Business Intelligence

Leandro de Oliveira Capela

Machine Learning e segurança pública: classificando municípios brasileiros a partir de indicadores socioeconômicos

Monografia de Final de Curso

22/09/2020

Monografia apresentada ao Departamento de Engenharia Elétrica da PUC-Rio como parte dos requisitos para a obtenção do título de Especialização em Business Intelligence.

Orientadora:

Manoela Rabello Kohler

Resumo

Esta monografia consiste em um estudo supervisionado aplicado de *Machine Learning* com o objetivo de desenvolver um modelo de classificação que diferencie municípios brasileiros como portadores de violência endêmica ou não. A classificação se baseia em uma métrica definida pela Organização Mundial da Saúde (OMS), onde uma taxa de homicídios superior a 10 por 100 mil habitantes em um ano é típica de violência endêmica. O trabalho foi desenvolvido partir de um *dataset* com 87 colunas de dados socioeconômicos a partir de múltiplas fontes, como grau de escolaridade por faixa etária, taxa de desemprego, arrecadação com impostos, número de empresas por atividade econômica, ativos bancários, entre outros, que estão detalhados no dicionário de dados presente no apêndice. Foram empregadas diversas etapas de pré-processamento, como criação de variáveis binárias a partir de categóricas, tratamento de valores faltantes, tratamento de *outliers*, balanceamento, normalização e redução de dimensionalidade. Após combinar e testar todas essas técnicas, os resultados foram comparados, frente ao uso de Regressão Logística, *Support Vector Machine* (SVM), *Random Forest* e *k-Nearest Neighbors* (k-NN). O modelo de classificação com melhor desempenho foi construído com *Random Forest*, onde os atributos, tratados com balanceamento, normalização, *Kernel* PCA e t-SNE explicam mais de 80% da classificação.

Abstract

This work consists of a supervised study of applied Machine Learning, aiming to develop a classification model that differentiates Brazilian cities as having endemic violence or not. The label is based on a metric defined by the World Health Organization (WHO), where a homicide rate greater than 10 per 100 thousand inhabitants in a year is typical of endemic violence. The work was developed from a dataset with 87 features of socioeconomic data from multiple sources, such as education level by age group, unemployment rate, tax collection, number of companies by economic activity, banking assets, among others, which are detailed in the data dictionary in the appendix. Several pre-processing steps were applied, such as creating binary variables from categorical variables, treating missing values, treating outliers, balancing, normalization and dimensionality reduction. After combining and testing all these techniques, the results were compared against the use of Logistic Regression, Support Vector Machine (SVM), Random Forest and k-Nearest Neighbors (k-NN). The classification model with the best performance was built with Random Forest, where the features, treated with balancing, normalization, Kernel PCA and t-SNE explain more than 80% of the classification.

Sumário

1. Introdução

O debate sobre segurança pública é recorrente, especialmente no Brasil. Em geral, é raro haver um entendimento claro – tanto da população, quanto dos governantes – sobre os fatores que explicam o fato de uma cidade ser menos segura e, por consequência, impactar negativamente na qualidade de vida das pessoas. Com isso, há o risco de não se direcionar políticas públicas adequadas para mitigar esse problema.

Na comunidade científica, existe uma diversidade de publicações que empregam indicadores sociais, seja com dados tabulares – numéricos ou categóricos – ou não-estruturados – como imagens, por exemplo – para explicar ou predizer comportamentos em relação à segurança, desenvolvimento ou condição socioeconômica de uma ou mais regiões.

Alves, Ribeiro e Rodrigues (2018) aplicaram análise preditiva a dados de crimes, identificaram a correlação entre crime e métricas urbanas e quantificaram a importância de métricas urbanas na predição de crimes no Brasil, usando um regressor *Random Forest*. Foi apontado que o analfabetismo e o desemprego eram as variáveis mais relevantes para explicar a criminalidade.

Sousa, Del-Fiaco e Berton (2018) fizeram uma análise de *cluster* das taxas de homicídios no em Goiás entre os anos de 2002 e 2014, considerando variáveis sociodemográficas no estado, que era o quinto colocado em homicídios no Brasil no último ano estudado. Além de analisar a correlação, foram empregados três algoritmos de agrupamento: *K-means*, métodos baseados em densidade e métodos hierárquicos. Os resultados indicaram que as taxas de homicídio eram mais elevadas em cidades vizinhas a grandes centros urbanos, ainda que apresentassem melhores indicadores socioeconômicos.

Jean *et al*. (2016) criaram um modelo de predição de pobreza a partir de imagens de satélite, comparando localidades à luz do dia e à noite. Os pesquisadores apresentam uma rede neural convolucional treinada para identificar nas imagens elementos que explicam até 75% da variação de indicadores econômicos a nível local, em cinco países africanos.

Shafizadeh-Moghadam *et al*. (2017), por fim, combinaram aprendizado de máquina, modelos baseados em árvores e modelos estatísticos a autômatos celulares para simular o crescimento urbano da região metropolitana de Teerã, capital do Irã. Foram comparados: redes neurais artificiais (ANN), *Support Vector Regression* (SVR), *Random Forest*, árvores de classificação e regressão (CART), regressão logística e *splines* de regressão multivariada adaptativas (MARS).

1.1. Motivação

Historicamente, o combate à violência foi tratado de diversas formas no Brasil, seja a nível federal, estadual ou municipal. Ao longo do tempo, houve abordagens desde incentivo à pesquisa até o uso de força policial. Em geral, a postura se mostra mais reativa que preventiva, e quando ocorre o segundo caso, a tomada de decisão apresenta pouco aprofundamento no que os dados podem explicar sobre a questão.

Segundo Silvares (2019), o conceito de segurança pública começou a ser usado no Brasil durante a Ditadura Militar (1964-1985) e, nesse contexto, se destacavam seu caráter reativo a incidentes e a repressão militar. Na visão de Nascimento e Teixeira (2016), a situação atual da segurança pública no país possui traços desse período.

Na atual década, o tema ganhou importância. Entre 2011 e 2016, foram realizadas cinco edições do projeto “Pensando a Segurança Pública”, com o objetivo de incentivar a produção de conhecimento científico no tema (BRASIL, 2016). Porém, não há registros de sua continuidade nos anos seguintes.

Desde agosto de 2019, está em curso o projeto “Em Frente, Brasil”, que se propõe a “articular políticas públicas entre a União, estados e municípios que trabalham na redução da violência e da criminalidade” (BRASIL, 2019). Segundo o Ministério do Desenvolvimento Social (2020), o foco seria “na saúde, na educação, no esporte, na cultura, na habitação e na geração de empregos” e, em sua primeira fase, foram escolhidos cinco municípios, um em cada região.

Não cabe no escopo deste trabalho debater aspectos específicos da implementação ou da eficácia de políticas públicas. Todavia, os indícios permitem uma análise crítica da tomada de decisão, mais diretamente no que se refere ao uso de dados nesse processo.

1.2. Objetivos do trabalho

Este trabalho tem como objetivo geral criar um modelo de classificação, de modo a gerar previsões significativamente confiáveis relacionadas à segurança pública nos municípios brasileiros a partir de diversos indicadores socioeconômicos locais.

Os objetivos específicos deste projeto são:

- realizar análise exploratória dos dados de métricas urbanas dos municípios brasileiros;
- testar diferentes algoritmos de *Machine Learning*;
- descrever os processos de implementação de um projeto de *Machine Learning* aplicado a métricas urbanas e
- analisar e demonstrar um modelo preditivo de classificação a partir de métricas urbanas.

1.3. Descrição do trabalho

Neste projeto, desenvolveram-se diversos modelos de classificação, com quatro diferentes algoritmos, de modo a escolher o modelo com melhor desempenho após comparar os resultados de cada um deles. Ao longo do processo, foram realizadas cinco etapas: desde fases iniciais de organização de dados, passando por análise exploratória, diversas etapas de pré-processamento, aplicação de algoritmos e comparação de métricas de desempenho entre modelos.

A fase de organização de dados, ou integração da base de dados, consistiu na descrição das principais fontes de dados, do dicionário de dados e das principais transformações que geraram novos atributos. Além disso, foram sintetizados alguns dos tratamentos pontuais realizados em parte dos registros.

Na etapa seguinte, de análise exploratória, o processo de análise foi descrito, mencionando as tecnologias empregadas, informações quantitativas da variável resposta, detalhes sobre variáveis categóricas que seriam transformadas, um panorama geral das principais medidas-resumo a partir do *dataset* e *insights* gerais que podem ser tirados a partir de medidas centrais da base.

O terceiro passo é um dos mais importantes do projeto: o pré-processamento. Nessa seção, foram descritos detalhadamente os processos de limpeza da base, tratamento de valores faltantes, tratamento de *outliers*, balanceamento, normalização e redução de dimensionalidade, mostrando, principalmente, os impactos de cada processo sobre os dados e de que forma cada etapa pode contribuir no modelo final.

A quarta parte é a de aplicação de algoritmos. Nela, foram descritos o processo utilizado para implementar cada um dos algoritmos: Regressão Logística, *Support Vector Machine* (SVM), *Random Forest* (RF) e *k-Nearest Neighbors* (k-NN), informando a biblioteca e os principais pacotes utilizados.

Por fim, a avaliação de modelos apresenta as métricas de avaliação – acurácia, precisão, *recall* e F1 – e o principal recurso visual utilizado, a matriz de confusão, para medir o desempenho dos modelos. Também são apresentados resultados dos testes, mostrando tanto o impacto individual de cada etapa de pré-processamento, quanto de combinações entre elas, na *performance* dos modelos.

1.4. Organização da monografia

Este trabalho é composto de cinco capítulos adicionais, descritos a seguir:

- o capítulo 2 apresenta a descrição do problema, contendo a descrição global dos aspectos relevantes da questão e das soluções que são atualmente empregadas;
- o capítulo 3 discorre sobre as metodologias aplicadas no estudo;
- o capítulo 4 descreve a arquitetura do sistema proposto, justificando as alternativas escolhidas no desenvolvimento da solução e descrevendo os passos da implementação;
- o capítulo 5 relata os melhores resultados obtidos, comparando com outros resultados e discutindo a apresentação desses números e
- o capítulo 6 suscita as conclusões a partir dos resultados, resgata os objetivos iniciais, destaca os resultados obtidos e as limitações do projeto, além de indicar sugestões ou relatos a respeito de trabalhos posteriores a este.

2. Descrição do Problema

O problema tratado nesta monografia consiste em entender quais variáveis são mais significativas para explicar a taxa de homicídios nos municípios do Brasil. Para isso, será criado um modelo supervisionado, de classificação.

A classificação no modelo se baseia em rotular um município de acordo com sua taxa estimada de homicídios por 100 mil habitantes. Neste projeto, os dados de homicídios têm como fonte o Atlas da Violência, estudo publicado anualmente pelo Instituto de Pesquisa Econômica Aplicada (Ipea) e elaborado em conjunto com o Fórum Brasileiro de Segurança Pública (FBSP). A Figura 1 mostra os municípios com as maiores taxas de homicídios em tons mais escuros de vermelho.

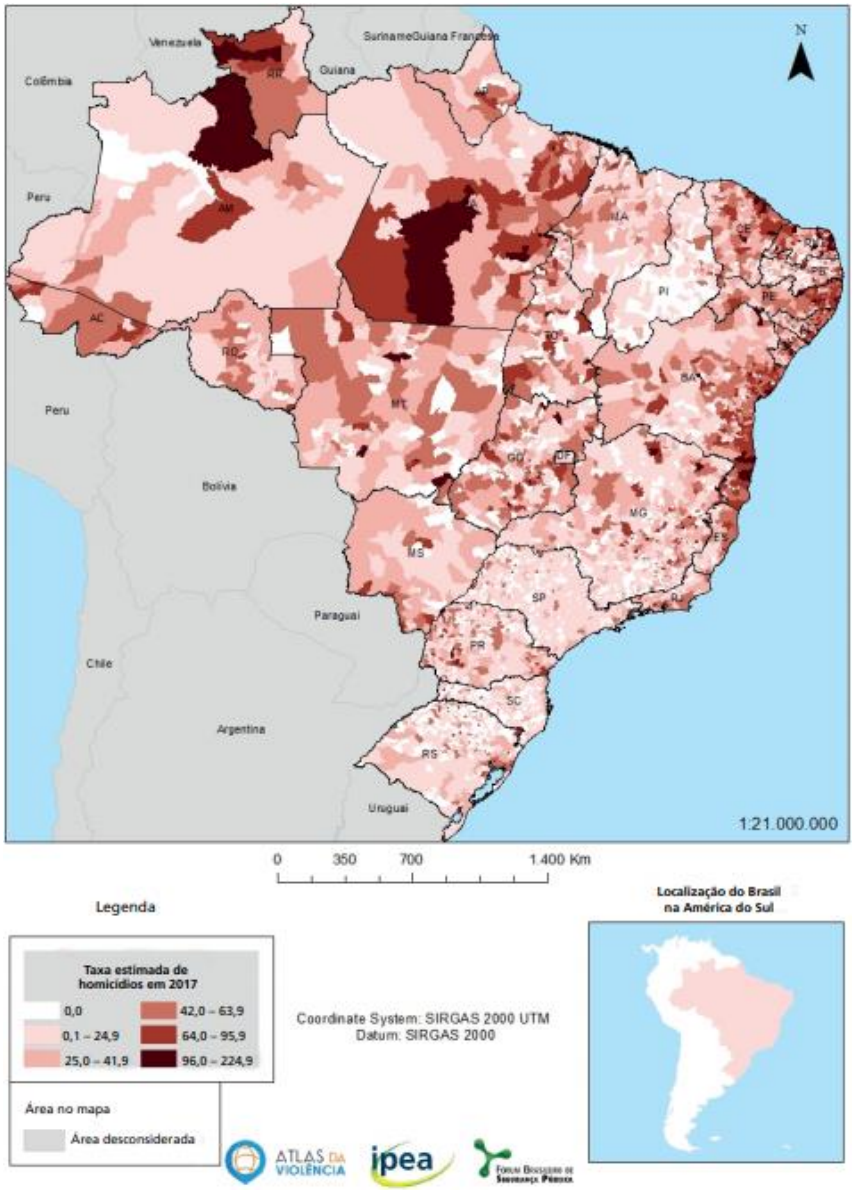

Figura 1 – Mapa indicando a taxa estimada de homicídios por 100 mil habitantes em 2017

Fonte: Ipea (2019)

2.1. Cálculo da taxa de homicídios

A principal variável desta monografia é a taxa de homicídios. O Ipea (2019, p. 7) indica que, em seu estudo, a taxa de homicídios é calculada pelo somatório do total de homicídios reportados e do total de homicídios ocultos, divididos pela população, que por sua vez é dividida por 100 mil. O total de homicídios reportados se dá pela soma entre o número de óbitos por agressão e o número de óbitos por intervenção legal e está indicado na base de dados apenas como “homicídios”.

$$Taxa\ de\ homicídios = \frac{Homicídios + Homicídios\ ocultos}{\left(\frac{População}{100\,000}\right)}$$

Os homicídios ocultos, por sua vez, são mortes violentas com causa indeterminada (MVCIs). Em estudos voltados aos municípios, é importante que esses óbitos sejam levados em conta “pelo fato de alguns poucos homicídios ocultados pela classificação incorreta alterarem significativamente a taxa de homicídio local, fazendo com que municípios relativamente violentos sejam considerados pacíficos” (IPEA, 2019, p. 7).

Além do impacto na classificação dos municípios, há outro elemento que reforça o emprego do número de homicídios ocultos na taxa de homicídios de um município. De acordo com Cerqueira (2013 *apud* IPEA, 2019, p. 7), 73,9% das MVCIs em 2010 se tratavam de homicídios com classificação incorreta. Dessa forma, nesta monografia, a taxa de homicídios seguirá o cálculo apontado pelo Ipea (2019).

2.2. Rótulo de classificação

A principal variável estudada é numérica, mas o modelo desenvolvido é de classificação. Para isso, será extraído um rótulo de classificação a partir da taxa de homicídios. O parâmetro adotado para a classificação se baseia em uma definição mencionada pelo Banco Mundial (2016), que diz que a Organização Mundial da Saúde (OMS) considera uma taxa de homicídios maior ou igual a 10 para cada 100 mil habitantes como típica de violência endêmica.

Portanto, neste estudo, a classificação utilizada será “mais violento” para os municípios com taxa de homicídios maior ou igual a 10 e “menos violento” para os municípios abaixo desse valor. Vale notar que, por mais que simplificar a classificação com os nomes “pacífico” e “violento” possa parecer tentador, uma taxa próxima de 10 não denota que um local seja necessariamente pacífico.

