o que pode ser mensurado através dos conceitos da teoria de informação, como a entropia, uma medida da incerteza sobre uma variável aleatória que nos fornece a quantidade de bits, em média, necessária para descrevê-la (QUINTINO, 2017).

A entropia $H(Y)$ de uma variável aleatória discreta y é dada pela equação 3, onde $p(y)$ é a probabilidade de um evento y ocorrer.

$$H(Y) = -\sum_{y} p(y)\log(p(y)) \tag{3}$$

Esta grandeza é dada em bits quando utilizado o logarítmo de base 2, e em nats se usado o logarítmo natural.

Interessante notar que dado que a entropia é função da probabilidade de ocorrência de um evento $Y = y$, $p(y)$, quanto maior a probabilidade, menor a informação de Shannon, pois temos mais informações sobre o evento. Por outro lado, a entropia de uma variável aleatória uniforme tem seu valor máximo, pois a incerteza é maior já que temos menos informações sobre cada evento, logo, informações sobre esta variável são muito úteis. Resumidamente, sobre a entropia podemos dizer que "O conceito básico é que ela nos fala quanta informação extra obteríamos se soubessemos o valor do atributo." (MARSLAND, 2014, p. 251, tradução própria).

Portanto, a escolha do atributo mais informativo pode ser entendida como escolher o atributo que nos fornece maior informação média, ou seja, o atributo com maior entropia.

A entropia não é a única forma utilizada para criar um *ranking* atributos, o algoritmo CART (Árvores de Classificação e Regressão - *Classification and Regression Trees*), uma das variações do algoritmo de árvores de decisão mais conhecido, o ID3, utiliza uma medida de informação diferente, a impureza Gini (*Gini Impurity*). Segundo Marsland, "A impureza sugere que o objetivo de uma árvore de decisão é ter cada folha representando um conjunto de dados pertencentes à mesma classe, sem que haja classificações equivocadas. Isso é conhecido como pureza." (MARSLAND, 2014, p. 260, tradução própria).

Dado um atributo k, com probabilidade $p(i)$ dos dados pertencerem a uma classe i, onde $i \in \{1, ..., c\}$, calcula-se a Impureza Gini (ou Índice Gini) para cada um dos ramos do nó através da equação 4.

$$I_k = \sum_{i=1}^{c} p(i)(1 - p(i)) \tag{4}$$

A partir do cálculo da impureza gini, parte-se para a operação de divisão,

responsável por dividir o atributo k em dois conjuntos k_L e k_R. O objetivo é identificar a melhor divisão, capaz de maximizar a redução da impureza gini (XIA et al., 2008), através da equação 5.

$$\Delta I = I_k - I_k(k_L)p(k_L) - I_k(k_R)p(k_R) \tag{5}$$

A redução da impureza obtida para cada um dos atributos é utilizada para definir quais deles irão compor os nós da árvore. A profundidade do nó em que um dado atributo é utilizado na árvore pode ser empregada para estimar a sua importância (LOUPPE, 2014). Atributos utilizados no topo da árvore contribuem para a classificação de uma fração maior dos dados do problema.

Uma das grandes vantagens da utilização da floresta aleatória é a sua capacidade de gerar variedade nos classificadores utilizados. O modelo é capaz de gerar aleatoriedade a partir de um conjunto de dados, para isso cada uma das árvores componentes do modelo é treinada com um subconjunto de dados diferente, o que é possível de se obter através da utilização de *bootstrap*, método que consiste em amostragens com reposição a partir do conjunto original de dados. Além disso, a aleatoriedade é ainda maior devido à limitação de escolhas que cada um dos nós das árvores podem fazer, pois é dado apenas um subconjunto aleatório de atributos para cada árvore (MARSLAND, 2014). Os resultados individuais de cada uma das árvores componentes da floresta aleatória são então combinados através da média dos valores obtidos.

2.3.2 AdaBoost

Os algoritmos do tipo *boosting* utilizam uma combinação sequencial de classificadores individuais, diferentemente do processo realizado pela floresta aleatória. Marsland explica que seu funcionamento consiste na divisão da base de treinamento em três partes, e então realiza-se o treinamento do primeiro classificador usando a primeira parte dos dados e valida-se com a segunda parte. Todos os dados classificados erroneamente são utilizados para formar uma nova base, juntamente com uma fração equalizada dos dados classificados corretamente. Em seguida um segundo classificador é treinado com os dados desta base recém criada e então ambos classificadores são testados com a terceira parte dos dados. Caso os resultados apresentados por ambos classificadores seja o mesmo os dados são descartados, caso contrário os dados são utilizados para compor uma nova base, que será utilizada para treinar um terceiro classificador (MARSLAND, 2014).

O principal algoritmo de *boosting* se chama AdaBoost (Impulso Adaptativo -

Adaptive Boosting) e foi descrito inicialmente por Freund e Schapire em meados da década de 1990 (MURPHY, 2012). A vantagem do AdaBoost está na inclusão de pesos aos dados de acordo com a dificuldade que os classificadores anteriores encontraram para classificá-lo.

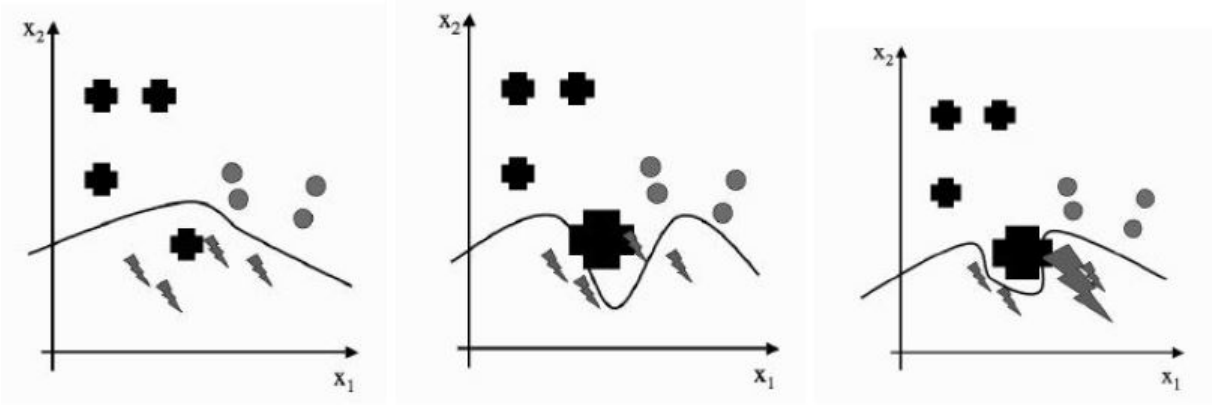
No processo de classificação, os pesos são iniciados com o mesmo valor, $1/N$, sendo N a quantidade de dados utilizados. A cada iteração o erro ϵ é computado como a soma dos valores classificados erroneamente, e os pesos dos dados incorretos são multiplicados por um fator $\alpha = (1 - \epsilon)/\epsilon$. Feito isso, todos os dados são normalizados para que sua soma seja igual a um. O treinamento termina após um determinado número de iterações, ou quando todos os dados forem classificados corretamente, ou ainda quando um dos pontos possuir mais da metade do peso disponível (MARSLAND, 2014). A Figura 6 ilustra a atribuição de pesos durante o processo de classificação com o AdaBoost. Os dados que são classificados erroneamente recebem maior peso, aqui ilustrados com ícones maiores, para que o próximo classificador dê mais importância à eles. O algoritmo completo adaptado de Marsland (2014), pode ser visto abaixo.

Algoritmo AdaBoost

- Inicialize todos os pesos com $1/N$, onde N é a quantidade de dados utilizados
- Enquanto $0 < \epsilon_t < \frac{1}{2}$ ($t < T$, número máximo de iterações)
  - treine o classificador em $\{S, w_n^t\}$, gerando as hipóteses $h_t(x_n)$ para os dados x_n
  - calcule o erro de treinamento ϵ_t = $\sum_{n=1}^{N} w_n^{(t)} I(y_n \neq h_t(x_n))))$
    onde $I(y_n \neq h_t(x_n)$ é uma função que retorna 1 se o valor predito e o real são iguais, e 0 caso contrário
  - defina o fator $\alpha_t = log(\frac{1-\epsilon_t}{\epsilon_t})$
  - atualize os pesos $w_n^{t+1} = w_n^t exp(\alpha_t I(y_n \neq h_t(x_n)))/Z_t$
    onde Z_t é uma constante de normalização
- Saída $f(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$

O algoritmo AdaBoost se mostrou muito bem sucedido na resolução de problemas de classificação de duas classes, e sua utilização em problemas de múltiplas classes envolve a redução em diversos problemas de duas classes (ZHU et al., 2009). Zhu et al propuseram a criação de um classificador que extenda o algoritmo AdaBoost

Figura 6 – Classificação utilizando AdaBoost



Fonte: Marsland (2014, p. 269)

para problemas de múltiplas classes sem que seja necessário reduzi-los em problemas de duas classes.

O algoritmo desenvolvido chama-se AdaBoost-SAMME (*Adaptive Boosting - Stagewise Additive Modeling using Multi-class Exponential loss function*), que pode ser traduzido como Impulso Adaptativo - Modelagem Aditiva em Fases usando a função de perda Exponencial para Múltiplas classes, e inclui um ajuste sutil na aplicação dos pesos para que sejam consideradas as múltiplas classes do problema. Neste trabalho não serão abordadas as formulações matemáticas e justificativas estatísticas deste ajuste.

2.4 AVALIAÇÃO DE DESEMPENHO DO CLASSIFICADOR

Uma das etapas mais importantes do processo de KDD é a avaliação do modelo de aprendizado de máquina, nesta etapa mede-se quão bem o modelo foi capaz de aprender. Além disso, conforme discutido anteriormente, espera-se que o modelo seja capaz de generalizar para dados que não estão presentes no conjunto de treinamento, por isso, a avaliação é realizada com dados de teste, uma fração dos dados originais do problema que não são utilizados no treinamento do modelo (MARSLAND, 2014).

Basicamente a avaliação compara os dados de teste com as predições realizadas pelo modelo, porém diversos aspectos podem ser levados em consideração. A seguir são apresentadas métricas utilizadas para a avaliação dos modelos de classificação supervisionada.

Tabela 1 – Matriz de confusão para um problema de classificação

		Predita		
	Classe	C_1	C_2	C_3
Verdadeira	C_1	5	1	0
	C_2	1	4	1
	C_3	2	0	4

Fonte: Adaptado de Marsland (2014, p. 22)

Tabela 2 – Matriz de confusão para um problema de classificação binária

		Predita	
		Positivo	Negativo
Verdadeira	Positivo	TP	FN
	Negativo	FP	TN

Fonte: Adaptado de Chawla et al (2002, p. 323)

2.4.1 Matriz de Confusão

A matriz de confusão é um método de avaliação bastante simples e muito utilizado em problemas de classificação, consiste na criação de uma matriz de dimensão CxC, onde C denota a quantidade de classes do problema, as colunas representam as classes preditas pelo modelo, enquanto as linhas são as classes reais do problema. Dado que $(i, j) \in (1, ..., C)$, cada um dos elementos (i, j) da matriz representa a quantidade de casos pertencentes a uma classe i e preditos como membros da classe j. Os elementos da diagonal principal, onde $i = j$, são os valores corretamente preditos pelo modelo (MARSLAND, 2014). A Tabela 1 exibe a matriz de confusão para um problema de classificação com três classes.

Com a matriz de confusão é possível identificar os Verdadeiros Negativos (TN - *True Negative*), Falsos Positivos (FP - *False Positive*), Falsos Negativos (FN - *False Negative*) e Verdadeiros Positivos (TP - *True Positive*). Considerando por simplicidade um modelo de classificação binária, a matriz de confusão pode ser interpretada conforme ilustrado na Tabela 2.

2.4.2 Acurácia e Taxa de Erro

A matriz de confusão fornece uma visão bastante ampla sobre o desempenho do modelo de classificação, porém por vezes é necessário traduzi-la em um único número. A acurácia é calculada pela soma dos elementos da diagonal principal dividida pela soma de todos os elementos da matriz de confusão (MARSLAND, 2014), conforme

exibido na Equação 6.

$$Acurácia = \frac{\#TP + \#TN}{\#TP + \#FP + \#TN + \#FN} \tag{6}$$

Em outros termos, a acurácia pode ser interpretada como o percentual de casos preditos corretamente pelo modelo.

Pode-se ainda calcular a taxa de erro do classificador através da Equação 7.

$$Erro = 1 - Acurácia \tag{7}$$

Similarmente, a taxa de erro indica o percentual de casos preditos incorretamente pelo modelo.

2.4.3 Métricas Complementares de Acurácia

De acordo com Marsland, "O problema da acurácia é que não nos diz tudo sobre os resultados, uma vez que transforma quatro números em apenas um." (MARSLAND, 2014, p. 23, tradução própria). Pode-se contornar esta limitação e obter mais informações à respeito do desempenho do modelo com a utilização de métricas complementares, como precisão (também conhecida como confiabilidade positiva) e *recall* (também conhecido como sensibilidade) (MARSLAND, 2014; MONARD e BARANAUSKAS, 2003).

