UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

GUSTAVO RIBEIRO KREMER

Algoritmos de Aprendizado de Máquina Aplicados a Dados Públicos para Obtenção de Insights em Segurança Pública

Monografia apresentada como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação

Orientadora: Profa. Dra. Renata Galante
Co-orientadora: Profa. Dra. Daniela Brauner

Porto Alegre
2023

AGRADECIMENTOS

A minha família, por me darem o suporte necessário para me dedicar e aproveitar as oportunidades durante o longo período da graduação.

Ao Instituto de Física, o Parque Zenit e a PRAE por terem acreditado no meu trabalho e me proporcionado as bolsas, que foram fundamentais para enriquecer a minha formação e cobrir meus gastos durante o processo.

Aos professores que participaram da minha jornada acadêmica e estenderam a mão pra ajudar, especialmente a Renata Galante e Daniela Brauner pela incrível orientação neste trabalho, e Renato Ribas pela parceria ao longo da graduação.

Aos meus colegas extensionistas e de graduação, especialmente dos projetos Reconecta e CRC, pela amizade, parceria e convivência diária que nos fortalece frente a intensa rotina acadêmica. Especialmente Marco Antônio, pela infinita parceria para todos desafios da graduação e da vida.

Aos Rondonistas da Operação Rondon Minas Gerais, pela mais transformadora experiência de sempre, por me mostrarem um outro jeito de ver a vida, e por me encherem, até hoje, de amizade, carinho e compreensão.

E aqueles um dia estiveram de passagem, mas escolheram parar e ficar como amigos, agradeço por terem me dado toda força e determinação para chegar até aqui.

RESUMO

Em todo o território do estado do Rio Grande do Sul, ocorrem diariamente milhares de ocorrências criminais, os registros de tais acontecimentos são volumosos e representam uma grande parcela dos problemas da sociedade moderna. Entre esses acontecimentos, os crimes de furto e roubo de veículos se destacam pelo impacto que causam na sociedade. Esse trabalho explora a utilização de aprendizado de máquina para entender como podem ser utilizados para identificar padrões e prever os crimes de subtração de veículos. São experimentados diferentes modelos de aprendizado não supervisionado e supervisionado aplicados a dados do município de Porto Alegre. Os dados foram disponibilizados pela brigada militar por meio da Lei de Acesso à Informação. Os experimentos com aprendizado não supervisionado conseguiram agrupar as subtrações por localização e valor do veículo, identificando padrões nas ocorrências dos fatos. Após os experimentos, foi desenvolvida uma visualização interativa com *Dashboards*, para visualizar os agrupamentos e permitir assim a obtenção de insights e estratégias de combate ao crime. Os modelos de aprendizado supervisionado obtiveram acurácia de até 67% na predição de qual região da cidade os veículos subtraídos serão encontrados.

Palavras-chave: Aprendizado de máquina. dados governamentais. visualização. inteligência artificial. base de dados.

Machine Learning Algorithms Applied to Public Data from Criminal Occurrences to Obtain Insights in Public Safety

ABSTRACT

Throughout the territory of the state of Rio Grande do Sul, thousands of criminal occurrences occur daily, the records of such events are voluminous and represent a large portion of the problems of modern society. Among these events, the crimes of vehicle theft stand out for the impact they cause on society. In order to understand how they can be used, this work explores the use of machine learning to understand patterns and predict vehicle theft crimes. Different models of unsupervised and supervised learning applied to subtraction data from vehicles in the city of Porto Alegre are experimented. The data were made available by the military brigade through the Access to Information Act. Experiments with unsupervised learning were able to group subtractions by location and vehicle value, identifying patterns in the occurrence of facts. After the experiments, an interactive visualization was developed with Dashboards, to visualize the groupings and thus allow the obtaining of insights and strategies to combat crime. The supervised learning models achieved an accuracy of up to 67% in predicting which region of the city the subtracted vehicles will be found.

Keywords: Machine Learning, Public Data.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AM	*Aprendizado de Máquina*
BID	*Banco Interamericano de Desenvolvimento*
BM	*Brigada Militar*
GB	*Gradient Boosting*
KNN	*K-Nearest Neighbors*
LAI	*Lei de Acesso a Informação*
PPC	*Paridade do Poder de Compra*
PPG	*Programa de Pós Graduação*
RS	*Rio Grande do Sul*
SSPRS	*Secretaria de Segurança Pública do Rio Grande do Sul*

SUMÁRIO

1 INTRODUÇÃO

A frase "Eu só quero é ser feliz e andar tranquilamente na favela onde eu nasci...", presente na letra do famoso funk de Cidinho & Doca (DOCA, 1994), evidencia o descontentamento da população Brasileira com a segurança pública, uma das maiores dores da sociedade moderna, problema capaz de atingir todas as bolhas e classes sociais. Os custos do crime e da violência são altos no Brasil, um estudo feito pelo BID, Banco Interamericano de Desenvolvimento, estima que o montante equivale a 3,14% do PIB do país, considerando apenas o impacto direto (CAPRIOLO, 2017).

Segundo a Constituição Federal, a segurança pública é dever do Estado e direito de todos (BRASIL, 1988), porém o Brasil se destaca por seu alto gasto com segurança privada, o que pode ser entendido como indício de insatisfação da população sobre o serviço de segurança prestado pelo governo (CAPRIOLO, 2017). Ainda de acordo com o estudo de 2014 feito por Capriolo, do custo total da criminalidade: 48% recaiu sobre o gasto privado, 16% sobre o custo social - que é composto pelo custo da vitimização e da renda não gerada - e 36% sobre gastos públicos. O total dos custos do crime e da violência na América Latina é duas vezes maior que a de países desenvolvidos, totalizando $ 509 Dólares PPC per capita no Brasil.

Atualmente, existem sistemas responsáveis pela coleta e armazenamento dos registros criminais, que são feitos por diferentes órgãos de segurança pública ou até mesmo pelo próprio cidadão, via delegacia online. A Secretaria de Segurança Pública do Rio Grande do Sul - de acordo com a Lei nº 15.610 - com intuito de ampliar a transparência ativa de informações de interesse público referentes à segurança do estado, (SSPRS, 2023) publica mensalmente dados descaracterizados dos registros criminais armazenados nos sistemas governamentais. Além dos dados em formato de tabelas, também são disponibilizados dicionários de dados e relatórios estatísticos, importantes ferramentas que auxiliam os gestores nas tomadas de decisões.

De acordo com a Lei de Acesso à Informação, os dados da SSPRS fazem parte da transparência ativa, que ocorre quando o dado público é divulgado pelo órgão de forma aberta. Dados de órgãos públicos que não

estão disponíveis para consulta podem ser solicitados, desde que não apresentem risco à segurança da sociedade ou do Estado e estejam de acordo com a Lei Geral de Proteção dos Dados, devendo o pedido conter a identificação do requerente e a especificação da informação requerida (BRASIL, 2011).

Mesmo considerando a taxa de subnotificações, que no Brasil chega a 80,1% (DATAFOLHA, 2013), esses dados são numerosos e estão disponíveis nos bancos de dados dos órgãos públicos. A utilização de dados pelos gestores permite que eles possam tomar decisões com maior embasamento e, consequentemente, suas organizações se tornam mais efetivas e eficientes (SHAMIM et al., 2019). Dessa forma, essas informações tornam-se vitais para melhor direcionar as ações e políticas.

Dentre os crimes de maior impacto social, encontram-se os roubos e furtos. As taxas de subnotificação de roubos e furtos no Brasil são altas, apesar disso, as subtrações[1] de veículos em específico fogem à regra, as taxas de notificação de roubo de carro e moto são de 90 e 80% respectivamente, furto de carros e moto, 70,3% e 69,5% respectivamente (DATAFOLHA, 2013).

Com o objetivo de aprimorar o processo de tomada de decisão dos órgãos de segurança pública, podemos utilizar algoritmos de aprendizado de máquina. Existem diferentes tipos de aprendizado de máquina, e a escolha depende do tipo de problema a ser resolvido. Algoritmos de aprendizado supervisionado tem o objetivo aprender a prever, a partir de exemplos rotulados, a classe ou atributo de um dado (MAHESH, 2020), logo aprendizado supervisionado pode ser utilizado para prever e entender tendências futuras. Em aprendizado não supervisionado, não existem rótulos, eles são utilizados principalmente para tarefas de agrupamento ou redução de dimensionalidade (MAHESH, 2020), logo, podem ser utilizados para encontrar padrões e identificar tendências.

O objetivo deste trabalho é aplicar técnicas de aprendizado de máquina em dados públicos de registros criminais de roubos e furtos[2] de veículos no município de Porto Alegre, para melhor entender como ocorrem e quais ações podem ser tomadas para mitigá-los. Os dados foram fornecidos pela Brigada Militar via LAI, por intermédio do tenente-coronel Roberto

[1] Nesse trabalho, a palavra "subtração"é utilizada como sinônimo para roubo e furto.
[2] A diferença entre esses dois crimes é que roubo é uma subtração com violência, furto não.

Donato. Um dos motivos para a escolha desse conjunto de crimes em específico é o fato de tais acontecimentos possuírem características de fácil quantificação, como valor, ano do modelo, posição geográfica, marca e cor, atributos que podem ser utilizados em algoritmos de aprendizado de máquina.

Os experimentos com aprendizado não supervisionado com K-Médias se mostraram úteis para identificar padrões de uma forma mais generalista, sugerindo que a cidade seja separada em 6 agrupamentos, utilizando a localização geográfica e o valor dos veículos envolvidos. Dentre os algoritmos de aprendizado supervisionado testados - K-Nearest Neighbors, Árvore de Decisão, Floresta Aleatória e *Gradient Boosting* - o que obteve melhor performance foi o Gradient Boosting, prevendo com 64% de acurácia balanceada quando um veículo será ou não recuperado, e com 67% a região da cidade que ele será levado.

O Capítulo 2 introduz a fundamentação teórica das técnicas aplicadas. O Capítulo 3 apresenta uma revisão dos principais trabalhos relacionados a aprendizado de máquina aplicado a dados públicos de criminalidade. O Capítulo 4 explica a metodologia de cada etapa da análise, apresentando o fluxograma com o processo implementado. O Capítulo 5 analisa os resultados obtidos, observando de forma empírica o desempenho dos modelos. Por fim, o Capítulo 6 revisa o que foi desenvolvido, refletindo sobre a adequação das técnicas aos dados e possíveis aplicações, além de apontar trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo introduz os conceitos e as ferramentas utilizados para o desenvolvimento do trabalho. O capítulo apresenta os conceitos na seção 2.1, e as tecnologias e ferramentas utilizadas na seção 2.2.

2.1 Conceitos

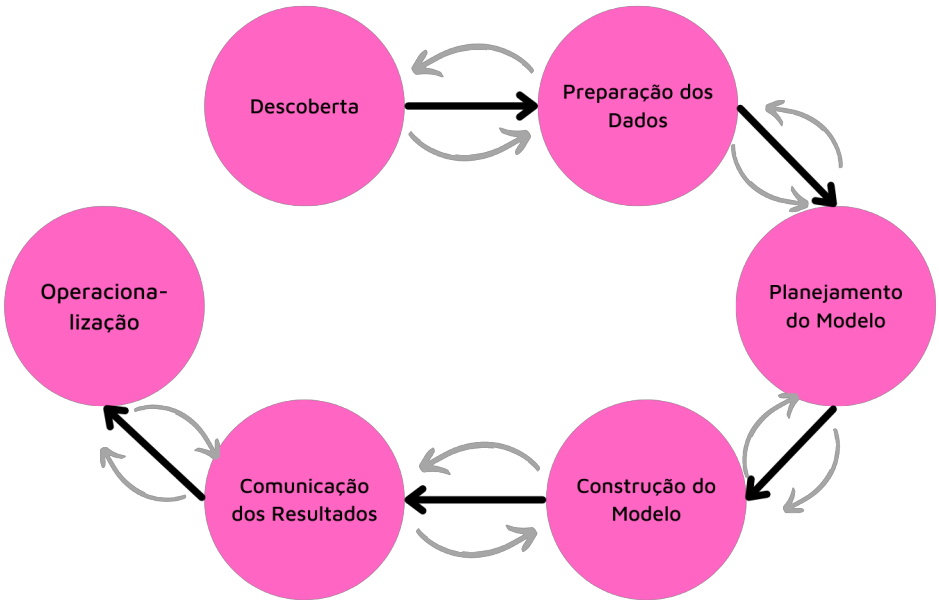
Essa seção apresenta os conceitos utilizados para o desenvolvimento deste trabalho. São explicados os tipos de aprendizado de máquina e os algoritmos utilizados, as métricas utilizadas para avaliar os modelos e o ciclo de vida de um projeto de análise de dados, que foi utilizado como base para a metodologia.

2.1.1 Ciclo de Vida de um Projeto de Análise de Dados

O ciclo de vida de um projeto de análise de dados é composto de seis etapas, ilustradas na Figura 2.1. É possível avançar e retornar para o desenvolvimento de cada uma, conforme for necessário (EMC, 2015). Neste trabalho é utilizada uma variação desse ciclo de vida. Segue uma breve explicação de cada etapa.

- **Descoberta**: A equipe investiga o problema, são feitas reuniões com as partes interessadas para entender o problema, formular hipóteses e encontrar as possíveis fontes de dados.
- **Preparação dos dados**: A equipe extrai e prepara os dados para a análise, isso inclui a extração dos dados, transformações nos atributos e limpeza.
- **Planejamento do modelo**: A equipe determina quais métodos, técnicas e fluxos serão utilizados para construir os modelos. A equipe também explora os atributos para descobrir relações entre as variáveis.
- **Construção do modelo**: A equipe desenvolve os modelos, separa os dados em treino, teste e validação, executando os fluxos estabelecidos na etapa anterior. Nessa etapa, os modelos são testados e avaliados, a

Figura 2.1: Ciclo de vida de um projeto de análise de dados.



Fonte:Elaborado pelo autor.

equipe deve sempre avaliar se os dados e as ferramentas utilizadas são suficientes e estão corretos, ou se será necessário dar um passo pra trás.

- **Comunicação dos resultados**: Nessa etapa, a equipe comunica os resultados para as partes interessadas, avaliando se o projeto obteve sucesso ou não. Nessa etapa, são apresentadas as descobertas e a utilidade delas.
- **Operacionalização**: Nessa etapa final, a equipe entrega os relatórios e as ferramentas desenvolvidas, nessa etapa opcionalmente pode ser desenvolvido um protótipo ou projeto piloto que implementa os modelos desenvolvidos.

2.1.2 Aprendizado de Máquina

De acordo com Arthur Samuel, pioneiro na área de inteligência artificial, aprendizado de máquina pode ser definido como o campo de estudo que habilita o computador a aprender sem ser explicitamente programado (MAHESH, 2020). Existem três tipos principais de Aprendizado de Máquina: Supervisionado, Não Supervisionado e por Reforço (LUDERMIR, 2021). Neste trabalho, são utilizados apenas aprendizado supervisionado e não

supervisionado.

2.1.3 Aprendizado Não Supervisionado

Algoritmos de aprendizado não supervisionado, diferente do supervisionado, não passam por um processo de treinamento para estimar uma classe alvo, não existem estimativas corretas ou erradas, próximas ou distantes, pois não existe supervisão. Ao invés disso, os algoritmos tentam por si só encontrar e apresentar estruturas interessantes nos dados (MAHESH, 2020). Aprendizado não supervisionado é utilizado para agrupamentos e redução de dimensionalidade nos dados, neste trabalho são utilizados agrupamentos para tentar encontrar grupos e padrões nos dados.

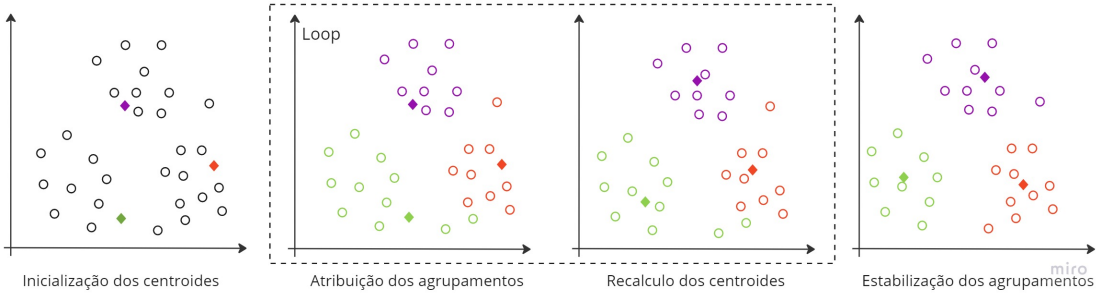
2.1.3.1 KMeans

Para resolver o conhecido problema dos agrupamentos, o K-Médias, do inglês *K-Means*, é um dos algoritmos de aprendizado não supervisionado mais simples. É um algoritmo de agrupamentos de propósito geral que separa os dados em K grupos, minimizando a inércia, que é soma dos quadrados das distâncias dentro dos *clusters* (PEDREGOSA et al., 2011).