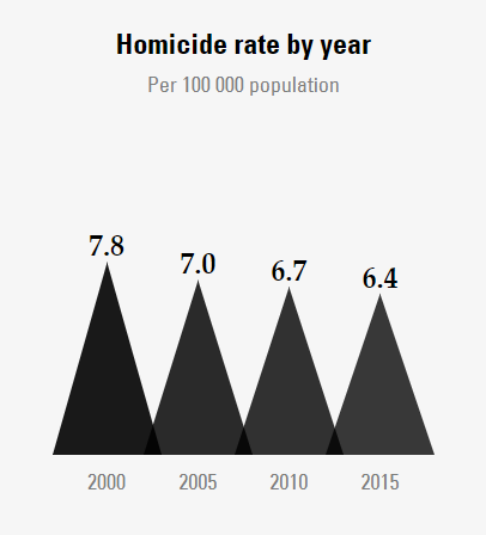

Figura 2 – Taxa mundial de homicídios por 100 mil habitantes por ano
Fonte: OMS (2017)

A Figura 2, retirada do último levantamento da OMS (2017), aponta que a taxa mundial era de 6,4 homicídios a cada 100 mil habitantes em 2015. Segundo o relatório, a grande maioria dos países da Europa e América do Norte apresentava taxas abaixo da média global. No Brasil, a taxa era de 30,5 em 2015 (OMS, 2017) e 34,9 em 2017 (IPEA, 2019).

2.3. Demais variáveis

As outras variáveis presentes no *dataset* serão descritas de forma mais detalhada no capítulo sobre a arquitetura do sistema proposto. No entanto, nesta parte, cabe ressaltar o seu papel na solução do problema em questão.

O intuito é, dentre as variáveis analisadas, selecionar aquelas que sejam mais relevantes, ofereçam o melhor desempenho ao modelo e expliquem melhor o resultado do rótulo de classificação. Para isso, serão empregadas diversas técnicas de mineração de dados, que serão expostas com mais detalhes no capítulo de metodologia.

Caso esta análise viesse a ser utilizada como suporte à decisão em políticas públicas de segurança, encontrar as melhores variáveis explicativas teria como principal função incentivar o direcionamento de políticas públicas a setores da sociedade que possam contribuir na redução da violência. Por outro lado, há de se ter o cuidado de não assumir uma relação direta de causa e efeito baseando-se unicamente na correlação entre variáveis. É preciso realizar validações adicionais para chegar a conclusões definitivas e que estão além do escopo deste estudo.

3. Metodologias

A parte principal da análise é voltada para a criação do modelo de classificação usando duas classes. A ideia é testar quatro algoritmos e comparar seus indicadores de desempenho. Os algoritmos são:

- Regressão Logística;
- *Support Vector Machine* (SVM);
- *Random Forest* (RF) e
- *k-Nearest Neighbors* (k-NN).

As etapas iniciais, como integração de dados, análise exploratória e pré-processamento, e a etapa final, de avaliação, serão abordadas no capítulo seguinte, que apresenta a arquitetura do sistema proposto. Nesta seção, o foco será apenas nos algoritmos. Os modelos criados a partir desses algoritmos também serão tratados no próximo capítulo.

3.1. Regressão Logística

De acordo com Sainani (2014), a regressão logística foi concebida de maneira similar à regressão linear, de modo a buscar ajustar uma reta – ou seja, um intercepto e uma inclinação – aos dados, como apontado na Figura 3. No entanto, como o resultado tem apenas dois níveis, o ajuste linear não é otimizado nesse formato.

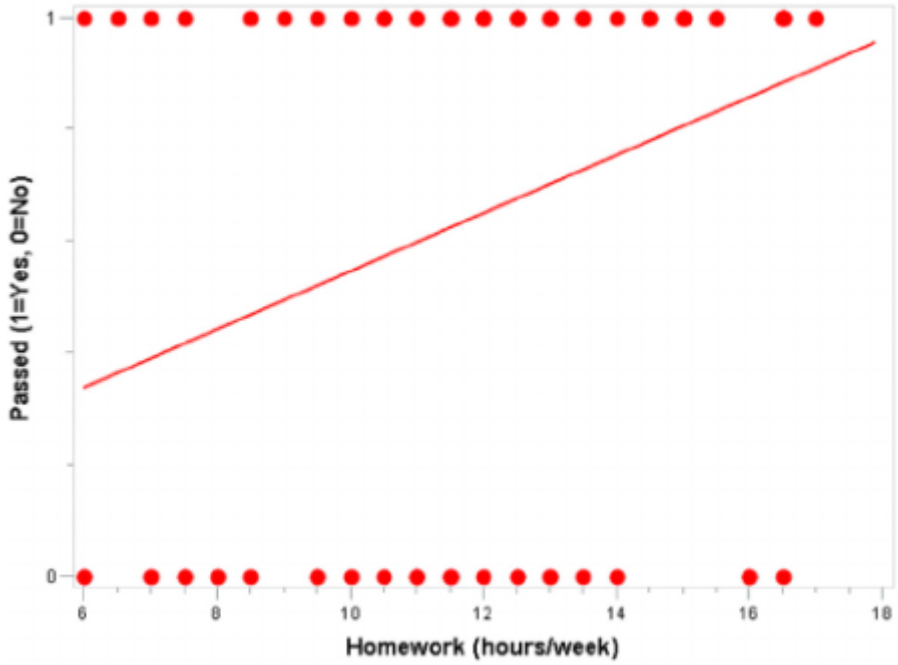

Figura 3 – Demonstração do ajuste linear da regressão logística aos resultados binários

Fonte: Sainani (2014, p. 1158)

Por esse motivo, ao invés de ajustar uma reta diretamente ao resultado binário, a regressão logística usa uma transformação do resultado, chamada *logit*. O *logit* é diretamente relacionado à probabilidade do resultado. No entanto, não se limita a valores entre 0 e 1, e pode assumir qualquer valor real.

$$logit = \ln\left(\frac{p}{1-p}\right)$$

Vale observar que, para calcular o *logit* manualmente e apresentar em um gráfico, é preciso agrupar os dados em percentis, por exemplo, e calcular as probabilidades para cada grupo. Todavia, o algoritmo não faz divisões arbitrárias, mas sim usa cálculo para encontrar a equação da reta que traz o melhor ajuste.

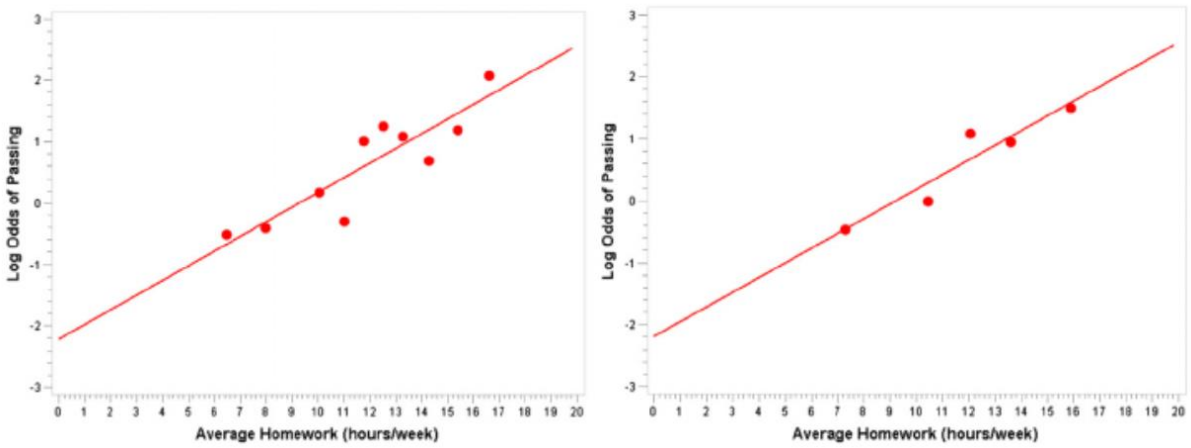

Figura 4 – Representação do ajuste linear com *logit*, agrupado em decis (esquerda) e quintis (direita)
Fonte: Sainani (2014, p. 1158)

Na Figura 4, é visível que, a partir do cálculo do *logit*, o ajuste linear se torna mais adequado. A regressão logística traz informação sobre a relação entre variáveis individuais e o resultado binário. Ela também pode ser utilizada para calcular probabilidades em uma predição e para gerar curvas *Receiver Operating Characteristic* (ROC), que refletem a habilidade de diferenciação do modelo.

3.2. *Support Vector Machine* (SVM)

Segundo Guenther e Schonlau (2016), de forma análoga à regressão logística, SVMs foram inicialmente concebidos para classificação com duas classes. Esta abordagem foi posteriormente estendida a resultados contínuos e classificações com mais de duas classes, mas nesta aplicação, a classificação se dá em duas classes. Portanto, este será o foco desta seção.

No sistema analisado, são cerca de 90 variáveis explicativas. Para simplificar a representação gráfica, será demonstrado um exemplo com duas variáveis, *x1* e *x2*. Portanto, nesse caso, as observações se distribuem em um espaço bidimensional. Para introduzir a ideia do algoritmo, é preciso, também, trazer as definições de hiperplano e margem.

A dimensão de um hiperplano varia de acordo com o número de dimensões do espaço onde está inserido. Sua definição diz que um hiperplano em um espaço de dimensão *n* é a translação de um espaço gerado de dimensão *n* – 1. Ou seja, quando a dimensão é 2, uma reta é um hiperplano. Por sua vez, em um espaço tridimensional, um plano é um hiperplano, e assim sucessivamente (CABRAL; GOLDFELD, 2012, p. 56).

A margem, no contexto desta análise, é a distância mínima entre o hiperplano e o ponto mais próximo dele (GUENTHER; SCHONLAU, 2016, p. 918). Portanto, em um espaço bidimensional, se trata da menor distância entre a reta e a observação que mais se aproxima dela. O SVM atua de modo a encontrar a reta que apresenta a maior margem. Ou seja, é um problema de otimização com o objetivo de maximizar o valor da margem. O hiperplano, por sua vez, é o elemento que separa as classes nesse espaço vetorial.

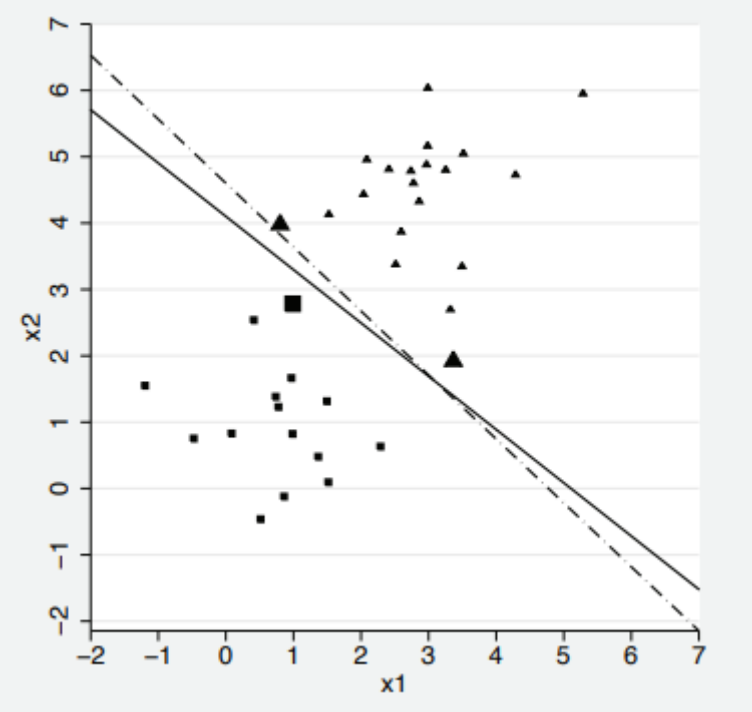

Figura 5 – Gráfico demonstrando a atuação de um SVM
Fonte: Guenther e Schonlau (2016, p. 919)

Na Figura 5, as classes são representadas pelas observações nas formas de quadrado e triângulo. As formas de maior tamanho são denominadas vetores de suporte. Tratam-se dos pontos que são usados como referência para medir a margem e determinar o hiperplano

escolhido pelo algoritmo na construção do modelo. A linha tracejada representa uma solução possível e a linha sólida é a solução ótima.

3.3. *Random Forest* (RF)

Breiman (2001) define *Random Forest* como um classificador composto de um conjunto de classificadores estruturados em árvore, contendo vetores aleatórios identicamente distribuídos e independentes, e cada árvore emite um voto unitário para eleger a classe mais popular nas variáveis de entrada.

O algoritmo *Random Forest* é uma ferramenta de modelagem natural e não-linear que fornece estimativas sobre a hierarquia de variáveis na classificação. Com isso, é também capaz de estimar a contribuição de cada variável no resultado. RF é apontado como um algoritmo de alta acurácia de previsão, aceitável tolerância a *outliers* e ruído e com facilidade de evitar problemas de *overfitting* (WANG *et al*., 2015, p. 1131).

Algoritmos RF combinam *bagging* e um método específico onde o subespaço aleatório é conduzido em cada nó da árvore de classificação e regressão (CART). Além disso, usam particionamento recursivo para gerar várias CARTs e agregar os resultados. Cada árvore, por sua vez, é construída de forma independente usando uma amostra *bootstrap*[1] dos dados de treino (ZHANG; SUGANTHAN, 2014).

3.4. *K-Nearest Neighbors* (k-NN)

Segundo Liu e Gopalakrishnan (2017), para determinar o valor de uma variável para uma determinada amostra, o k-NN extrai a média ponderada da variável das *k* amostras mais próximas. Uma das principais vantagens desse algoritmo é a capacidade de ser usado com dados numéricos e categóricos. No caso de variáveis categóricas, é utilizada a moda para estipular valores.

Por outro lado, Piegl e Tiller (2002) demonstram que, embora a eficácia do algoritmo não dependa do valor de *k*, a velocidade de processamento é bastante afetada pela quantidade de vizinhos apontados como parâmetro. Concluiu-se que, quando o valor de *k* é superior a 20% do tamanho da base de dados, o tempo de processamento é diretamente proporcional a esse tamanho, elevado ao quadrado. Entre 10% e 20%, a proporção é linear, e até 5%, é inferior a linear.

[1]*Bootstrap* é um método não-paramétrico de amostragem que possui a capacidade de estimar corretamente a variância da mediana amostral e apresenta desempenho superior ao do método *cross-validation* ao estimar taxas de erro em um problema de diferenciação linear (EFRON, 1979, p. 1).

4. Arquitetura do Sistema Proposto

O sistema proposto, representado na Figura 6, consiste em um fluxo de cinco etapas: integração de bases de dados, análise exploratória, pré-processamento, aplicação de algoritmos e avaliação de modelos. A primeira parte foi realizada com auxílio do Excel. Já nas etapas seguintes, o ambiente de desenvolvimento integrado (IDE) utilizado foi o JupyterLab e a linguagem de programação Python.

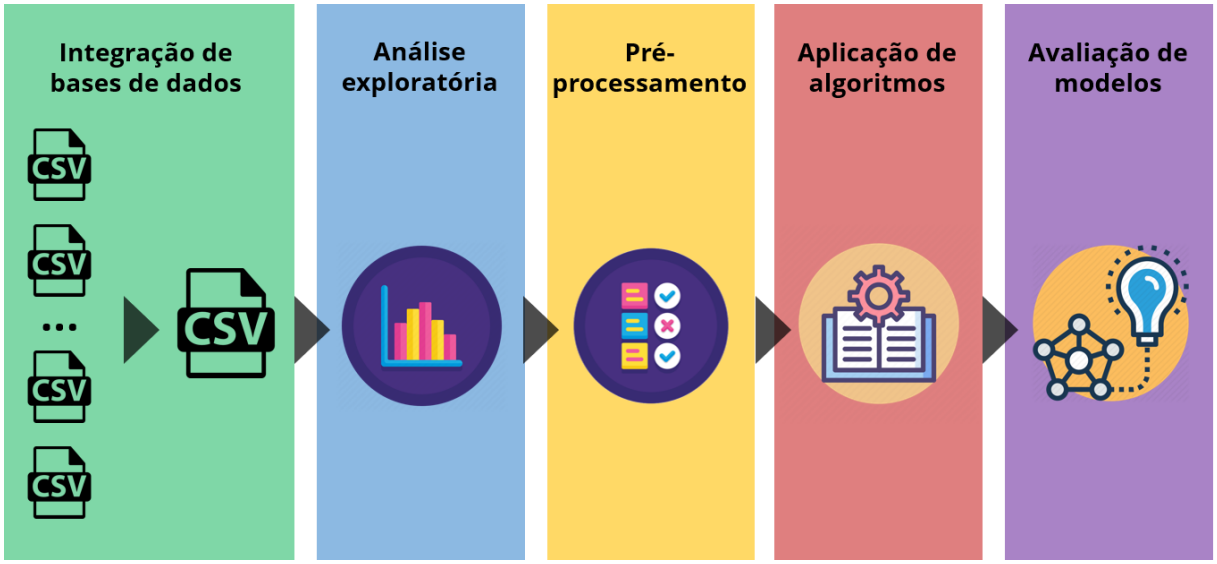

Figura 6 – Diagrama representando a arquitetura do sistema proposto

Fonte: Elaboração própria

4.1. Integração da base de dados

A maior parte dos dados foi obtida dentro da plataforma Kaggle[2], com origem em diversas fontes primárias. A principal delas é o Instituto Brasileiro de Geografia e Estatística (IBGE). As outras são órgãos como Agência Nacional de Telecomunicações (Anatel), Banco Central do Brasil, Departamento Nacional de Trânsito (Denatran), Ministério do Turismo, Programa das Nações Unidas para o Desenvolvimento (Pnud), Tesouro Nacional e companhias como Empresa Brasileira de Correios e Telégrafos (ECT), Uber, McDonald's e Walmart.