A precisão ou confiabilidade positiva é a razão de verdadeiros positivos entre todos os casos classificados como positivos. O cálculo da precisão é feito através da Equação 8.

$$Precisão = \frac{\#TP}{\#TP + \#FP} \tag{8}$$

O *recall* ou sensibilidade é a razão de verdadeiros positivos entre todos os casos positivos do problema, ou seja, a probabilidade da predição ser C_K dado que pertence à classe C_K. Esta métrica pode ser calculada utilizando a Equação 9.

$$Recall = \frac{\#TP}{\#TP + \#FN} \tag{9}$$

2.4.4 *F-Measure*

Também conhecida como *F-Score* e *F1-Score*, é uma métrica que pode ser calculada através da média harmônica da precisão e *recall* (MURPHY, 2012), conforme exibido na Equação 10.

$$F = 2\frac{precisão \times sensibilidade}{precisão + sensibilidade} \tag{10}$$

De acordo com Murphy, para problemas de múltiplas classes há duas formas de generalizar a *F-Measure*, macro e micro média F-Measure (MURPHY, 2012). A macro média F-Measure fornece um valor médio considerando o cálculo da *F-Measure* para cada uma das classes, conforme exibido na Equação 11.

$$F_{macro} = \sum_{c=1}^{C} \frac{F(c)}{C} \tag{11}$$

Diferentemente da macro média, a micro média *F-Measure* é calculada analisando os dados de todas as classes juntas e possui o mesmo valor da micro média precisão e micro média *recall*, conforme Equação 12. Portanto pode-se obtê-la a partir do cálculo de uma destas duas métricas, considerando-se os dados da diagonal principal da matriz de confusão como TP e os demais como FP ou FN.

$$F_{micro} = Precisão_{micro} = Recall_{micro} \tag{12}$$

Ainda de acordo com Murphy, é preferível utilizar a macro média quando for necessário considerar o mesmo peso para todas as classes e a micro média quando trabalhar com classes desbalanceadas (MURPHY, 2012).

3 REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta uma síntese de estudos já realizados com foco na aplicação de técnicas de ML à dados de acidentes de trânsito.

Chong et al (2005) investigaram a performance das Redes Neurais Artificiais, Árvores de Decisão, SVM e um modelo híbrido, que combina Redes Neurais Artificiais e Árvores de Decisão, aplicados à dados de acidentes de trânsito registrados entre 1995 e 2000 nos Estados Unidos para predição da gravidade das lesões resultantes, classificando-as em Sem Lesão, Possivelmente Lesionado, Lesão Não Incapacitante, Lesão Incapacitante e Lesão Fatal. Neste estudo foram considerados os registros de acidentes com impacto frontal, que correspondem a 98,70% dos registros da base de dados. Os dados foram separados de acordo com suas respectivas classes e comparados com todas as demais classes, de modo que a classe selecionada seja a classe positiva, e as demais, negativas. Os melhores atributos foram determinados através da utilização do método *chi-squared*, capaz de determinar a dependência das variáveis de entrada e respostas do problema. A análise de desempenho dos modelos levou em consideração a acurácia e os resultados revelaram que o modelo híbrido obteve melhor desempenho para as classes Lesão Não Incapacitante, Lesão Incapacitante e Lesão Fatal. Para as classes Sem Lesão e Possivelmente Lesionado o modelo Árvores de Decisão apresentou os melhores resultados.

Yasaswini et al (2018) utilizaram Redes Neurais Convolucionais para identificar os fatores que contribuem para a ocorrência de acidentes de trânsito com vítimas fatais. Neste estudo foram utilizados dados de acidentes de trânsito ocorridos nos Estados Unidos, dos quais foram selecionados os atributos Forma de Colisão, Condições de Iluminação, Condições Climáticas, Condições da Superfície da Via, Limite de Velocidade e Direção Sob Efeito de Álcool. O estudo teve como resultado a definição do fator de risco, alto ou baixo, para cada uma das combinações dos atributos considerados. Os resultados da análise foram comparados através das métricas acurácia, precisão, *recall* e *F-measure* com resultados obtidos anteriormente para a utilização do modelo Naive Bayes. A comparação entre os resultados indica que a performance do modelo Redes Neurais Convolucionais foi muito semelhante ao Naive Bayes, considerando acurácia e *F-measure*, porém identificou-se diferenças consideráveis para as métricas precisão e *recall*. Em síntese os resultados obtidos pelo modelo demonstram a eficiência da aplicação de Redes Neurais Convolucionais ao problema estudado.

Perone (2015) realizou um estudo comparativo utilizando Regressão Logística,

SVM, Naive Bayes, KNN e Floresta Aleatória para a construção de modelos capazes de predizer se um acidente de trânsito resultaria em vítimas lesionadas ou não, a partir de dados de acidentes de trânsito de Porto Alegre, Rio Grande do Sul, ocorridos em 2013. Os resultados dos modelos foram avaliados considerando os valores da AUC (Área Abaixo da Curva) da ROC (Característica de Operação do Receptor), Precisão, *Recall* e *F-measure*. Os modelos construídos utilizando Regressão Logística e SVM apresentaram os melhores resultados de AUC e das médias de Precisão, *Recall* e *F-measure*. O modelo Floresta Aleatória apresentou performance bastante semelhante a apresentada pelos modelos Regressão Logística e SVM, e forneceu também a lista dos atributos mais relevantes para a classificação, que incluem os campos *MOTO*, *AUTO*, *TIPO_ACID ATROPELAMENTO*, *LOCAL LOGRADOURO* e *LOCAL CRUZAMENTO*.

Shanthi e Ramani (2012) realizaram a comparação da performance dos classificadores C4.5, CR-T, ID3, CS-CRT, CS-MC4, Naïve Bayes e Floresta Aleatória, juntamente com a aplicação dos métodos de seleção de atributos CFS, FCBF, MIFS, MODTree e *Ranking* de Atributos, além da combinação com o classificador Arc-X4 Meta, para identificação dos fatores que influenciam a gravidade das lesões em vítimas de acidentes de trânsito, utilizando dados dos Estados Unidos registrados em 2009. A classificação foi realizada utilizando as classes Fatal, Incapacitante, Não-Incapacitante, Possivelmente Lesionado, Nenhum e Desconhecido. A classificação dos dados sem a utilização dos métodos de seleção de atributos revelou que a Floresta Aleatória apresentou melhor performance, com taxa de erro de 14,2%, seguido pelo C4.5, com 15,38%, enquanto o CS-CRT obteve o pior desempenho, 33,54%. Após aplicação dos métodos de seleção de atributos, observou-se que o melhor resultado foi obtido na aplicação do *Ranking* de atributos, que através da eliminação de quatro campos conseguiu reduzir a taxa de erro de classificação da Floresta Aleatória para 5,17%. Por último foi utilizado o classificador Arc-X4 combinado com os classificadores já utilizados, obtendo os melhores resultados na combinação com a Floresta Aleatória, com taxa de erro de 0,27%. A crítica para este estudo fica por conta do uso de atributos que estão relacionados apenas a uma classe, os acidentes fatais, o que influencia nos resultados da classificação, aumentando a acurácia devido à alta taxa de acertos para esta classe, esta característica pode ser vista nas matrizes de confusão apresentadas no estudo, onde é possível verificar que para a classe Fatal foram classificados corretamente em torno de 100% dos casos.

Beshah e Hill (2010) buscaram identificar a contribuição de fatores relacionados às vias na gravidade das lesões de vítimas de acidentes de trânsito ocorridos na Etiópia, classificados em Fatal, Lesão Grave, Lesão Leve e Danos Materiais. Neste estudo foram construídos modelos usando Árvores de Decisão, KNN e Naive Bayes a partir de

uma base de dados reduzida composta por dez atributos, obtida através da utilização de técnicas de seleção de atributos. O processo de treinamento e testes foi realizado usando validação cruzada 10-fold e a performance dos modelos foi avaliada considerando acurácia e AUC. Os modelos obtiveram valores de acurácia muito semelhantes, com o melhor resultado tendo sido obtido pelo KNN, 0,8082. Os valores de AUC obtidos apresentaram maiores discrepâncias, com melhores resultados também obtidos pelo KNN, com AUC de 0,9650 para a classe Fatal, enquanto o segundo melhor modelo, o Naive Bayes, obteve AUC de 0,8550. O estudo utilizou ainda a classe *PART*, disponível no WEKA, que torna possível representar os padrões identificados pelo modelo. A representação indicou que a maior parte dos cenários favoráveis para a ocorrência de acidentes com lesão grave incluem estradas planas e retas.

Banerjee e Khadem (2019) aplicaram Redes Neurais Artificiais para a identificação dos fatores que influenciam a gravidade das lesões ocasionadas em acidentes de trânsito relacionados à ingestão de álcool, para este fim foram utilizados dados de acidentes ocorridos na Carolina do Norte entre 2012 e 2015. Neste estudo foram criados dois modelos, um utilizando três classes - Lesões Fatais e Incapacitantes, Lesões não Incapacitantes e Possivelmente Lesionado, e Sem lesões - e um binário - Lesionado e Não-Lesionado - ambos com quarenta e nove neurônios na camada de entrada e uma camada escondida com 6 neurônios. O modelo binário apresentou acurácia de 68,22% enquanto o modelo com três classes apresentou 63,58%. Ambos modelos apontaram que capotamento, excesso de velocidade, colisão com árvores e ausência de *airbag* são fatores que ocasionam lesões fatais, incapacitantes e não incapacitantes. Os modelos também corroboraram ao indicar que a presença de *airbag* e utilização do cinto de segurança estão relacionados à vítimas sem lesões.

Wahab e Jiang (2019) realizaram um estudo sobre a aplicação dos modelos Árvores de Decisão, Floresta Aleatória e KNN para classificar a gravidade das lesões decorrentes de acidentes de trânsito envolvendo motocicletas, utilizando os dados registrados em Gana entre 2011 e 2015, os resultados obtidos pelos modelos foram comparados também com o modelo estatístico Multinomial Logit. Os modelos de aprendizado de máquina implementados foram validados através da técnica validação cruzada 10-fold e avaliados por suas matrizes de confusão, taxa de verdadeiros positivos, taxa de falsos positivos, precisão, *recall*, AUC e acurácia. Os melhores resultados foram obtidos pelo modelo Floresta Aleatória, com acurácia de 73,91%, os demais modelos de aprendizado de máquina apresentaram resultados ligeiramente inferiores. O pior desempenho foi observado para o modelo estatístico Multinomial Logit, com acurácia de 52,04%. Neste estudo avaliou-se também a importância dos atributos na predição da gravidade das lesões através da taxa de ganho, identificando que os

atributos *Tipo de Localização*, *Hora do Acidente*, *Tipo de Região*, *Tipo de Colisão* e *Tipo da Superfície* estão entre os mais relevantes.

4 METODOLOGIA

Neste capítulo será apresentada a abordagem metodológica do desenvolvimento deste estudo através da descrição das etapas de aquisição dos dados, pré-processamento e aplicação dos modelos de ML, além das técnicas e ferramentas utilizadas.

4.1 AQUISIÇÃO DOS DADOS

Os dados de acidentes de trânsito do ano de 2015 utilizados para a elaboração deste trabalho foram disponibilizados pela Administração Nacional de Segurança de Tráfego Rodoviário (NHTSA - *National Highway Traffic Safety Administration*) em setembro de 2016 e são os dados mais atuais disponíveis.

Os dados estão disponíveis em arquivos CSV (*Comma Separated Values*) e arquivos de dados SAS. Por conveniência, optou-se por utilizar a opção de arquivos CSV. O conjunto de dados é composto por 26 arquivos CSV contendo dados sobre os acidentes, veículos e pessoas envolvidas, conforme detalhado na Tabela 3. Além da disponibilização dos dados, a NHTSA também fornece um manual de usuário com informações detalhadas sobre os dados como definições dos atributos, códigos e seus significados. O documento possui ainda um diagrama que ilustra os relacionamentos entre as diferentes bases de dados, como pode ser visto na Figura 7.

Após entendimento inicial dos dados disponíveis para análise optou-se por utilizar as bases *accident*, *person* e *vehicle*, pois este conjunto de dados traz as principais informações à respeito dos envolvidos nos acidentes. A Tabela 4 apresenta um resumo das informações das tabelas selecionadas.

4.2 PRÉ-PROCESSAMENTO DOS DADOS

4.2.1 Tratamento de Nulos das Bases Originais

Inicialmente as bases de dados selecionadas - *accident*, *person* e *vehicle* - foram carregadas para o *Jupyter Notebook*. A primeira análise realizada em cada uma das três bases foi a verificação da presença de nulos, identificando a quantidade para cada um dos atributos.

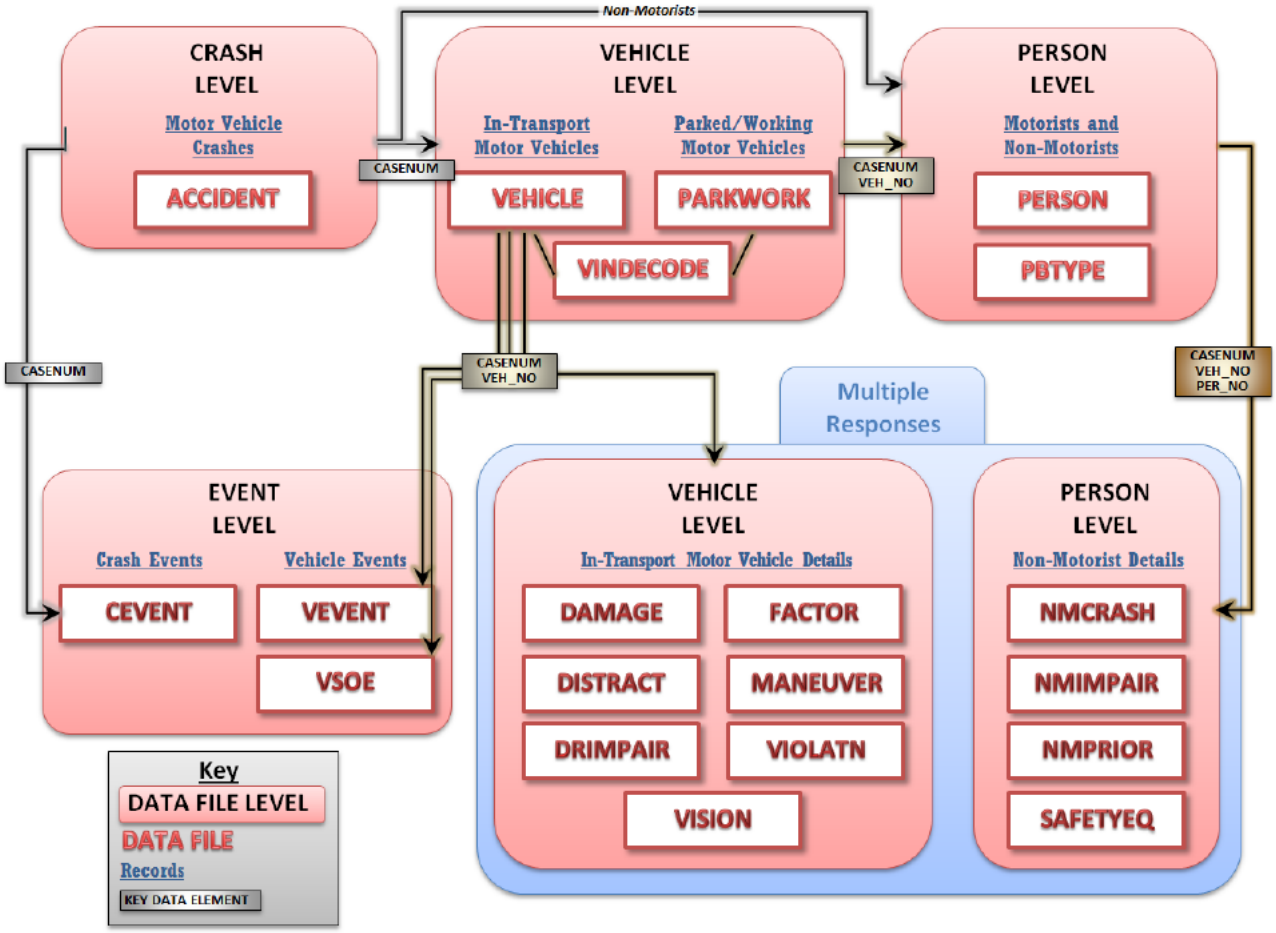
Na base *accident* foram identificados 24.008 registros contendo nulos, o que