Para montar os agrupamentos, o algoritmo realiza essa sequência de etapas:

1. Escolhe k amostras aleatoriamente, esses serão os centros dos agrupamentos, chamados centróides;
2. Calcula a distância de cada instância até cada centróides, cada centróide forma um agrupamento, e as instâncias são atribuídas ao agrupamento cujo centróide está mais próximo;
3. Recalcula os centróides a partir da média de todos os integrantes do agrupamento.
4. O algoritmo entra em *loop* retornando ao segundo passo, onde as instâncias são atribuídas aos centróides mais próximos, e repetindo até que o os agrupamentos fiquem estáveis, isso é, até que os centróides não se movam mais.

Figura 2.2: Exemplo execução do algoritmo K-Médias.



Fonte: Elaborado pelo autor.

Diferentemente de outros algoritmos de agrupamento mais sofisticados, o *KMeans* não determina o número ideal de agrupamentos (K), é necessário passar K como parâmetro. Caso não se saiba qual valor de K informar, existem técnicas que avaliam a qualidade dos agrupamentos, então é possível utilizá-las para descobrir qual valor melhor se adapta.

2.1.3.2 Métricas de Avaliação - Aprendizado Não Supervisionado

No caso dos algoritmos de aprendizado não supervisionado de agrupamento, apesar de não existir um número correto ou errado para a quantidade de grupos, é possível avaliar qual é capaz de distinguir os dados da melhor forma. Para isso, é definida uma faixa de valores de K a serem testados, o que possuir melhor desempenho é o escolhido.

Dentre os métodos disponíveis para avaliação do algoritmo K-Médias, o escolhido para esse trabalho é o método do cotovelo. O método do cotovelo ajuda cientistas a encontrar o número ótimo de agrupamentos para diferentes valores de K especificados (YELLOWBRICK, 2023).

Para encontrar a quantidade ideal de agrupamentos, são plotados a pontuação de distorção, calculada pela soma dos quadrados das distâncias até o centróide de cada agrupamento por K, e se procura pelo ponto da curva com maior inclinação (parecendo-se com um cotovelo). Esse ponto representa o melhor ganho de variância em relação ao número de clusters, logo, o número ideal para K.

2.1.4 Aprendizado Supervisionado

Aprendizado supervisionado é uma tarefa de aprendizado de máquina que mapeia as entradas para uma saída baseada em pares de exemplos de entrada e saída (MAHESH, 2020). O objetivo do algoritmo é construir um classificador que possa determinar corretamente a classe de novos exemplos ainda não rotulados (LUDERMIR, 2021).

Os dados de entrada precisam ser divididos em treinamento e teste, os dados de treinamento são utilizados como exemplos para o modelo para realizar as predições, os dados de teste são usados para medir o desempenho desse modelo tentando realizar as predições. Todos os algoritmos aprendem algum tipo de padrão a partir dos dados de treinamento e os usam para fazer as predições nos dados de teste (LUDERMIR, 2021)

Existem dois tipos de algoritmos de aprendizado supervisionado: os que resolvem problemas de classificação e os que resolvem problemas de regressão. Modelos de classificação identificam qual categoria um objeto pertence, e modelos de regressão predizem valores contínuos associados a objetos (PEDREGOSA et al., 2011).

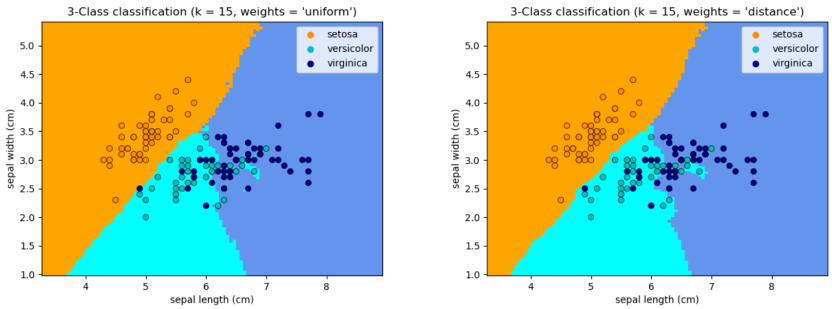
2.1.4.1 Classificador K-Nearest Neighbors

O algoritmo *K-Nearest Neighbors* (KNN) consiste em um algoritmo de aprendizado baseado em instância. Isso porque nenhum modelo é construído, o algoritmo apenas armazena as instâncias de treinamento. Quando recebe uma instância, o KNN verifica as k instâncias mais próximas e classifica conforme a classe da maioria (PEDREGOSA et al., 2011).

KNN é um dos algoritmos mais utilizados, por sua simplicidade, ele é fácil de implementar e possui poucos hiperparâmetros, por isso o KNN é muito usado para comparação com algoritmos mais sofisticados. Uma das restrições desse algoritmo é que ele não escala bem com grandes volumes de dados, torna-se lento e suscetível a *overfitting* nesses casos (IBM, 2023).

Os parâmetros utilizados no algoritmo são o K, o peso dos atributos e o método de cálculo das distâncias. Caso o K seja um valor muito pequeno deixa o algoritmo mais suscetível a ruídos, um valor muito alto deixa as fronteiras entre as classes menos definidas (PEDREGOSA et al., 2011). O

Figura 2.3: Exemplo de modelo KNN.



Fonte: Scikit-learn: *Nearest Neighbors* (PEDREGOSA et al., 2011).

peso dos atributos pode ser uniforme, por distância ou customizado. A fórmula do cálculo das distâncias também é um parâmetro, a fórmula de cálculo mais comum é a Distância Euclidiana (IBM, 2023).

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

2.1.4.2 Classificador Árvore de Decisão

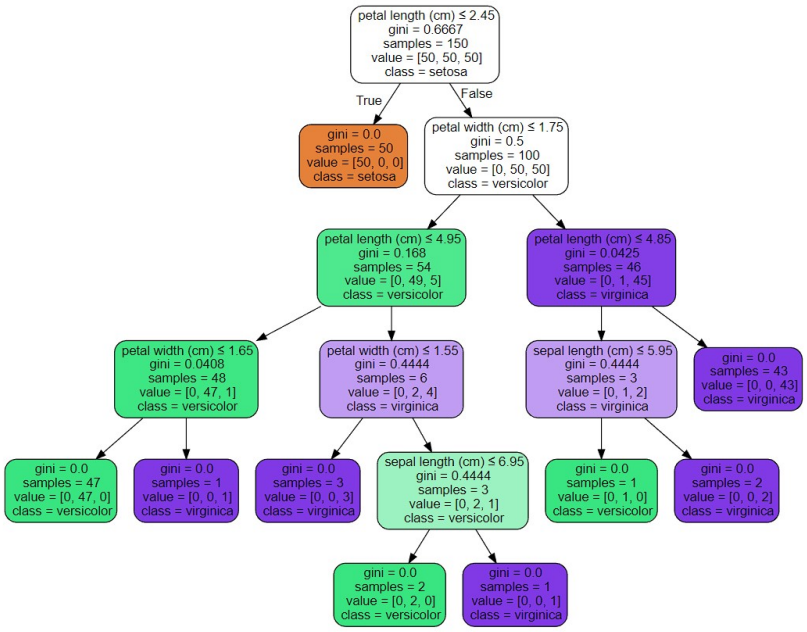
Uma Árvore de Decisão é um modelo que prediz o valor alvo a partir do aprendizado de simples regras de decisão inferidas sobre os atributos (PEDREGOSA et al., 2011). Árvores de decisão são simples de entender e interpretar, podem ser considerados modelos caixa branca.

Para montar a árvore, se inicia pela *feature* com o maior ganho de informação em relação ao atributo alvo, então, o *dataset* é dividido entre as classes, então, para cada filho desse *split*, um nova *feature* é encontrada a partir do mesmo cálculo do ganho de informação, e assim sucessivamente até que a árvore tenha atingido a profundidade máxima. Dentro os hiperparâmetros desse algoritmo, estão: o método de cálculo do ganho de informação, a profundidade máxima da árvore - uma árvore muito profunda pode causar *overfitting* - e a quantidade mínima de amostras para *split*.

2.1.4.3 Classificador Floresta Aleatória

O algoritmo Floresta Aleatória combina diversos modelos de Árvore de Decisão, implementando um modelo de *ensemble*. *Ensemble* é uma técnica de aprendizado supervisionado que consiste em combinar um conjunto de

Figura 2.4: Exemplo de modelo de Árvore de Decisão.



Fonte: Scikit-learn: *Decision Tree Classifier* (PEDREGOSA et al., 2011).

classificadores para formar um classificador mais generalista e robusto (PEDREGOSA et al., 2011).

Uma Floresta Aleatória cria e combina Árvores de Decisão, formadas de forma aleatória normalmente utilizado recortes do *dataset*, gerados a partir de um *bootstrap*. Para fazer uma predição, cada árvore calcula a sua classe predita e a predição final é feita a partir da votação das classes preditas de todas as árvores. A Floresta Aleatória é um modelo robusto e resistente a *overfitting*.

Florestas Aleatórias possuem muitos hiperparâmetros, dentre eles: o número de árvores geradas, o método de formar os recortes dos dados - se será utilizado *bootstrap* ou os dados completos - e todos os outros parâmetros que uma Árvore de Decisão já possui.

2.1.4.4 Gradient Boosting

O *Gradient Boosting* é um algoritmo capaz de criar modelos eficazes que podem ser usado para problemas de regressão e classificação em uma variedade de áreas (PEDREGOSA et al., 2011). O *Gradient Boosting* consiste em um conjunto de Árvores de Decisão, similar a Floresta Aleatória, que juntas formam um modelo de regressão.

O algoritmo gera uma árvore por vez, depois que a primeira árvore é gerada, é feito o cálculo do erro gerado por essa árvore, e então, a cada iteração uma nova árvore é gerada com o objetivo de diminuir esse erro, o erro então é recalculado, realizando assim o processo de *boosting*. O cálculo do erro é gerado pela "função de perda".

Para gerar um modelo de classificação, o treinamento ocorre de forma similar a um modelo de regressão, porém as classes são mapeadas para o valor predito, enquanto a "função de perda" possui o mesmo papel. Mesmo para classificação, o *Gradient Boosting* se comporta como um modelo de regressão (PEDREGOSA et al., 2011).

Entre os hiperparâmetros, é possível especificar:

- O número de estimadores - que é igual ao número de iterações e de árvores geradas;
- A taxa de aprendizado - ou *shrinkage*;
- O método de cálculo da função de perda - que pode ser erro absoluto, erro ao quadrado ou outras métricas;
- Profundidade máxima das árvores.

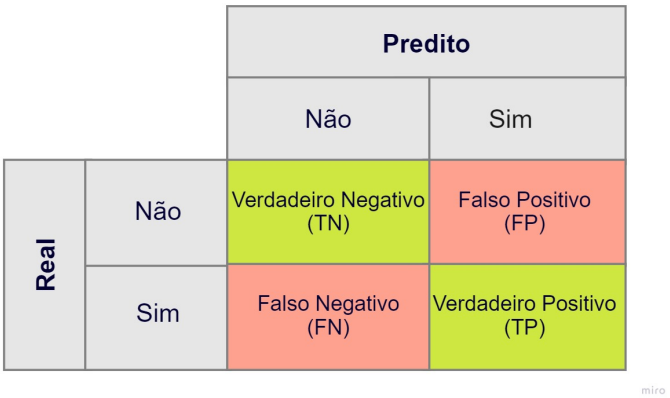
Existem implementações otimizadas do algoritmo *Gradient Boosting*, que facilitam a execução de *datasets* muito extensos, dentre elas o *HistGradientBoosting* do *framework* Scikit-learn, e a implementação do *framework* LightGBM, que possuem tempo de execução e consumo de memória consideravelmente menores do que a implementação clássica. Ambas as versões utilizam estruturas de dados baseadas em números inteiros (*histograms*), que são formados pelos atributos contínuos que são separados em compartimentos de valores inteiros para serem inseridos nas estruturas. Dessa forma, a complexidade do cálculo do ganho é constante e o uso de memória é reduzido (LIGHTGBM, 2023), pois realiza apenas operações com números inteiros.

2.1.4.5 Métricas de Avaliação de Modelos Aprendizado Supervisionado de Classificação

A matriz de confusão é gerada contabilizando os valores preditos, contabilizando os erros e acertos para cada classe. Na Figura 2.5 está a

matriz de confusão para um classificador binário. A partir da matriz, é possível calcular as principais métricas utilizadas para medir o desempenho do algoritmo, são elas:

Figura 2.5: Matriz de confusão.



Fonte: Elaborado pelo autor

- Acurácia - taxa de acertos geral do modelo, calculada pela fórmula:

$$\frac{TP + TN}{TP + FP + TN + FN}$$

- Precisão - taxa de acertos dentre os preditos como classe positiva, calculada pela fórmula:

$$\frac{TP}{TP + FP}$$

- Revocação ou Sensibilidade - taxa de acertos dentre os realmente pertencentes a classe positiva, calculada pela fórmula:

$$\frac{TP}{TP + FN}$$

- Especificidade - taxa de acertos dentre os realmente pertencentes a classe negativa, calculada pela fórmula:

$$\frac{TN}{TN + FP}$$

- F1 Score - balanceamento entre Precisão e Recall, calculada pela fórmula:

$$\frac{2 \times \mathit{Precisão} \times \mathit{Revocação}}{\mathit{Precisão} + \mathit{Revocação}}$$

- Acurácia Balanceada - média da taxa de acertos de cada classe, utilizada em *datasets* desbalanceados, calculada pela fórmula:

$$\frac{1}{2}\frac{TP}{TP+FN}+\frac{1}{2}\frac{TN}{TN+FP}$$

2.1.4.6 Avaliação de Importância de Atributos

Embora o principal objetivo de modelos de predição normalmente seja obter a melhor performance possível, para algumas aplicações é preciso ter um certo nível de interpretabilidade, pois isso permite ter um melhor entendimento de como o modelo funciona, possibilitando melhorias e correções. Além disso, entender como o modelo funciona contribui para evidenciar quais são as características mais importantes, possibilitando o melhor entendimento do problema (CAIRES, 2022), isso é válido tanto para modelos de classificação quanto de regressão.

Os modelos gerados com Floresta de Decisão são os que possuem melhor interpretabilidade, pois um ser humano é capaz de entender quais atributos levaram o modelo a tomar a decisão apenas verificando a árvore. Porém, os algoritmos Floresta Aleatória e *Gradient Boosting*, apesar de serem baseados em Árvores de Decisão, não podem ser visualizados da mesma forma, pois normalmente são compostos por centenas de árvores, tornando-se modelos caixa preta.

Para dar a esses modelos algum nível de interpretação, é possível utilizar técnicas que estimam a importância dos atributos no processo de decisão. A importância de atributo, ou *feature importance*, é uma técnica muito utilizada em modelos de *ensemble*, importância de atributo já vem implementada na maioria das bibliotecas de aprendizado de máquina (CAIRES, 2022) - Scikit-learn e LightGBM possuem.

Para modelos que utilizam Árvores de Decisão, a importância de atributo pode ser calculada contando a quantidade de vezes que o atributo foi utilizado nas árvores de decisão que compõem o modelo. Essa métrica representa importância, pois assume que: quanto mais vezes a variável foi utilizada no modelo, maior o impacto que ela teve na predição (CAIRES, 2022).

2.2 Tecnologias

Essa seção apresenta as tecnologias utilizadas no desenvolvimento do trabalho. A linguagem de programação escolhida foi Python, pela facilidade e disponibilidades das bibliotecas e ferramentas descritas na sequência. O código escrito foi desenvolvido no formato de notebooks, com controle de versionamento com GitHub. Também foi utilizada a plataforma Tableau para geração de gráficos.

2.2.1 Python

Python é uma linguagem de programação, interpretada, orientada a objetos, com suporte múltiplos paradigmas de programação, como funcional e procedural (FOUNDATION, 2021). Python incorpora módulos, exceções, tipagem dinâmica, tipos de dados de alto nível e classes (FOUNDATION, 2021). A linguagem de programação Python é uma linguagem bem estabelecida e uma das mais populares para computação científica (PEDREGOSA et al., 2011).

Python possui diversas bibliotecas e *frameworks* desenvolvidos para as mais diversas aplicações, como: desenvolvimento Web, gráficos, mineração de dados e aprendizado de máquina. Python é conhecida por ser uma linguagem rápida para desenvolver, porém lenta para executar. Isso é válido para código escrito puramente em Python, porém grande parte das bibliotecas de ciência de dados disponíveis para Python não são escritas apenas em Python, mas sim em linguagens de baixo nível que executam de forma mais eficiente. Dessa forma, a biblioteca ou *framework* fornece uma interface na linguagem Python, facilitando o desenvolvimento e a integração, enquanto executam o código com outra linguagem, de forma mais otimizada, assim aproveitando o melhor dos dois mundos. Na sequência, segue uma rápida descrição das bibliotecas e *frameworks* utilizados.

2.2.2 Pandas

O conceito mais importante da biblioteca Pandas é o de *DataFrame*, que consiste em uma estrutura bidimensional com elementos rotulados (COMARELA et al., 2019). Um *DataFrame* é como uma planilha, cada linha representa um dado e cada coluna um atributo.

A vantagem de se utilizar *DataFrames* com Pandas é a flexibilidade e eficiência. Pandas permite importar, converter e combinar dados de diferentes tipos. Além disso, *Pandas* possui integração com outras ferramentas e bibliotecas comumente utilizadas no ecossistema de computação científica e ciência de dados do Python como: Seaborn, Scikit-learn e LightGBM (PANDAS, 2023).