Na mesma plataforma, foi fornecido um dicionário de dados, em inglês, contendo o detalhamento dos nomes das colunas, descrições dos campos, períodos de referência, unidades de medida e fontes. No desenvolvimento deste projeto, as informações foram traduzidas para o português. Além disso, foi realizada atualização de parte dos dados, adição

[2] <https://www.kaggle.com/crisparada/brazilian-cities/>

de 13 novos campos e transformação de 10 colunas de números absolutos para percentuais ou taxas. O dicionário de dados deste projeto se encontra no Apêndice A.

Ao analisar a base principal, foram observadas algumas inconsistências de registros, que demandaram tratamento. Algumas cidades com mais de uma grafia estavam duplicadas. Há casos na Bahia, com “Santa Teresinha” e “Santa Terezinha”, e Pernambuco, com “São Caitano” e “São Caetano”.

Em cada registro inconsistente, havia dados faltantes que se complementavam. Por exemplo, em uma linha, havia registros de atributos de população, mas não havia de atividade econômica. Em outra, ocorria o oposto. O comportamento se repetiu em todos os atributos desse município. Não havia sobreposição. Com isso, as linhas foram mescladas manualmente no Excel e foram mantidas as cidades com as grafias “Santa Terezinha”[3] e “São Caitano”[4], por serem as que constam no site do IBGE.

Outro tratamento empregado foi sobre o registro que continha como município a Lagoa dos Patos, no Rio Grande do Sul. Não há nenhum registro dessa localidade como município no site do IBGE. Existem apenas informações de área, como parte do território brasileiro. Com isso, o registro foi removido da base. A esse conjunto de dados, foram adicionados dados oriundos do Ipea e do Ministério da Saúde. Com isso, a base final totalizou 87 atributos.

4.2. Análise exploratória

No JupyterLab, a primeira etapa é a análise exploratória de dados. Essa etapa consiste em organizar e classificar os dados, calcular medidas resumo e visualizar o conjunto de dados. Inicialmente, foram importadas as bibliotecas NumPy e Pandas. Em seguida, é a base é carregada e armazenada em um *dataframe* do Pandas.

O passo seguinte é encontrar o tamanho do conjunto de dados: são 5.570 linhas – uma para cada município existente no Brasil em 2017 – e 87 colunas – entre nomes de municípios, identificadores e atributos. A partir desse momento, torna-se fundamental gerar a coluna rótulo.

Às 87 colunas existentes, somam-se duas: a taxa de homicídios, calculada pela fórmula da seção 2.1, e a coluna com a classificação, onde municípios com taxa de homicídios superior ou igual a 10 serão a classe dominante (3.815 registros) e possuem classificação “*endemic*” (endêmica) ou classe 1, e aqueles que estiverem abaixo desse valor apresentam classificação “*less violent*” (menos violenta), ou classe 0 e serão a classe rara (1.755 registros).

[3] <https://cidades.ibge.gov.br/brasil/ba/santa-terezinha/panorama>
[4] <https://cidades.ibge.gov.br/brasil/pe/sao-caitano/panorama>

Na sequência, são verificados os tipos de dados dos atributos. Observa-se que a grande maioria é numérica – mais precisamente, 82 – e, dentre essas, as contínuas prevalecem. No entanto, existem também sete variáveis categóricas no conjunto de dados: *CITY*, *STATE*, *REGIAO_TUR*, *CATEGORIA_TUR*, *RURAL_URBAN*, *GVA_MAIN* e *CLASS*.

A etapa que que se segue é gerar medidas-resumo de cada coluna numérica do conjunto de dados estudado, entre medidas centrais e outras métricas que trazem um panorama dos números, como a contagem de valores, média, desvio padrão, mínimo, máximo, mediana, primeiro e terceiro quartis.

	IBGE_ID	CAPITAL	IBGE_RES_POP_ESTR_PERC	IBGE_DU_RATE	IBGE_DU_RURAL_PERC	IBGE_POP_UP_PERC	IBGE_0-14_PERC	IBGE_15-59_PERC	IBGE_60+_PERC	IBGE_PLANTED_AREA	...	IBGE_EDU_3_PERC	IBGE_EDU_NA_PERC	IBGE_ESTIMATED_POP_2017	HOMICIDES
count	5.570000e+03	5570.000000	5565.000000	5563.000000	5563.000000	5565.000000	5565.000000	5565.000000	5565.000000	5.570000e+03	...	5565.000000	5565.000000	5.570000e+03	5570.000000
mean	3.253591e+06	0.004847	0.000759	0.296956	0.347884	0.625942	0.249089	0.630818	0.120093	1.417987e+04	...	0.041744	0.004329	3.728203e+04	11.608618
std	9.849103e+05	0.069461	0.005457	0.035449	0.214331	0.215368	0.045846	0.032088	0.033571	4.405924e+04	...	0.026144	0.005158	2.183999e+05	62.879470
min	1.100015e+06	0.000000	0.000000	0.033931	0.000000	0.041621	0.106383	0.471631	0.022546	0.000000e+00	...	0.001817	0.000000	8.120000e+02	0.000000
25%	2.512126e+06	0.000000	0.000000	0.277559	0.169573	0.462674	0.215288	0.608678	0.097993	9.102500e+02	...	0.022808	0.001054	5.501500e+03	0.000000
50%	3.146280e+06	0.000000	0.000000	0.301836	0.337229	0.634778	0.245177	0.632532	0.119218	3.471500e+03	...	0.036142	0.002897	1.163550e+04	2.000000
75%	4.119190e+06	0.000000	0.000699	0.321476	0.508431	0.806386	0.277108	0.654258	0.141037	1.119425e+04	...	0.054025	0.005856	2.528975e+04	7.000000
max	5.300108e+06	1.000000	0.377218	0.390541	0.954474	0.999491	0.440760	0.744760	0.421986	1.205669e+06	...	0.266880	0.064914	1.210692e+07	2145.000000

Figura 7 – Fragmento da tabela de medidas-resumo dos atributos numéricos

Fonte: Elaboração própria

Esse tipo de tabela, exemplificado na Figura 7, pode ser interessante para análises de colunas individuais e ter alguns *insights* gerais, como por exemplo, a informação de que, em 2017, os municípios brasileiros possuíam população estimada média de 37.282 habitantes, mas uma mediana de 11.636, a partir dos dados do Atlas da Violência (Ipea, 2019). Trata-se de um bom exemplo de como *outliers* podem afetar a leitura dos dados a partir de medidas centrais. Em seguida, foram elaborados gráficos para visualizar a distribuição de dados: um *boxplot* e um histograma, ambos mostrando a taxa de homicídios por 100 mil habitantes, a principal métrica do estudo.

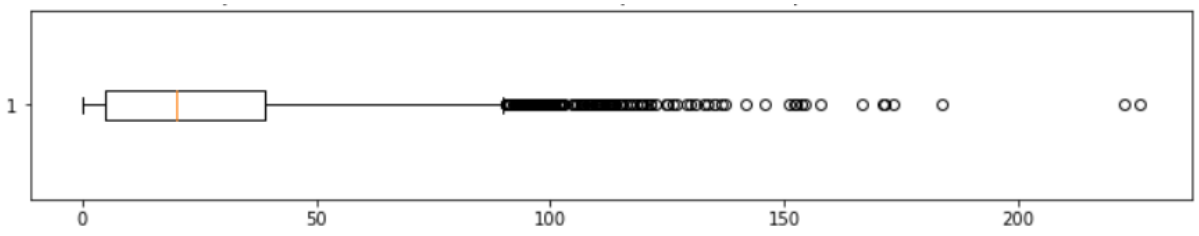

Figura 8 – *Boxplot* da taxa de homicídios dos municípios brasileiros por 100 mil habitantes

Fonte: Elaboração própria

A partir dessa análise, podemos observar que a mediana – representada pela linha vermelha dentro da caixa – está em torno de 20. Já o primeiro e terceiro quartis, entre aproximadamente 5 e 39. O intervalo entre as caudas, ou entre o mínimo e máximo do *boxplot*, vai de zero até um pouco acima de 90, como mostra a Figura 8.

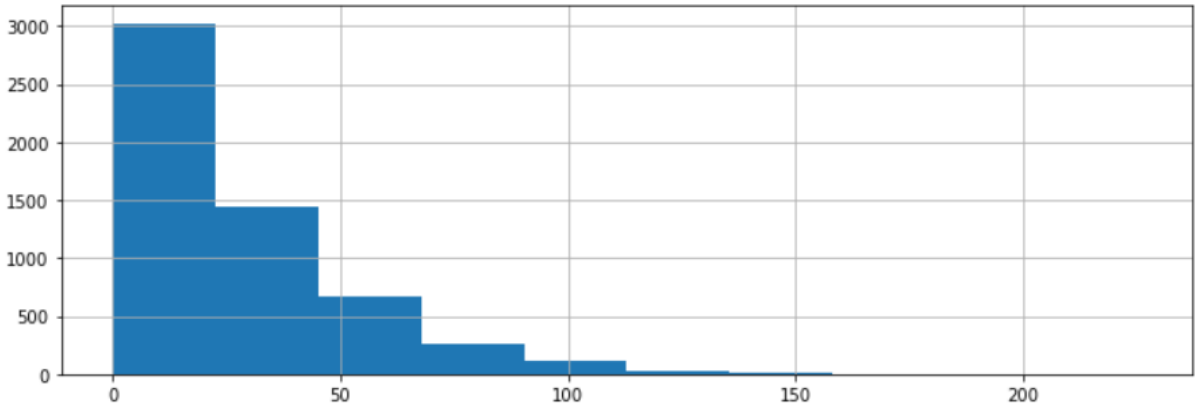

Figura 9 – Histograma da taxa de homicídios dos municípios brasileiros por 100 mil habitantes
Fonte: Elaboração própria

O histograma, apresentado na Figura 9, cujo agrupamento foi realizado de forma automática por meio da função *hist* da interface *Pyplot* da biblioteca *Matplotlib*, demonstra que as ocorrências são numerosas onde a taxa de homicídios é mais baixa, e se distribuem de forma mais heterogênea quando os valores são mais altos, tornando o formato da curva de distribuição de frequências próximo do exponencial.

4.3. Pré-processamento

Em seguida, foi feito o pré-processamento, utilizando técnicas de tratamento de dados como tratamento de valores faltantes, detecção de outliers, balanceamento, normalização e seleção de atributos. No entanto, a primeira etapa dessa fase é a organização ou limpeza da base. Assim, o conjunto de dados é preparado para as etapas de pré-processamento.

Vale a ressalva de que, neste projeto, as etapas de pré-processamento são usadas para testar o comportamento do modelo, primeiro em uma situação onde elas não são aplicadas, depois em situações onde cada uma delas é aplicada. Ou seja, primeiro os modelos são executados diretamente, sem nenhuma etapa de tratamento. Depois, são executados aplicando cada tipo de tratamento. Em seguida, podem ser testadas outras combinações, ou uma combinação de todos de uma vez só.

4.3.1. Limpeza da base

Inicialmente, foram tomadas decisões sobre os atributos categóricos, mostrados na Tabela 1. Importante observar que nenhum dos atributos categóricos possuía valores faltantes. Dando início à limpeza, a coluna *CITY*, com os nomes dos municípios, foi removida por não fazer sentido para a análise. A coluna *STATE*, com as siglas de cada unidade federativa (UF), foi substituída por um ID numérico, igual aos dois primeiros dígitos do ID do município, no mesmo padrão do IBGE (2019).

Tabela 1 – Atributos categóricos e as respectivas quantidades de categorias

Atributo	**Nº de categorias**
CITY	5.298
REGIAO_TUR	323
STATE	27
GVA_MAIN	10
CATEGORIA_TUR	6
RURAL_URBAN	6

Fonte: Elaboração própria

Restaram, portanto, quatro atributos categóricos a serem tratados: *REGIAO_TUR*, *GVA_MAIN*, *CATEGORIA_TUR*, *RURAL_URBAN*. Havia duas opções principais – variáveis *dummy* ou binárias. Em ambos os casos, são geradas variáveis que assumem o valor 1 ou 0. Mas há uma diferença crucial: nas variáveis *dummy*, a relação entre categorias e colunas é de um para um. Já nas variáveis binárias, para cada 2^n categorias, são criadas *n* colunas.

Tabela 2 – Comparação da quantidade de categorias criadas em cada método

Atributo	***Dummy***	**Binárias**
REGIAO_TUR	323	9
GVA_MAIN	10	4
CATEGORIA_TUR	6	3
RURAL_URBAN	6	3
Total	**345**	**19**

Fonte: Elaboração própria

Quando há duas categorias, basicamente não há diferença prática entre as abordagens. No entanto, quando esse número aumenta, a vantagem das variáveis binárias se torna

considerável. A decisão foi, portanto, pelas variáveis binárias, o que gerou 19 novos atributos, como mostra a Tabela 2. Assim, somando os atributos numéricos originais, o ID da UF e os binários, o total passa a ser 102. Por fim, os dados de homicídios, usados para gerar a variável de classificação, foram removidos, para não influenciar os modelos:

- *HOMICIDES*;
- *HIDDEN_HOMICIDES* e
- *HOMICIDE_RATE*.

4.3.2. Tratamento de valores faltantes

A etapa seguinte é o tratamento de valores faltantes. Dos 82 atributos originalmente numéricos, 29 continham valores faltantes (*missing values*, em inglês). Desses, em 26 a quantidade de *missing* correspondia a 0,14% de seus registros ou menos, como mostra a Figura 10. Nesses casos, os valores faltantes, em grande maioria, eram relativos a municípios que foram criados após o levantamento dos dados – por exemplo, municípios que se emanciparam após o Censo 2010, fonte de grande parte das informações analisadas.