Tabela 3 – Lista de tabelas e descrição

Nome da tabela	Descrição
accident	Dados do acidente
vehicle	Dados do veículo
person	Dados dos motoristas e não motoristas
parkwork	Dados do veículo (aplicável a veículos estacionados e a trabalho)
cevent	Eventos perigosos ocorridos em decorrência da colisão
vevent	Eventos perigosos ocorridos em decorrência da colisão para cada veículo
vsoe	Eventos perigosos ocorridos em decorrência da colisão para cada veículo
damage	Identifica a área do dano
distract	Identifica distração do motorista
drimpair	Identifica limitações do motorista
factor	Identifica fatores pré-existentes do veículo que possam ter contribuído para o acidente
maneuver	Identifica tentativa de esquiva antes do acidente
violatn	Identifica violações atribuídas ao motorista
vision	Identifica obstruções visuais
nmcrash	Identifica ações ou circunstâncias do não-motorista que possam ter contribuído para o acidente
nmimpair	Identifica limitações do não-motorista
nmprior	Identifica ações do não-motorista anterior ao seu envolvimento no acidente
safetyeq	Identifica os itens de segurança utilizados pelos não-motoristas no acidente
vindecode	Fornece especificações dos veículos envolvidos no acidente
pbtype	Dados dos pedestres e ciclistas envolvidos nos acidentes
miacc	Nada consta
midrvacc	Nada consta
miper	Nada consta
acc_ux	Dados do acidente com variáveis auxiliares
veh_ux	Dados do veículo com variáveis auxiliares
per_ux	Dados dos motoristas e não motoristas com variáveis auxiliares

Fonte: Fatality Analysis Reporting System (FARS) Analytical User's Manual 1975-2015, 2015

Figura 7 – Diagrama de dados e relacionamentos



Fonte: Fatality Analysis Reporting System (FARS) Analytical User's Manual 1975-2015, 2015

Tabela 4 – Tabelas selecionadas

Tabela	Atributos	Registros
accident	52	32.538
person	68	81.620
vehicle	102	49.478

Fonte: Autoria própria

representa 73,78% dos registros da base, todos no atributo *TWAY_ID2*. De acordo com a documentação dos dados, este atributo fornece o nome da segunda via para os acidentes ocorridos em intersecção de vias. Devido ao fato da base possuir outros atributos mais precisos para identificar a localização do acidente optou-se pela remoção dos campos *TWAY_ID* e *TWAY_ID2*.

A análise da base *person* permitiu identificar dez atributos com valores nulos: *MAKE*, *IMPACT1*, *BODY_TYP*, *MOD_YEAR*, *TOW_VEH*, *SPEC_USE*, *EMER_USE*, *ROLLOVER*, *FIRE_EXP* e *MAK_MOD*. Ao todo foram identificados 7.064 registros com valores nulos, 8,65% dos registros da base. A consulta à documentação dos dados mostrou que estes atributos fornecem informações a respeito dos veículos, por isso também estão disponíveis na base *vehicle*, desta forma, optou-se por remover todos os 10 atributos e utilizar os registros originados da base *vehicle*.

Por fim, a análise da base *vehicle* identificou 167 registros nulos, o que representa 0,34% dos registros, distribuidos em 7 atributos: *VIN_12*, *VIN_11*, *VIN_10*, *VIN_9*, *VIN_8*, *VIN_7* e *VIN_6*. Todos os dados nulos foram identificados em atributos que fornecem o Número de Identificação do Veículo (VIN - *Vehicle Identification Number*), que é um identificador único fornecido pelos fabricantes à cada veículo produzido (RAK, 2019). Este identificador é composto por informações que identificam o país de fabricação, fabricante, ano do modelo etc. Devido ao fato de que a base possui outros atributos redundantes quanto a identificação do veículo, optou-se pela remoção de todos os atributos com informações do VIN.

A Tabela 5 apresenta de forma sintetizada todos os atributos removidos durante a análise de nulos. As dimensões das bases após o tratamento de nulos pode ser vista na Tabela 6.

4.2.2 Montagem da Base Completa e Tratamento de Nulos

Após o tratamento dos nulos das três bases, foi possível realizar a junção dos dados para obter a base de dados completa a ser utilizada neste estudo. Para esta finalidade utilizou-se das funções da biblioteca *Pandas*, disponível na linguagem *Python*. Antes porém foram identificados os atributos em comum entre as bases, removendo-os para impedir a duplicação de dados na nova base. A Tabela 7 lista todos os campos removidos previamente à junção dos dados.

A tabela completa pôde ser obtida através da junção das bases *accident*, *person* e *vehicle*, utilizando os atributos *ST_CASE* e *VEH_NO*, conforme especificado na documentação dos dados, vide Figura 7, dando origem à base *accident_person_vehicle*, composta por 81.620 registros e 164 atributos.

Após a geração da nova base foi verificado novamente a presença de nulos, pois a junção de bases através dos comandos *Left Join* e *Right Join* em casos em que não haja a total correspondência entre as bases acaba por gerar campos não preenchidos. A análise identificou a presença de 7.350 registros nulos, dos quais

Tabela 5 – Atributos removidos após análise de nulos

Base	Atributo	Quantidade de Registros Nulos
accident_df	TWAY_ID	0
accident_df	TWAY_ID2	24.008
person_df	MAKE	7.064
person_df	IMPACT1	7.064
person_df	BODY_TYP	7.064
person_df	MOD_YEAR	7.064
person_df	TOW_VEH	7.064
person_df	SPEC_USE	7.064
person_df	EMER_USE	7.064
person_df	ROLLOVER	7.064
person_df	FIRE_EXP	7.064
person_df	MAK_MOD	7.064
vehicle_df	MAK_MOD	0
vehicle_df	VIN	0
vehicle_df	VIN_1	0
vehicle_df	VIN_2	0
vehicle_df	VIN_3	0
vehicle_df	VIN_4	0
vehicle_df	VIN_5	0
vehicle_df	VIN_6	2
vehicle_df	VIN_7	8
vehicle_df	VIN_8	28
vehicle_df	VIN_9	50
vehicle_df	VIN_10	103
vehicle_df	VIN_11	131
vehicle_df	VIN_12	167

Fonte: Autoria própria

7.064 se tratam de vítimas que não ocupavam veículos, portanto não possuem dados referentes à veículos, os demais 286 eram ocupantes de veículos, porém não possuem estes dados, se tratando possivelmente de um erro na captura dos dados.

Para a correção da presença de nulos dividiu-se a base gerada em três: *data_complete*, contendo os registros totalmente preenchidos, *data_missing_ocupante*, composta pelos 286 registros de ocupantes de veículos, e *data_missing_nao_ocupante*, com os 7.064 registros de vítimas não ocupantes de veículos.

Os nulos da base *data_missing_ocupante* foram substituídos pela média dos campos, valores calculados a partir da base *data_complete*. Esta operação não pôde ser realizada para os campos não-numéricos *RAIL*, *MCARR_ID2* e *MCARR_ID*, portanto foram removidos de todas as três bases.

Tabela 6 – Dimensão dos dados após tratamento de nulos

Tabela	Número de atributos	Quantidade de Registros
accident	50	32.538
person	58	81.620
vehicle	88	49.478

Fonte: Autoria própria

A correção dos nulos da base *data_missing_nao_ocupante* foi realizada por meio de outro procedimento, pois diferente dos dados da base *data_missing_ocupante*, não seria adequado preencher os nulos com a média dos valores da base *data_complete* pois estariamos atribuindo valores de categorias que não existem para este subconjunto de dados, por exemplo atribuir a informação de fabricante de veículo para um vítima que não estava em um automóvel. O mais adequado neste caso é criar uma nova categoria exclusiva, representativa deste subconjunto, para cada um dos atributos da base. Para isso, identificou-se o maior valor para cada um dos atributos, e criou-se uma nova categoria somando uma unidade ao número identificado.

Após a correção dos dados, foi gerada uma nova base, *data_complete_new*, formada pela união das três bases, com 81.620 registros e 160 atributos.

Antes de seguir para a próxima etapa do pré-processamento de dados foi realizada uma análise geral dos atributos presentes na base, confrontando-os com a documentação dos dados a fim de identificar campos que devessem ser descartados. Este estudo identificou 14 atributos com potencial de influenciar nos resultados da classificação, pois trazem informações adicionais para uma classe específica, lesões fatais, podendo enviesar os resultados. A Tabela 8 apresenta os campos identificados e suas descrições, além disso também foram descartados os registros de classes que não serão consideradas neste estudo, são elas: “Lesionado, gravidade desconhecida”, “Morto antes da colisão”, “Não reportado” e “Desconhecido”.

Após a remoção destes atributos e registros teve-se como resultado uma base com 80.571 registros e 146 atributos.

4.2.3 Separação das Bases de Treinamento e Teste

Tendo posse da base completa seguiu-se para a separação das bases de treinamento e teste. Nesta tarefa foi utilizada a função *train_test_split* do pacote *sklearn.model_selection*. Após consulta à documentação da função optou-se por utilizar os parâmetros *test_size* = 0.20, *random_state* = 0 e *shuffle* = *True*.

Tabela 7 – Atributos duplicados removidos

Base	Atributo
accident_df	COUNTY
accident_df	DAY
accident_df	FUNC_SYS
accident_df	HARM_EV
accident_df	HOUR
accident_df	MAN_COLL
accident_df	MINUTE
accident_df	MONTH
accident_df	RUR_URB
accident_df	SCH_BUS
accident_df	STATE
accident_df	VE_FORMS
vehicle_df	BODY_TYP
vehicle_df	DAY
vehicle_df	EMER_USE
vehicle_df	FIRE_EXP
vehicle_df	HARM_EV
vehicle_df	HOUR
vehicle_df	MAN_COLL
vehicle_df	MINUTE
vehicle_df	MOD_YEAR
vehicle_df	MONTH
vehicle_df	ROLLOVER
vehicle_df	SPEC_USE
vehicle_df	STATE
vehicle_df	TOW_VEH
vehicle_df	VE_FORMS
vehicle_df	IMPACT1
vehicle_df	MAKE

Fonte: Autoria própria

As bases geradas têm suas informações listadas na Tabela 9.

4.2.4 Balanceamento de Dados

A análise inicial dos dados de treinamento permitiu identificar um desbalanceamento entre as classes do problema, o que pode ser visto na distribuição de frequência das classes apresentada na Figura 8.

A maior parte dos dados pertence à classe 4, Lesão Fatal, e 0, Sem Lesões Aparente, enquanto as classes 1, Possivelmente Lesionado, 2, Suspeita de Lesão Leve, e 3, Suspeita de Lesão Séria, possuem uma quantidade menor de registros.

Tabela 8 – Atributos com informações exclusivas para vítimas fatais

Atributo	Descrição
DEATH_TM	Hora e minuto do óbito
DEATH_YR	Ano do óbito
DEATH_MO	Mês do óbito
DEATH_DA	Dia do óbito
DEATH_HR	Hora do óbito
DEATH_MN	Minuto do óbito
LAG_HRS	Tempo entre o acidente e o óbito
LAG_MINS	Minutos entre o acidente e o óbito, em complemento às horas
RACE	Raça registrada no certificado de óbito
HISPANIC	Origem hispânica registrada no certificado de óbito
WORK_INJ	Óbito durante jornada de trabalho
DOA	Óbito no local do acidente ou a caminho do hospital
DEATHS	Número de óbitos no veículo
FATALS	Número de óbitos no acidente

Fonte: Autoria própria

Tabela 9 – Bases de treinamento e teste

Base	Registros	Atributos
X_train	64.456	145
X_test	16.115	145
y_train	64.456	1
y_test	16.115	1

Fonte: Autoria própria

Tabela 10 – Bases de treinamento após aplicação do SMOTE

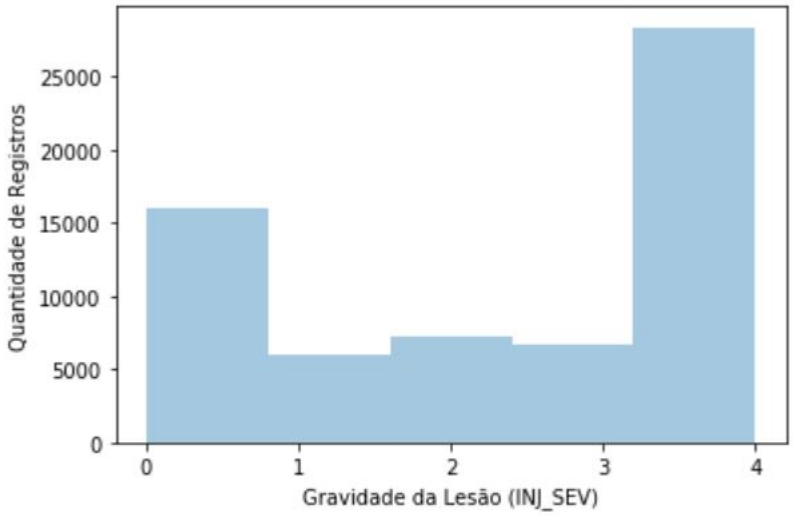
Base	Registros	Atributos
X_res	132.785	145
y_res	132.785	1

Fonte: Autoria própria

Para contornar esta característica dos dados do problema optou-se por realizar o *oversample* dos dados utilizando a técnica *SMOTE*, que nos estudos conduzidos por CHAWLA et al apresentaram ótimos resultados (CHAWLA et al., 2002).