2.2.3 Scikit-learn

Scikit-learn é um *framework* para Python que integra uma ampla gama de algoritmos de aprendizado de máquina estado-da-arte para problemas de média escala de aprendizado supervisionado e não supervisionado (PEDREGOSA et al., 2011). O Scikit-learn disponibiliza métodos para gerar, treinar e avaliar modelos de aprendizado de máquina, utilizando linguagem de alto nível com foco na usabilidade e performance. Neste trabalho, foi utilizada a implementação do Scikit-Learn dos algoritmos *KNN*, Árvore de Decisão e Floresta Aleatória, assim como os métodos de avaliação dos modelos.

2.2.4 LightGBM

LightGBM é um *framework* de *Gradient Boosting* que implementa algoritmos de aprendizado baseados em árvores. LightGBM foi construída para rodar de forma distribuída e eficiente, com menos uso de memória, treinamento mais rápido e suporte a processamento paralelo, permitindo a manipulação de dados de grande escala (KE et al., 2017). Neste trabalho, foi utilizada a implementação do LightGBM para os algoritmos de *Gradient*

Boosting.

2.2.5 Seaborn

Seaborn é uma biblioteca de visualização de dados para Python, ela oferece uma interface em alto nível para gerar gráficos estatísticos, orientada a *DataFrames* (WASKOM, 2021). Seaborn pode ser utilizada em conjunto com código escrito no formato de notebooks, facilitando o processo de análise exploratória, como foi feito no caso deste trabalho.

2.2.6 Notebooks

Notebook, também conhecido como Jupyter Notebook, é uma ferramenta para desenvolvimento de código que permite que a execução seja feita em partes. Desenvolvido para facilitar o compartilhamento e reprodução de análise de dados, o notebook está sendo cada vez mais usado por cientistas que querem manter registro do seu trabalho (SHEN, 2014).

Entre as vantagens de se utilizar o formato de notebooks estão: a facilidade de construir código enquanto o executa, a documentação do processo com *Markdown* entre os blocos e a capacidade de executar apenas certos blocos do código, sem precisar rodar todo o código do começo toda as vezes. Nesse trabalho, todo o código em Python foi desenvolvido no formato de notebooks.

2.2.7 Visualização de Dados com Tableau

Tableau é um software capaz de gerar visualizações, como gráficos e tabelas, e combiná-los no formato de painéis interativos. Com o objetivo de tornar os dados mais acessíveis e compreensíveis, a visualização de dados na forma de painéis, também conhecidos como *Dashboards*, é a ferramenta ideal para muitas empresas analisarem e compartilharem informações (TABLEAU, 2023).

O Tableau permite que sejam desenvolvidos e combinados conjuntos de

painéis, que juntos formam uma história, que pode ser disponibilizada para acesso do usuário usando o próprio Tableau. Neste trabalho, o Tableau é utilizado na etapa da análise exploratória.

2.2.8 Controle de Versionamento com Git e GitHub

Controle de versionamento é um sistema que registra as alterações feitas em um conjunto de arquivos ao longo do tempo, para que seja possível restaurar versões anteriores (CHACON; STRAUB, 2014). O controle de versionamento pode ser implementado com a ferramenta Git e gerenciado pelo GitHub.

Criado por Linus Torvalds, *Git* é um sistema de gerenciamento de versões de código (MOREIRA, 2016). *Git* é uma ferramenta poderosa utilizada para controlar, modificar e unir versões de projetos de software. Git pode ser utilizado de forma local, apenas para o controle, ou pode ser hospedado em um servidor, possibilitando a colaboração com diferentes ambientes de desenvolvimento.

GitHub é uma plataforma de hospedagem de repositórios Git, ele pode ser usado para hospedagem e compartilhamento de código. O GitHub funciona como uma rede social, que permite que usuários acessem e contribuam para os projetos (GITHUB, 2023). Com GitHub é possível integrar o acesso de diferentes usuários ao projeto, ou integrar o acesso de um mesmo usuário em diferentes ambientes de desenvolvimento.

Neste trabalho, Git foi utilizado para fazer o controle de versionamento. O repositório foi hospedado no GitHub, com isso foi possível realizar o desenvolvimento de diferentes ambientes.

3 TRABALHOS RELACIONADOS

Neste Capítulo, são apresentados os principais trabalhos que exploram o uso de ciência de dados e aprendizado de máquina em dados abertos ou governamentais, com o objetivo de prever tendências ou obter *insights* sobre problemas sociais.

3.1 INFERÊNCIA PREDITIVA GEOESPACIAL DA CRIMINALIDADE EM PORTO ALEGRE: UMA ABORDAGEM DE APRENDIZADO DE MÁQUINA

A dissertação de mestrado do PPG da faculdade de Economia da UFRGS de (JONER, 2020) analisa o uso de inteligência artificial, por meio de aprendizado de máquina, como ferramenta de combate à criminalidade. O trabalho publicado em 2020, analisa os dados de crimes violentos letais intencionais que ocorreram de 2005 e 2019 no município de Porto Alegre, os dados foram disponibilizados pelo Observatório da Segurança Pública do Governo do Estado do Rio Grande do Sul.

A proposta de Joner H. é separar a cidade em clusters, utilizando aprendizado não supervisionado com o algoritmo *KMeans*, e então prever a quantidade de crimes que ocorrerão em cada região formada, utilizando aprendizado supervisionado. Os modelos utilizados são algoritmos de regressão, classificação, redes neurais profundas e *long shot-term memory*. Os resultados mostram que todos os modelos têm capacidade de predição.

Os testes foram realizados em diferentes números de *clusters*, a melhor performance foi obtida quando dividiu a cidade em um número menor de clusters, alcançando o coeficiente de determinação de 0.94 quando utilizando apenas 6 clusters, demonstrando que a metodologia possui potencial de prever tendências de forma generalizada, possibilitando otimizar a eficiência das políticas combate ao crime.

3.2 DETERMINANTES E PREDIÇÃO DE CRIMES DE HOMICÍDIOS NO BRASIL: UMA ABORDAGEM DE APRENDIZADO DE MÁQUINA

No trabalho desenvolvido na universidade de São Paulo, (LOPES; FELIX, 2019), é utilizado aprendizado de máquina para prever quantidades de homicídios em cidades Brasileiras. Utilizando aprendizado supervisionado, são testados modelos de regressão com os algoritmos: KNN, Árvore de Regressão, Florestas Aleatórias e *Boosting*. Técnicas de ensemble também foram utilizadas, o ajuste dos hiperparâmetros foi realizado de forma manual.

Os experimentos demostraram que os modelos com melhor desempenho, ou seja, que apresentaram os menores valores de MAE e RMSE, foram Florestas Aleatórias e *Boosting*, com 82% e 78% de variância explicada. O trabalho gera uma tabela de importância de cada fator para o modelo de Florestas Aleatórias, assim utilizando o modelo para analisar quais são os fatores que mais influenciam alterações nas taxas de crime. Por fim, o artigo analisa cada fator obtido com importância maior do que 5%, visível na Tabela 3.1, discutindo quais são as possíveis relações com os índices de homicídio.

Tabela 3.1: Importância de cada Fator no modelo de Florestas Aleatórias.

Fatores	Importância Média
População Jovem	8,99%
Saneamento Básico	8,28%
População Total	8,10%
População Economicamente Ativa	7,36%
População Urbana	7,15%
PIB	6,58%
Mulher Chefe da Família	5,60%
Pessoas Pobres entre 0 a 14 anos	5,30%
Proporção de Pessoas ganham até meio Salário Mínimo	5,23%

Fonte: (LOPES; FELIX, 2019)

3.3 EXPLORANDO APRENDIZAGEM SUPERVISIONADA EM DADOS HETEROGÊNEOS PARA PREDIÇÃO DE CRIMES

A dissertação de mestrado do PPG em informática da Pontifica Universidade Católica de Minas Gerais desenvolvida (CASTRO, 2020), utiliza aprendizado supervisionado para prever tendências de crimes. Neste trabalho, os crimes considerados são apenas furto e roubo.

A autora utiliza duas bases de dados, uma com registros criminais oficiais coletados com a Secretaria de Segurança do Estado de Minas de Gerais e outra com registros não oficiais de um site "Onde Fui Roubado". É feita uma análise de complementaridade entre os dois *datasets*, e as bases são combinadas. A autora realiza uma extensa análise exploratória em ambos os conjuntos de dados,

São cinco algoritmos de aprendizado supervisionado de classificação: *k-Nearest Neighbor* (k-NN), *Support Vector Machine* (SVM), *Random Forest* (RF), *eXtreme Gradient Boosting* (XGBoost) e a rede neural Long Short Term Memory (LSTM). Os modelos se propõem a prever a classe da tendência dos crimes diários em uma região, que pode assumir 3 valores: Aumentar, diminuir ou se manter do dia anterior. Para otimizar os hiperparâmetros, a autora utiliza o *GridSearch*.

Os modelos obtêm boas métricas que variam de 78% a 89%, dependendo da combinação das bases de dados utilizadas nos testes, destaca-se a técnica de LTSM se destaca por obter as métricas relacionadas a precisão. A autora destaca, entre as contribuições do trabalho: desenvolvimento do *crawler*, análise heterogênea e a combinação dos conjuntos de dados e a avaliação das cinco técnicas de aprendizado de máquina.

3.4 Padrões de Concentração Espacial de Roubos de Automóveis em Municípios da Grande João Pessoa a Partir de Técnicas de Aprendizado de Máquinas

O artigo (ANJOS et al., 2020) utiliza técnicas de aprendizado não supervisionado para encontrar forma de mapear o comportamento criminoso

e identificar padrões e tendências da atividade criminosa na região da grande João Pessoa. Os dados fornecidos são 5.385 registros de ocorrência de roubos e furtos de carros e motos, que aconteceram entre 2017 e 2019, disponibilizados aos pesquisadores pela Secretaria de Estado da Segurança e da Defesa Social da Paraíba.

As informações disponíveis para cada crime são apenas a data, o horário e a localização geográfica. A análise exploratória contabiliza o total de ocorrências e plota mapas de calor relacionando o mês, o dia da semana e a hora dos crimes com o número de ocorrências.

Para o aprendizado de máquina, a técnica utilizada é a de clusterização, com o algoritmo DBSCAN. Os clusters são formados utilizando somente a posição geográfica do fato, separando o *dataset* por dia da semana e turno.

3.5 Comparação entre os trabalhos

Joner (2020) e Castro (2020) seguem uma metodologia parecida, que consiste em separar regiões demográficas e tentar predizer as quantidades de crimes e tendências futuras, seja por meio de regressão ou classificação. O trabalho (LOPES; FELIX, 2019) se diferencia por abordar o mesmo problema, porém o objetivo é encontrar quais são os atributos de maior importância para entender as causas de um aumento ou diminuição dos homicídios. Esses três trabalhos têm foco em crimes violentos e predição de tendências com aprendizado supervisionado. O trabalho (ANJOS et al., 2020) é o único que possui foco em roubos e furtos de veículos, ele utiliza algoritmos de agrupamentos para agrupar os crimes por características.

Este trabalho se propõe a analisar o uso de aprendizado de máquina para entender e prever obter insights em subtrações de veículos, diferenciando-se dos trabalhos de (JONER, 2020) e (CASTRO, 2020), pois analisa o uso de aprendizado supervisionado para prever informações respeito da recuperação desses veículos ao invés de prever tendências. Assim, como o trabalho de (LOPES; FELIX, 2019) faz com os modelos de predição de homicídios, esse trabalho se propõe a descobrir e analisar quais são os atributos de maior importância para as subtrações de veículos. Na

parte de aprendizado não supervisionado, o trabalho se assemelha ao de (ANJOS et al., 2020), porém se diferencia por abordar outros aspectos que não foram explorados, como o preço e o tipo dos veículos. A Tabela 3.2 apresenta as principais pontos que contrastam este trabalho com os relacionados.

Tabela 3.2: Comparação entre os trabalhos

	(JONER, 2020)	**(LOPES; FELIX, 2019)**	**(CASTRO, 2020)**	**(ANJOS et al., 2020)**	**Este trabalho**
Aprendizado Supervisionado	**X**	**X**	**X**		**X**
Aprendizado Não Supervisionado	**X**			**X**	**X**
Importância dos atributos		**X**			**X**
Foco em subtrações de veículos				**X**	**X**

Fonte: Elaborado pelo autor.

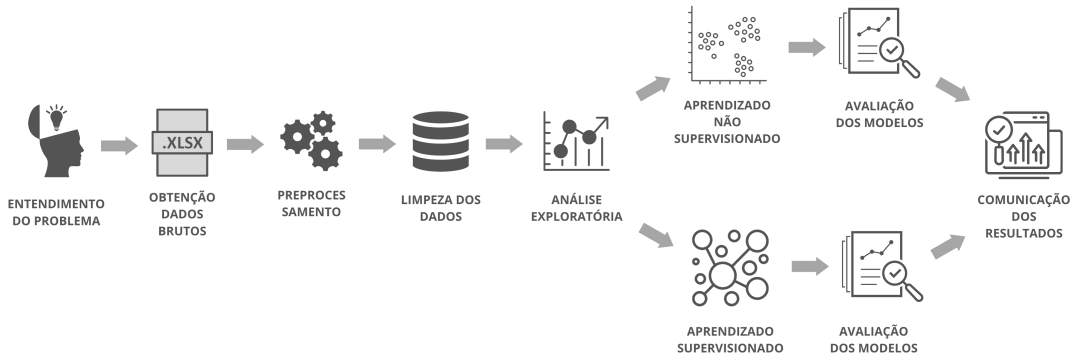
4 METODOLOGIA

O objetivo deste trabalho é realizar experimentos com algoritmos de aprendizado supervisionado e não supervisionado como forma de entender padrões e obter *insights* em segurança pública. Este capítulo descreve a metodologia proposta para abordar o problema, obter os dados, realizar os experimentos e analisar os resultados.

4.1 Visão Geral

A metodologia é baseada nas etapas do *Data Analytics Lifecycle* (EMC, 2015), o fluxograma da Figura 4.1 fornece um panorama geral do processo. O primeiro passo é entender o problema, e então obter os dados, que passam pelo pré-processamento e limpeza e, em seguida, é feita a análise exploratória. Depois da análise exploratória, é possível optar por diferentes estratégias de aprendizado de máquina. Para este trabalho, que analisa ocorrências de roubos e furtos de veículos, optou-se por explorar modelos de aprendizado supervisionado e não supervisionado. Na sequência, cada etapa é explicada e comentada, justificando as escolhas feitas.

Figura 4.1: Visão geral da metodologia do trabalho.



Fonte: Elaborado pelo autor

4.2 Entendimento do Problema

Segundo o Data Analytics Lifecycle (EMC, 2015), nesta etapa os esforços se concentram em aprender e investigar o problema, desenvolver o contexto e entendimento sobre as fontes de dados que estão disponíveis para o projeto. Para isso, é possível consultar fontes de dados públicos ou órgãos públicos para entender como os dados podem ser utilizados. Também é possível definir esta etapa como ”descobrimento” ao invés de ”entendimento” do problema.

Nesta etapa, também são definidos quem são as partes interessadas no projeto, também chamados de *stakeholders*. É fundamental identificar as partes interessadas, pois com elas é possível obter mais informações sobre o domínio do problema e identificar as metas e pontos chave para o desenvolvimento do projeto.

Neste trabalho, o problema explorado foram os crimes de roubo e furto de veículos no município de Porto Alegre. A principal parte interessada é o Comando de Inteligência da Brigada Militar, órgão público ao qual a proposta do trabalho foi apresentada.

4.3 Obtenção dos Dados

Após definir a motivação do trabalho e entender o problema, é necessário buscar os dados que alimentarão as análises. Caso os dados estejam disponíveis por transparência ativa, estão disponíveis para download em um site do governo. Caso o dado seja caracterizado como transparência passiva, é necessário solicitar ao órgão governamental que detém o acesso. Após o tempo de resposta, os dados para o projeto são enviados, normalmente em formatos de planilhas, ou, se for o caso, é fornecido acesso ao banco de dados.

Segundo a Lei de Acesso à Informação (WIKILAI, 2023), existem dois tipos de transparência.

- Transparência ativa - ocorre quando o dado público é divulgado pelo órgão de forma aberta.

- Transparência passiva - ocorre quando o cidadão solicita acesso a informações que não estão disponíveis em páginas oficiais de órgãos públicos por transparência ativa.

Nesta etapa, também é feita a verificação do dicionário de dados, geralmente disponibilizado para *download* junto aos dados no caso de transparência ativa. No caso da transparência passiva, pode ser que ele não exista. Caso não exista dicionário, ele pode ser construído com base nas informações fornecidas com o *dataset* e, se necessário, consultando o próprio fornecedor ou parte interessada.

Neste trabalho, os dados não estavam disponíveis por transparência ativa, por isso foi necessário solicita-los para a brigada militar. Os dados são referentes a subtrações de veículos no estado do Rio Grande do Sul, no período de janeiro de 2018 a dezembro 2022. Os dados foram extraídos do banco de dados governamental e enviados no formato de planilhas, sem incluir dicionário de dados.