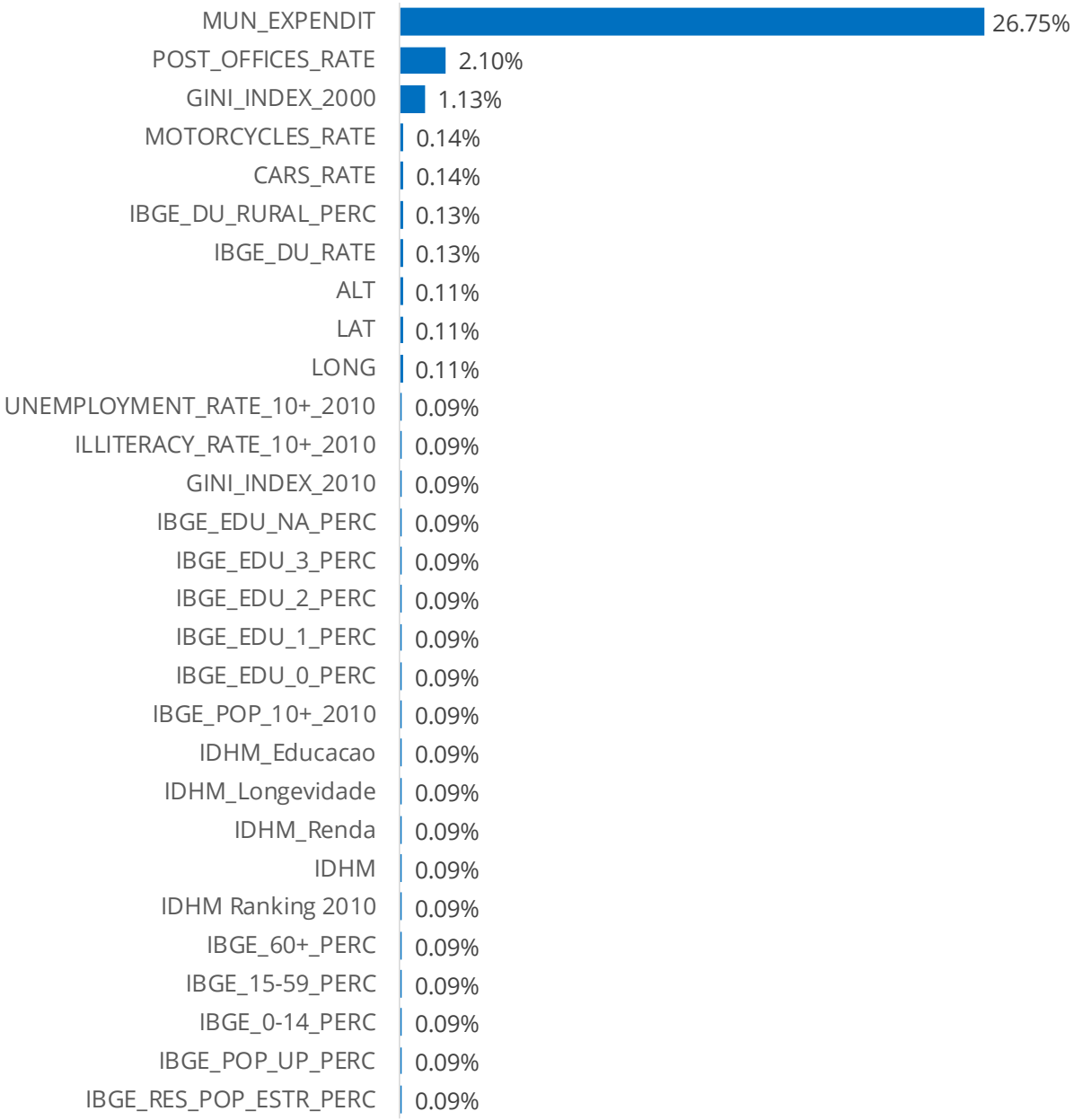

Figura 10 – Proporção de valores faltantes por atributo

Fonte: Elaboração própria

Entre os métodos mais comuns para tratamento de valores faltantes, estão três: eliminação de registros (linhas), substituição pela média – em alguns casos, pela mediana, ou pela moda, em categóricos – e eliminação de variáveis (colunas). No entanto, há risco de enviesar os resultados no primeiro e segundo métodos e, no terceiro, o potencial preditivo do modelo pode ser reduzido (ASSUNÇÃO, 2012, p. 9).

Em contrapartida, desenvolver modelos complexos para aplicar apenas no tratamento de valores faltantes, sobretudo em uma proporção tão reduzida quanto na base estudada, pode

representar um dispêndio desnecessário de tempo, recursos e processamento, com pouco ou nenhum resultado. Portanto, é preciso buscar o equilíbrio nesse quesito.

Nesse cenário, é importante observar que todos os atributos com valores faltantes são numéricos. Assim, para tomar decisões adequadas para esse tratamento, as variáveis foram divididas em quatro grupos, de acordo com a natureza dos dados: Taxa ou índice, Percentual composto, Geográfica e Valor absoluto. O detalhamento se encontra na Figura 11.

Taxa ou índice (0,28%)
IBGE_RES_POP_ESTR_PERC
IBGE_DU_RATE
IBGE_DU_RURAL_PERC
IBGE_POP_UP_PERC
IDHM Ranking 2010
IDHM
IDHM_Renda
IDHM_Longevidade
IDHM_Educacao
CARS_RATE
MOTORCYCLES_RATE
POST_OFFICES_RATE
IBGE_POP_10+_2010
GINI_INDEX_2000
GINI_INDEX_2010
ILLITERACY_RATE_10+_2010
UNEMPLOYMENT_RATE_10+_2010

Percentual composto (0,09%)
IBGE_0-14_PERC
IBGE_15-59_PERC
IBGE_60+_PERC
IBGE_EDU_0_PERC
IBGE_EDU_1_PERC
IBGE_EDU_2_PERC
IBGE_EDU_3_PERC
IBGE_EDU_NA_PERC

Geográfica (0,11%)
LONG
LAT
ALT

Valor absoluto (26,75%)
MUN_EXPENDIT

Figura 11 – Grupos de tratamento de valores faltantes
Fonte: Elaboração própria

Nos dois primeiros grupos, a média da proporção de valores faltantes por atributo é de 0,28% e 0,09%, respectivamente. Esses grupos se assemelham por serem compostos de variáveis relativas, como taxas proporcionais à população, índices que variam em uma escala limitada – caso do grupo com mais variáveis – ou percentuais onde uma combinação de atributos totaliza 1 (ou 100%) em cada linha – motivo do nome do segundo grupo.

As características que ambos possuem em comum – baixa proporção de valores faltantes e valores relativos, que, diferentemente de valores absolutos, podem fazer sentido para municípios de qualquer tamanho – tornam a decisão de substituir *missings* por medidas centrais mais confortável. O que os difere é a forma de tratamento: no grupo **Taxa ou índice**, a substituição é pela mediana, com objetivo de mitigar os efeitos da presença de *outliers*. Já

no grupo **Percentual composto**, a substituição é pela média, pois esta garante que, nesses casos, a combinação de colunas retornará a soma 100%. Quaisquer diferenças seriam fruto de arredondamentos.

Na sequência, o grupo de variáveis **Geográficas** – composto de latitude, longitude e altitude – também precisa de um tratamento especial. Porém, se alguma medida central de todos os registros fosse aplicada, apontaria para uma posição central no mapa do país, o que muito provavelmente não faria sentido para as UFs correspondentes a cada um dos registros. Portanto, o método aplicado é a substituição pela média dos municípios localizados na mesma UF e na mesma região de turismo.

Por fim, o grupo **Valor absoluto** contém apenas um atributo, *MUN_EXPENDIT*, correspondente às despesas dos municípios, que tem o maior número de *missings* (26,75%). Essa proporção é grande o suficiente para evitar a eliminação de linhas, porém pequena o suficiente para evitar eliminar a coluna – não podemos esquecer que são 73,25% de valores presentes.

A substituição por medidas centrais parece um caminho pouco preciso e com risco de forte enviesamento – caso sejam usadas métricas de todos os registros da coluna – ou excessivamente complexo e pouco efetivo – na ocasião de escolher atributos secundários para agrupar os registros, como nas Geográficas.

Sendo assim, uma opção equilibrada é determinar os valores faltantes dessa variável a partir de uma regressão linear, usando um subconjunto dos dados onde não há *missings* nessa coluna e deixando de utilizar nessa etapa os atributos diretamente relacionados a homicídios (total de homicídios registrados, total de homicídios ocultos, taxa de homicídios), para minimizar o risco de influenciar indevidamente o resultado final do modelo.

Vale o adendo de que a regressão linear empregada nesse caso não tem o mesmo rigor de um modelo de previsão, pois, no tratamento de valores faltantes, a ideia é obter valores aproximados que consigam estimar, com eficácia razoável, o comportamento dos valores faltantes com base nos valores presentes no restante da base.

As variáveis de entrada foram normalizadas usando o pacote *Preprocessing* da biblioteca *Scikit-Learn* e posteriormente filtradas para que fossem consideradas apenas as que apresentassem valores não-nulos na variável de resposta. Após algumas iterações, foram selecionadas as seguintes variáveis para estimar os gastos municipais (variável *MUN_EXPENDIT*):

- *COMP_J*;
- *COMP_L*;

- *COMP_M*;
- *COMP_N*;
- *COMP_O*;
- *COMP_R*;
- *GDP*;
- *GVA_INDUSTRY*;
- *GVA_SERVICES*;
- *HOMICIDES*;
- *PAY_TV*;
- *Pr_Agencies*;
- *Pu_Agencies* e
- *Pu_Bank*.

Após utilização do pacote *LinearRegression* da biblioteca *Scikit-Learn*, a equação para determinar o valor de gastos municipais teve o valor aproximado abaixo, onde as variáveis x_1 a x_{14} são as variáveis listadas acima, em ordem, e $\hat{Y}$ é a variável *MUN_EXPENDIT*.

$$\hat{Y} = (-6.96 \cdot 10^{-3} + 7.83x_1 + 1.75x_2 - 9.34x_3 + 6.94x_4 + 1.52x_5 + 4.64x_6 + 5.92x_7 + 2.21x_8 + 4.82x_9 + 1.06x_{10} + 10.22x_{11} + 11.23x_{12} - 2.53x_{13} + 0.118x_{14}) \cdot 10^9$$

4.3.3. Tratamento de *outliers*

Na etapa de tratamento de *outliers*, foram combinadas as técnicas de normalização, seleção de atributos (*feature selection*) e agrupamento (*clustering*) usando agrupamento espacial de baseado em densidade de aplicações com ruído, ou *density-based spatial clustering of applications with noise* (DBSCAN).

A partir da base sem valores faltantes, foi gerada uma matriz normalizada – ou seja, com valores entre 0 e 1 – usando novamente o pacote *Preprocessing* da biblioteca *Scikit-Learn*. A ideia é padronizar os valores para estabelecer uma base única de comparação entre cada atributo.

O passo seguinte é estabelecer quais atributos serão utilizados para a análise de *outliers*. Para isso, foi usada, a partir do pacote *Feature_selection* do *Scikit-Learn*, a técnica de seleção de atributos por limite de variância (ou *variance threshold*), que consiste em considerar apenas os atributos que atingem um valor mínimo de variância, descrito na equação abaixo. O valor foi obtido heuristicamente, a partir de testes com outros valores, como 0.7, 0.8, 0.9 e 0.95 no lugar de 0.72.

$$V_{min} = 0.72\ (1 - 0.72)$$

A partir dessa seleção, foram selecionados 10 atributos, dentro do universo de 105. Em outros testes, também foi avaliada a técnica *Principal Component Analysis* (PCA), que reduz a dimensionalidade – ou seja, a quantidade de atributos –, através da combinação de colunas. No entanto, na combinação de parâmetros escolhida, o PCA não se fez necessário, por já ter havido redução significativa na dimensionalidade e os resultados já se mostravam satisfatórios.

Por fim, foi aplicado o DBSCAN, com objetivo de agrupar os valores dos atributos baseando-se na densidade dos grupos, a partir do cálculo de distâncias euclidianas para os pontos vizinhos e, dessa forma, determinar como *outliers* os que estivessem mais distantes, considerados ruído. No total, foram encontrados 561 *outliers*, que representam cerca de 10% do total de registros.

4.3.4. Balanceamento

Em modelos de classificação, é muito comum que existam classes com uma quantidade maior de registros que outras. Quando isso ocorre, diz-se que a base de dados está desbalanceada. O desbalanceamento pode gerar erros de classificação e enviesamento dos resultados. Por isso, é importante avaliar a necessidade e os impactos de aplicar o balanceamento.

Neste desenvolvimento, foi escolhida a técnica de super-amostragem sintética da minoria, ou *Synthetic Minority Over-sampling Technique* (Smote), que consiste em gerar sinteticamente valores da classe rara, com o objetivo de balancear a base. Antes, a base tinha uma proporção de $3815/5570 = 68{,}5\%$ de registros da classe dominante (*endemic*, classe 1), como mostra a Figura 12. Depois do balanceamento, a proporção foi equilibrada com os dados sintéticos, explicitados na Figura 13.

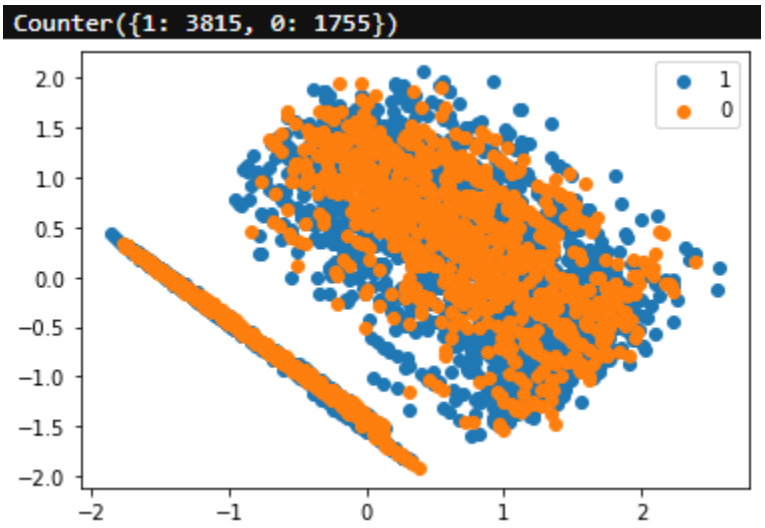

Figura 12 – Representação dos dados antes do balanceamento

Fonte: Elaboração própria

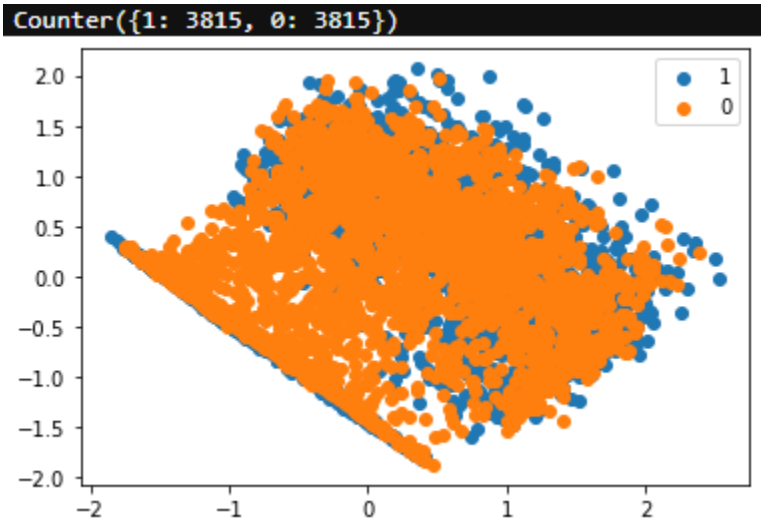

Figura 13 – Representação dos dados após o balanceamento

Fonte: Elaboração própria

4.3.5. Normalização

Conforme mencionado em etapas anteriores, a etapa de normalização é feita para que os valores do *dataset* passem a se situar entre 0 e 1. Para isso, em cada atributo, é feito um cálculo para que o valor mínimo passe a ser 0 e o valor máximo passe a ser 1. A transformação é feita usando o módulo *MinMaxScaler* do pacote *Preprocessing* da biblioteca *Scikit-Learn*. O cálculo realizado está descrito na equação abaixo.

$$x_{norm} = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

Onde *i* é o índice de cada linha e os máximos e mínimos são relativos a cada coluna, ou atributo. Ou seja, o valor normalizado de uma célula é a diferença entre o valor original da célula menos o mínimo daquela coluna dividida pela diferença entre o máximo e o mínimo da coluna.

4.3.6. Seleção de atributos

A etapa de seleção de atributos, referida em inglês como *feature selection*, consiste em selecionar os atributos que mais contribuem para a variância acumulada das classes de saída e, dessa forma, aumentar a probabilidade de aprimorar a acurácia do modelo. Foi utilizada a mesma variância-limite aplicada na análise de *outliers*.

$$V_{min} = 0.72\ (1 - 0.72)$$

A saída dessa etapa gerou uma tabela reduzida com 10 atributos:

- *CATEGORIA_TUR_2*;
- *CATEGORIA_TUR_3*;
- *GVA_MAIN_3*;
- *GVA_MAIN_4*;
- *REGIAO_TUR_6*;
- *REGIAO_TUR_7*;
- *REGIAO_TUR_8*;
- *REGIAO_TUR_9*;
- *RURAL_URBAN_2* e
- *RURAL_URBAN_3*.

4.3.7. Redução de dimensionalidade

Na execução da redução de dimensionalidade, a ideia é reduzir a quantidade de atributos sem perda de informação. O principal método para essa etapa é o PCA. Nele, é selecionada a quantidade desejada de dimensões e vários atributos originais são combinados em novos atributos, em uma quantidade reduzida.

Nessa etapa, não há descarte de dados. No entanto, a informação torna-se descaracterizada, tanto em rótulos, quanto em valores. Foi determinada heuristicamente a quantidade de 10 dimensões, numeradas de zero a nove, na Figura 14.