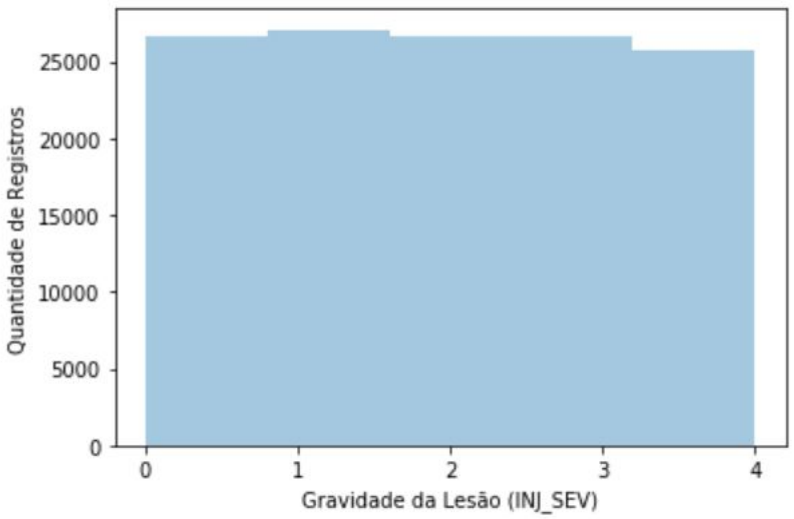
A aplicação da técnica foi realizada utilizando as bases *X_train* e *y_train* através da função *SMOTETomek*, presente no módulo *imblearn.combine*, com o parâmetro *random_state* = 42. Após a realização deste procedimento foram geradas as bases *X_res* e *y_res*, suas dimensões são apresentadas na Tabela 10 e sua distribuição de frequência na Figura 9

Figura 8 – Distribuição de frequência das classes na base de treinamento



Fonte: Autoria própria

Figura 9 – Distribuição de frequência das classes na base de treinamento após aplicação do *SMOTE*



Fonte: Autoria própria

4.2.5 Redução de Dimensionalidade

Nesta etapa do pré-processamento de dados buscou-se identificar os atributos com maior informação da base completa, tendo como benefício uma base de dimensões reduzidas, com eliminação de redundância, ruídos e dados irrelevantes.

Optou-se por realizar a seleção de atributos utilizando um método *embedded* disponível no algoritmo Floresta Aleatória, o *feature_importance*, método simples e de baixo custo computacional. Antes porém, buscaram-se os parâmetros ótimos para a

Tabela 11 – Valores avaliados no processo de identificação de parâmetros ótimos para a seleção de atributos com Floresta Aleatória

Parâmetro	Valores Testados
n_estimators	50, 100, 200, 300, 600, 1000
criterion	gini, entropy
max_depth	10, 30, 50, 100, 200

Fonte: Autoria própria

Tabela 12 – Atributos selecionados e seus valores de importância

Atributo	Importância	Atributo	Importância
HOSPITAL	0,0946	ATST_TYP	0,0141
DEFORMED	0,0267	EJECTION	0,0140
TOWED	0,0241	REST_USE	0,0139
P_CRASH2	0,0231	ALC_RES	0,0136
NUMOCCS	0,0214	PERSONS	0,0131
AIR_BAG	0,0183	DSTATUS	0,0128
MODEL	0,0168	COUNTY	0,0123
DR_ZIP	0,0161	PERMVIT	0,0116
LONGITUD	0,0158	VSPD_LIM	0,0111
ST_CASE	0,0157	DR_HGT	0,0107
ACC_TYPE	0,0157	MINUTE	0,0107
LATITUDE	0,0156	HARM_EV	0,0106
AGE	0,0152	PEDS	0,0103
PCRASH5	0,0149	M_HARM	0,0103
ALC_STATUS	0,0148	MILEPT	0,0102

Fonte: Autoria própria

construção do modelo Floresta Aleatória, maximizando o seu desempenho. Diversos valores foram testados para os parâmetros *n_estimators*, *criterion* e *max_depth* através do método *RandomizedSearchCV*, presente no módulo *sklearn.model_selection*, dos quais foram identificados como melhores os valores 200, gini, e 50, respectivamente. Os valores avaliados são apresentados na Tabela 11

Após definidos os melhores parâmetros, configurou-se o algoritmo Floresta Aleatória para criar um ranking de importância de atributos, do qual foram selecionados os atributos com importância maior ou igual a 0,01, resultando em uma base contendo 30 atributos, 21% dos atributos da base original. A lista de atributos selecionados e seus valores de importância podem ser vistos na Tabela 12.

4.3 APLICAÇÃO DE MODELOS DE ML

Após o pré-processamento dos dados deu-se início a etapa de aplicação dos modelos de ML. Visando melhor rendimento da aplicação em um computador com configurações de hardware simples, optou-se por utilizar os modelos do tipo *ensemble*, no caso a Floresta Aleatória e AdaBoost.

4.3.1 Base Completa

Inicialmente, para efeitos comparativos, foi realizada a modelagem da base completa do problema utilizando o algoritmo Floresta Aleatória com os parâmetros ótimos identificados anteriormente.

Após o treinamento do modelo foram computadas as métricas de acurácia, precisão, *recall* e *F-Measure* através das predições realizadas para os dados presentes na base de teste, também foi gerada a matriz de confusão da classificação, de modo a obter o maior número de informações a respeito do desempenho do modelo.

4.3.2 Base Reduzida

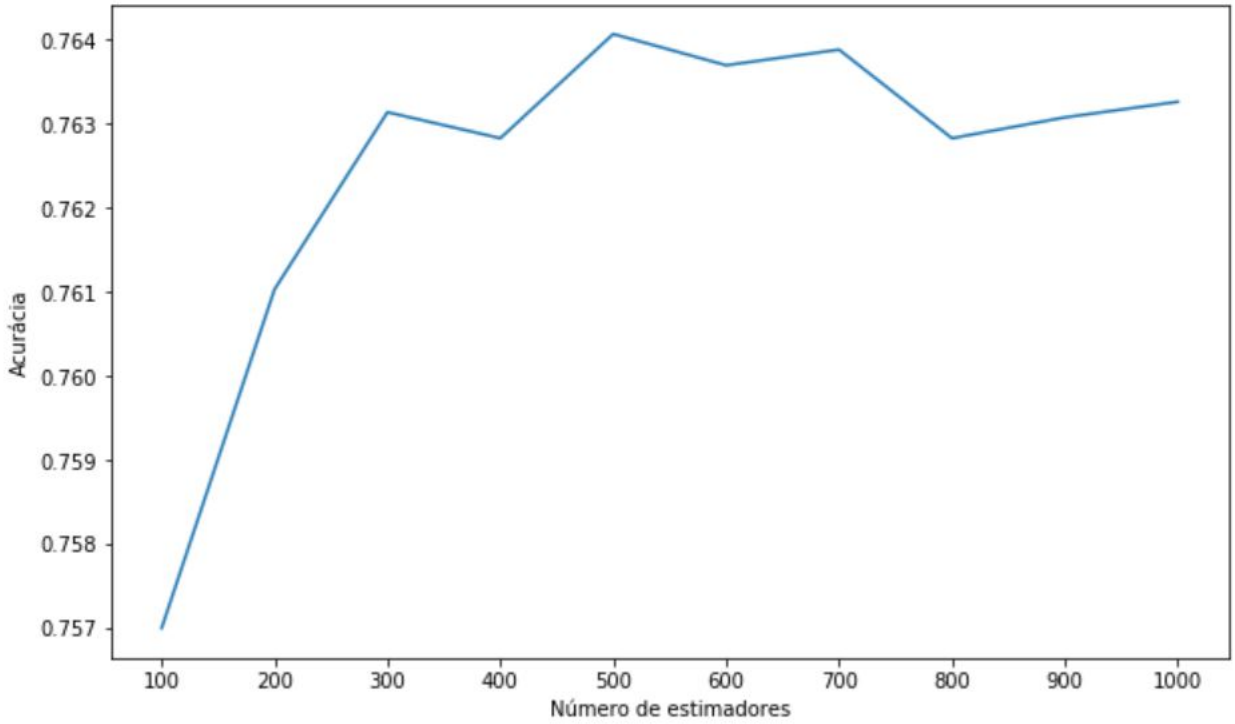
Foi realizada a modelagem da base reduzida utilizando o algoritmo Floresta Aleatória, ainda utilizando os parâmetros ótimos. Em seguida foram computadas as métricas acurácia, precisão, *recall* e *F-Measure* através das predições realizadas para os dados presentes na base de teste reduzida, além de gerada a matriz de confusão.

Visando melhorar a classificação de classes que não tiveram bons resultados com o modelo Floresta Aleatória, foi feita a modelagem da base reduzida utilizando o AdaBoost com o classificador Árvores de Decisão. Foram utilizados os mesmos parâmetros usados no classificador Floresta Aleatória, obtidos através do método *RandomizedSeachCV*. Buscou-se também identificar para o parâmetro *n_estimators* o valor que maximizasse a acurácia do classificador, tendo como resultado o valor 500, conforme ilustrado na Figura 10.

As mesmas métricas acima mencionadas foram calculadas para o modelo.

Por fim, tendo como objetivo a comparação de desempenho com os modelos Ensemble, foi realizada a modelagem da base reduzida utilizando o classificador Árvores de Decisão. Neste classificador foram utilizados os parâmetros *depth* = 50 e *criterion* = gini. O desempenho do modelo foi avaliado através das métricas acurácia, precisão, *recall* e *F-Measure*, além da matriz de confusão.

Figura 10 – Avaliação do parâmetro *n_estimators* para o modelo AdaBoost



Fonte: Autoria própria

4.4 CÓDIGO

Os códigos desenvolvidos para implementação dos procedimentos detalhados nesta seção estão disponíveis em um repositório no *github* e podem ser acessados através deste *link*.

5 RESULTADOS E DISCUSSÕES

Neste capítulo serão apresentados os resultados obtidos para os modelos implementados utilizando as bases completa e reduzida, e comparados seus desempenhos e tempos de treinamento. Será também avaliada a redução de dimensionalidade e sua influência na performance dos classificadores.

5.1 DESEMPENHO DOS MODELOS

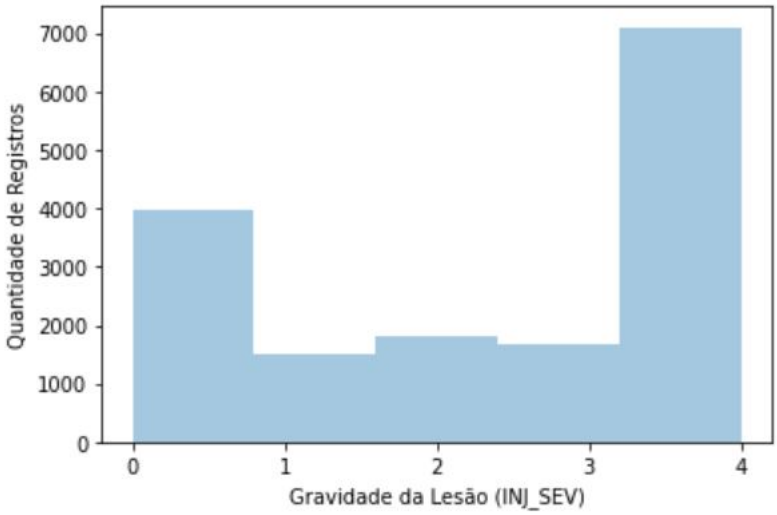
A avaliação dos modelos foi realizada através da utilização dos classificadores treinados para a predição dos dados de teste e análise dos resultados, comparando os valores preditos com os valores reais através das métricas acurácia, precisão, *recall*, *F-Measure* e matriz de confusão.

Visando a preservação das características do problema estudado, os dados de teste não foram submetidos ao processo de balanceamento de classes (CHAWLA et al., 2002), conforme exibido na Figura 11.

5.1.1 Floresta Aleatória aplicada à base completa

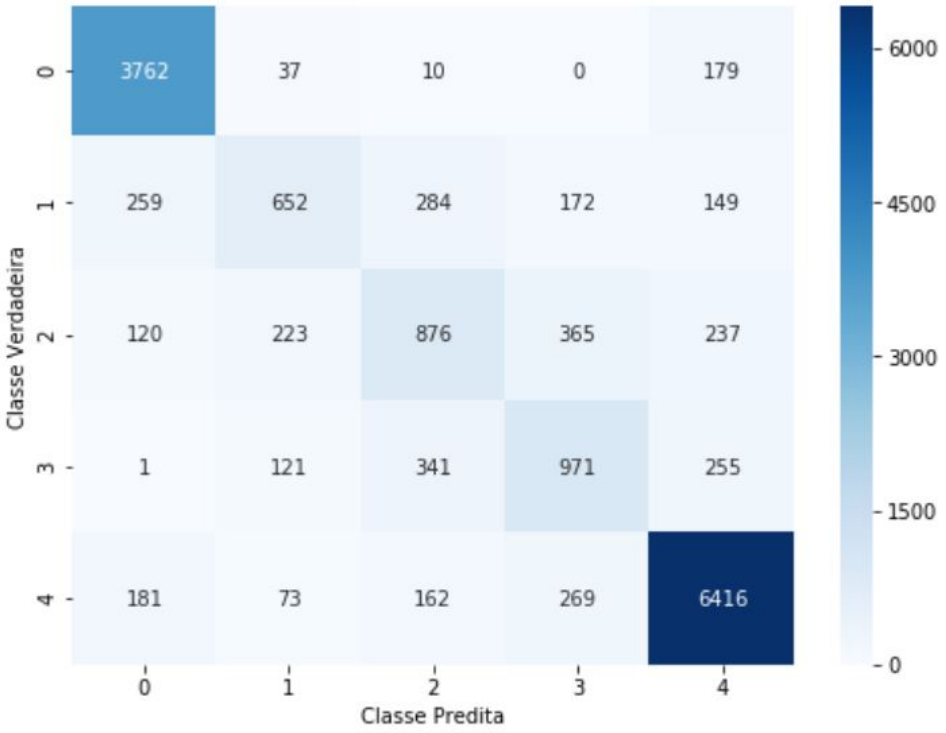
O modelo criado a partir da base completa apresentou bons resultados, dos quais pode-se destacar acurácia de 0,7866 e macro média *F-Measure* de 0,6719 para os dados de teste.