4.4 Pré-processamento

A etapa de pré-processamento tem como objetivo preparar os dados para que possam ser utilizados nos modelos. Caso os dados sejam fornecidos no formato de planilhas, é necessário converter as planilhas em *dataframes*, e combiná-las, caso necessário. Nesta etapa também podem ocorrer as transformações nas *features*, como, por exemplo, decompor o atributo de data: em ano, mês e dia.

Neste trabalho, o pré-processamento começa importando os dados em *dataframes* e convertendo os tipos dos atributos. Depois, um *script* é executado para aferir o tipo e o preço de mercado dos veículos da base, adicionando assim mais informações que serão utilizadas nas próximas etapas.

4.5 Limpeza

A etapa de limpeza dos dados inclui realizar o tratamento de valores faltantes, caso não sejam considerados significativos podem ser removidos. Alguns modelos possuem estratégias para tratar valores faltantes, deixá-los também pode ser uma opção. Também é necessário realizar as padronizações, todos os atributos textuais precisam padronizar a codificação e mesma formatação. Neste trabalho, a limpeza consiste em colocar todos os atributos em maiúsculo, sem pontuações, e sem acentos, e reduzir atributos categóricos com muitos valores diferentes para apenas os mais frequentes.

Segundo o *Data Analytics Lifecycle*, a etapa de preparação dos dados - que inclui o pré-processamento e a limpeza - é uma das etapas mais custosas, muitas vezes tomando 50% do tempo do projeto (EMC, 2015). Ainda segundo o livro, nesta fase ocorre o primeiro contato com a base de dados, esse é um momento crítico pois é essencial se familiarizar com os dados e prepará-los da forma correta para que sirvam de base para as próximas etapas.

O pré-processamento e a limpeza foram realizados com a linguagem de programação Python, no formato de *notebooks*. A biblioteca Pandas foi usada para importar as tabelas fornecidas para *dataframes*.

4.6 Análise Exploratória

A análise exploratória é a etapa na qual são verificadas as relações e o comportamento das variáveis, para entender o domínio do problema e criar as primeiras hipóteses. A análise exploratória é como um trabalho de detetive (TUKEY, 1977), pois é um importante passo para entender os dados e serve como base para todo o resto do projeto.

Para a análise, foram gerados diversos gráficos verificando a relação dos atributos mais relevantes com a taxa de recuperação e outras possíveis correlações. Para realizar essa etapa, os dados, depois de preparados, foram importados em um *notebook Python*. A biblioteca Pandas foi utilizada para manipular os *data frames*, e a biblioteca Seaborn para gerar os gráficos.

A ferramenta de análise de dados Tableau foi utilizada para gerar *Dashboards* complementares a análise exploratória. O objetivo é que esses

painéis sejam utilizados para que o próprio cliente possa participar, de alguma forma, do processo de análise exploratória.

4.7 Aprendizado Não Supervisionado

O objetivo do aprendizado não supervisionado é encontrar padrões em dados sem rótulo (AMIDI; AMIDI, 2018). Nesta etapa, o aprendizado supervisionado é utilizado para explorar os dados e tentar encontrar padrões que não são facilmente percebidos.

O algoritmo de aprendizado não supervisionado aplicado neste trabalho é o de clusterização com o K-Médias. Antes de executar o algoritmo, é necessário selecionar as features que serão analisadas e normaliza-las. Para analisar os resultados dos clusters gerados pelo K-Médias e encontrar o valor ideal de K é utilizado o método do cotovelo.

Após a geração dos *clusters*, são verificados se eles expressam alguma característica ou tornam algum comportamento visível. Diferentes combinações de variáveis podem ser testadas, para investigar as hipóteses e tentar obter *insights*.

Neste trabalho, as visualizações dos *clusters* foram geradas com a biblioteca Seaborn. Posteriormente, para tornar a análise mais interativa a facilitar a exploração, os agrupamentos foram inseridos em *dashboards* construídos com a ferramenta Tableau.

4.8 Aprendizado Supervisionado

A etapa de aprendizado supervisionado consiste em construir um modelo que identifique os padrões que não são perceptíveis pelo ser humano para, então, prever resultados baseados nos atributos do conjunto de dados. Esse modelo é treinado com exemplos rotulados, e então tenta prever os rótulos de objetos novos não rotulados, com o objetivo de prever corretamente o valor de cada um (MONARD; BARANAUSKAS, 2003).

Existem dois tipos de modelo de aprendizado supervisionado:

- Classificação - cujo objetivo é prever a qual classe pertence um dado.

- Regressão - cujo objetivo é prever um valor contínuo.

É importante entender como os atributos e as características de um dado influenciam o comportamento do modelo. Essa interpretabilidade é importante pois contribui para a colaboração entre os humanos e a IA (MICROSOFT, 2023). Existem modelos que são mais facilmente interpretados, como, por exemplo, as árvores de decisão, já outros, como as redes neurais, não são tão simples de se entender. Logo, é necessário levar em consideração a interpretabilidade do modelo ao fazer a escolha, pois existe potencial de contribuição.

Esse trabalho se propõe a analisar o desempenho de diferentes algoritmos. No caso dos modelos de classificação, é comum analisar a matriz de confusão e as principais métricas como acurácia, precisão, sensibilidade, especificidade e revocação. No caso de modelos de regressão, as métricas utilizadas são MAE e MRSE. Neste trabalho, foram utilizados os seguintes algoritmos para geração dos modelos: KNN, Árvore de Decisão, Floresta de Decisão e *Gradient Boosting*.

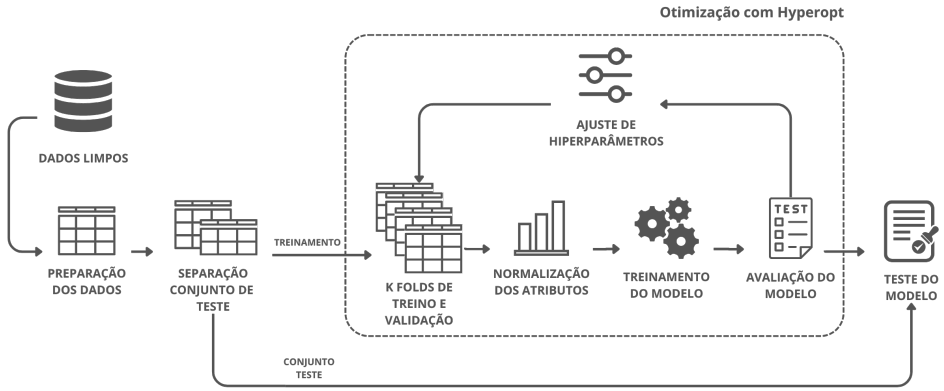
O processo de treinamento dos modelos pode ser visualizado no fluxograma da Figura 4.2. Para esse trabalho, foram realizados dois modelos de aprendizado supervisionado, utilizando a mesma metodologia, ambos modelos de classificação. O primeiro modelo tem o objetivo de prever se o veículo subtraído será recuperado ou não, e o segundo prevê para onde esse veículo será levado.

Antes de separar os dados em treino e validação, é preciso separar parte dos dados para realizar teste do modelo. Os dados do conjunto de teste não são utilizados em nenhuma parte do treinamento ou otimização de hiperparâmetros, pois o objetivo é verificar como o modelo se comporta com dados que inéditos. Depois de separar o conjunto de teste, para realizar a separação dos conjuntos de treino e validação, foi utilizada a validação cruzada com K-Folding.

A normalização dos atributos é realizada de forma isolada no treinamento, os dados do conjunto de treinamento não utilizam os dados do conjunto de validação para normalizar os valores, pois isso pode ser uma forma de alimentar o modelo com informações que ele não teria acesso. Os dados do conjunto de validação utilizam os valores do conjunto de

treinamento para as normalizações, pois são informações que o modelo sabe antes de realizar a predição.

Figura 4.2: Etapas para geração dos modelos de aprendizado supervisionado.



Fonte: Elaborado pelo autor.

4.8.1 Validação Cruzada

Para evitar o *overfiting* e melhorar a confiança nas métricas do modelo gerado pode ser utilizada a validação cruzada. A validação cruzada pode ser implementada com os métodos K-Folds, dentre eles é possível escolher: o *KFold* Estratificado, que é uma boa opção para equilibrar as classes entre os *Folds*, ou o *TimeSeriesSplit*, que pode ser a melhor opção por aproximar o cenário real quando os dados representam eventos cronológicos.

Neste trabalho, o método de *KFolding* escolhido foi o Estratificado, com *ShuffleSplit*, pois o conjunto de validação possui desbalanceamento entre as classes preditas, e o K-Fold estratificado é capaz de manter esse desbalanceamento constante entre os conjuntos de validação.

4.8.2 Otimização dos Hiperparâmetros

Após o modelo ser avaliado, ele pode ter seus hiperparâmetros ajustados, com o objetivo de tornar o algoritmo mais adaptável aos dados, assim melhorando a performance do modelo. Existem técnicas que testam combinações de valores de hiperparâmetros, de forma exaustiva ou aleatória, como o *Grid Search* e o *Random Search*. Existem alternativas de técnicas que implementam heurísticas para procurar em um espaço de busca pela melhor combinação dos hiperparâmetros, para utilizá-las é necessário especificar uma métrica para otimizar, como por exemplo a acurácia balanceada, e então os valores para os parâmetros que serão verificados.

A otimização dos hiperparâmetros foi feita com a biblioteca *Hyperopt-sklearn*, ela implementa uma busca heurística para aproximar a melhor combinação de valores para os parâmetros. Os testes são executados comparando os modelos antes e depois da otimização.

4.8.3 Teste do Modelo

Por fim, após serem realizados os ajustes dos hiperparâmetros e coletadas as métricas, é feito teste do modelo com os dados que foram previamente separados. Esses dados foram selecionados antes das etapas de validação cruzada e ajuste de hiperparâmetros. É possível que o desempenho seja um pouco inferior, por serem dados nunca antes vistos pelo modelo, porém é fundamental testar que o modelo funciona com dados inéditos.

Em ambos os modelos de aprendizado supervisionado o conjunto de teste é equivalente a 10% dos dados. O conjunto de teste foi separado de forma cronológica, ou seja, os dados retirados são dos crimes que ocorreram nos últimos 6 meses do conjunto, para que não ocorra vazamento de informação entre as etapas de treinamento e teste.

4.8.4 Avaliação dos Resultados

Para avaliar os resultados obtidos nos modelos de aprendizado supervisionado compara-se as métricas obtidas pelo modelo, verifica-se quais desempenharam dentro e fora do esperado. É importante analisar a complexidade do modelo, verificando se é factível a sua implementação. Também pode se considerar como utilizar a interpretabilidade dos modelos gerados para obter mais informações e *insights* a respeito dos dados e do problema.

5 EXPERIMENTOS E RESULTADOS

Este capítulo relata o resultado da aplicação da metodologia proposta no capítulo 4. A metodologia é aplicada no problema das ocorrências criminais de subtrações e recuperações de veículos no estado do Rio Grande do Sul.

5.1 Entendimento do Problema

A motivação do trabalho vem da publicação dos registros de ocorrência disponibilizados pela SSPRS. Com o objetivo de entender o problema e se existe a possibilidade de desenvolver modelos de aprendizado de máquina em cima desses dados, foi feita uma reunião com o comando de inteligência da brigada militar, onde foi apresentado o projeto e a proposta de cooperação. A ideia foi bem recebida, com a sugestão envolver no estudo apenas uma categoria de crimes: roubo e furto de veículos. Os dados foram solicitados diretamente para a brigada militar, por intermédio do tenente-coronel Roberto Donato.

5.2 Obtenção dos Dados

A base de dados foi entregue no formato de planilha *xlsx*. Cada linha na tabela registra um fato envolvendo um veículo, que pode ser um furto, um roubo ou uma recuperação. Não foi fornecido dicionário de dados, então foi criado um com base nas informações fornecidas, representado na Tabela 5.1. Nota-se que o mesmo veículo pode ser subtraído e recuperado diversas vezes, a única forma de identificá-lo é pelo identificador único fornecido.

A base fornecida pelo órgão público possui atributos com informações redundantes, como por exemplo as colunas “UF” e “Nome Pais”. Existem também informações que não são úteis para a abordagem escolhida, como por exemplo a coluna “Unidade”, que indica em qual batalhão da polícia militar foi feito o registro, essa informação pode ser útil para outras análises, mas foge do escopo da abordagem escolhida. Todos esses atributos foram descartados logo na primeira etapa do pré-processamento, e não entram para o dicionário

Tabela 5.1: Dicionário dos dados brutos.

Nome da Coluna	**Descrição**	**Tipo**
Data/Hora Fato	Data da ocorrência, com horário, dia, mês e e ano.	Numérico ordinal
Nome Rua Nro	Logradouro e número onde ocorreu o fato	Categórico nominal
Bairro	Bairro onde ocorreu o fato.	Categórico nominal
Município	Município onde ocorreu o fato.	Categórico nominal
Fato Abreviado	Se o crime é considerado furto ou roubo	Categórico nominal
Tipo Fato	Se o registro corresponde a uma tentativa ou um crime consumado	Categórico nominal
Latitude	Latitude da posição geográfica onde ocorreu o fato.	Numérico cardinal
Longitude	Longitude da posição geográfica onde ocorreu o fato.	Textual categórico
Tipo veículo	Tipo de veículo envolvido, não padronizado. Exemplo: Carro, motocicleta	Textual categórico
Ano Fabricação	Ano de fabricação do veículo envolvido.	Numérico ordinal
Ano Modelo	Ano do modelo do veículo envolvido.	Numérico ordinal
Cor	Cor do veículo envolvido.	Textual categórico
Marca	Marca e modelo do veículo envolvido.	Textual categórico
Identificador veículo	Identificador único de veículo.	Numérico categórico

Fonte: Elaborado pelo autor.

de dados brutos da Tabela 5.1.

5.3 Pré-processamento

O pré-processamento foi feito utilizando a linguagem de programação Python, com a biblioteca pandas. Todas as tabelas foram importadas no formato de *data frames*.

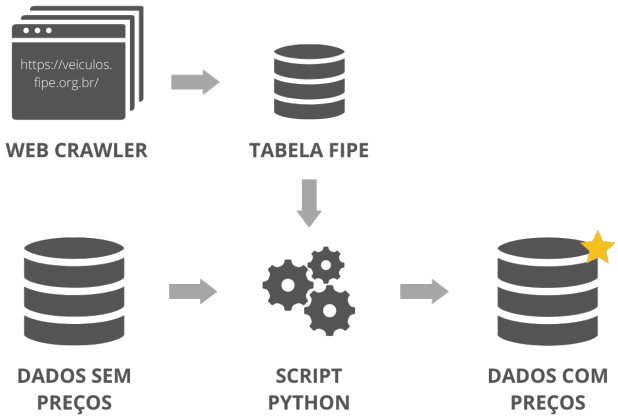
Os registros das subtrações e das recuperações não estão agregados, então o primeiro passo é descobrir se um veículo envolvido em um roubo ou furto foi recuperado posteriormente. Para essa aferição, é necessário separar as ocorrências em dois *dataframes*, um para os roubos e furtos e outro para as recuperações. Em seguida, ambos os *dataframes* são cruzados, buscando pelo identificador único de cada veículo e comparando as datas, se ocorreu algum registro de recuperação após a data da subtração, então esse veículo é registrado como recuperado. Se não encontra, é marcado como não recuperado. Nesta etapa registra também a latitude, longitude e data dessa recuperação.

Uma das principais características a se considerar é o valor dos veículos envolvidos nas ocorrências, então, antes de qualquer análise, percebeu-se que poderia ser um atributo de interesse. Essa informação não consta na base de dados, mas é possível estimá-la a partir da marca, modelo e ano, informações que existem na base. A Tabela Fipe registra os preços médios dos veículos do mercado nacional (ECONÔMICAS, 2023), ela foi usada como referência para estimar os preços.

Para adicionar essa informação à base, foi necessário criar um algoritmo que afere o valor de cada registro dos veículos na base. Tendo em vista que a Fipe não fornece serviço de *api* ou base de dados para *download*, foi utilizado um *web scraper* para consultar e armazenar todos os modelos e valores disponíveis na Tabela. O fluxograma da Figura 5.1 explica o processo. O mês de referência utilizado é dezembro de 2022.

Infelizmente, a informação de modelo dos veículos é registrada na base de dados de forma não padronizada, não permitindo consultas diretas ao *data frame* da Tabela Fipe. Para resolver esse problema foi criado um *script* em *Python* que acessa cada registro do *data frame* de subtrações, comparando todas as palavras contidas no campo marca com os registros de modelo da Fipe. Em cada registro de ocorrência, cada palavra do campo é verificada

Figura 5.1: Fluxograma do processo de estimação dos preços.



Fonte: Elaborado pelo autor.

se está contida no campo “Modelo” da Tabela Fipe, caso esteja é adicionado 1 ponto ao contador de similaridade desse modelo na Tabela Fipe. O mesmo *script* também adiciona 1 ponto caso o ano do modelo do veículo da ocorrência seja igual ao ano do modelo da Fipe.

Após o *script* comparar todas as palavras do campo “Marca” do *data frame* das subtrações com toda a Tabela Fipe, é contabilizado qual ou quais veículos obtiveram a maior pontuação de similaridade. Caso exista mais de um modelo com a maior pontuação, o valor computado é a média entre eles. Além da aproximação do valor, esse *script* também é utilizado para registrar o tipo de veículo, que é um atributo categórico que representa se o automóvel é um carro, moto ou caminhão.