	0	1	2	3	4	5	6	7	8	9
0	-6.028688e+09	-2.641404e+09	-4.657715e+07	149654.198685	-68496.363586	-65049.359855	56241.077598	19015.024489	-45508.568643	-65292.027690
1	-6.028711e+09	-2.641417e+09	-5.689417e+07	-10553.603612	-87880.503080	-112939.795301	54558.139854	38755.214936	-31506.227319	-29796.886831
2	-5.984299e+09	-2.580811e+09	-3.746338e+07	-10557.986287	22722.811211	-80508.761627	58834.649015	-20822.840140	-16490.930592	-39585.350560
3	-5.922680e+09	-2.282021e+09	-1.465254e+07	132914.637172	24290.933468	197140.789883	100012.834432	-103220.991260	647.021083	-17157.647933
4	-5.820802e+09	-1.864826e+09	7.854364e+07	382544.320667	-43044.565472	301070.239941	41830.785500	-332579.716560	372360.977127	557895.410988
...	...	...	...	...	...	...	...	...	...	...
5565	-5.931647e+09	-2.377551e+09	-1.228002e+07	279727.388146	19298.658877	89744.346066	84895.022726	-94039.823937	46075.325038	-53444.539544
5566	-6.028732e+09	-2.641428e+09	-6.630899e+07	-63958.201148	40611.000074	-123177.042109	55367.314370	32381.173891	-43909.756549	-67349.639858
5567	-6.028711e+09	-2.641417e+09	-5.693806e+07	37339.246919	94214.684174	-69724.067521	57966.871619	14823.715773	-41750.009422	-137.528456
5568	-5.906216e+09	-2.122921e+09	1.727640e+06	667246.323074	309873.241027	270203.082696	70370.533803	-142146.478426	-3766.864515	-39526.718005
5569	-6.028714e+09	-2.641419e+09	-5.843817e+07	-17196.985566	37593.890773	-94310.158326	63719.826795	17551.353353	-39935.717989	-26817.750414

Figura 14 – Captura de tela do *dataframe* auxiliar gerado após aplicação de PCA

Fonte: Elaboração própria

4.4. Aplicação de algoritmos

Inicialmente, cada etapa de pré-processamento foi aplicada uma de cada vez para ser avaliada individualmente em cada algoritmo. Para cada etapa de pré-processamento, foram gerados *dataframes* auxiliares. Todos foram divididos em treino e teste, onde 25% dos dados foram selecionados aleatoriamente para compor a base de teste.

Essa etapa, assim como a implementação dos algoritmos, foi realizada a partir de pacotes da biblioteca *Scikit-Learn*. Para a separação em bases de treino e teste, foi empregado o módulo *Train_test_split* do pacote *Model_selection*. Com objetivo de gerar as métricas de avaliação e a matriz de confusão em todos os algoritmos, foram importados o pacote *Metrics* e a biblioteca *Scikitplot*, respectivamente.

Na regressão logística, foi usado o módulo *LogisticRegression* do pacote *Linear_model*. No SVM, foi importado o pacote de mesmo nome. Já para o *Random Forest*, o módulo *RandomForestClassifier* do pacote *Ensemble* foi utilizado. Por fim, no k-NN, foi usado o módulo *KNeighborsClassifier* do pacote *Neighbors*.

4.5. Avaliação de modelos

Cada modelo foi avaliado a partir da matriz de confusão e com base em quatro métricas principais: acurácia, precisão, *recall* e F1 *score*. Segundo Wang, Hu & Zhai (2018), cada uma dessas métricas é calculada a partir das predições corretas, mas com algumas diferenças.

$$Acurácia = \frac{Número\ de\ predições\ corretas}{Total\ de\ predições}$$

$$Precisão_C = \frac{Número\ de\ predições\ corretas\ da\ classe\ c}{Número\ de\ predições\ como\ classe\ c}$$

$$Recall_C = \frac{Número\ de\ predições\ corretas\ da\ classe\ c}{Número\ de\ elementos\ da\ classe\ c}$$

$$F1\ score = \frac{1}{2}\ 2\sum_{c=0}^{1}\frac{Recall_C \times Precisão_C}{Recall_C + Precisão_C}$$

A matriz de confusão consiste em cruzar as classificações reais com as classificações preditas. No caso da Figura 15, existem muitas classificações incorretas da classe 0. O perfil de matriz de confusão desejado é o que tem mais classificações corretas, ou seja, onde as classificações reais coincidem com as da predição.

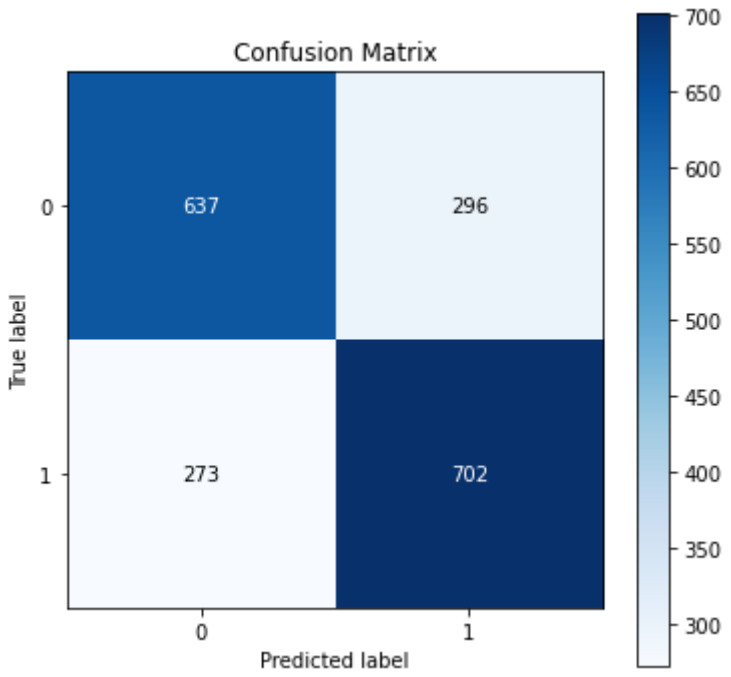

Figura 15 – Exemplo de matriz de confusão

Fonte: Elaboração própria

No exemplo acima, as métricas seriam calculadas da seguinte forma:

- a acurácia seria de $(702 + 637)/(702 + 637 + 296 + 273) = 0.7018$;
- a precisão da classe 1 seria de $702/(702 + 296) = 0.7034$;
- o *recall* da classe 1 seria de $702/(702 + 273) = 0.7200$ e
- o F1 *score* da classe 1 totalizaria $2 \times (0.7200 \times 0.7034)/(0.7200 + 0.7034) = 0.7116$.

A interpretação dessas métricas se baseia em analisar o desempenho de cada modelo a partir de seus resultados. A acurácia é a métrica mais simples para indicar quantas predições foram feitas corretamente. Já uma baixa precisão indica muitos falsos positivos – chamados de erro do tipo I – e um baixo *recall* denota alta ocorrência de falsos negativos – conhecidos como erro do tipo II.

Por sua vez, o F1 *score*, por definição, é uma média ponderada da precisão e do *recall*. No entanto, deve ser usado com cautela. Caso haja grande diferença entre as proporções de falsos negativos e falsos positivos, é fundamental que precisão e *recall* estejam presentes individualmente na análise.

Na primeira rodada de resultados, cada etapa de pré-processamento foi executada três vezes, para comparar as transformações de forma assertiva. Os resultados foram armazenados em uma tabela, onde as células com os maiores resultados apresentam tons mais fortes de verde. Para manter os registros mais legíveis, o conteúdo foi dividido entre as Tabelas 3 a 6.

Tabela 3 – Resultados da primeira rodada de pré-processamento com Regressão Logística

		Regressão Logística			
	Tentativa	**Acurácia**	**Precisão**	**Recall**	**F1**
Valores faltantes	#1	0.6820	0.6820	1.0000	0.8109
Valores faltantes	#2	0.6798	0.6798	1.0000	0.8094
Valores faltantes	#3	0.7021	0.7021	1.0000	0.8250
Outliers	#1	0.6752	0.6752	1.0000	0.8061
Outliers	#2	0.6872	0.6877	0.9988	0.8146
Outliers	#3	0.6983	0.6983	1.0000	0.8224
Balanceamento	#1	0.5105	0.5105	1.0000	0.6759
Balanceamento	#2	0.4921	0.4921	1.0000	0.6596
Balanceamento	#3	0.4953	0.4953	1.0000	0.6625
Normalização	#1	0.7358	0.7618	0.8942	0.8227
Normalização	#2	0.7358	0.7613	0.9041	0.8266
Normalização	#3	0.7337	0.7571	0.8980	0.8215
Seleção de atributos	#1	0.6913	0.7143	0.9186	0.8037
Seleção de atributos	#2	0.6906	0.6954	0.9661	0.8087
Seleção de atributos	#3	0.7035	0.7113	0.9583	0.8165
PCA	#1	0.5032	0.8532	0.3252	0.4709
PCA	#2	0.4192	0.8652	0.1640	0.2757
PCA	#3	0.4221	0.8537	0.1842	0.3030

Fonte: Elaboração própria

Tabela 4 – Resultados da primeira rodada de pré-processamento com SVM

		SVM			
	Tentativa	**Acurácia**	**Precisão**	**Recall**	**F1**
Valores faltantes	#1	0.7035	0.7035	1.0000	0.8260
Valores faltantes	#2	0.6784	0.6784	1.0000	0.8084
Valores faltantes	#3	0.6863	0.6863	1.0000	0.8140
Outliers	#1	0.6983	0.6983	1.0000	0.8224
Outliers	#2	0.6744	0.6744	1.0000	0.8055
Outliers	#3	0.6776	0.6776	1.0000	0.8078
Balanceamento	#1	0.5147	0.8824	0.0160	0.0314
Balanceamento	#2	0.5000	1.0000	0.0083	0.0165
Balanceamento	#3	0.4969	1.0000	0.0041	0.0083
Normalização	#1	0.7373	0.7508	0.9320	0.8316
Normalização	#2	0.7251	0.7363	0.9306	0.8221
Normalização	#3	0.7229	0.7385	0.9153	0.8174
Seleção de atributos	#1	0.6992	0.7028	0.9684	0.8145
Seleção de atributos	#2	0.6892	0.6870	0.9817	0.8083
Seleção de atributos	#3	0.6949	0.6941	0.9754	0.8110
PCA	#1	0.6676	0.6676	1.0000	0.8007
PCA	#2	0.6805	0.6805	1.0000	0.8099
PCA	#3	0.6820	0.6820	1.0000	0.8109

Fonte: Elaboração própria

Tabela 5 – Resultados da primeira rodada de pré-processamento com *Random Forest*

		Random Forest			
	Tentativa	**Acurácia**	**Precisão**	**Recall**	**F1**
Valores faltantes	#1	0.7444	0.7431	0.9592	0.8374
Valores faltantes	#2	0.7330	0.7376	0.9476	0.8295
Valores faltantes	#3	0.7337	0.7287	0.9672	0.8311
Outliers	#1	0.7103	0.7130	0.9487	0.8141
Outliers	#2	0.7215	0.7235	0.9452	0.8196
Outliers	#3	0.7326	0.7365	0.9448	0.8278
Balanceamento	#1	0.7049	0.6951	0.7279	0.7111
Balanceamento	#2	0.7144	0.7085	0.7212	0.7148
Balanceamento	#3	0.7002	0.6897	0.7370	0.7126
Normalização	#1	0.7394	0.7388	0.9680	0.8380
Normalização	#2	0.7337	0.7350	0.9537	0.8302
Normalização	#3	0.7150	0.7211	0.9449	0.8180
Seleção de atributos	#1	0.6813	0.6813	1.0000	0.8104
Seleção de atributos	#2	0.6834	0.6834	1.0000	0.8119
Seleção de atributos	#3	0.6927	0.6927	1.0000	0.8185
PCA	#1	0.7071	0.7055	0.9938	0.8252
PCA	#2	0.7093	0.7164	0.9584	0.8199
PCA	#3	0.7014	0.7064	0.9752	0.8193

Fonte: Elaboração própria

Tabela 6 – Resultados da primeira rodada de pré-processamento com k-NN

		k-NN			
	Tentativa	**Acurácia**	**Precisão**	**Recall**	**F1**
Valores faltantes	#1	0.6547	0.7257	0.7937	0.7582
Valores faltantes	#2	0.6597	0.7188	0.8205	0.7663
Valores faltantes	#3	0.6820	0.7488	0.8231	0.7842
Outliers	#1	0.6592	0.7189	0.8132	0.7632
Outliers	#2	0.6720	0.7322	0.8248	0.7758
Outliers	#3	0.6409	0.7278	0.7760	0.7511
Balanceamento	#1	0.6834	0.7301	0.6027	0.6603
Balanceamento	#2	0.7170	0.7497	0.6379	0.6893
Balanceamento	#3	0.6792	0.6957	0.6265	0.6592
Normalização	#1	0.7093	0.7628	0.8454	0.8020
Normalização	#2	0.6985	0.7470	0.8444	0.7927
Normalização	#3	0.7042	0.7575	0.8310	0.7925
Seleção de atributos	#1	0.6820	0.7207	0.8779	0.7915
Seleção de atributos	#2	0.6719	0.7246	0.8314	0.7743
Seleção de atributos	#3	0.6705	0.7208	0.8509	0.7805
PCA	#1	0.6633	0.7213	0.8226	0.7686
PCA	#2	0.6583	0.7215	0.8030	0.7601
PCA	#3	0.6691	0.7270	0.8242	0.7726

Fonte: Elaboração própria

Na avaliação dos resultados, uma informação bastante importante para ter em mente é que, sem balanceamento, a proporção de ocorrências é de aproximadamente 68,5% registros da classe dominante e 31,5% da classe rara. Com isso, nos casos onde a acurácia e a precisão estão em torno de 0.685 (nas bases desbalanceadas) ou 0.5 (na base balanceada) e o *recall* está muito próximo de 1, significa que o modelo está classificando praticamente todos os resultados com a classe dominante.

4.5.1. Após tratar valores faltantes e *outliers*

A etapa de tratamento de valores faltantes foi a primeira. O *dataframe* gerado nessa etapa foi usado como base para as demais etapas. É possível notar na Tabela 7 que o tratamento de *outliers*, realizado logo em seguida, não impactou significativamente os resultados em nenhum dos quatro algoritmos.

Tabela 7 – Média dos resultados iniciais após tratar *missing* e *outliers*

		Valores faltantes	Outliers
Regressão Logística	**Acurácia**	0.6880	0.6869
	Precisão	0.6880	0.6871
	Recall	1.0000	0.9996
	F1	0.8151	0.8143
SVM	**Acurácia**	0.6894	0.6834
	Precisão	0.6894	0.6834
	Recall	1.0000	1.0000
	F1	0.8161	0.8119
Random Forest	**Acurácia**	0.7370	0.7215
	Precisão	0.7364	0.7243
	Recall	0.9580	0.9462
	F1	0.8327	0.8205
k-NN	**Acurácia**	0.6655	0.6574
	Precisão	0.7311	0.7263
	Recall	0.8124	0.8047
	F1	0.7696	0.7634

Fonte: Elaboração própria

Nessa parte, os modelos com *Random Forest* e k-NN tiveram desempenho ligeiramente superior aos demais. As respectivas matrizes de confusão estão apresentadas nas Figuras 16 e 17, abaixo.