Figura 11 – Distribuição de frequência das classes na base de teste



Fonte: Autoria própria

Figura 12 – Matriz de confusão para o modelo Floresta Aleatória utilizando a base completa



Fonte: Autoria própria

A Figura 12 apresenta a matriz de confusão para o modelo. Visando complementar a análise, foram geradas matrizes de confusão binárias para cada uma das classes, apresentadas na Figura 13. Ambas figuras mostram que as classes 0, Sem Lesões Aparente, e 4, Lesão Fatal, obtiveram ótimos resultados, enquanto as classes intermediárias 1, Possivelmente Lesionado, 2, Suspeita de Lesão Leve, e 3, Suspeita de Lesão Séria, apresentaram resultados inferiores.

Por fim, os valores de precisão, *recall* e *F-Measure* para cada uma das classes do problema, apresentados na Tabela 13, quantificam quão bem o modelo classificou cada uma das classes do problema e proveem a mesma conclusão, os melhores resultados foram obtidos para as classes 0 e 4.

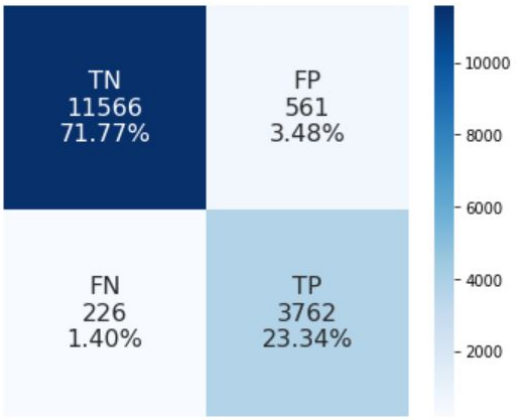
5.1.2 Floresta Aleatória aplicada à base reduzida

O modelo criado com o algoritmo Floresta Aleatória e base reduzida apresentou desempenho muito semelhante ao modelo criado a partir da base completa, com acurácia de 0,7755 e macro média *F-Measure* de 0,6615 para os dados de teste.

A Figura 14 apresenta a matriz de confusão para o modelo, a Figura 15, as matrizes de confusão binárias.

Figura 13 – Matrizes de confusão binárias para o modelo Floresta Aleatória utilizando a base completa

(a) Classe 0 - Sem Lesões Aparente



(b) Classe 1 - Possivelmente Lesionado

(c) Classe 2 - Suspeita de Lesão Leve

(d) Classe 3 - Suspeita de Lesão Séria

(e) Classe 4 - Lesão Fatal

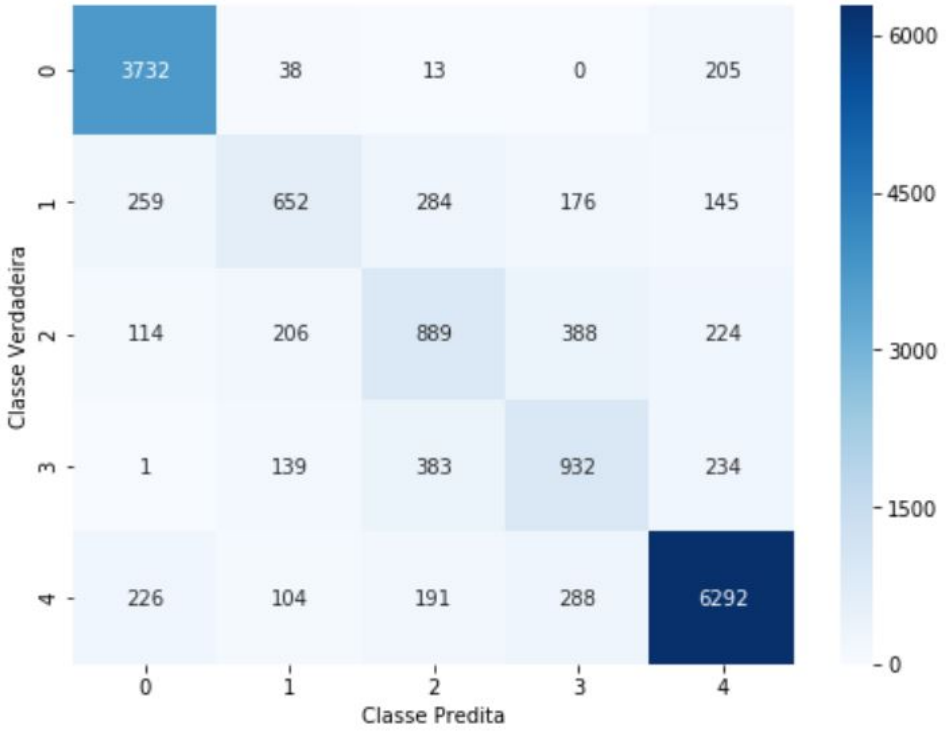
Fonte: Autoria própria

Tabela 13 – Métricas de avaliação para o modelo Floresta Aleatória utilizando a base de dados completa

Classe	Precisão	*Recall*	*F-Measure*
0	0,8702	0,9433	0,9053
1	0,5895	0,4301	0,4973
2	0,5236	0,4810	0,5014
3	0,5464	0,5749	0,5603
4	0,8867	0,9035	0,8950

Fonte: Autoria própria

Figura 14 – Matriz de confusão para o modelo Floresta Aleatória utilizando a base reduzida

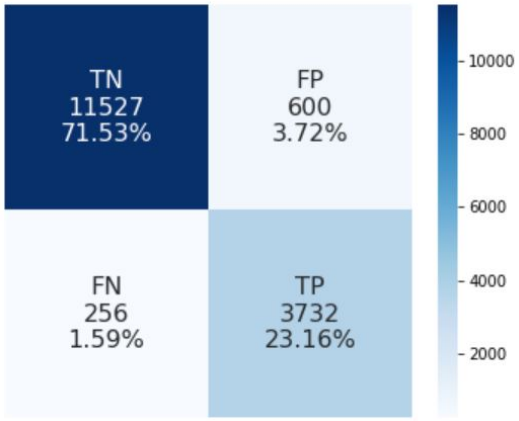


Fonte: Autoria própria

Este modelo apresentou o mesmo resultado do modelo Floresta Aleatória utilizando a base completa. As matrizes de confusão mostram que as classes 0, Sem Lesões Aparente, e 4, Lesão Fatal, obtiveram ótimos resultados, enquanto as classes intermediárias 1, Possivelmente Lesionado, 2, Suspeita de Lesão Leve, e 3, Suspeita de Lesão Séria, apresentaram resultados inferiores. A Tabela 14 quantifica estes resultados.

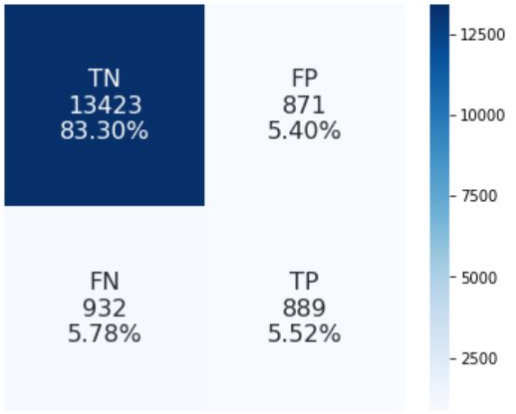
Figura 15 – Matrizes de confusão binárias para o modelo Floresta Aleatória utilizando a base reduzida

(a) Classe 0 - Sem Lesões Aparente



(b) Classe 1 - Possivelmente Lesionado

(c) Classe 2 - Suspeita de Lesão Leve



(d) Classe 3 - Suspeita de Lesão Séria

(e) Classe 4 - Lesão Fatal

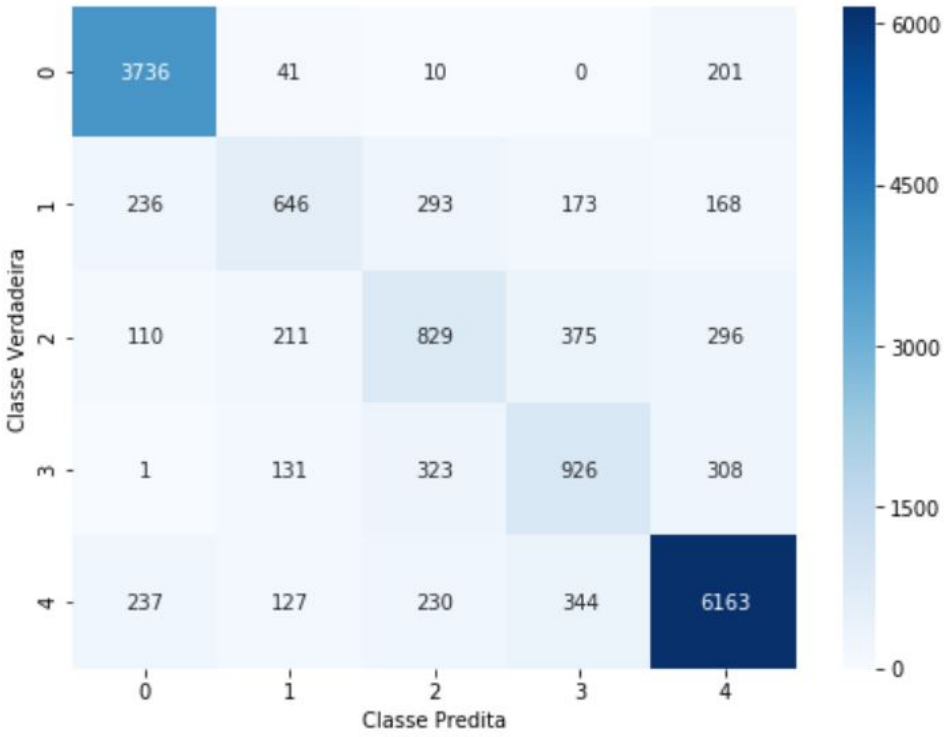
Fonte: Autoria própria

Tabela 14 – Métricas de avaliação para o modelo Floresta Aleatória utilizando a base de dados reduzida

Classe	Precisão	*Recall*	*F-Measure*
0	0,8614	0,9358	0,8971
1	0,5724	0,4301	0,4911
2	0,5051	0,4882	0,4965
3	0,5224	0,5518	0,5367
4	0,8862	0,8861	0,8861

Fonte: Autoria própria

Figura 16 – Matriz de confusão para o modelo AdaBoost utilizando a base reduzida



Fonte: Autoria própria

5.1.3 AdaBoost aplicada à base reduzida

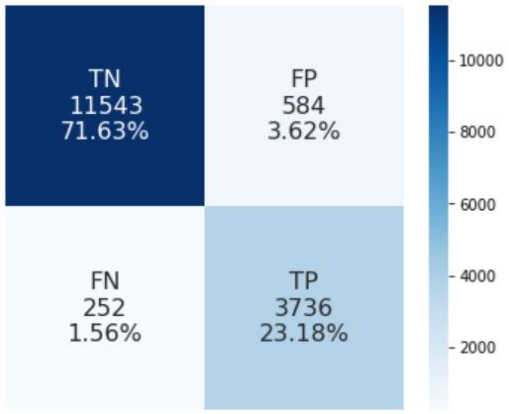
O modelo criado utilizando o algoritmo AdaBoost aplicado à base reduzida teve desempenho similar aos modelos apresentados anteriormente, com acurácia de 0,7632 e macro média *F-Measure* de 0,6499 para os dados de teste.

A Figura 16 apresenta a matriz de confusão para o modelo, a Figura 17, as matrizes de confusão binárias. A Tabela 15 apresenta os resultados calculados para o modelo.

Os resultados obtidos por este modelo são os mesmos vistos para os dois

Figura 17 – Matrizes de confusão binárias para o modelo AdaBoost utilizando a base reduzida

(a) Classe 0 - Sem Lesões Aparente



(b) Classe 1 - Possivelmente Lesionado

(c) Classe 2 - Suspeita de Lesão Leve

(d) Classe 3 - Suspeita de Lesão Séria

(e) Classe 4 - Lesão Fatal

Fonte: Autoria própria

Tabela 15 – Métricas de avaliação para o modelo AdaBoost utilizando a base de dados reduzida

Classe	Precisão	*Recall*	*F-Measure*
0	0,8648	0,9368	0,8994
1	0,5588	0,4261	0,4835
2	0,4920	0,4552	0,4729
3	0,5093	0,5482	0,5281
4	0,8636	0,8679	0,8658

Fonte: Autoria própria

classificadores apresentados anteriormente, com ótimas predições para as classes 0 e 4, e resultados inferiores para as classes 1, 2 e 3.

5.1.4 Árvores de Decisão aplicadas à base reduzida

O modelo criado utilizando o algoritmo Árvores de Decisão aplicado à base reduzida teve desempenho inferior ao Floresta Aleatória e AdaBoost, obtendo acurácia de 0,7057 e macro média *F-Measure* de 0,5900 para os dados de teste.

A Figura 18 apresenta a matriz de confusão para o modelo, a Figura 19, as matrizes de confusão binárias. A Tabela 16 apresenta os resultados calculados para o modelo.