5.4 Limpeza

Os dados contêm 100.903 crimes consumados em todo o estado do Rio Grande do Sul, desse total existem muitos fatos com atributos importantes faltando, existem também *outliers* e valores absurdos, como “6020” no ano modelo do veículo, todos esses dados foram removidos da base. Também foi feita uma análise manual para verificar se todas as ocorrências registradas em Porto Alegre de fato ocorreram dentro dos limites do município. Também

foram removidas instâncias que não conseguiram uma boa estimativa no processo de precificação (qualquer uma com pontuação menor do que 3 no atributo “contagem”). Sobraram 93.576 dados relativos a todo o estado, filtrando apenas a capital gaúcha restam 30.837 registros.

Os atributos categóricos foram reduzidos da seguinte forma:

- “Fabricante”: Mantidas apenas as 10 mais comuns, as outras recebem o valor ’OUTRA’;
- “Cor”: Mantidas apenas as 7 mais comuns, as outras recebem o valor ’OUTRA’;
- “Bairro”: Mantidos apenas os 20 mais comuns, os outros recebem o valor ’OUTRO’;

Na Tabela 5.2, segue o dicionário dos dados limpos, que são utilizados na análise exploratória e nos algoritmos de aprendizado de máquina.

Tabela 5.2: Dicionário dos dados limpos.

Nome da Coluna	**Descrição**	**Tipo**
Fato	Se o crime é considerado furto ou roubo	Categórico nominal
Data_Hora_fato	Data da ocorrência, com horário, dia, mês e e ano.	Numérico ordinal
Municipio	Município onde ocorreu o fato.	Categórico nominal
Bairro	Bairro onde ocorreu o fato.	Categórico nominal
Latitude	Posição geográfica onde ocorreu a subtração.	Numérico cardinal
Longitude	Posição geográfica onde ocorreu a subtração.	Textual categórico
Tipo_modelo	Tipo de veículo envolvido, padronizado conforme Tabela Fipe Exemplo: Carro, motocicleta	Textual categórico
Ano_modelo	Ano do modelo do veículo envolvido.	Numérico ordinal
Cor	Cor do veículo envolvido.	Textual categórico
Fabricante	Fabricante do veículo envolvido.	Textual categórico
Identificador veículo	Identificador único de veículo.	Numérico categórico
Valor	Valor do veículo envolvido	Numérico Ordinal
Recuperado	Registro se o veículo foi ou não recuperado, 0 ou 1	Categórico Ordinal
Municipio Rec	Municipio onde o veículo foi recuperado	Categórico nominal
Latitude Rec	Posição geográfica onde o veículo foi recuperado.	Numérico contínuo
Longitude Rec	Posição geográfica onde o veículo foi recuperado.	Numérico cardinal

Fonte: Elaborado pelo autor.

5.5 Análise Exploratória

A primeira observação feita diz respeito ao balanceamento do *dataset*. Na Tabela 5.3, é possível observar a taxa de recuperação de veículos nas 10 cidades do estado com maior número absoluto de subtrações. Nota-se que algumas cidades fogem do padrão, como Passo Fundo onde mais de 70% dos carros roubados são recuperados, mas no geral o percentual de recuperações se mantém relativamente balanceado. Na capital gaúcha, que é o foco deste trabalho, dos 30.837 carros roubados, 16.907 foram recuperados, resultando em um índice de 54,8% de recuperação.

Tabela 5.3: Quantidade de crimes e recuperações de veículos em municípios do Rio Grande do Sul.

Município	Total de Subtrações	Recuperados
PORTO ALEGRE	30.837	54,83%
CAXIAS DO SUL	6.096	59,53%
CANOAS	4.730	52,79%
NOVO HAMBURGO	4.316	58,20%
SÃO LEOPOLDO	3.982	59,84%
VIAMÃO	3.950	60,15%
ALVORADA	3.918	58,27%
GRAVATAÍ	3.104	50,55%
PASSO FUNDO	1.906	70,93%
PELOTAS	1.900	64,58%

Fonte: Elaborado pelo autor.

Em relação ao tipo de crime, observa-se que roubos de veículos possuem uma taxa de recuperação muito maior do que furtos. Na Tabela 5.4, é possível verificar a relação entre tipo de crime com os índices de recuperação.

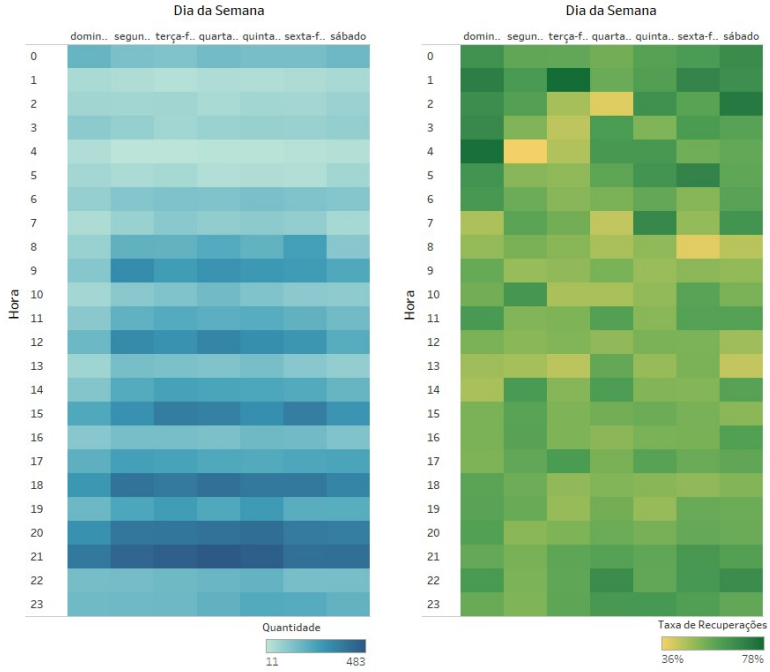
Tabela 5.4: Quantidade de crimes e recuperações em Porto Alegre por tipo de crime.

Tipo de Crime	Total de Subtrações	Recuperados
ROUBO DE VEÍCULO	19.429	60,09%
FURTO DE VEÍCULO	11.408	45,86%

Fonte: Elaborado pelo autor.

Também é possível observar os horários e os dias da semana que esses crimes acontecem. A maior parte dos crimes acontece no período do fim da tarde e noite, sendo quarta-feira as 21 horas o horário com maior número de ocorrências.

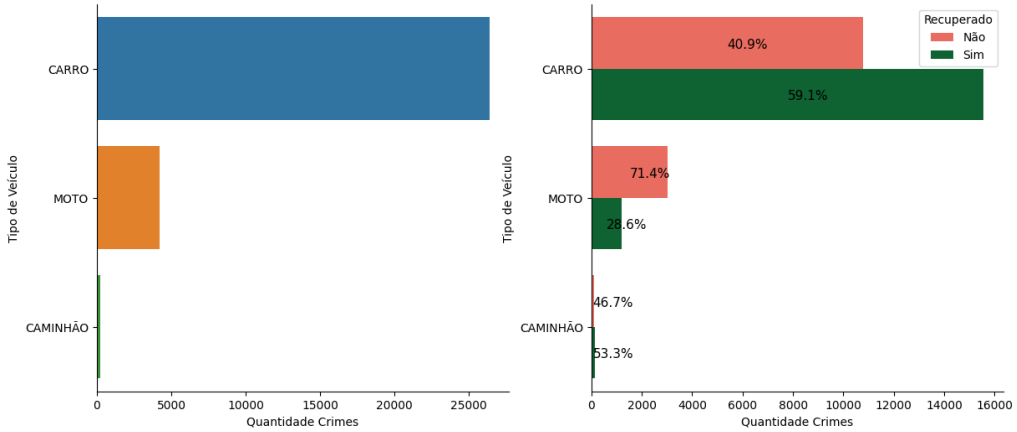
Figura 5.2: Quantidade de crimes e taxas de recuperação por horário e dia da semana.



Fonte: Elaborado pelo autor.

O tipo de veículo também é um diferencial, carros são consideravelmente mais encontrados do que motocicletas, 60% dos carros subtraídos são recuperados, em contraste, 70% das motos não são encontradas. Caminhões são muito pouco subtraídos. É possível verificar os valores na Figura 5.3.

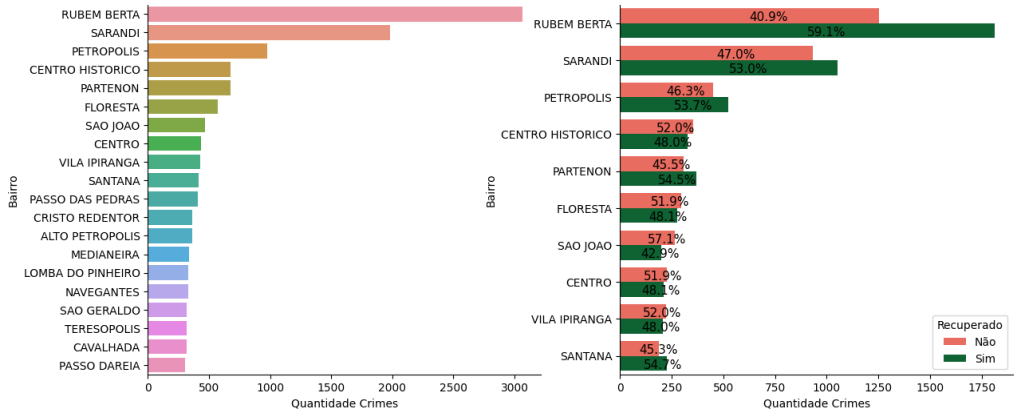
Figura 5.3: Quantidade de crimes e recuperações agrupados por tipo de veículo.



Fonte: Elaborado pelo autor.

Observando a quantidade de crimes por cada bairro de Porto Alegre, é possível notar que alguns bairros, como Rubem Berta, possuem uma taxa de recuperação de veículos superior, enquanto outros, como Centro Histórico, São João e Flores possuem uma taxa de recuperação muito menor. Isso pode estar relacionado a possíveis rotas de fuga e destinos que o transgressor utilizou após cometer o delito. Na Figura 5.4 estão a quantidade absoluta de ocorrências e os índices de recuperação nos 10 bairros com maior quantidade de subtrações.

Figura 5.4: Quantidade de crimes e recuperações agrupados por bairros de Porto Alegre.

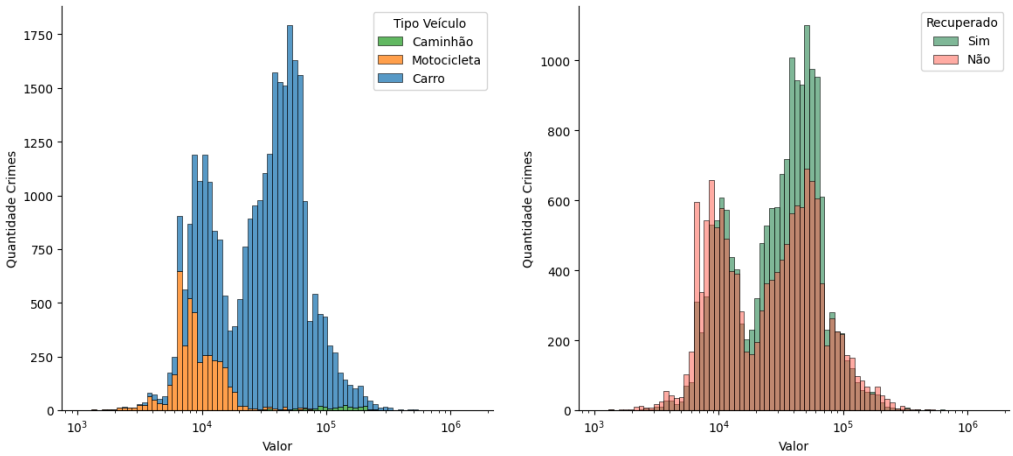


Fonte: Elaborado pelo autor.

Analisando o valor dos veículos subtraídos, percebe-se que a taxa de recuperação é menor em veículos de até 10 mil reais, enquanto veículos de

10 a 100 mil reais são os mais recuperados. Essa taxa pode estar relacionada com o fato de veículos de menores valores normalmente são motocicletas, que possuem menor taxa de recuperação do que carros e caminhões. Veículos acima de 100 mil reais, carros de luxo, possuem menos chances de serem recuperados. Na Figura 5.5, é possível observar na esquerda o histograma com a quantidade de veículos de cada valor, colorido conforme tipo de veículo, e a direita o histograma com a quantidade de veículos, colorido conforme recuperado ou não.

Figura 5.5: Histogramas dos valores e recuperações de veículos em Porto Alegre, escala logarítmica.

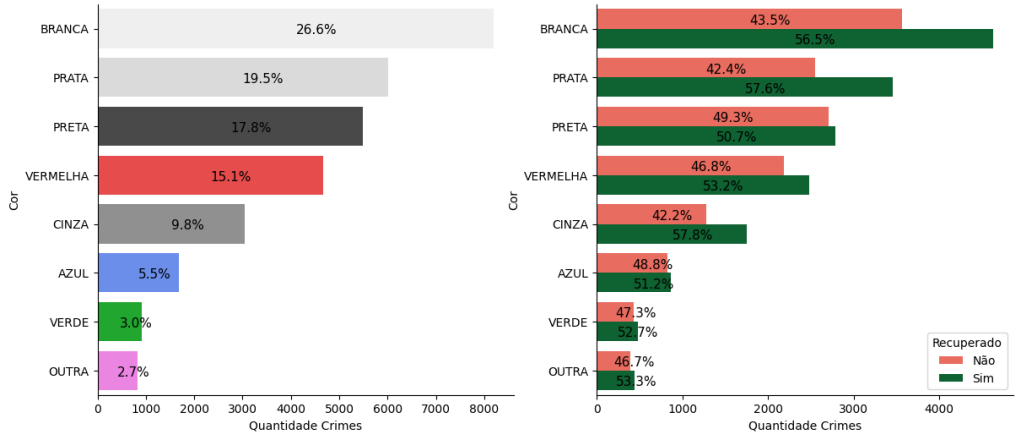


Fonte: Elaborado pelo autor.

Uma hipótese verificada nesta etapa foi a seguinte: carros com cores chamativas e incomuns tendem a ter uma taxa de recuperação maior, pois é mais fácil identificá-los visualmente. Plotando a taxa de recuperação agrupada pelas cores dos veículos, verifica-se que esse comportamento aparentemente não se confirma na cidade de Porto Alegre. Inclusive, em alguns casos, automóveis com cores mais raras possuem menores índices de recuperação.

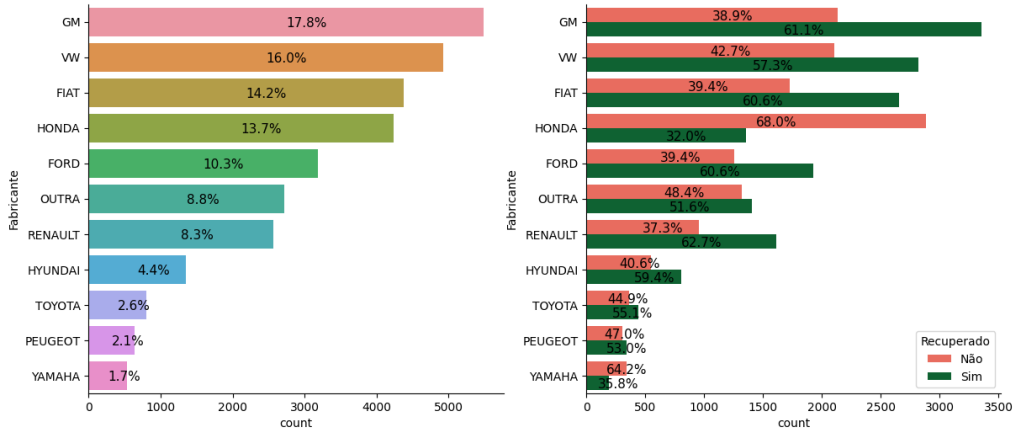
Observando as 10 marcas mais comuns no *dataset*, as duas marcas com menor taxa de recuperação de veículos são as únicas fabricantes de motocicletas, Honda e Yamaha. Todas as outras marcas de automóveis possuem índice de recuperação maior do que 50%. Existe a hipótese de que a marca do veículo tem relação com a probabilidade de recuperação, levando em consideração que existem diferentes redes de desmanche e venda das peças desses veículos.

Figura 5.6: Quantidade de crimes e recuperações de veículos em Porto Alegre, por cor.



Fonte: Elaborado pelo autor.