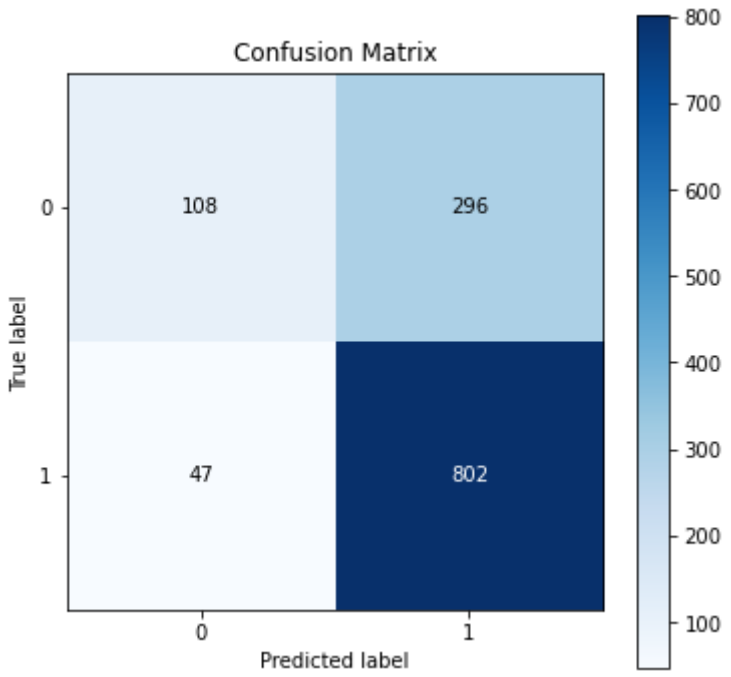

Figura 16 – Matriz de confusão do modelo com *Random Forest* após tratamento de *outliers*

Fonte: Elaboração própria

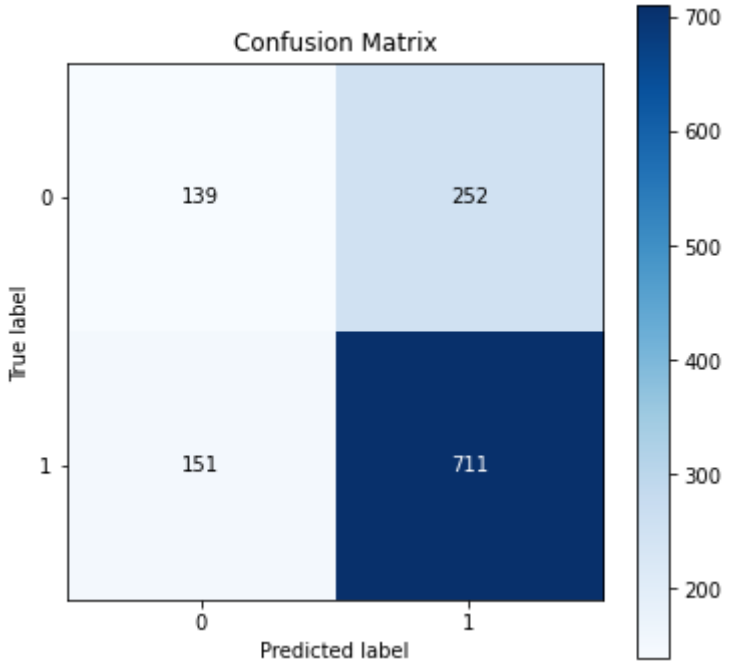

Figura 17 – Matriz de confusão do modelo com *k-NN* após tratamento de *outliers*

Fonte: Elaboração própria

Nesse ponto, o k-NN, apesar de ter acurácia mais baixa que todos os demais – média de 0.67 contra 0.69 da Regressão Logística e SVM e 0.74 do *Random Forest* –, mostra uma boa

vantagem ao ter *recall* mais baixo, nesse contexto. Na Regressão Logística e no SVM, praticamente não há ocorrência de falsos negativos porque o modelo praticamente não aponta negativos. Ou seja, nesse contexto, até esse ponto, usando Regressão Logística e SVM, os respectivos modelos encontraram dificuldade em diferenciar as classes.

Os modelos com *Random Forest* e k-NN classificaram corretamente mais elementos da classe rara. No caso do k-NN, o desempenho foi um pouco melhor nesse aspecto. Isso significa que, até esse ponto, o modelo com k-NN mostra mais efetividade ao diferenciar as classes. Isso é refletido, em parte, na precisão, que possui níveis semelhantes em ambos. Na matriz da Figura 16, o modelo *Random Forest* gerou 12% de predições na classe rara, enquanto, na Figura 17, o modelo com k-NN gerou 23% de predições na classe rara.

4.5.2. Após balanceamento

Nessa etapa, ao balancear os rótulos das classes, a Tabela 8 mostra que a Regressão Logística continuou com o mesmo comportamento: classificou todos com a classe dominante. A acurácia e precisão diminuíram em comparação às etapas anteriores porque a proporção de elementos com a classe dominante diminuiu para 50%. O SVM, por sua vez, inverteu: passou a classificar quase todos como classe rara. Por isso, o *recall* foi tão baixo e a precisão tão alta.

Tabela 8 – Média dos resultados iniciais após balanceamento

		Balanceamento
Regressão Logística	Acurácia	0.4993
	Precisão	0.4993
	Recall	1.0000
	F1	0.6660
SVM	Acurácia	0.5038
	Precisão	0.9608
	Recall	0.0095
	F1	0.0187
Random Forest	Acurácia	0.7065
	Precisão	0.6978
	Recall	0.7287
	F1	0.7128
k-NN	Acurácia	0.6932
	Precisão	0.7251
	Recall	0.6223
	F1	0.6696

Fonte: Elaboração própria

A Tabela 8 e a Figura 18 explicitam que o *Random Forest* apresentou comportamento similar à etapa anterior, mas com redução no *recall* e com predição relativamente balanceada em cada classe: em torno de 50% para cada. Ou seja, o modelo aprimorou a capacidade de diferenciação de classes. Aparentemente, trata-se de um efeito direto do balanceamento.

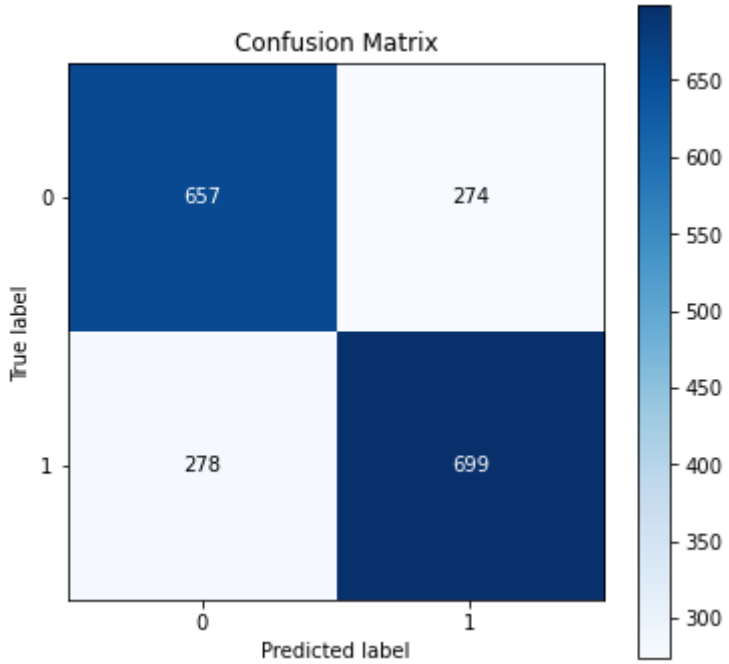

Figura 18 – Matriz de confusão do modelo com *Random Forest* após balanceamento

Fonte: Elaboração própria

Já o k-NN, por sua vez, teve redução drástica no *recall* porque passou a classificar muitos falsos negativos e passou a classificar mais registros como pertencentes à classe rara: cerca de 60%, contra 40%, como mostra a Figura 19. É um sinal de alerta para os passos seguintes.

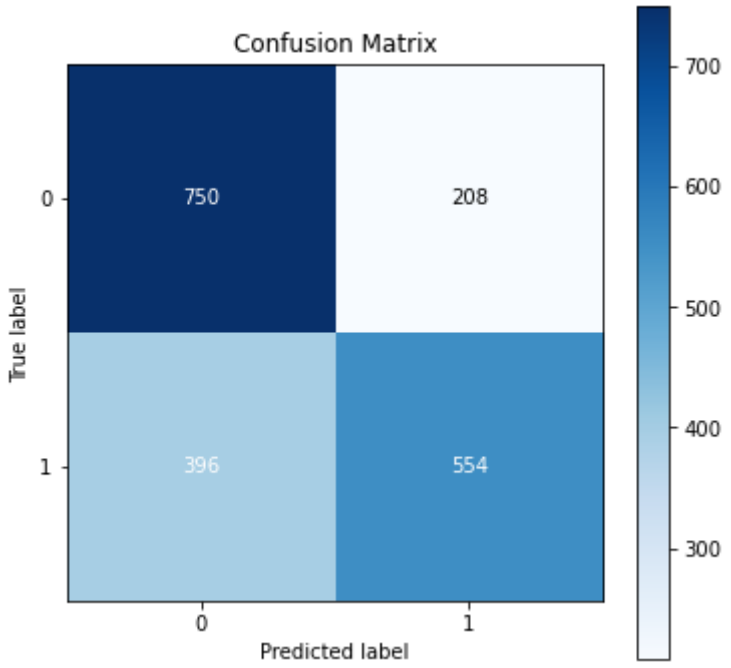

Figura 19 – Matriz de confusão do modelo com k-NN após balanceamento

Fonte: Elaboração própria

4.5.3. Após normalização

Com a normalização, a comparação entre algoritmos se torna mais equilibrada, como demonstra a Tabela 9. No entanto, nesses casos, índices mais altos de *recall* mais alto significaram maior concentração de classificações na classe dominante. As Figuras 20 e 21 mostram as matrizes de confusão de dois modelos, onde é possível observar como esse comportamento se manifesta.

Tabela 9 – Média dos resultados iniciais após normalização

		Normalização
Regressão Logística	Acurácia	0.7351
	Precisão	0.7601
	Recall	0.8988
	F1	0.8236
SVM	Acurácia	0.7284
	Precisão	0.7419
	Recall	0.9259
	F1	0.8237
Random Forest	Acurácia	0.7294
	Precisão	0.7316
	Recall	0.9556
	F1	0.8287
k-NN	Acurácia	0.7040
	Precisão	0.7557
	Recall	0.8403
	F1	0.7957

Fonte: Elaboração própria

O pior caso é o *Random Forest*, na Figura 20, com cerca de 90% de registros da base de teste classificados como municípios com violência endêmica (classe 1, dominante). O contraponto é o modelo com k-NN, na Figura 21, com cerca de 75% classificados como classe 1, para uma proporção de 68,5% na base de treino.

Vale observar que o parâmetro de proporção de classificados é um complemento à análise do *recall* e que foi adicionado ao longo da avaliação. Isso se deveu ao índice excessivamente elevado do *recall* em boa parte dos casos e que, por não vir acompanhado de bons resultados em outras métricas, indicou um sintoma de problemas nos outros modelos.

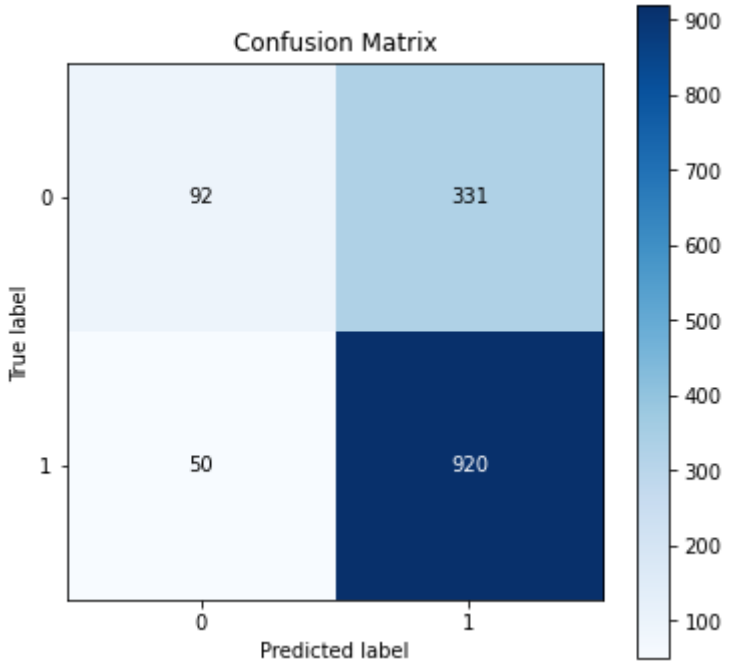

Figura 20 – Matriz de confusão do modelo com *Random Forest* após normalização

Fonte: Elaboração própria

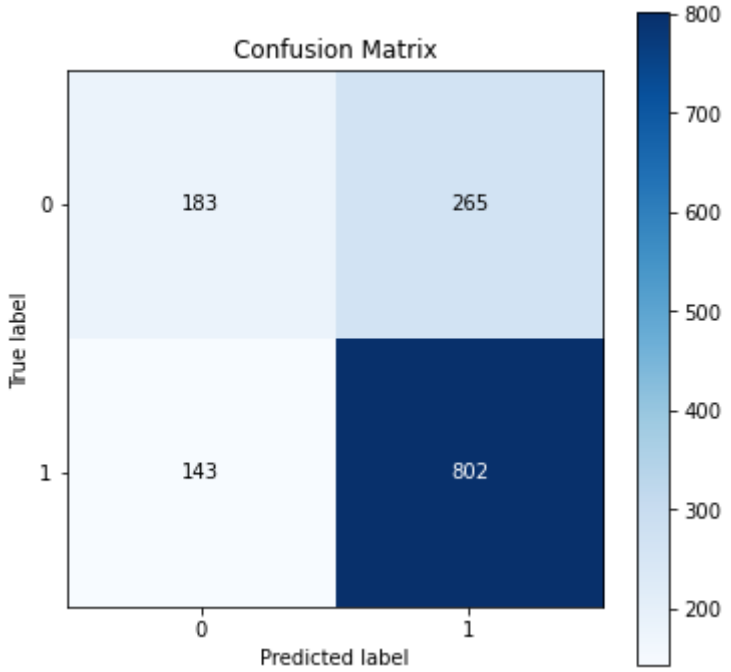

Figura 21 – Matriz de confusão do modelo com k-NN após normalização

Fonte: Elaboração própria

4.5.4. Após seleção de atributos e PCA

Como mostra a Tabela 10, a aplicação de *feature selection* e, em seguida, de PCA, isoladamente, não surtiram efeitos positivos significativos nos modelos. Na Regressão Logística e SVM, os resultados da seleção de atributos foram ligeiramente melhores, mas muito similares aos resultados após tratamento de *missing values* e de *outliers*, assim como o k-NN. Já no *Random Forest*, houve piora. Todos foram classificados como classe dominante.

Tabela 10 – Média dos resultados iniciais após seleção de atributos e PCA

		Seleção de atributos	PCA
Regressão Logística	**Acurácia**	0.6951	0.4482
	Precisão	0.7070	0.8573
	Recall	0.9476	0.2245
	F1	0.8096	0.3499
SVM	**Acurácia**	0.6944	0.6767
	Precisão	0.6946	0.6767
	Recall	0.9752	1.0000
	F1	0.8113	0.8072
Random Forest	**Acurácia**	0.6858	0.7059
	Precisão	0.6858	0.7094
	Recall	1.0000	0.9758
	F1	0.8136	0.8215
k-NN	**Acurácia**	0.6748	0.6636
	Precisão	0.7220	0.7233
	Recall	0.8534	0.8166
	F1	0.7821	0.7671

Fonte: Elaboração própria

Quanto aos efeitos do PCA, no SVM, *Random Forest* e k-NN, os resultados ficaram praticamente inalterados. Na Regressão Logística, houve um caso curioso, mostrado na Figura 22, parecido com o que o SVM sofreu após o balanceamento: a grande maioria dos registros da base de teste passou a ser classificada como classe rara (menos violentos, classe 0).

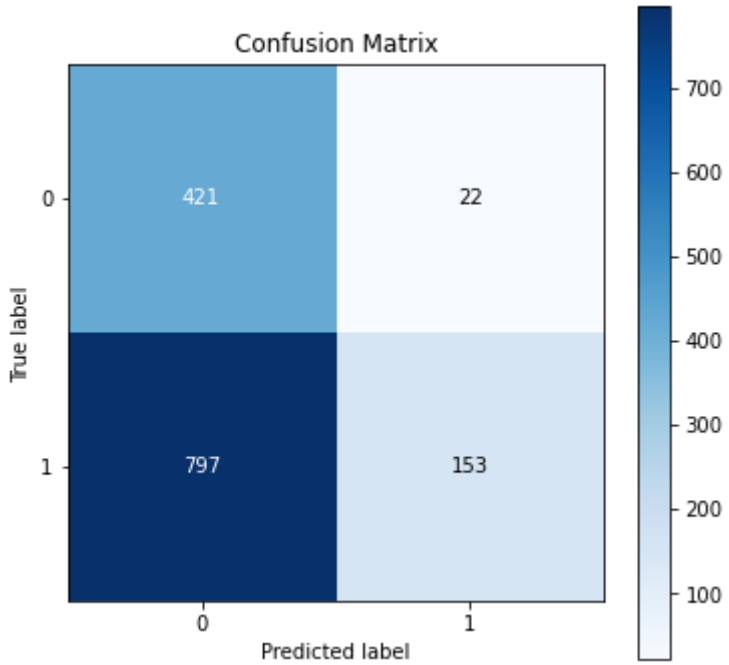

Figura 22 – Matriz de confusão do modelo com Regressão Logística após *PCA*

Fonte: Elaboração própria

4.5.5. Combinação de técnicas de pré-processamento

Na etapa anterior da análise, os resultados de balanceamento e normalização separados se mostraram promissores, especialmente usando *Random Forest*, conforme Tabelas 8 e 9. Com isso, elas foram combinadas para buscar melhorar os resultados, o que de fato aconteceu, como mostra a Tabela 11. Embora os valores isolados mostrem redução em parte das métricas, a precisão e o *recall* anteriores, quando estiveram próximos ou acima de 0.9, eram um sinal de baixa qualidade na diferenciação de classes. Com a combinação – na última coluna –, os modelos melhoraram sua capacidade de classificar e houve maior equilíbrio entre as métricas.