Tabela 16 – Métricas de avaliação para o modelo Árvores de Decisão utilizando a base de dados reduzida

Classe	Precisão	*Recall*	*F-Measure*
0	0,8528	0,8531	0,8530
1	0,3528	0,4142	0,3811
2	0,4052	0,4481	0,4256
3	0,4483	0,4541	0,4512
4	0,8698	0,8110	0,8394

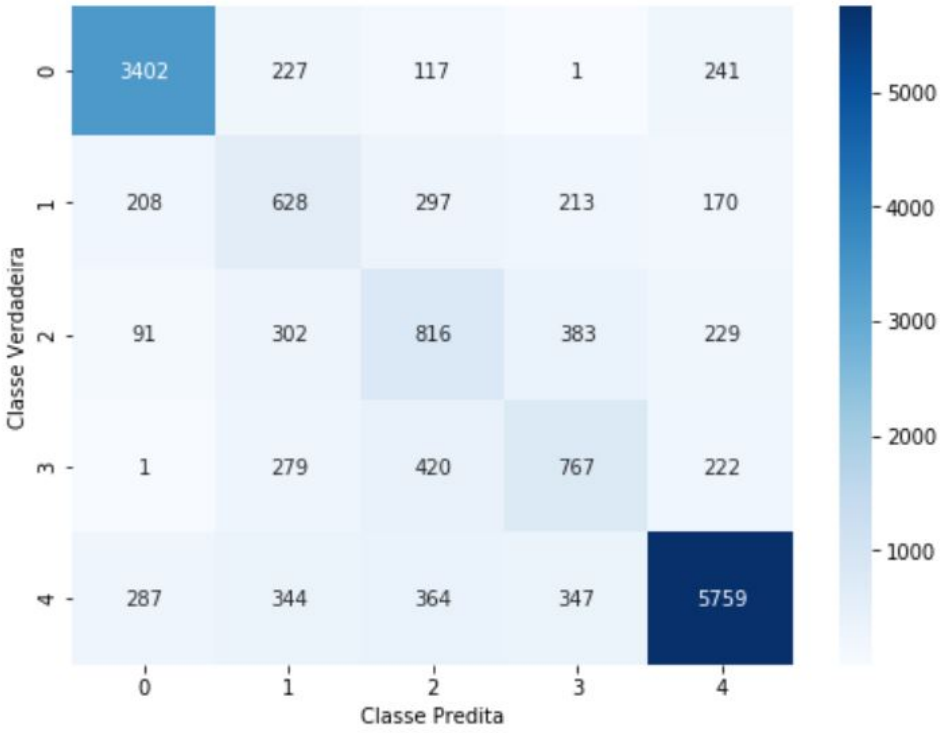
Fonte: Autoria própria

Os resultados obtidos para o modelo Árvores de Decisão apresentaram o mesmo padrão visto nos modelos anteriores, com as melhores predições para as classes 0 e 4, e resultados inferiores para as classes 1, 2 e 3.

5.1.5 Comparação entre os modelos

Nesta seção será analisado o desempenho dos modelos utilizados considerando a macro média *F-Measure* e o tempo necessário para realizar o treinamento,

Figura 18 – Matriz de confusão para o modelo Árvores de Decisão utilizando a base reduzida



Fonte: Autoria própria

Tabela 17 – Comparativo entre os modelos desenvolvidos

Modelo	Macro média *F-Measure*	Tempo
Floresta Aleatória - Base Completa	0,6719	0:03:54
Floresta Aleatória - Base Reduzida	0,6615	0:02:29
AdaBoost - Base Reduzida	0,6499	0:33:37
Árvores de Decisão - Base Reduzida	0,5900	0:00:05

Fonte: Autoria própria

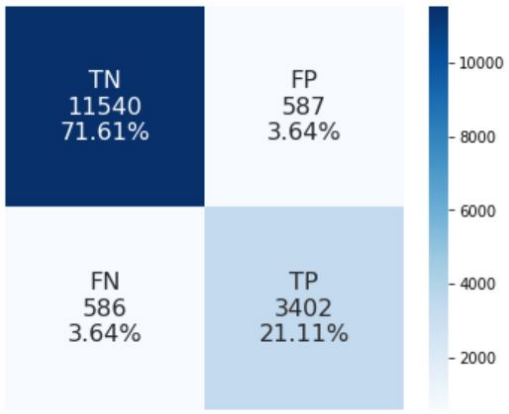
buscando apontar o melhor custo benefício. A Tabela 17 consolida os valores de macro média *F-Measure* e tempo de treinamento para cada um dos modelos.

O modelo Floresta Aleatória criado a partir da base completa obteve bom desempenho na classificação dos dados, com o maior valor de macro média *F-Measure* entre os modelos comparados, além disso não demandou muito tempo para o processo de treinamento, mesmo utilizando uma base muito maior. Os resultados deste modelo podem ser considerados como metas a serem alcançadas pelos demais modelos.

Os modelos Floresta Aleatória e AdaBoost treinados com a base reduzida também apresentaram bons desempenhos na classificação dos dados, com resultados muito próximos ao obtido pelo classificador Floresta Aleatória treinado com a base

Figura 19 – Matrizes de confusão binárias para o modelo Árvores de Decisão utilizando a base reduzida

(a) Classe 0 - Sem Lesões Aparente

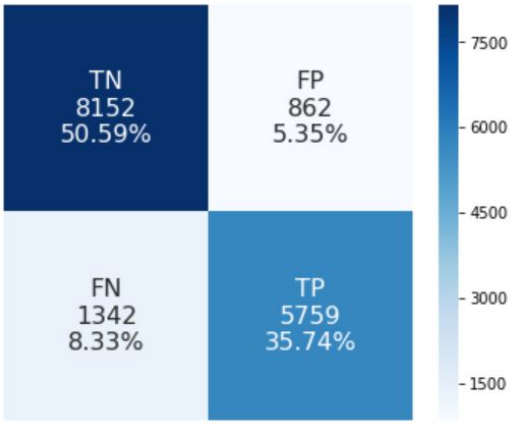


(b) Classe 1 - Possivelmente Lesionado

(c) Classe 2 - Suspeita de Lesão Leve

(d) Classe 3 - Suspeita de Lesão Séria

(e) Classe 4 - Lesão Fatal



Fonte: Autoria própria

completa. Destes modelos o destaque positivo atribui-se ao Floresta Aleatória, que além de bons resultados de classificação, apresentou o menor tempo de treinamento. O AdaBoost, apesar dos resultados de classificação semelhantes, apresentou tempo de processamento treze vezes maior que o Floresta Aleatória.

A discrepância entre o tempo necessário para treinar o AdaBoost já era esperada devido à arquitetura do modelo, que apresenta um processamento sequencial. Cada etapa do processo é dependente da anterior, onde é avaliado o desempenho e atribuição de pesos como forma de reforçar o treinamento das classes que apresentam os piores resultados. A recompensa obtida em troca do maior tempo de treinamento é um melhor desempenho na predição de classes que em outros modelos não apresentam bons resultados, o que não foi observado neste estudo. O resultados obtidos apontam para uma possível indissociabilidade entre as classes intermediárias a partir dos dados do problema.

A comparação com o modelo Árvores de Decisão evidencia que a combinação de classificadores através de métodos *ensemble* é capaz de melhorar significativamente os resultados obtidos.

5.2 REDUÇÃO DE DIMENSIONALIDADE

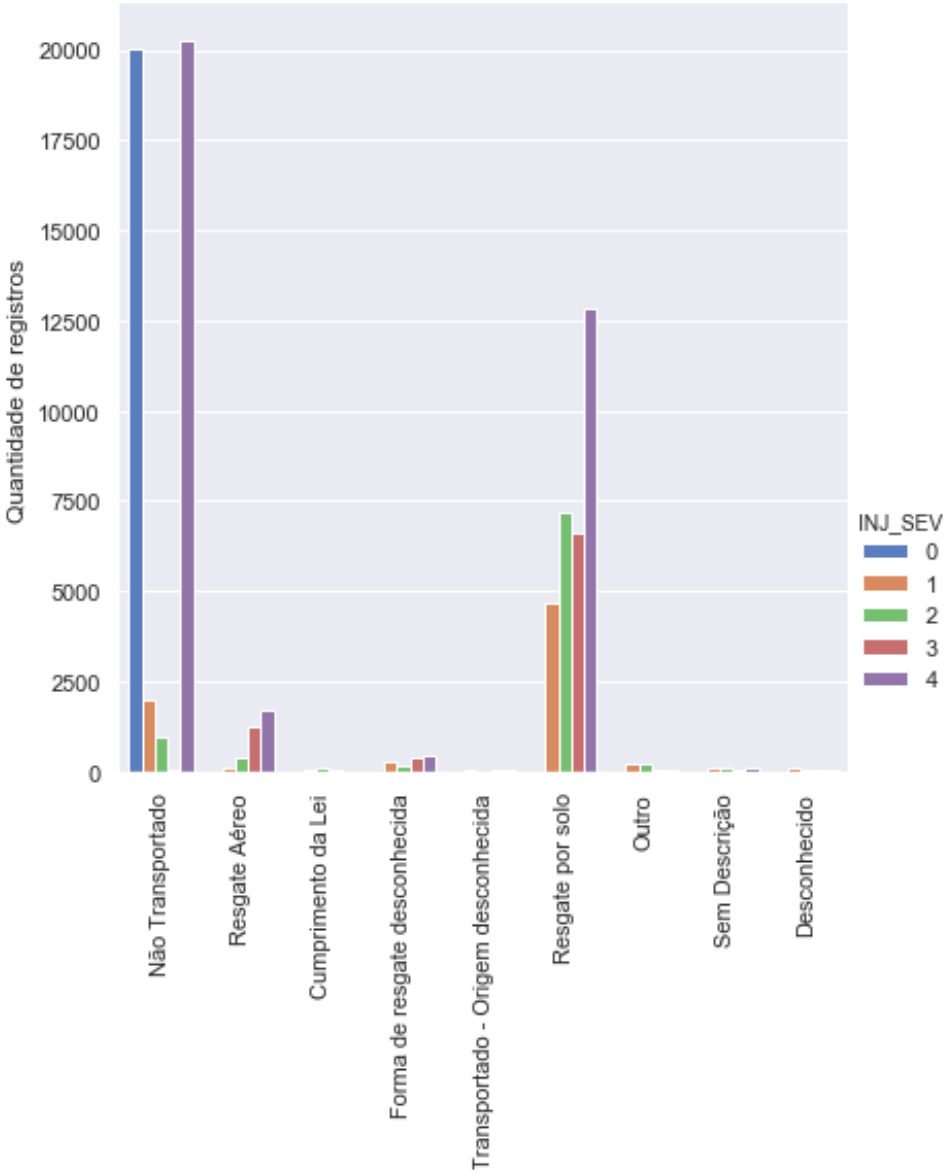
A redução de dimensionalidade realizada neste estudo se mostrou bastante eficaz na tarefa de selecionar os dados mais importantes, eliminando dados irrelevantes, redundantes e ruidosos. A escolha por um método do tipo *embedded* teve como grande vantagem a velocidade de processamento, aspecto de grande relevância no desenvolvimento deste trabalho, sempre pautado pela utilização de métodos e técnicas capazes de serem executadas em computadores com configurações de hardware modestas.

5.2.1 Atributos mais importantes e resultados

A aplicação desta técnica permitiu o descarte de 79% dos atributos da base original, mantendo apenas trinta atributos. Esta redução, embora drástica, não resultou em limitação significativa da capacidade de predição dos modelos treinados com este subconjunto de dados.

Buscando identificar a existência de padrões entre os dados considerados mais significativos e a gravidade das lesões, foi realizada uma análise estatística destes atributos. Esta análise consiste na verificação da distribuição dos dados de acordo com cada uma das classes do problema.

Figura 20 – Análise do atributo *HOSPITAL*

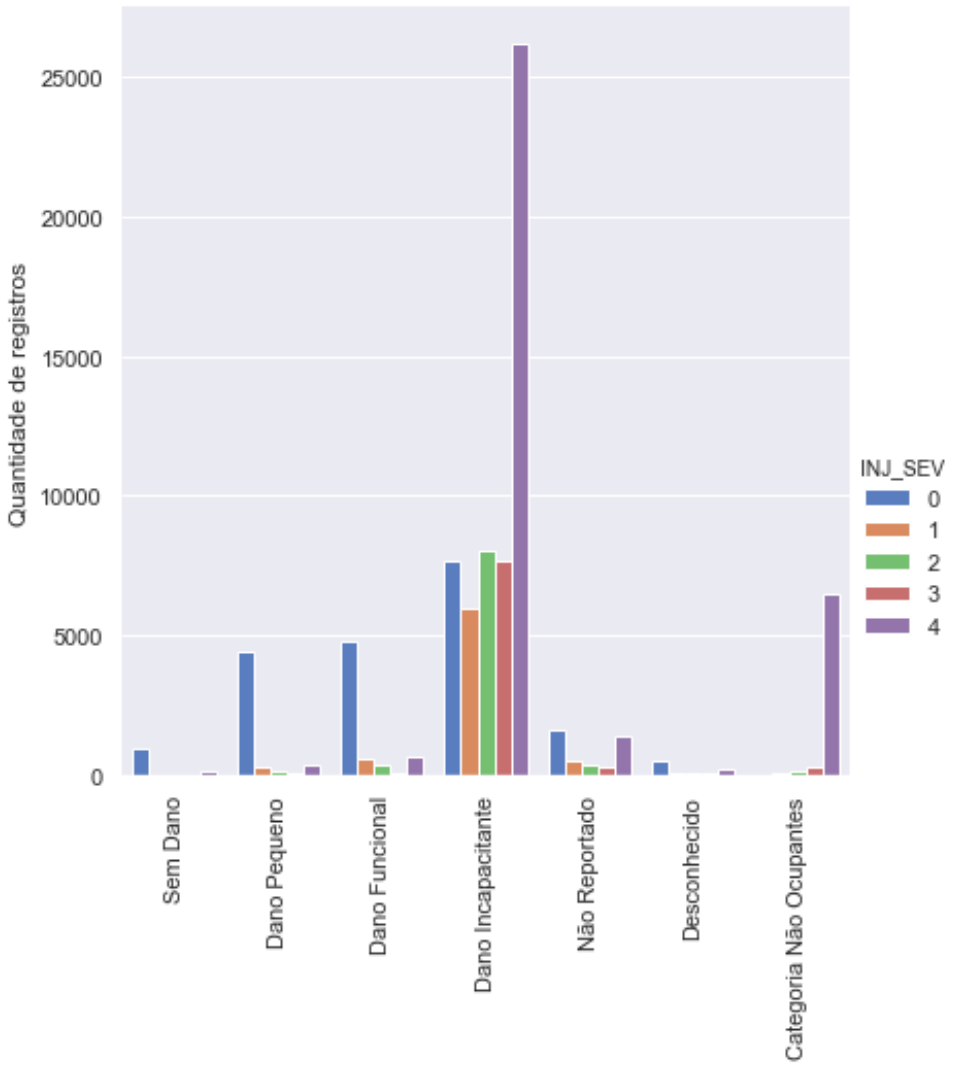


Fonte: Autoria própria

O atributo *HOSPITAL* identifica o meio de transporte ou atendimento médico fornecido às vítimas e sua análise exibiu um padrão que reforça o motivo pelo qual foi identificado como o mais relevante, todas as vítimas pertencentes à classe 0, “Sem Lesões Aparente”, não foram transportadas ao hospital, como pode ser visto na Figura 20. Além disso, a maior parte dos vítimas pertencentes à classe 4, “Lesão Fatal”, não foram transportadas ao hospital ou foram resgatadas por solo.