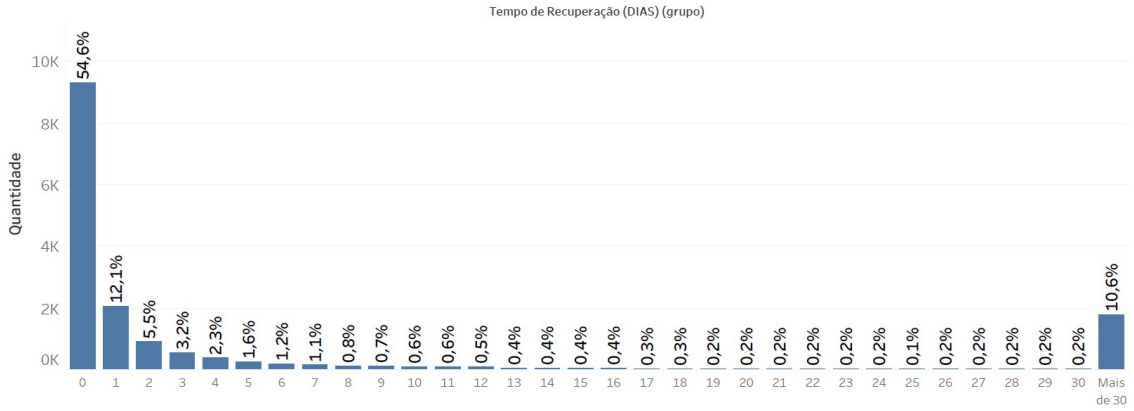
Figura 5.7: Quantidade de crimes e recuperações de veículos em Porto Alegre, por marca.



Fonte: Elaborado pelo autor.

Outro atributo importante a se observar é o tempo que leva até um veículo subtraído ser recuperado. Para descobrir esse valor, foi aplicada uma função que calcula a diferença de tempo do registro do crime e do registro da recuperação. Na Figura 5.8, nota-se que a grande maioria das recuperações acontece em um curto intervalo após a subtração, sendo que mais da metade foram recuperados em menos de 24 horas, e 90% recuperados no período de até 30 dias do acontecimento do crime.

Figura 5.8: Histograma do tempo de recuperação dos veículos subtraídos em Porto Alegre.



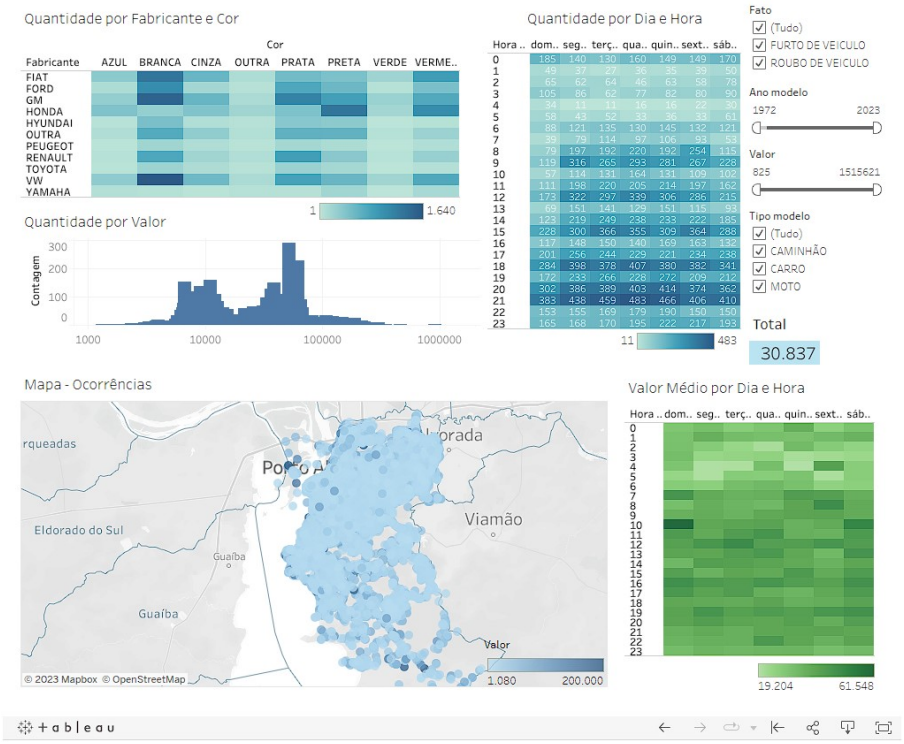
Fonte: Elaborado pelo autor.

5.5.1 Painéis

Os gráficos estáticos gerados com a biblioteca Seaborn e Tableau cumpriram seu papel na análise exploratória, porém são ferramentas pouco interativas. Para aumentar a interatividade e explorabilidade da análise exploratória, foram desenvolvidos painéis de dados, também conhecidos como *dashboards*, usando a ferramenta Tableau, combinando as visualizações geradas e adicionando filtros.

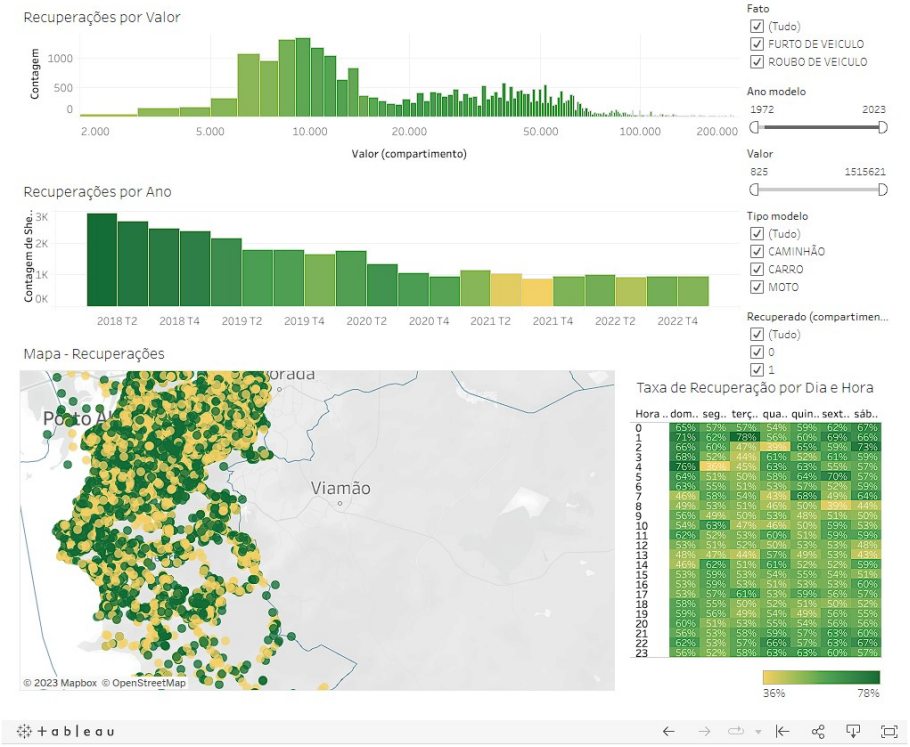
Foram gerados dois painéis, o primeiro aborda o comportamento dos atributos por número total de subtrações e o segundo demonstra o comportamento das variáveis em relação às taxas de recuperação dos veículos. Uma captura de tela de cada painel pode ser observada nas Figuras 5.9 e 5.10. Esses painéis foram desenvolvidos para consumo interno da Brigada Militar.

Figura 5.9: Painel quantidade de subtrações de veículos em Porto Alegre.



Fonte: Elaborado pelo autor.

Figura 5.10: Painel recuperações de veículos em Porto Alegre.



Fonte: Elaborado pelo autor.

5.5.2 Avaliação da Análise Exploratória

Com a análise exploratória percebe-se que diversos atributos podem ter relação com a as chances de um carro ser ou não recuperado, desde o local onde foi roubado a sua cor, demonstrando assim um cenário promissor para utilização de modelos de aprendizado de máquina. Na sequência, são investigado como esses dados se comportam quando são aplicados modelos de aprendizado supervisionados e não supervisionados, para verificar se essas hipóteses a respeito das características são identificadas.

5.6 Aprendizado Não Supervisionado

Após a análise exploratória, algumas questões foram levantadas: É possível identificar características do modo de operação dos agentes responsáveis por cometer esses delitos, ou nos agentes responsáveis pelas recuperações? Se sim, é possível identificar e agrupar os crimes de forma que se consiga isolar algum perfil ou comportamento? Para responder essa pergunta, algoritmos de agrupamentos podem ser utilizados para tentar isolar e identificar esses padrões.

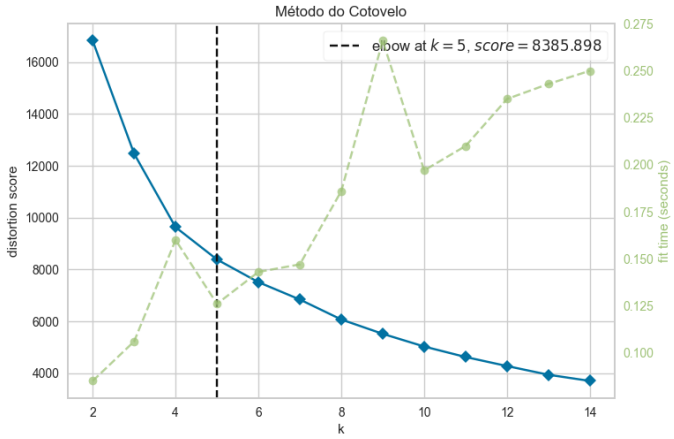
Para esse trabalho foi utilizado o algoritmo *KMeans*, implementado pela biblioteca Scikit-learn. Os dados, pós limpeza, foram normalizados, tendo em vista que o cálculo do algoritmo *KMeans* fica enviesado quando existem diferentes escalas entre as *features*. Os dados utilizados no primeiro agrupamento são “Valor”, “Latitude”, “Longitude”, “Hora”, “Recuperado” e “Fato”, foram normalizados da seguinte forma:

- Valor, foi normalizado utilizando escala logarítmica com o transformador PowerFit, disponibilizado pela biblioteca scikit-learn;
- Latitude e Longitude foram normalizadas utilizando o transformador *MinMax*, também disponibilizado pela scikit-learn;
- Hora foi mapeado para dois atributos: “Hora_sin” e “Hora_cos”, dessa forma o atributo terá um comportamento circular.
- Recuperado mapeado para 0 e 1 quando corresponde a “não recuperado” ou “recuperado”, respectivamente;
- Fato mapeado para 0 e 1 quando corresponde a “furto” ou “roubo”, respectivamente;

Para encontrar o número ideal de agrupamentos foi utilizado o método do cotovelo, disponibilizado pela biblioteca *Yellowbrick*. O número encontrado para o primeiro conjunto de agrupamentos foi 5. É possível verificar o gráfico e o cotovelo da curva na Figura 5.11.

Na Figura 5.12, é possível observar quais foram os *clusters* gerados. Infelizmente o agrupamento se concentrou nos atributos binários que correspondem ao tipo de crime e recuperação dos veículos. Esse resultado não é satisfatório, pois o algoritmo não levou em conta os atributos de valor,

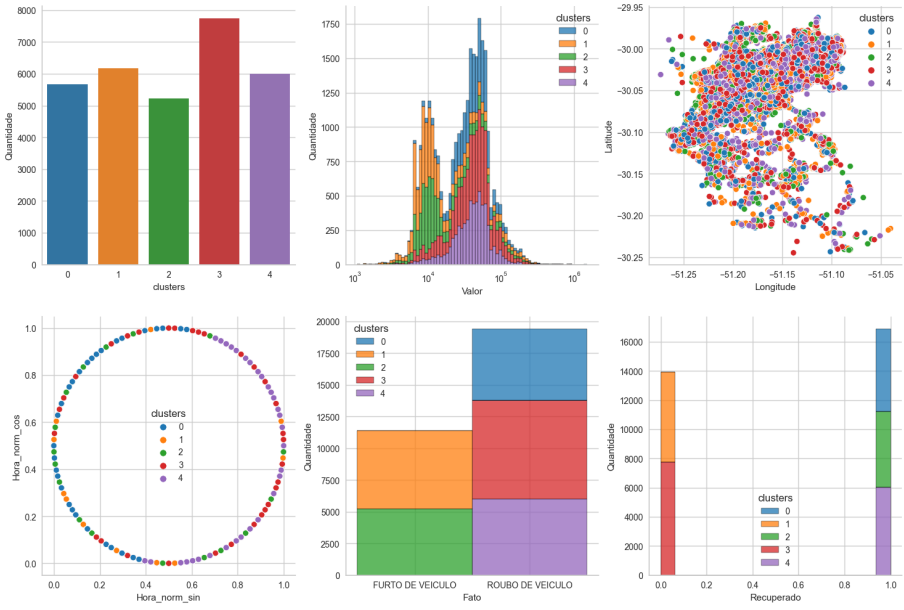
Figura 5.11: Método do cotovelo para o primeiro agrupamento.



Fonte: Elaborado pelo autor.

localização geográfica e hora do dia. Para tentar tornar o resultado desse modelo mais significativo, foi testada outra combinação de atributos.

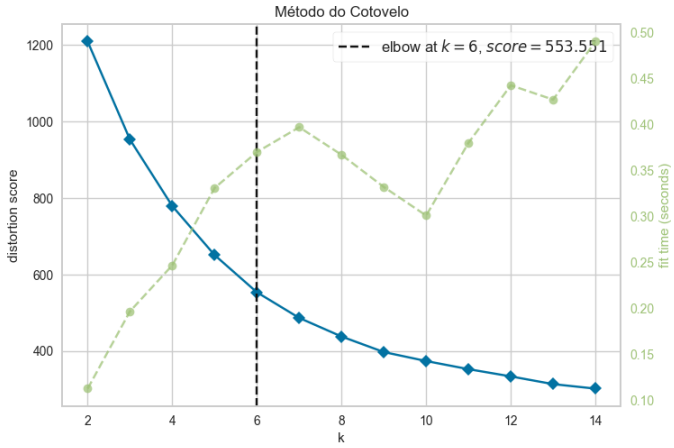
Figura 5.12: Visualização do primeiro agrupamento gerado.



Fonte: Elaborado pelo autor.

Um segundo agrupamento com o modelo o algoritmo K-Médias, mas dessa vez com menos atributos, foi gerado, apenas os atributos relativos à posição geográfica e valor dos veículos foram utilizados. Dessa forma, se espera facilitar o entendimento dos padrões com um ambiente mais simples. O método do cotovelo foi novamente executado, resultando em 6 *clusters* como número ideal de agrupamentos, visível na Figura 5.13.

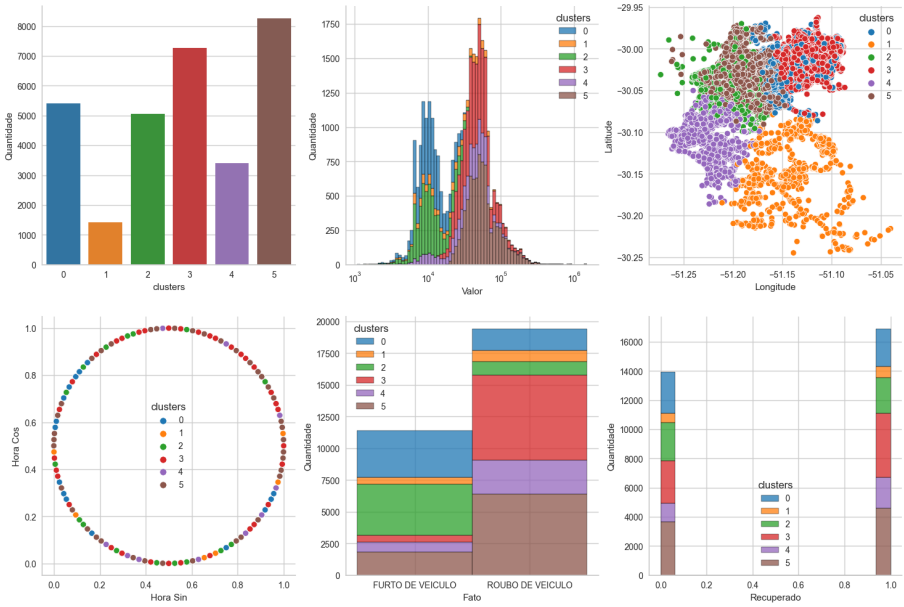
Figura 5.13: Método do cotovelo para o segundo agrupamento.



Fonte: Elaborado pelo autor.

É possível observar que o algoritmo dividiu a cidade em 4 regiões, entre elas, as duas regiões superiores - 0, 2, 3 e 5 - compostas por dois *clusters* cada, partindo as regiões em carros de menor e maior valor. Nas regiões inferiores, 1 e 4, não houve essa separação por valor. Algumas observações podem ser feitas a respeito de cada conjunto:

Figura 5.14: Visualização do segundo agrupamento gerado.



Fonte: Elaborado pelo autor.