Tabela 11 – Média dos resultados iniciais após combinar balanceamento e normalização

		Balanceamento	Normalização	Balanceamento + Normalização
Regressão Logística	Acurácia	0.4993	0.7351	0.7084
	Precisão	0.4993	0.7601	0.7082
	Recall	1.0000	0.8988	0.7094
	F1	0.6660	0.8236	0.7088
SVM	Acurácia	0.5038	0.7284	0.7682
	Precisão	0.9608	0.7419	0.7464
	Recall	0.0095	0.9259	0.8128
	F1	0.0187	0.8237	0.7782
Random Forest	Acurácia	0.7065	0.7294	0.7091
	Precisão	0.6978	0.7316	0.7002
	Recall	0.7287	0.9556	0.7322
	F1	0.7128	0.8287	0.7158
k-NN	Acurácia	0.6932	0.7040	0.7264
	Precisão	0.7251	0.7557	0.7590
	Recall	0.6223	0.8403	0.6644
	F1	0.6696	0.7957	0.7084

Fonte: Elaboração própria

Em seguida, foram adicionadas outras técnicas de pré-processamento ao estudo. Além do já utilizado PCA, testaram-se *Kernel* PCA e Incorporação Estocástica de Vizinhos t-Distribuída, ou *t-distributed Stochastic Neighbor Embedding* (t-SNE). Em cada uma delas, seus principais hiperparâmetros foram testados. PCA e *Kernel* PCA têm em comum o parâmetro de número de componentes (N), e em ambos foram experimentados os valores de 10% a 90% do total de atributos, ou seja, para 102 *features*, $N \in \{10, 20, 30, 40, 51, 61, 71, 81, 91\}$.

No *Kernel* PCA, há, ainda, o parâmetro Kernel, que pode assumir os valores "*linear*", "*poly*", "*rbf*", "*sigmoid*", "*cosine*" e "*precomputed*". Todos, exceto "*precomputed*", foram testados. Já no t-SNE, os parâmetros testados foram a dimensão do espaço incorporado (números inteiros de 1 a 3) e a perplexidade, que é relativa ao número de vizinhos mais próximos considerados na análise. Recomendam-se valores entre 5 e 50, e o valor padrão é 30. Para *datasets* maiores, devem ser usados valores maiores[5]. Foram testados valores de 5 a 60.

[5] Informação extraída do guia do usuário de implementação de t-SNE, pela *Scikit-Learn*. Disponível em <https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html>. Acessado em 13 set. 2020.

Com relação aos algoritmos, Regressão Logística e k-NN pouco progrediram após os testes combinando balanceamento e normalização. Portanto, as combinações de três ou mais técnicas foram avaliadas apenas com SVM e *Random Forest*, como mostra a Tabela 12.

Tabela 12 – Resultados após combinar três ou mais técnicas de pré-processamento

		Balanceamento + Normalização + PCA-80	Balanceamento + Normalização + Kernel PCA-60%	Balanceamento + Normalização + PCA-80 + t-SNE	Balanceamento + Normalização + Kernel PCA-60% + t-SNE	Balanceamento + Normalização + t-SNE
SVM	**Acurácia**	0.7835	0.7762	0.7799	0.7799	0.7799
	Precisão	0.7628	0.7609	0.7668	0.7668	0.7668
	Recall	0.8169	0.8122	0.8128	0.8128	0.8128
	F1	0.7890	0.7858	0.7892	0.7892	0.7892
Random Forest	**Acurácia**	0.8092	0.8019	0.8035	0.8097	0.7966
	Precisão	0.8224	0.8130	0.8197	0.8275	0.8157
	Recall	0.7841	0.7894	0.7849	0.7890	0.7735
	F1	0.8028	0.8011	0.8019	0.8078	0.7941

Fonte: Elaboração própria

4.5.6. *Cross-validation*

Em todos os casos de combinação entre três ou mais técnicas de pré-processamento, apresentados na Tabela 12, houve muita similaridade nos resultados. No SVM, a combinação com uma ligeira vantagem em relação às demais foi com balanceamento, normalização e PCA com 80 dimensões. Já no modelo com *Random Forest*, foi a combinação de balanceamento, normalização, *Kernel* PCA com a quantidade de dimensões igual a 60% da quantidade de atributos e *kernel sigmoid*, e t-SNE com perplexidade 30.

Nesses casos, os algoritmos foram implementados utilizando os parâmetros padrão. Para buscar variações nos resultados, também foram testadas diversas combinações de hiperparâmetros, utilizando *cross-validation* por meio do módulo *GridSearchCV*, do pacote *Model_selection*, da biblioteca *Scikit-Learn*. Por demandar um alto custo computacional, este teste foi feito com apenas uma combinação de técnicas de pré-processamento para cada algoritmo, que foram justamente as com melhor desempenho na etapa anterior, destacadas na Tabela 12.

As cinco combinações de hiperparâmetros com os melhores resultados do *Random Forest* e SVM estão detalhadas nas Tabelas 13 e 14. É importante notar que, no *Random Forest*, a profundidade máxima (*max_depth*), a mínima redução de impureza para divisão de nó (*min_impurity_decrease*) e o número mínimo de amostras por nó folha (*min_samples_leaf*) tiveram melhor desempenho com seus valores padrão. O mesmo ocorreu com o *kernel* no SVM.

Por outro lado, em cada modelo, houve valores diferentes dos valores padrão, como mostram as Tabelas 13 e 14. No *Random Forest*, a quantidade de árvores (*n_estimators*) variou entre 300 e 500, onde o padrão é 100; o número máximo de atributos (*max_features*) com valores "*auto*" e "*log2*", onde o primeiro é o padrão; e a heurística booleana de reutilizar a solução da iteração anterior (*warm_start*) com ambas as opções entre as melhores combinações. Já no SVM, custo (*C*) de 10 ou 1000, onde o padrão é 1, e tendo sido testados valores como 50, 100 e 500, com desempenho inferior; peso da classe (*class_weight*) com *None* e "*balanced*" entre os melhores e *gamma* entre "*auto*", 0.1 e 0.3, onde o primeiro caso é o *default*.

Tabela 13 – *Top* 5 combinações de hiperparâmetros do *Random Forest*

Random Forest					
n_estimators	max_depth	min_samples_leaf	min_impurity_decrease	max_features	warm_start
500	None	1	0	auto	True
500	None	1	0	log2	True
300	None	1	0	auto	True
300	None	1	0	log2	False
500	None	1	0	auto	False

Fonte: Elaboração própria

Tabela 14 – *Top* 5 combinações de hiperparâmetros do SVM

SVM			
C	kernel	class_weight	gamma
1000	rbf	None	auto
1000	rbf	balanced	auto
10	rbf	balanced	0.1
10	rbf	None	0.1
10	rbf	None	0.3

Fonte: Elaboração própria

5. Resultados

Os melhores resultados foram do modelo com *Random Forest*, na Tabela 15 e Figura 24. Foi utilizada a primeira linha de hiperparâmetros da Tabela 13, combinando balanceamento, normalização, *Kernel* PCA com 61 atributos e t-SNE de perplexidade 30. A combinação da primeira linha da Tabela 14, com SVM, chegou próxima, mas teve uma concentração maior de classificações na classe dominante, como mostra a Figura 23, e isso trouxe impacto nas métricas de desempenho, apresentadas na Tabela 15.

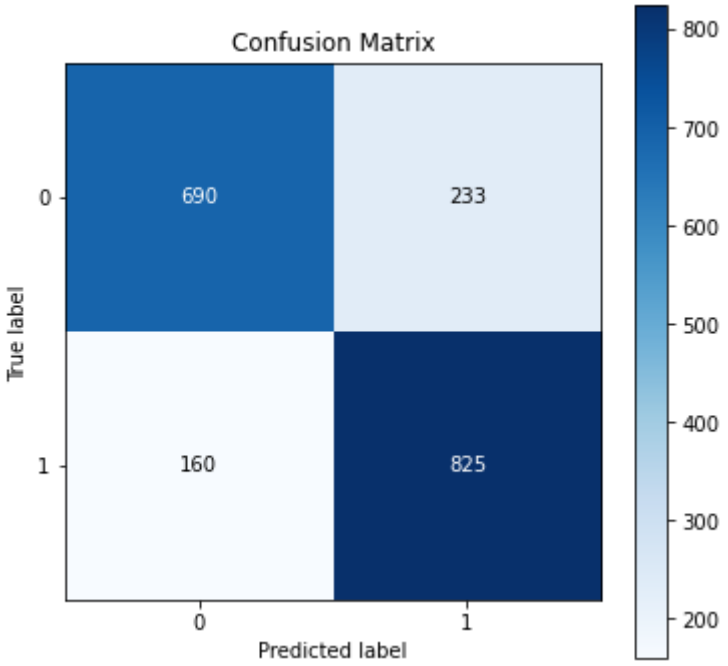

Figura 23 – Matriz de confusão do modelo de SVM com melhor desempenho

Fonte: Elaboração própria

A concentração de predições na classe 1 pelo modelo com SVM, na Figura 23 – 59% das incorretas e 55% do total –, mesmo após balanceamento, mostra que as mesmas dificuldades das fases anteriores perduraram, ainda que em menor intensidade. Por outro lado, no modelo com *Random Forest*, na Figura 24, o equilíbrio é maior: 55% das incorretas são na classe rara e cada classe tem cerca de 50% das predições totais.

Para efeito de comparação, os outros modelos, com Regressão Logística e k-NN, demonstraram ainda mais dificuldade de diferenciar classes, concentrando proporções de classificação de uma classe superiores a 70% – no caso do k-NN, na Figura 21 – e 80% – no caso da Regressão Logística, na Figura 22. O *Random Forest*, até então, sofria de um efeito semelhante, mas conseguiu melhorar seu desempenho com a combinação de técnicas, o que não ocorreu com Regressão Logística e k-NN.

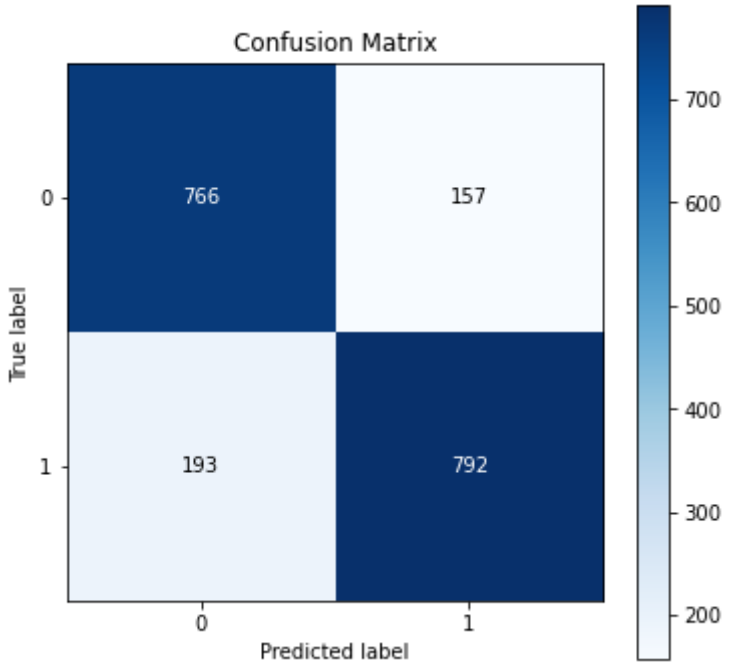

Figura 24 – Matriz de confusão do modelo de *Random Forest*, com melhor desempenho geral

Fonte: Elaboração própria

A diferença do melhor resultado do *Random Forest* para o melhor do SVM recai nas métricas de desempenho, na Tabela 15: o *recall* do SVM foi inflado pela concentração de classificações da classe 1, o que refletiu também no F1. Com esse acúmulo, as outras métricas do SVM não tiveram bons resultados.

Por sua vez, ainda que o *Random Forest* tenha se beneficiado na precisão por ter tido uma concentração menor de classificações incorretas na *label* 1, conseguiu apresentar uma *performance* visivelmente mais consistente, tanto na matriz de confusão da Figura 24, quanto nas métricas da Tabela 15.

Tabela 15 – Métricas de desempenho comparando resultados de SVM e *Random Forest*

	SVM	Random Forest
Acurácia	0.7940	0.8166
Precisão	0.7798	0.8346
Recall	0.8376	0.8041
F1 Score	0.8076	0.8190

Fonte: Elaboração própria

6. Conclusões e trabalhos futuros

Ao longo deste estudo, foi notado que a análise exploratória proporcionou *insights* para as etapas seguintes, como a possível necessidade de balanceamento, com a proporção de cerca de 70% e 30% entre as classes, a necessidade de criação de variáveis binárias a partir de atributos categóricos e a observação da natureza dos dados para criar o formato de agrupamento de atributos no tratamento de valores faltantes.

Com a tarefa de testar quatro diferentes algoritmos de *Machine Learning* e descrever os processos de implementação do projeto, foi possível criar, analisar e demonstrar um modelo de classificação capaz de gerar previsões significativamente confiáveis relacionadas à segurança pública nos municípios brasileiros a partir de diversos indicadores socioeconômicos locais.

Neste estudo, observou-se, ainda, que a combinação de técnicas de pré-processamento se mostrou fundamental para melhorar o desempenho dos modelos. Utilizando técnicas isoladas, boa parte dos modelos demonstrou grande dificuldade em diferenciar classes. Mas, com as combinações, o desempenho inicial foi relativamente uniformizado. Testando mais combinações, foi possível atingir o melhor resultado.

Em paralelo, a partir desta série de experimentos, pode-se afirmar que os atributos analisados – e detalhados no Apêndice A – explicam cerca de 80% da classificação de um município como portador de violência endêmica ou não. Trata-se de um número que pode ser melhorado, talvez investindo em um processo de *feature selection* mais robusto, ou buscar abordagens de custo computacional mais alto, mas com respostas mais precisas.

Outra sugestão seria aplicar mais análises, como a curva ROC, para analisar mais a fundo a capacidade de diferenciação de cada modelo, o método não-paramétrico de amostragem *bootstrap*, para estimar a variância da mediana amostral, e verificações adicionais para apurar a possível ocorrência de *overfitting*.

Por fim, os resultados e a metodologia deste projeto podem ser levados em consideração em trabalhos futuros que busquem, por exemplo, determinar quais variáveis explicativas são mais relevantes para essa classificação. Ou, ainda, agrupar os municípios em *clusters*, de acordo com suas *features* e, com isso, sugerir tratamentos diferentes para cada grupo.

Referências Bibliográficas

ALVES, Luiz G. A.; RIBEIRO, Haroldo V.; RODRIGUES, Francisco A. Crime prediction through urban metrics and statistical learning. *In*: **Physica A**: Statistical Mechanics and its Applications, v. 505, p. 435-443, 01 set. 2018. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0378437118304059>. Acesso em: 03 maio 2020.

ASSUNÇÃO, Fernando. **Estratégias para tratamento de variáveis com dados faltantes durante o desenvolvimento de modelos preditivos**. São Paulo, 2012. 74 f. Dissertação (Mestrado em Estatística). Instituto de Matemática e Estatística, Universidade de São Paulo, 2012. Disponível em: <https://www.teses.usp.br/teses/disponiveis/45/45133/tde-15082012-203206/publico/DissertacaoFernando_vfinal.pdf>. Acesso em: 04 jul. 2020.

BANCO MUNDIAL. **Urban Violence**: A Challenge of Epidemic Proportions. 06 set. 2016. <https://www.worldbank.org/en/news/feature/2016/09/06/urban-violence-a-challenge-of-epidemic-proportions>. Acesso em: 21 jun. 2020.

BRASIL. Pensando a Segurança Pública: Promovendo o conhecimento científico em Segurança Pública. **Ministério da Justiça**. 2016. Disponível em: <https://www.novo.justica.gov.br/sua-seguranca/seguranca-publica/analise-e-pesquisa/pensando-a-seguranca>. Acesso em: 03 maio 2020.

BRASIL. Governo federal lança programa Em Frente, Brasil para combater a criminalidade. **Ministério da Cidadania**. 29 ago. 2019. Disponível em: <http://mds.gov.br/area-de-imprensa/noticias/2019/agosto/governo-federal-lanca-programa-em-frente-brasil-para-combater-a-criminalidade>. Acesso em: 03 maio 2020.

BRASIL. 'Em Frente, Brasil' ingressa em nova fase e foca na prevenção à violência em 2020. **Ministério do Desenvolvimento Social**. 24 jan. 2020. Disponível em: <https://desenvolvimentosocial.gov.br/noticias/2018em-frente-brasil2019-ingressa-em-nova-fase-e-foca-na-prevencao-a-violencia-em-2020>. Acesso em: 03 maio 2020.