Os atributos *DEFORMED* e *TOWED* fornecem informações sobre o nível de dano sofrido pelos veículos e as razões por terem ou não sido rebocados, respectivamente. Ambos atributos corroboram com a afirmação de que veículos com maiores

Figura 21 – Análise do atributo *DEFORMED*



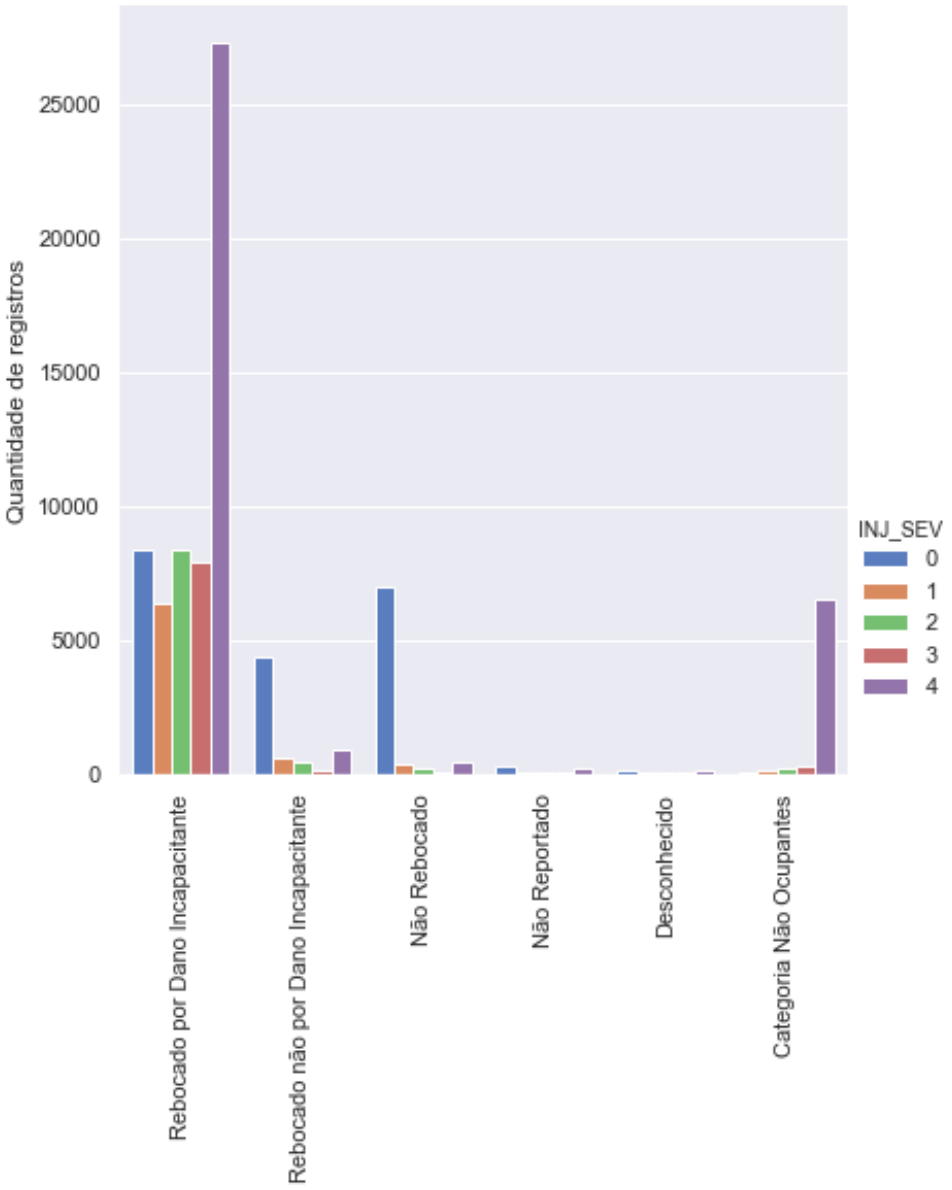
Fonte: Autoria própria

danos, e que por este motivo tenham sido rebocados, estão relacionados à maior parte dos registros de vítimas fatais, conforme exibido nas Figuras 21 e 22.

Estes atributos possuem também uma classe denominada “Categoria Não Ocupantes”, gerada no pré-processamento de dados para as vítimas não ocupantes de veículos que não possuíam dados para o atributo por ser relacionado exclusivamente à veículos. Por este motivo foi gerada uma categoria específica para estes casos. Os próximos atributos analisados também possuem esta informação.

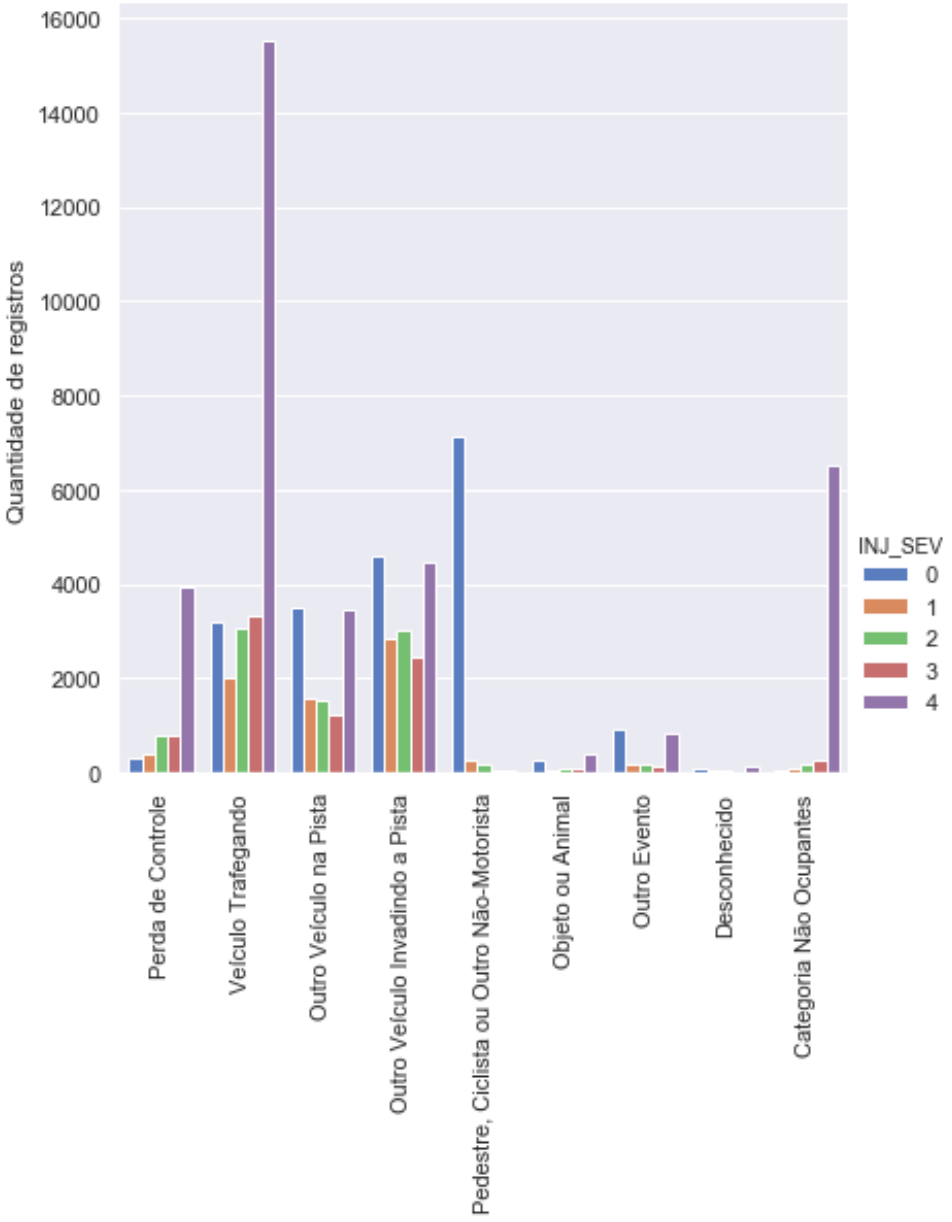
O atributo *P_CRASH2* apresenta o evento crítico que ocasionou o acidente e está representado em dois níveis, categorias dos eventos e os eventos propriamente ditos. A análise das categorias dos eventos mostra que a maior parte das lesões fatais está relacionada à acidentes envolvendo veículos trafegando, e suas ações como

Figura 22 – Análise do atributo *TOWED*



Fonte: Autoria própria

Figura 23 – Análise do atributo *P_CRASH2*



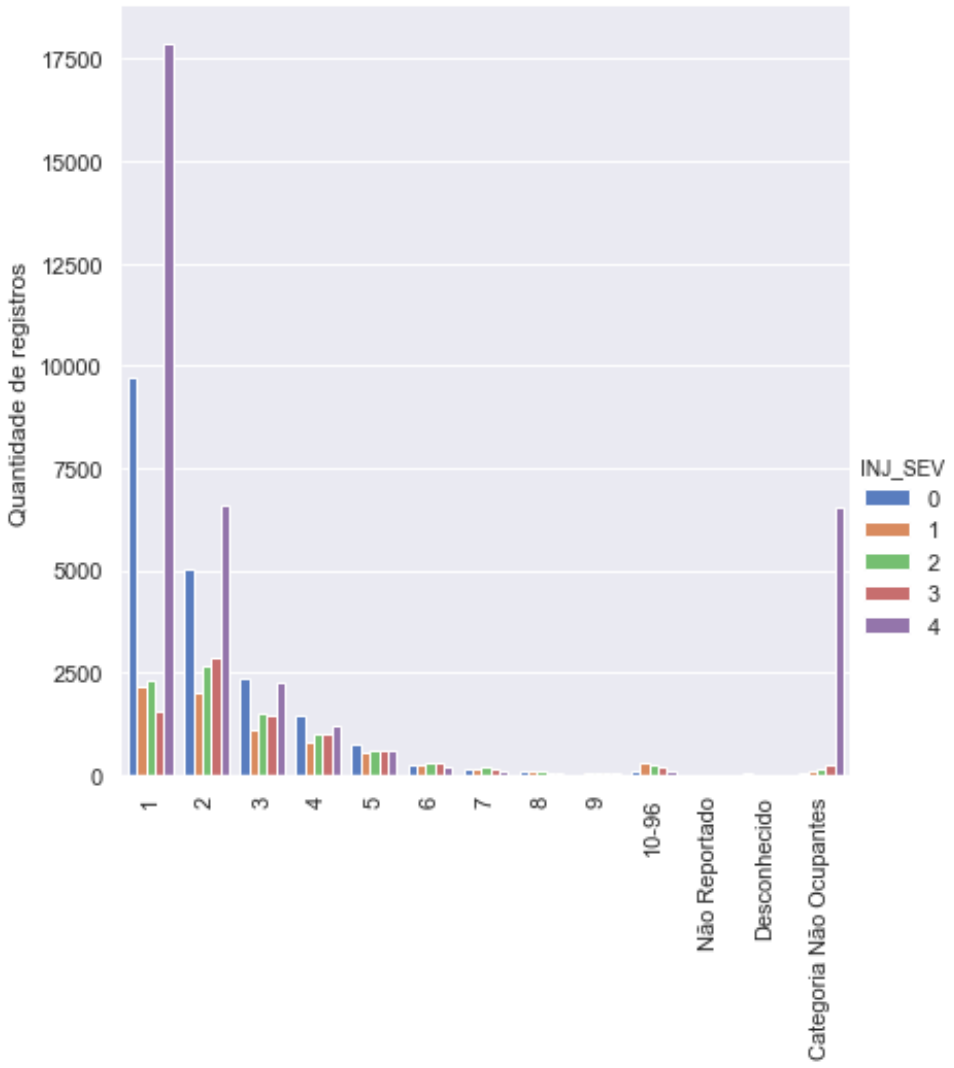
Fonte: Autoria própria

conversões, passagem por intersecções etc. Além disso, acidentes que tiveram como causadores pedestres, ciclistas e outros não-motoristas apresentaram em sua maior parte vítimas sem lesões aparentes, conforme apresentado na Figura 23.

O atributo *NUMOCCS* apresenta uma contagem do número de ocupantes nos veículos e sua análise mostrou que a maior parte dos acidentes com lesões fatais possuem até dois ocupantes por veículo, conforme Figura 24.