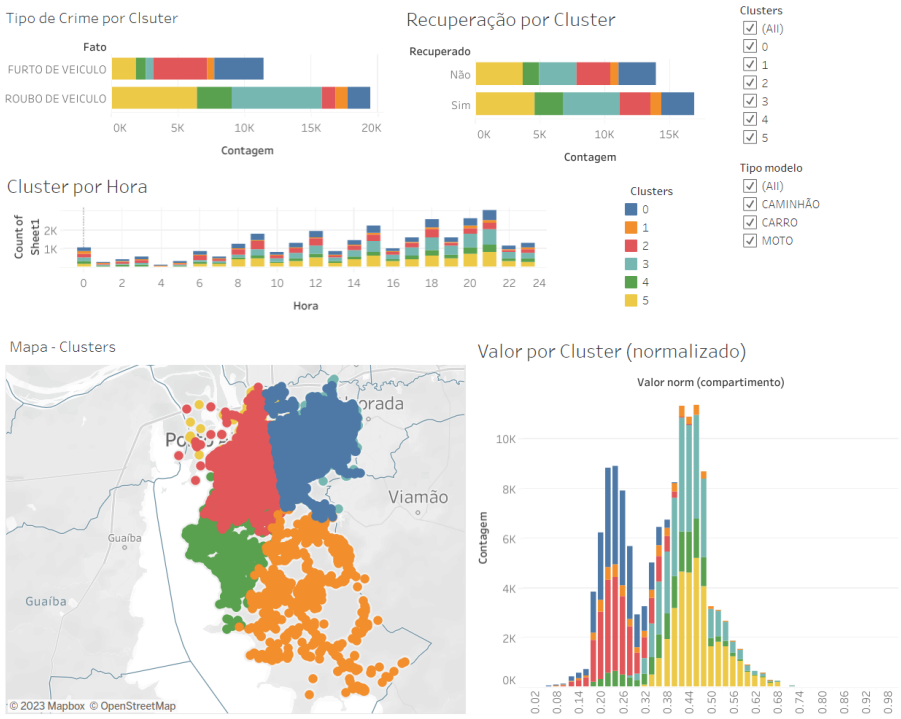
- O *cluster* 0, representa a zona norte da cidade, contendo carros de até 30 mil reais aproximadamente, a maioria desses carros foram furtados;
- O *cluster* 3 representa a zona norte da cidade contendo carros a partir

de 10 mil reais, a grande maioria desses carros foram roubados;

- O *cluster* 2 representa a zona central da cidade, contendo carros de valor mais baixo, até 30 mil reais aproximadamente, a maioria desses carros foi furtado;
- O *cluster* 4 representa a zona Sul da cidade, contendo carros de quase todas as faixas de valor, a maioria desses carros foi roubado;
- O *cluster* 1 representa os dados da zona Extremo Sul da cidade, contendo carros de quase todas as faixas de valor, é o *cluster* mais bem dividido entre roubos e furtos.

Depois de definir os agrupamentos com o algoritmo *KMeans*, foi criado um *dashboard* interativo com os principais atributos observados. Esse Dashboard foi adicionado ao conjunto de *dashboards* gerados com a análise exploratória. O objetivo desse painel é facilitar a visualização e fornecer uma interface interativa para explorar os agrupamentos gerados. Na Figura 5.15, consta uma captura de tela do painel.

Figura 5.15: Painel de visualização dos agrupamentos.



Fonte: Elaborado pelo autor.

Observou-se que, apesar do atributo “tipo de crime” não ter sido incluído como *feature* no modelo, o algoritmo os separou de forma

considerável entre roubo e furto. Percebe-se que furtos ocorreram muito mais em agrupamentos de carros de baixo valor, enquanto roubos ocorreram em agrupamentos de carros de alto valor.

Não foi possível correlacionar o atributo “hora” com o arranjo desses agrupamentos. Outro atributo que não foi possível perceber relação foram as recuperações de veículos, a classe se manteve razoavelmente bem equilibrada entre os *clusters* não apresentando relação com os agrupamentos.

5.7 Aprendizado Supervisionado

Após a realização da análise exploratória, surgiram as seguintes hipóteses: A partir das informações da ocorrência e do veículo envolvido, é possível prever se o mesmo será recuperado ou não? E em caso positivo, é possível prever onde ele será recuperado? A primeira pergunta pode ser testada com um modelo de classificação binária, classificando o veículo entre recuperado ou não. A segunda hipótese corresponde a um modelo de regressão, para prever a localização da recuperação do veículo.

5.7.1 Modelo de Classificação: Recuperação do Veículo

Para prever se o veículo subtraído será recuperado ou não, pode-se utilizar um modelo de classificação binária. A proposta é treinar diversos modelos usando uma gama de diferentes algoritmos, para comparar o desempenho e escolher o que melhor se adapta ao problema. Foram utilizados os seguintes algoritmos de classificação: KNN, Árvore de Decisão, Floresta de Decisão e *Gradient Boosting*. Na Figura 4.2 está o fluxograma de como foi implementado o treinamento dos modelos e análise do desempenho.

A implementação do *Gradient Boosting* escolhida foi a da biblioteca LightGBM, enquanto as outras técnicas utilizadas foram as implementações da biblioteca Scikit-learn. Todo o desenvolvimento dessa etapa foi feito na linguagem Python, no formato de *notebooks*.

5.7.1.1 Treino, Validação e Teste

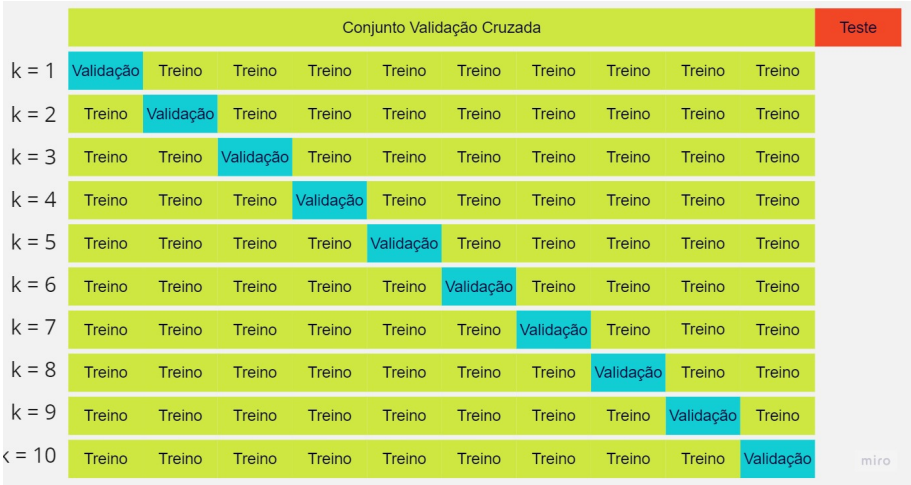
Com os dados já limpos, é necessário separar uma parte do *dataset* para o treinamento com validação cruzada e outra para o teste do modelo. Foi separado 10% do conjunto para o teste. Como esses fatos acontecem de forma cronológica, existe muita relação com a sazonalidade, então os dados do conjunto de teste são as ocorrências mais recentes. Do total de 60 meses de registros, a amostra de teste equivale a, aproximadamente, os últimos 6 meses de ocorrências.

Durante a etapa de treinamento, o modelo utiliza validação cruzada.

A validação cruzada foi feita utilizando 10 folds. Na Figura 5.16 é possível visualizar a separação entre treinamento, validação e teste, e os conjuntos de folds utilizados como treino e validação.

Os dados são ordenados em ordem cronológica, logo pegar dados em sequência deixa a classe alvo desbalanceada, podendo prejudicar o processo de treinamento, por isso, apesar da Figura 5.16 representar o *KFolding* padrão, foi utilizado *StratifiedShuffleSplit*, que mantém a proporção de instâncias da classe alvo e mistura as amostras. Cogitou-se utilizar o *TimeSeriesSplit* como estratégia de *KFolding*, porém ocasionaria o mesmo problema de desbalanceamento.

Figura 5.16: Conjuntos de *Folds* gerados com *ShuffleSplit KFolding*.



Fonte: Elaborado pelo autor.

5.7.1.2 Atributos Categóricos

Após importar os dados limpos da base, é necessário tratar os atributos que são categóricos, neste caso foi utilizado o transformador *oneHotEncoder*. Entre algoritmos escolhidos nesta etapa do trabalho, o único que possui suporte nativo para atributos categóricos é o *gradient boosting*, neste caso ele foi treinado sem utilizar o transformador, apenas foi necessário converter o tipo dos atributos para "*category*".

5.7.1.3 Normalização

Assim como na etapa de aprendizado não supervisionado, é preciso normalizar os atributos, pois alguns modelos são sensíveis à diferença nas

escalas. Todas as normalizações são feitas depois de separar os dados em treino e validação, de forma que os dados de validação são normalizados baseando-se apenas nos dados de treino. Isso é feito para que os dados de treinamento não sejam vazados, tendo em vista que o modelo de predição tenta prever fatos que ainda não aconteceram.

Para os atributos numéricos ordinais foi utilizado o transformador *MinMaxScaler*, que apenas escala os valores para o intervalo [0-1]. O atributo Valor, por possuir escala log-normal, foi normalizado com o *PowerTransformer*. Seguem abaixo os métodos utilizados para transformar cada *feature*.

- MinMaxScaler: Latitude, Longitude, Ano, Mes, Hora, Ano_modelo,
- PowerTransformer: Valor;
- OneHotEncoding ou Categorização: Cor, Fato, Tipo_modelo, Fabricante, Bairro;

5.7.1.4 Ajuste dos Hiperparâmetros

A otimização dos hiperparâmetros foi feita com a biblioteca *Hyperopt-sklearn*, a heurística escolhida é sugerida pelo próprio algoritmo, foram permitidas no máximo 150 iterações. Como cada modelo foi treinado utilizando a validação cruzada, os resultados das métricas é a média dos valores obtidos ao treinar e validar o modelo em cada em cada *fold*. A métrica especificada para otimização foi a acurácia balanceada.

A seguir, na Tabela 5.5 são comparadas as métricas obtidas com o treinamento de cada modelo, para cada um foram registrados os resultados das métricas com os hiperparâmetros padrões (*Default*) e otimizados (*Hyperopt*).

5.7.1.5 Teste do Modelo

Ao fazer o teste do modelo, realizado após o treinamento e ajuste dos hiperparâmetros, as métricas, disponíveis na Tabela 5.6, foram um pouco menores do que o encontrado durante o treinamento. Isso pode ser explicado pelo fato do conjunto de teste ser correspondente aos últimos meses registrados, logo, os veículos classificados como recuperados, não foram

Tabela 5.5: Resultado modelos de aprendizado supervisionado.

Modelo	Acurácia Balanceada	Precisão	Revocação	F1
KNN - Default	0,57	0,65	0,61	0,63
KNN - Hyperopt	0,59	0,72	0,61	0,66
Árvore de Decisão - Default	0,56	0,60	0,60	0,60
Árvore de Decisão - Hyperopt	0,62	0,88	0,60	0,71
Floresta Aleatória - Default	0,61	0,74	0,63	0,68
Floresta Aleatória - Manual	0,62	0,76	0,63	0,69
Gradient Boosting - Default	0,64	0,81	0,64	0,72
Gradient Boosting - Hyperopt	0,64	0,84	0,63	0,72

Fonte: Elaborado pelo autor.

recuperados ainda, mas podem ser recuperados no futuro, conforme mostrado no gráfico da Figura 5.8, diminuindo assim a acurácia do modelo.

Tabela 5.6: Resultado do teste do primeiro modelo com *Gradient Boosting*.

Teste	Acurácia Balanceada	Precisão	Revocação	F1
Gradient Boosting - Hyperopt	0.60	0.62	0.58	0.60

Fonte: Elaborado pelo autor.

5.7.1.6 Avaliação dos Resultados

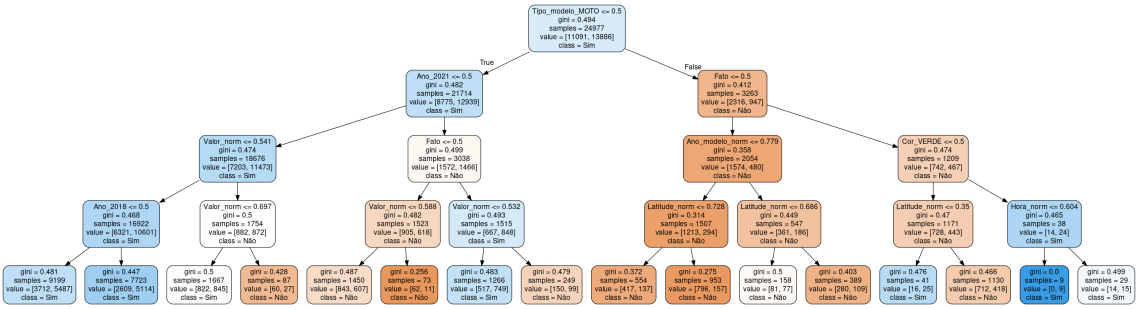
Os algoritmos de aprendizado não supervisionado comportaram-se da forma esperada, houve um ganho significativo de performance quando foi aplicada a otimização dos hiperparâmetros pelo *Hyperopt*, no caso do Floresta Aleatória a melhora não foi tão expressiva, pelo fato do ajuste manual não ser tão efetivo.

Dentre os algoritmos, o pior desempenho foi o *KNN*, por ser uma técnica simples que depende muito de seleção de *features*, e é sensível a técnica de *OneHotEncoder*, para melhorar o desempenho seria necessário testar diferentes pesos para os atributos.

A Árvore de Decisão conseguiu um desempenho relativamente alto

pós-otimização, existem alguns atributos na base que conseguem diferenciar com alguma significância quando um veículo será ou não recuperado, possibilitando que seja encontrada uma árvore expressiva. Curiosamente, a árvore otimizada possui apenas 4 níveis de profundidade. Este modelo é especialmente útil para ajudar a entender quais características levam um veículo a ser ou não recuperado, pois é que possui melhor interpretabilidade dentre os modelos de aprendizado de máquina. Na Figura 5.17 está a árvore gerada pelo modelo.

Figura 5.17: Árvore de decisão gerada pelo modelo otimizado.



Fonte: Elaborado pelo autor.

A Floresta Aleatória, como o esperado, teve um desempenho pouco superior à Árvore de Decisão, quando comparando o caso *default*, por implementar a técnica de *ensemble*, foi capaz de chegar a um modelo um pouco mais acurado. Porém, não foi possível explorar ao máximo o potencial deste modelo com a otimização dos hiperparâmetros.

O modelo que obteve melhor desempenho foi o *Gradient Boosting*, este resultado era o esperado, pois o algoritmo implementa *ensemble* e *boosting*. Uma vantagem deste algoritmo é que ele é o único com suporte a *features* categóricas de forma nativa, não sendo necessário inserir colunas adicionais para a codificação dos atributos, isso contribuiu bastante, pois a os dados possuem diversos atributos categóricos.

A versão otimizada sofreu pouca diferença de ganho de performance comparado com a versão *default*. Os parâmetros *default* da biblioteca LightGBM já se adaptam bem ao problema, não sendo tão expressiva a melhora com a otimização.

O algoritmo *gradient boosting* possui uma ferramenta que permite gerar a taxa de importância dos atributos utilizados no modelo, permitindo

verificar quais fatores foram considerados mais importantes no processo de classificação. Na Tabela 5.7, é possível ver o nível de importância de cada fator considerada pelo modelo.

Tabela 5.7: Taxas de importância dos fatores no modelo de *gradient boosting* de classificação binária.

Fatores	**Importância**
Valor	15,95%
Latitude	15,58%
Longitude	14,91%
Hora	11,92%
Ano Modelo	9,28%
Fabricante	7,16%
Mês	6,78%
Ano	4,64%
Dia Semana	4,55%
Fato	4,00%
Cor	3,15%
Tipo	2,09%

Fonte: Elaborado pelo autor.

5.7.2 Aprendizado Supervisionado: Onde o Veículo Será Recuperado

Após a geração do primeiro modelo, que prevê se o veículo será recuperado, o próximo passo foi criar um modelo que prevê onde esse veículo será recuperado. Este problema pode ser interpretado como um problema de regressão, onde o modelo é construído tentando prever as coordenadas geográficas onde o veículo será encontrado, o problema dessa abordagem é que será necessário construir dois modelos - para latitude e longitude - ao invés de apenas um modelo, e estes modelos trabalhariam de forma separada. Uma outra opção é separar a cidade em regiões, dessa forma transformando em um problema de classificação.

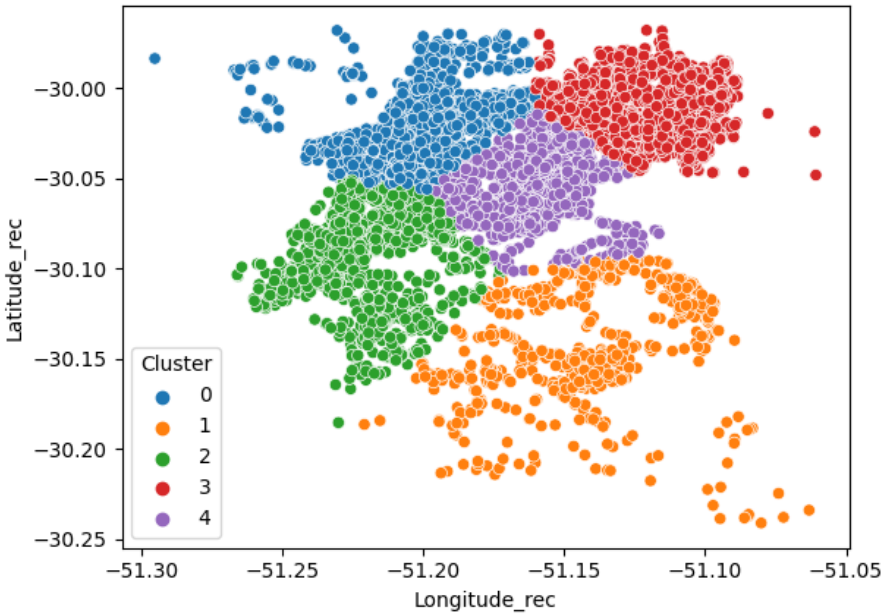
Nesta etapa, a análise foi feita apenas nos dados dos veículos que

foram recuperados. Uma restrição adicionada neste experimento foi utilizar apenas os veículos subtraídos e recuperados em Porto Alegre, e em até 72 horas depois da subtração. O objetivo dessas restrições é tentar identificar padrões sobre este cenário em específico, de veículos subtraídos e logo em seguida recuperados, ainda no município.

Ao realizar a filtragem, do total de 30.837 subtrações de veículos, 16.907 foram recuperados, destes, 11.816 foram recuperados dentro dos limites de Porto Alegre, mesmo município onde ocorreu a subtração. Destes 11.816, 9.117 foram recuperados em até 72 horas depois do crime.

Para dividir a cidade em regiões, foi utilizado o algoritmo de agrupamentos K-Médias, mesmo algoritmo de aprendizado não supervisionado que já foi utilizado anteriormente para identificação de padrões. Dessa vez, os dados que serão utilizados no agrupamento foram apenas latitude e longitude da recuperação do veículo. Utilizando novamente o método do cotovelo, o número ideal de agrupamentos encontrado é cinco, o resultado pode ser observado na Figura 5.18

Figura 5.18: Agrupamentos formados com o K-Médias.



Fonte: Elaborado pelo autor.

Os algoritmos: KNN, Árvore de Decisão, Floresta Aleatória e *Gradient Boosting* foram novamente testados, dessa vez para o problema de classificação multiclasse.

Foi utilizada a mesma técnica de validação cruzada com 10 *folds*, o mesmo processo de normalização dos atributos e ajuste de hiperparâmetros. A Figura 4.2 demonstra o processo de treinamento dos modelos, idêntico aos modelos gerados na seção anterior.

O cálculo das métricas foi a única parte que precisou ser alterada em relação à seção anterior, pois dessa vez o problema é não é mais classificação binária. O cálculo da precisão, Revocação e F1-Score são feitos a partir da média de cada métrica para cada classe (*macro-average*). O cálculo da acurácia balanceada ainda é a média da acurácia de cada classe. As métricas do desempenho dos modelos no treinamento, com 10 folds, podem ser visualizadas na Tabela 5.8

Tabela 5.8: Resultado modelos de aprendizado supervisionado multiclasse.

Modelo	**Acurácia Balanceada**	**Precisão**	**Revo-cação**	**F1**
KNN - Default	0,38	0,41	0,38	0,38
KNN - Hyperopt	0,41	0,42	0,41	0,42
Árvore de Decisão - Default	0,49	0,49	0,49	0,49
Árvore de Decisão - Hyperopt	0,62	0,64	0,62	0,63
Floresta Aleatória - Default	0,62	0,62	0,62	0,62
Floresta Aleatória - Manual	0,62	0,65	0,62	0,63
Gradient Boosting - Default	0,67	0,65	0,63	0,65
Gradient Boosting - Hyperopt	0,67	0,66	0,63	0,64

Fonte: Elaborado pelo autor.

5.7.2.1 Teste do Modelo

Para testar o modelo, foi utilizado o conjunto separado no inicio do treinamento, as métricas obtidas podem ser verificadas na Tabela 5.9. Nota-se que as métricas ficaram muito próximas aos valores obtidos no treinamento, afirmando assim o desempenho do modelo de *Gradient Boosting*.

Tabela 5.9: Resultado do teste do segundo modelo com *Gradient Boosting*.

Teste	Acurácia Balanceada	Precisão	Revocação	F1
Gradient Boosting - Hyperopt	0.67	0.66	0.62	0.62

Fonte: Elaborado pelo autor.

5.7.2.2 Avaliação dos resultados

No geral, comparando os algoritmos com eles mesmos, os algoritmos performaram da forma esperada, novamente o pior foi o KNN, Árvore de Decisão e Floresta Aleatória obtiveram um desempenho superior, e o melhor foi o *Gradient Boosting*. Por ser um problema de classificação multiclasse, esperava-se observar valores menores para as métricas de desempenho, porém *Gradient Boosting* conseguiu um resultado bem superior ao esperado.

Novamente, a otimização de hiperparâmetros com o Hyperopt resultou em um ganho significativo de performance, nos algoritmos KNN e Árvore de Decisão. Para Floresta Aleatória e *Gradient Boosting* o ganho não foi significante, pelos mesmos motivos discutidos na Sessão 5.7.1.6.

É possível fazer a análise de fatores de importância, para verificar quais atributos foram mais utilizados pelo modelo de *gradient boosting*. Na Tabela 5.10 nota-se que os atributos considerados pelo modelo são similares aos gerados pelo modelo de classificação binária, porém com um pouco mais de importância para Longitude e Latitude.

Tabela 5.10: Taxas de importância dos fatores no modelo de *gradient boosting* de classificação multiclasse.

Fatores	**Importância**
Longitude	21,96%
Latitude	21,52%
Valor	17,86%
Hora	11,81%
Ano Modelo	9,13%
Ano	6,05%
Mês	4,65%
Fabricante	2,43%
Dia Semana	2,06%
Cor	1,35%
Fato	0,93%
Tipo Modelo	0,26%

Fonte: Elaborado pelo autor.

5.8 Sumarização dos Experimentos

A análise exploratória cumpriu seu papel, pois foram geradas visualizações que evidenciaram diversas informações em relação as subtrações de veículos, como o horário, o valor dos veículos e o tempo médio para um veículo ser recuperado. Ressalta-se que as observações, especialmente as relacionadas aos preços dos veículos obtido com o cruzamento dos dados com o banco da tabela Fipe, traz a tona uma visão nova a respeito do problema, que não era possível de ser observado antes pelos órgãos de segurança pública.

Nos experimentos com aprendizado não supervisionado, no geral, o algoritmo dividiu a cidade em grandes agrupamentos genéricos, não conseguindo identificar pequenos grupos e perfis. Isso se deve ao fato de os testes com o algoritmo K-Médias não terem encontrado bons agrupamentos com mais do que três atributos, e nem terem encontrado bons agrupamentos com uma grande quantidade de clusters. Apesar disso, essa generalização é

útil para separar entender como os acontecimentos se comportam nas diferentes regiões da cidade.

No primeiro experimento com aprendizado supervisionado, que prevê se o veículo será recuperado, os algoritmos no geral obtiveram métricas um pouco abaixo do esperado, levando em conta que é um problema de classificação binária. Um preditor ingênuo consegue uma pontuação mínima para a acurácia balanceada de 0.5 apenas chutando o mesmo valor sempre, e o modelo com o melhor desempenho obteve uma métrica pouco superior: 0.64. Avalia-se que o motivo disso seja a quantidade de fatores desconhecidos que influenciam os acontecimentos, fora o fato de existirem diversos agentes com interesses diferentes envolvidos no processo, como o criminoso, os receptadores e os agentes de segurança responsáveis pela recuperação.

O segundo experimento com aprendizado supervisionado, que prevê onde o veículo será recuperado, os algoritmos obtiveram métricas melhores, apesar de ser um problema de classificação multiclasse, as métricas chegaram a 0.67 de acurácia balanceada. Avalia-se que este modelo obteve uma boa performance, e tem potencial para prever em tempo real onde os veículos subtraídos serão levados.

6 CONCLUSÃO

Com o objetivo de verificar a viabilidade de utilizar algoritmos de aprendizado de máquina para obter *insights* a respeito do problema proposto, o trabalho apresentou a metodologia e aplicou os experimentos com diferentes algoritmos de aprendizado de máquina para investigar roubos e furtos de veículos que ocorreram nos últimos 5 anos no município de Porto Alegre.

Em aprendizado não supervisionado, o trabalho apresentou os experimentos com o algoritmo K-Médias para agrupamento dos crimes. Em aprendizado supervisionado, o trabalho apresenta os experimentos com os algoritmos KNN, Árvore de Decisão, Floresta Aleatória e *Gradient Boosting* para duas abordagens do problema de predição de recuperação de veículos subtraídos. Alguns dos modelos não obtiveram o desempenho esperado, como os modelos que classificam entre recuperado e não recuperado, mas vale ressaltar que isso também é uma contribuição, pois para saber quais técnicas funcionam ou não é necessário primeiro explorar e testar.

Dentre os resultados obtidos com os experimentos, destacam-se:

- As percepções e o entendimento a respeito dos dados observados durante a análise exploratória.
- O *Dashboard* construído para complementar a análise exploratória;
- O entendimento de como algoritmos de agrupamento de aprendizado não supervisionado podem ser utilizados para traçar perfis e entender padrões nas subtrações de veículos;
- O entendimento de como algoritmos de aprendizado supervisionado podem ser utilizados para entender padrões, e aprimorar o processo de recuperação dos veículos;
- Como a interpretabilidade dos modelos pode ser aplicada sobre algoritmos baseados em Árvores podem ser utilizados para obter informações adicionais sobre o funcionamento do modelo e o problema.

Os resultados encontrados neste trabalho têm potencial para dar suporte a uma série de processos. Os *dashboards* gerados na análise exploratória podem ser integrados diretamente no banco de dados das

ocorrências, fornecendo visualizações e informações que dão suporte à tomada de decisão. Os resultados gerados pelos agrupamentos, podem ser utilizados para auxiliar no direcionamento de políticas públicas, como melhor distribuição de reforço de patrulhamento e instalação de câmeras de cercamento eletrônico. O aprendizado supervisionado como modelo de predição pode ser utilizado para prever possíveis rotas de fuga utilizadas pelo criminoso, para interceptar e recuperar veículos subtraídos em tempo real.

Esse trabalho não se propõe a executar a etapa de operacionalização, logo, como trabalhos futuros existem diversas oportunidades de implementação e integração dos modelos estudados com os sistemas da própria BM e da SSPRS. Também existe a perspectiva de expansão para outras regiões, enquanto o trabalho se limitou apenas ao município de Porto Alegre. O problema e os dados existem no país todo, como, por exemplo, Laboratório de Inteligência de Dados de Gravataí, onde o autor atua como assistente de pesquisa, que já planeja replicar o trabalho com dados do próprio município.

Por fim, espera-se que esse trabalho seja utilizado como base para outros estudos e projetos relacionados a aprendizado de máquina e segurança pública e que sejam somados aos esforços do combate à criminalidade para que um dia traga um pouco mais de felicidade aos brasileiros.

REFERÊNCIAS

AMIDI, S.; AMIDI, A. **Dicas de aprendizado não supervisionado**. 2018. Disponível na Internet: <https://stanford.edu/~shervine/l/pt/teaching/cs-229/dicas-aprendizado-nao-supervisionado>.

ANJOS, O. R. dos; LIMA, R. O.; FILHO, S. C. L.; ALMEIDA, A. T. C. de; RAMALHO, H. M. de B. Padrões de concentração espacial de roubos de automóveis em municípios da grande joão pessoa a partir de técnicas de aprendizado de máquinas. **Teoria e Prática em Administração**, v. 11, n. 2, p. 28–45, set. 2020. Disponível na Internet: <https://periodicos.ufpb.br/index.php/tpa/article/view/50891>.

BRASIL. Artigo nº 144. **CONSTITUIÇÃO DA REPÚBLICA FEDERATIVA DO BRASIL DE 1988**, 1988. Disponível na Internet: <https://www.planalto.gov.br/ccivil_03/constituicao/constituicao.htm>.

BRASIL. Lei nº 12.527. **LEI Nº 12.527, DE 18 DE NOVEMBRO DE 2011**, 2011. Disponível na Internet: <https://www.planalto.gov.br/ccivil_03/_ato2011-2014/2011/lei/l12527.htm>.

CAIRES, D. de O. Técnicas de interpretabilidade para aprendizado de máquina : um estudo abordando avaliação de crédito e detecção de fraude. 2022. Disponível na Internet: <https://www.teses.usp.br/teses/disponiveis/55/55137/tde-16122022-180337/pt-br.php>.

CAPRIOLO, D. **Os custos de bem-estar do crime no Brasil:um país de contrastes**. Banco Interamericano de Desenvolvimento, 2017. Disponível na Internet: <https://publications.iadb.org/pt/node/17466>.

CASTRO, U. R. M. de. Explorando aprendizagem supervisionada em dados heterogÊneos para prediÇÃo de crimes. 2020. Disponível na Internet: <biblioteca.pucminas.br/teses/Informatica_UrsulaRosaMonteiroDeCastro_8666.pdf>.

CHACON, S.; STRAUB, B. **Pro git: Everything you need to know about Git**. 2nd edition. ed. Apress, 2014. Disponível na Internet: <https://git-scm.com/book/en/v2>.

COMARELA, G.; FRANCO, G.; TROIS, C.; LIBERATO, A.; MARTINELLO, M.; CORRÊA, J. H.; VILLAÇA, R. Introdução à ciência de dados: Uma visão pragmática utilizando python, aplicações e oportunidades em redes de computadores. **Sociedade Brasileira de Computação**, 2019.

DATAFOLHA. **Pesquisa Nacional de Vitimização**. Centro de Estudos de Criminalidade e Segurança Pública (CRISP), 2013. Disponível na Internet: <https://www.crisp.ufmg.br/wp-content/uploads/2013/10/Sumario_SENASP_final.pdf>.

DOCA, C. . **Rap da Felicidade**. 1994. Disponível na Internet: <https://www.youtube.com/watch?v=7pD8k2zaLqk>.

ECONÔMICAS, F. I. de P. **Preço Médio de Veículos**. 2023. Disponível na Internet: <https://veiculos.fipe.org.br/>.

EMC, E. S. [S.l.]: John Wiley I& Sons, Inc., 2015.

FOUNDATION, P. S. **Python**. 2021. Disponível na Internet: <https://www.python.org/>.

GITHUB. **GitHub, Let's build from here**. 2023. Disponível na Internet: <https://github.com/>.

IBM. **K-Nearest Neighbors Algorithm**. 2023. Disponível na Internet: <https://www.ibm.com/topics/knn>.

JONER, H. Inferência preditiva geoespacial da criminalidade em porto alegre : uma abordagem de aprendizado de máquina. 2020. Disponível na Internet: <https://lume.ufrgs.br/handle/10183/217875>.

KE, G.; MENG, Q.; FINLEY, T.; WANG, T.; CHEN, W.; MA, W.; YE, Q.; LIU, T.-Y. Lightgbm: A highly efficient gradient boosting decision tree. **Advances in neural information processing systems**, v. 30, p. 3146–3154, 2017.

LIGHTGBM. **LightGBM documentation**. 2023. Disponível na Internet: <https://lightgbm.readthedocs.io/en/latest/index.html>.

LOPES, L.; FELIX, S. Determinantes e prediÇÃo de crimes de homicÍdios no brasil: Uma abordagem de aprendizado de mÁquina. **REVISTA BRASILEIRA DE BIOMETRIA**, v. 37, 06 2019.

LUDERMIR, T. B. Inteligência artificial e aprendizado de máquina: estado atual e tendências. **Estudos Avançados**, SciELO Brasil, v. 35, p. 85–94, 2021.

MAHESH, B. Machine learning algorithms-a review. **International Journal of Science and Research (IJSR).[Internet]**, v. 9, p. 381–386, 2020.

MICROSOFT. **Interpretabilidade do Modelo**. 2023. Disponível na Internet: <https://learn.microsoft.com/pt-br/azure/machine-learning/how-to-machine-learning-interpretability>.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. Em: **Sistemas Inteligentes Fundamentos e Aplicações**. 1. ed. Barueri-SP: Manole Ltda, 2003. p. 89–114. ISBN 85-204-168.

MOREIRA, J. P. Desbravando o git e o github. **Redin-Revista Educacional Interdisciplinar**, v. 5, n. 1, 2016.

PANDAS. **Pandas Ecosystem**. 2023. <https://pandas.pydata.org/community/ecosystem.html>.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

SHAMIM, S.; ZENG, J.; SHARIQ, S. M.; KHAN, Z. Role of big data management in enhancing big data decision-making capability and quality among chinese firms: A dynamic capabilities view. **Information Management**, v. 56, n. 6, p. 103135, 2019. ISSN 0378-7206. Disponível na Internet: <https://www.sciencedirect.com/science/article/pii/S0378720618302854>.

SHEN, H. Interactive notebooks: Sharing the code. **Nature**, v. 515, p. 151–2, 11 2014.

SSPRS. **Dados abertos**. 2023. Disponível na Internet: <https://ssp.rs.gov.br/dados-abertos>.

TABLEAU. Tableau. 2023. Disponível na Internet: <https://www.tableau.com/>.

TUKEY, J. W. **Exploratory Data Analysis**. [S.l.: s.n.], 1977.

WASKOM, M. L. seaborn: statistical data visualization. **Journal of Open Source Software**, The Open Journal, v. 6, n. 60, p. 3021, 2021. Disponível na Internet: <https://doi.org/10.21105/joss.03021>.

WIKILAI, F. S. Transparência passiva. 2023. Disponível na Internet: <https://wikilai.fiquemsabendo.com.br/wiki/TransparÃłncia_passiva>.

YELLOWBRICK. **Elbow Method**. 2023. Disponível na Internet: <https://www.scikit-yb.org/en/latest/api/cluster/elbow.html>.