Uma característica presente em todos os atributos analisados é que os valores intermediários de gravidade de lesão, 1, 2 e 3, sempre apresentam valores bastante

Figura 24 – Análise do atributo *NUMOCCS*



Fonte: Autoria própria

próximos, sem nenhum destaque entre as classes. Esta característica pode ser responsável pelo baixo desempenho dos modelos na classificação destas classes, pois indicam que não é possível identificar padrões que relacionem os atributos à estas classes.

Por fim, muitos outros padrões podem estar presentes nestes dados, sendo necessária uma análise muito mais complexa para obter êxito na identificação de todos, tarefa inviável para atuação humana. Além disso, os resultados obtidos pelos modelos também são influenciados pelos dados e o pré-processamento realizado sobre eles, o que pode em diversas situações aumentar a complexidade do processo de KDD, tornando os resultados extraídos pelos algoritmos ainda mais difíceis de serem observados nos dados originais do problema.

6 CONCLUSÃO

Este trabalho teve como principais objetivos identificar os atributos mais relevantes para classificação da gravidade das lesões ocasionadas por acidentes de trânsito e avaliar os resultados da aplicação de técnicas de aprendizado de máquina, dando ênfase aos métodos que utilizam conjuntos de classificadores para obtenção de melhores resultados, denominados *ensemble*.

O pré-processamento dos dados utilizados se mostrou bastante eficaz na superação das características do problema que poderiam contribuir para um mau desempenho dos classificadores. Nesta etapa, a utilização da função *SMOTETomek* permitiu realizar o balanceamento dos dados das classes sem a necessidade de descarte de dados, processo que pode ocasionar perda de informações úteis, nem replicação de dados, o que pode causar superajuste do modelo. Bons resultados também foram obtidos na redução de dimensionalidade através da função *feature_importance*, presente na Floresta Aleatória, processo no qual foram eliminados 79% dos atributos da base original, sem limitação significativa de desempenho e com redução no tempo de treinamento. Além disso, o processo de redução de dimensionalidade permitiu também identificar que os atributos *HOSPITAL*, *DEFORMED*, *TOWED*, *P_CRASH2* e *NUMOCCS* são os mais relevantes para a classificação da gravidade das lesões.

A utilização dos métodos *ensemble* foi bastante satisfatória dado sua robustez e baixo custo computacional para implementação, além de ter obtido resultados significativamente maiores que os obtidos pelo modelo Árvores de Decisão, evidenciando que a combinação de classificadores é capaz de melhorar os resultados da classificação. Não houve variações consideráveis entre os resultados dos modelos *ensemble* implementados. O modelo Floresta Aleatória construído com a base completa apresentou o melhor desempenho, seguido pelo modelo Floresta Aleatória com a base reduzida, e por último o AdaBoost com a base reduzida. Ambos modelos apresentaram ótimos resultados para as classes 0, Sem Lesões Aparente, e 4, Lesão Fatal, porém para as classes 1, 2 e 3, Possivelmente Lesionado, Suspeita de Lesão Leve e Suspeita de Lesão Séria, respectivamente, obtiveram resultados inferiores. Este comportamento indica uma possível indissociabilidade dos dados das classes 1, 2 e 3, o que justifica a baixa performance dos classificadores analisados neste estudo quando aplicados à estas classes.

Propõe-se como objeto de estudo futuro a aplicação dos modelos de aprendizado de máquina ao mesmo conjunto de dados utilizado neste trabalho, porém,

considerando um problema com três classes: Sem Lesões Aparente, classe 0, Lesionado, agrupamento das classes 1, 2 e 3, e Lesão Fatal, classe 4. Neste cenário, possivelmente serão obtidos melhores resultados do que os identificados no presente estudo, dado que a junção das classes aqui apontadas como inseparáveis não prejudicará o desempenho geral do classificador. Outra possível linha de estudo consiste na transformação do problema abordado em um problema binário, com as classes Lesionado e Não Lesionado, onde espera-se também obter melhores resultados pelos mesmos motivos apresentados acima.

Recomenda-se ainda a realização de estudos com diferentes abordagens, como a realização da classificação da gravidade das lesões utilizando apenas informações possíveis de serem obtidas no momento da solicitação do resgate. Este modelo poderia ser utilizado para, a partir da predição da gravidade das lesões, fornecer o tipo de resgate que minimalize-a, otimizando o direcionamento das equipes de resgate de acordo com a necessidade das vítimas.

Outra possibilidade de estudo futuro se dá pela aplicação da metodologia apresentada neste estudo à dados de acidentes de trânsito de diferentes localidades do mundo, como forma de identificar os fatores que influenciam na determinação da gravidade das lesões e comparar com os resultados obtidos neste trabalho, de modo a encontrar similaridades e diferenças entre os padrões.

Diante dos fatos apresentados no decorrer deste trabalho e dos pontos ressaltados neste capítulo, demonstra-se que os objetivos propostos foram integralmente cumpridos.

REFERÊNCIAS

ABDALLAH, Z. S.; DU, L.; WEBB, G. I. Data preparation. In: ____. **Encyclopedia of Machine Learning and Data Mining**. Boston, MA: Springer US, 2017. p. 318–327. ISBN 978-1-4899-7687-1. Disponível em: <https://doi.org/10.1007/978-1-4899-7687-1_62>. Acesso em: 19 abr. 2020.

BANERJEE, S.; KHADEM, N. Factors influencing injury severity in alcohol-related crashes: A neural network approach using hsis crash data. **Ite Journal**, v. 89, 01 2019. Disponível em: <https://www.researchgate.net/publication/328697902_Factors_Influencing_Injury_Severity_in_Alcohol-Related_Crashes_A_Neural_Network_Approach_Using_HSIS_Crash_Data>. Acesso em: 25 jun. 2019.

BESHAH, T.; HILL, S. Mining road traffic accident data to improve safety: Role of road-related factors on accident severity in ethiopia. In: **AAAI Spring Symposium: Artificial Intelligence for Development**. [s.n.], 2010. Disponível em: <https://www.researchgate.net/publication/221250993_Mining_Road_Traffic_Accident_Data_to_Improve_Safety_Role_of_Road-Related_Factors_on_Accident_Severity_in_Ethiopia>. Acesso em: 25 jun. 2019.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. Springer, 2006. ISBN 978-0387-31073-2. Disponível em: <http://research.microsoft.com/en-us/um/people/cmbishop/prml/>. Acesso em: 24 jul. 2019.

CHANDRASHEKAR, G.; SAHIN, F. A survey on feature selection methods. **Computers Electrical Engineering**, v. 40, p. 16–28, Jan 2014. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0045790613003066>. Acesso em: 18 fev. 2020.

CHAWLA, N. V.; BOWYER, K. W.; HALL, L. O.; KEGELMEYER, W. P. Smote: Synthetic minority over-sampling technique. **Journal of Artificial Intelligence Research**, AI Access Foundation, El Segundo, CA, USA, v. 16, n. 1, p. 321–357, jun. 2002. ISSN 1076-9757. Disponível em: <https://arxiv.org/pdf/1106.1813.pdf>. Acesso em: 26 ago. 2020.

CHONG, M.; ABRAHAM, A.; PAPRZYCKI, M. Traffic accident analysis using machine learning paradigms. **Informatica**, v. 29, n. 1, 2005. Disponível em: <https://www.researchgate.net/publication/220166391_Traffic_Accident_Analysis_Using_Machine_Learning_Paradigms>. Acesso em: 13 jul. 2019.

DIETTERICH, T. G. Ensemble methods in machine learning. In: **Proceedings of the First International Workshop on Multiple Classifier Systems**. Berlin, Heidelberg: Springer-Verlag, 2000. (MCS '00), p. 1–15. ISBN 3540677046. Disponível em: <https://link.springer.com/chapter/10.1007/3-540-45014-9_1#citeas>. Acesso em: 08 ago. 2020.

GARCÍA, S.; LUENGO, J.; HERRERA, F. **Data Preprocessing in Data Mining**. Springer International Publishing, 2014. (Intelligent Systems Reference Library). ISBN 9783319102474. Disponível em: <https://books.google.com.br/books?id=SbFkBAAAQBAJ>. Acesso em: 29 abr. 2020.

GARCÍA, S.; RAMIREZ-GALLEGO, S.; LUENGO, J.; BENITEZ, J. M.; HERRERA, F. Big data preprocessing: methods and prospects. **Big Data Analytics**, v. 1, n. 1, p. 9, Nov 2016. ISSN 2058-6345. Disponível em: <https://doi.org/10.1186/s41044-016-0014-0>. Acesso em: 14 abr. 2020.

GUYON, I.; ELISSEEFF, A. An introduction to variable and feature selection. **J. Mach. Learn. Res.**, MIT Press, v. 3, p. 1157–1182, 2003. ISSN 1533-7928. Disponível em: <http://portal.acm.org/citation.cfm?id=944919.944968>. Acesso em: 29 jul. 2020.

LOUPPE, G. **Understanding Random Forests: From Theory to Practice**. 2014. Disponível em: <https://arxiv.org/abs/1407.7502>. Acesso em: 19 ago. 2020.

MARSLAND, S. **Machine Learning: An Algorithmic Perspective, Second Edition**. 2nd. ed. CRC Press, 2014. (Chapman & Hall). ISBN 9781466583337. Disponível em: <https://books.google.com.br/books?id=6GvSBQAAQBAJ>. Acesso em: 19 jul. 2020.

MURPHY, K. P. **Machine Learning: A Probabilistic Perspective**. Cambridge, MA: The MIT Press, 2012. (Adaptive Computation and Machine Learning series). ISBN 9780262018029. Disponível em: <https://books.google.com.br/books?id=NZP6AQAAQBAJ>. Acesso em: 20 abr. 2020.

ORGANIZAÇÃO MUNDIAL DA SAÚDE (OMS). **Global status report on road safety 2018**. Geneva, Switzerland: [s.n.], 2018. Disponível em: <https://www.who.int/violence_injury_prevention/road_safety_status/2018/en/>. Acesso em: 06 jul. 2019.

PERONE, C. Injury risk prediction for traffic accidents in Porto Alegre/RS, Brazil. Fev 2015. Disponível em: <https://arxiv.org/abs/1502.00245>. Acesso em: 13 jul. 2019.

QUINTINO, B. A. A informação mútua como medida de dependência não linear na estrutura de rede do mercado brasileiro de ações. **Dissertação (Mestrado em Administração de Organizações) - Faculdade de Economia, Administração e Contabilidade de Ribeirão Preto, Universidade de São Paulo, Ribeirão Preto**, 2017. Disponível em: <https://teses.usp.br/teses/disponiveis/96/96132/tde-25012018-094429/pt-br.php>. Acesso em: 29 abr. 2020.

RAK, R. Vehicle identification number – anatomy of error occurrence. **Journal of Physics: Conference Series**, IOP Publishing, v. 1303, p. 012146, aug 2019. Disponível em: <https://doi.org/10.1088%2F1742-6596%2F1303%2F1%2F012146>. Acesso em: 23 mai. 2020.

SHALEV-SHWARTZ, S.; BEN-DAVID, S. **Understanding Machine Learning: From Theory to Algorithms**. Cambrigde University Press, 2014. ISBN 9781107057135. Disponível em: <https://books.google.com.br/books?id=ttJkAwAAQBAJ>. Acesso em: 23 jul. 2019.

SHANTHI, S.; RAMANI, R. G. Feature relevance analysis and classification of road traffic accident data through data mining techniques. **Proceedings of the World Congress on Engineering and Computer Science**, v. 1, 10 2012. Disponível em: <http://www.iaeng.org/publication/WCECS2012/WCECS2012_pp122-127.pdf>. Acesso em: 14 jul. 2020.

URBANOWICZ, R. J.; MEEKER, M.; La Cava, W.; OLSON, R. S.; MOORE, J. H. Relief-based feature selection: Introduction and review. **Journal of Biomedical Informatics**, v. 85, p. 189 – 203, 2018. ISSN 1532-0464. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1532046418301400>. Acesso em: 04 abr. 2020.

U.S. DEPARTMENT OF TRANSPORTATION (USDOT) - NATIONAL HIGHWAY TRAFFIC SAFETY ADMINISTRATION (NHTSA). **2015 Traffic Fatalities Data, Dataset**. Washington D.C., 2015. Disponível em: <https://www.kaggle.com/nhtsa/2015-traffic-fatalities>. Acesso em: 26 jun. 2019.

U.S. DEPARTMENT OF TRANSPORTATION (USDOT) - NATIONAL HIGHWAY TRAFFIC SAFETY ADMINISTRATION (NHTSA). **Fatality Analysis Reporting System (FARS) Analytical User's Manual 1975-2015**. Washington D.C., 2015. Disponível em: <https://www.kaggle.com/nhtsa/2015-traffic-fatalities?select=docs>. Acesso em: 26 jun. 2019.

WAHAB, L.; JIANG, H. A comparative study on machine learning based algorithms for prediction of motorcycle crash severity. **PLOS ONE**, Public Library of Science, v. 14, n. 4, p. 1–17, 04 2019. Disponível em: <https://doi.org/10.1371/journal.pone.0214966>. Acesso em: 27 ago. 2020.

XIA, F.; ZHANG, W.; LI, F.; YANG, Y. Ranking with decision tree. **Knowledge and Information Systems**, v. 17, n. 3, p. 381–395, Dec 2008. ISSN 0219-3116. Disponível em: <https://doi.org/10.1007/s10115-007-0118-y>. Acesso em: 13 ago. 2020.

YASASWINI, L.; MAHESH, G.; REDDY, S.; SRINIVAS, L. Identifying road accidents severity using convolutional neural networks. **International Journal of Computer Sciences and Engineering**, v. 6, p. 354–360, 07 2018. Disponível em: <https://www.researchgate.net/publication/327073561_Identifying_Road_Accidents_Severity_using_Convolutional_Neural_Networks>. Acesso em: 14 jul. 2019.

ZHU, J.; ZOU, H.; ROSSET, S.; HASTIE, T. Multi-class adaboost. **Statistics and Its Interface**, v. 2, n. 3, p. 349–360, 2009. ISSN 1938-7989. Disponível em: <https://dx.doi.org/10.4310/SII.2009.v2.n3.a8>. Acesso em: 19 ago. 2020.