BREIMAN, Leo. Random Forests. **Machine Learning**, v. 45, p. 5-32, 2001. Disponível em: <https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf>. Acesso em: 28 jun. 2020.

CABRAL, Marco. A. P.; GOLDFELD, Paulo. **Curso de Álgebra Linear**: Fundamentos e Aplicações. 3 ed. Rio de Janeiro: Instituto de Matemática da UFRJ, 2012. Disponível em: <https://www.labma.ufrj.br/~mcabral/livros/livro-alglin/alglin-material/CursoAlgLin-livro.pdf>. Acesso em: 26 jun. 2020.

CERQUEIRA, D. **Mapa dos homicídios ocultos no Brasil**. Brasília: Ipea, 2013. (Texto para discussão, n. 1848).

CONSELHO NACIONAL DO MINISTÉRIO PÚBLICO. **Relatório Nacional da Execução da Meta 2**: um diagnóstico da investigação de homicídios no país. 2012. Disponível em: <https://www.cnmp.mp.br/portal/images/stories/Enasp/relatorio_enasp_FINAL.pdf>. Acesso em: 21 jun. 2020.

EFRON, B. Bootstrap methods: another look at the jackknife. **The Annals of Statistics**, v. 7, n. 1, p. 1-26, Stanford University, 1979. Disponível em: <https://projecteuclid.org/download/pdf_1/euclid.aos/1176344552>. Acesso em: 29 jun. 2020.

GUENTHER, N.; SCHONLAU, M. Support Vector Machines**. Stata Journal**, *[s. l.]*, v. 16, n. 4, p. 917–937, 2016. Disponível em: <https://bit.ly/2A6rHTS>. Acesso em: 26 jun. 2020.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. Sistema IBGE de Recuperação Automática. 2019. Disponível em: <https://sidra.ibge.gov.br>. Acesso em: 08 maio 2020.

INSTITUTO DE PESQUISA ECONÔMICA APLICADA. **Atlas da Violência**: Retratos dos Municípios Brasileiros. Rio de Janeiro: Ipea, jul. 2019. Disponível em: <https://www.ipea.gov.br/atlasviolencia/arquivos/downloads/7047-190802atlasdaviolencia2019municipios.pdf>. Acesso em: 18 maio 2020.

JEAN, Neal *et al*. Combining satellite imagery and Machine Learning to predict poverty. *In*: **Science**, v. 353, n. 6301, p. 790-794, 19 ago. 2016. Disponível em: <https://science.sciencemag.org/content/353/6301/790>. Acesso em: 03 maio 2020.

LIU, Yuzhe; GOPALAKRISHNAN, Vanathi. An Overview and Evaluation of Recent Machine Learning Imputation Methods Using Cardiac Imaging Data. **Data**, v. 2, n. 8, p. 1-15, 2017. Disponível em: <https://www.mdpi.com/2306-5729/2/1/8/pdf>. Acesso em: 29 jun. 2020.

NASCIMENTO, D. E.; TEIXEIRA, M. A. N. Segurança pública e desenvolvimento local: Experiências do Brasil, Colômbia e Japão. **Revista Brasileira de Planejamento e Desenvolvimento**, Curitiba, v. 5, n. 3, p. 365-385, set./dez. 2016. Disponível em: <https://periodicos.utfpr.edu.br/rbpd>. Acesso em: 03 maio 2020.

ORGANIZAÇÃO MUNDIAL DA SAÚDE. **WHO Global Health Estimates (2015 update)**: Homicide. 2017. Disponível em: <https://apps.who.int/violence-info/homicide/>. Acesso em: 21 jun. 2020.

PIEGL, L. A.; TILLER, W. Algorithm for finding all k nearest neighbors. **Computer-Aided Design**, v. 34, n. 2, p. 167-172, 2002. Disponível em: <http://www.sciencedirect.com/science/article/pii/S001044850000141X>. Acesso em: 29 jun. 2020.

SAINANI, Kristin L. Logistic Regression. **PM&R**, v. 6, n. 12, p. 1157-1162, 2014. Disponível em: <https://doi.org/10.1016/j.pmrj.2014.10.006>. Acesso em: 28 jun. 2020.

SHAFIZADEH-MOGHADAM, H. *et al*. Coupling Machine Learning, tree-based and statistical models with cellular automata to simulate urban growth. **Computers, Environment and Urban Systems**, v. 64, p. 297-308, jul. 2017. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0198971516302265>. Acesso em: 22 jun. 2020.

SILVARES, A. C. Políticas públicas em segurança no Brasil: avanços e novos desafios. **Revista Científica Doctum**: Direito. Caratinga. v. 1, n. 3, 2019. Disponível em: <http://revista.doctum.edu.br/index.php/DIR/article/viewFile/242/218>. Acesso em: 03 maio 2020.

SOUSA, S. B. S.; DEL-FIACO, R. C.; BERTON, L. Cluster analysis of homicide rates in the Brazilian state of Goias from 2002 to 2014. **Anais da XLIV Conferência Latino-americana de Informática (Clei)**. 2018. Disponível em: <https://arxiv.org/abs/1811.06366>. Acesso em: 22 jun. 2020.

WANG, Z. *et al*. Flood hazard risk assessment model based on Random Forest. **Journal of Hydrology**, v. 527, p.1130-1141, 2015. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0022169415004217>. Acesso em: 28 jun. 2020.

WANG, Z.; HU, M.; ZHAI, G. Application of Deep Learning Architectures for Accurate and Rapid Detection of Internal Mechanical Damage of Blueberry Using Hyperspectral Transmittance Data. **Sensors**, v. 18, n. 4, p. 1126-1140, 2018. Disponível em: <https://www.mdpi.com/1424-8220/18/4/1126/htm>. Acesso em: 17 ago. 2020.

ZHANG, L.; SUGANTHAN, P. N. Random Forests with ensemble of feature spaces. **Pattern Recognition**, v. 47, n. 10, p. 3429-3437, 2014. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0031320314001307>. Acesso em: 29 jun. 2020.

APÊNDICE A – DICIONÁRIO DE DADOS

Campo	Descrição	Período de Referência	Unidade de Medida	Fonte(s)
CITY	Nome do município			-
STATE	Sigla da UF			-
IBGE_ID	ID do município	2017		https://sidra.ibge.gov.br/
CAPITAL	1 se for capital da UF			-
IBGE_RES_POP_ESTR_PERC	Percentual de estrangeiros na população residente	2010	-	https://sidra.ibge.gov.br/tabela/1497
IBGE_DU_RATE	Taxa de unidades domésticas (UDs) por habitante	2010	-	https://sidra.ibge.gov.br/tabela/3495 & https://sidra.ibge.gov.br/tabela/1497
IBGE_DU_RURAL_PERC	Percentual de UDs rurais no total de UDs	2010	-	https://sidra.ibge.gov.br/tabela/3495
IBGE_POP_UP_PERC	Percentual da população residente em áreas com ordenamento urbano regular	2010	-	https://sidra.ibge.gov.br/tabela/3365 & https://sidra.ibge.gov.br/tabela/1497
IBGE_0-14_PERC	Percentual da população residente em áreas com ordenamento urbano regular (até 14 anos)	2010		https://sidra.ibge.gov.br/tabela/3365
IBGE_15-59_PERC	Percentual da população residente em áreas com ordenamento urbano regular (15 a 59 anos)	2010		https://sidra.ibge.gov.br/tabela/3365
IBGE_60+_PERC	Percentual da população residente em áreas com ordenamento urbano regular (acima de 60 anos)	2010		https://sidra.ibge.gov.br/tabela/3365
IBGE_PLANTED_AREA	Área plantada (hectares)	2017	Hectare (ha)	https://sidra.ibge.gov.br/tabela/5457
IBGE_CROP_PRODUCTION_$	Produção agrícola	2017	R$ 1.000	https://sidra.ibge.gov.br/tabela/5457

Campo	Descrição	Período de Referência	Unidade de Medida	Fonte(s)
IDHM Ranking 2010	Posição no *ranking* do Índice de Desenvolvimento Humano Municipal (IDHM)	2010	-	http://www.br.undp.org/content/brazil/pt/home/idh0.html
IDHM	IDHM	2010	-	http://www.br.undp.org/content/brazil/pt/home/idh0.html
IDHM_Renda	Componente Renda do IDHM	2010	-	http://www.br.undp.org/content/brazil/pt/home/idh0/rankings/idhm-municipios-2010.html
IDHM_Longevidade	Componente Longevidade do IDHM	2010	-	http://www.br.undp.org/content/brazil/pt/home/idh0/rankings/idhm-municipios-2010.html
IDHM_Educacao	Componente Educação do IDHM	2010	-	http://www.br.undp.org/content/brazil/pt/home/idh0/rankings/idhm-municipios-2010.html
LONG	Longitude	2010	-	ftp://geoftp.ibge.gov.br/organizacao_do_territorio/estrutura_territorial/localidades
LAT	Latitude	2010	-	ftp://geoftp.ibge.gov.br/organizacao_do_territorio/estrutura_territorial/localidades
ALT	Altitude	2010	Metros (m)	ftp://geoftp.ibge.gov.br/organizacao_do_territorio/estrutura_territorial/localidades
PAY_TV	Número de usuários de TV paga	2019-03	-	https://cloud.anatel.gov.br/index.php/s/TpaFAwSw7RPfBa8?path=%2FTV_por_Assinatura%2FPor_Municipio
FIXED_PHONES	Número de usuários de telefone fixo	2019-03	-	https://cloud.anatel.gov.br/index.php/s/TpaFAwSw7RPfBa8?path=%2FTelefonia_Fixa%2FPor_Municipio
AREA	Área do município	2018	km²	https://www.ibge.gov.br/geociencias/organizacao-do-territorio/estrutura-territorial/15761-areas-dos-municipios.html?t=acesso-ao-produto&c=1
REGIAO_TUR	Região turística	2017	-	http://dados.turismo.gov.br/mapa-do-turismo-brasileiro
CATEGORIA_TUR	Cluster da região turística	2017	-	http://dados.turismo.gov.br/mapa-do-turismo-brasileiro
RURAL_URBAN	Tipologia Rural-Urbana	2016	-	https://www.ibge.gov.br/estatisticas/economicas/contas-nacionais/9088-produto-interno-bruto-dos-municipios.html?t=downloads

Campo	Descrição	Período de Referência	Unidade de Medida	Fonte(s)
GVA_AGROPEC	Valor adicionado bruto (VAB) da agropecuária	2016	R$ 1.000	https://www.ibge.gov.br/estatisticas/economicas/contas-nacionais/9088-produto-interno-bruto-dos-municipios.html?t=downloads
GVA_INDUSTRY	VAB da indústria	2016	R$ 1.000	https://www.ibge.gov.br/estatisticas/economicas/contas-nacionais/9088-produto-interno-bruto-dos-municipios.html?t=downloads
GVA_SERVICES	VAB de serviços	2016	R$ 1.000	https://www.ibge.gov.br/estatisticas/economicas/contas-nacionais/9088-produto-interno-bruto-dos-municipios.html?t=downloads
GVA_PUBLIC	VAB de serviços públicos	2016	R$ 1.000	https://www.ibge.gov.br/estatisticas/economicas/contas-nacionais/9088-produto-interno-bruto-dos-municipios.html?t=downloads
GVA_TOTAL	VAB total	2016	R$ 1.000	https://www.ibge.gov.br/estatisticas/economicas/contas-nacionais/9088-produto-interno-bruto-dos-municipios.html?t=downloads
TAXES	Impostos	2016	R$ 1.000	https://www.ibge.gov.br/estatisticas/economicas/contas-nacionais/9088-produto-interno-bruto-dos-municipios.html?t=downloads
GDP	Produto Interno Bruto (PIB)	2016	R$ 1.000	https://www.ibge.gov.br/estatisticas/economicas/contas-nacionais/9088-produto-interno-bruto-dos-municipios.html?t=downloads
POP_GDP	População	2016	-	https://www.ibge.gov.br/estatisticas/economicas/contas-nacionais/9088-produto-interno-bruto-dos-municipios.html?t=downloads
GDP_CAPITA	PIB per capita	2016	-	https://www.ibge.gov.br/estatisticas/economicas/contas-nacionais/9088-produto-interno-bruto-dos-municipios.html?t=downloads

Campo	Descrição	Período de Referência	Unidade de Medida	Fonte(s)
GVA_MAIN[67]	Atividade com a maior contribuição no VAB	2016	-	https://www.ibge.gov.br/estatisticas/economicas/contas-nacionais/9088-produto-interno-bruto-dos-municipios.html?t=downloads
MUN_EXPENDIT	Gastos municipais	2016	R$	http://www.tesourotransparente.gov.br/ckan/dataset/dcam
COMP_TOT	Número total de empresas	2016	-	https://sidra.ibge.gov.br/tabela/993
COMP_A	Número de empresas: Agricultura, pecuária, produção florestal, pesca e aquicultura	2016	-	https://sidra.ibge.gov.br/tabela/993
COMP_B	Número de empresas: Indústrias extrativas	2016	-	https://sidra.ibge.gov.br/tabela/993
COMP_C	Número de empresas: Indústrias de transformação	2016	-	https://sidra.ibge.gov.br/tabela/993
COMP_D	Número de empresas: Eletricidade e gás	2016	-	https://sidra.ibge.gov.br/tabela/993
COMP_E	Número de empresas: Água, esgoto, atividades de gestão de resíduos e descontaminação	2016	-	https://sidra.ibge.gov.br/tabela/993
COMP_F	Número de empresas: Construção	2016	-	https://sidra.ibge.gov.br/tabela/993
COMP_G	Número de empresas: Comércio; reparação de	2016	-	https://sidra.ibge.gov.br/tabela/993

[6] Para as variáveis com as maiores atividades econômicas foram consideradas: agricultura, inclusive apoio à agricultura e a pós colheita; pecuária, inclusive apoio à pecuária; produção florestal, pesca e aquicultura; indústrias extrativas; indústrias de transformação eletricidade e gás, água, esgoto, atividades de gestão de resíduos e descontaminação; construção; comércio e reparação de veículos automotores e motocicletas; administração, defesa, educação e saúde públicas e seguridade social e demais serviços.

[7] A classe Demais serviços compreende a agregação dos setores: transporte, armazenagem e correio; alojamento e alimentação; informação e comunicação; atividades financeiras, de seguros e serviços relacionados; atividades imobiliárias; atividades profissionais, científicas e técnicas, administrativas e serviços complementares; educação e saúde privadas; artes, cultura, esporte e recreação e outras atividades de serviços, e serviços domésticos.

Campo	Descrição	Período de Referência	Unidade de Medida	Fonte(s)
	veículos automotores e motocicletas			
COMP_H	Número de empresas: Transporte, armazenagem e correio	2016	-	https://sidra.ibge.gov.br/tabela/993
COMP_I	Número de empresas: Alojamento e alimentação	2016	-	https://sidra.ibge.gov.br/tabela/993
COMP_J	Número de empresas: Informação e comunicação	2016	-	https://sidra.ibge.gov.br/tabela/993
COMP_K	Número de empresas: Atividades financeiras, de seguros e serviços relacionados	2016	-	https://sidra.ibge.gov.br/tabela/993
COMP_L	Número de empresas: Atividades imobiliárias	2016	-	https://sidra.ibge.gov.br/tabela/993
COMP_M	Número de empresas: Atividades profissionais, científicas e técnicas	2016	-	https://sidra.ibge.gov.br/tabela/993
COMP_N	Número de empresas: Atividades administrativas e serviços complementares	2016	-	https://sidra.ibge.gov.br/tabela/993
COMP_O	Número de empresas: Administração pública, defesa e seguridade social	2016	-	https://sidra.ibge.gov.br/tabela/993
COMP_P	Número de empresas: Educação	2016	-	https://sidra.ibge.gov.br/tabela/993
COMP_Q	Número de empresas: Saúde humana e serviços sociais	2016	-	https://sidra.ibge.gov.br/tabela/993
COMP_R	Número de empresas: Artes, cultura, esporte e recreação	2016	-	https://sidra.ibge.gov.br/tabela/993

Campo	Descrição	Período de Referência	Unidade de Medida	Fonte(s)
COMP_S	Número de empresas: Outras atividades de serviços	2016	-	https://sidra.ibge.gov.br/tabela/993
COMP_T	Número de empresas: Serviços domésticos	2016	-	https://sidra.ibge.gov.br/tabela/993
COMP_U	Número de empresas: Organismos internacionais e outras instituições extraterritoriais	2016	-	https://sidra.ibge.gov.br/tabela/993
HOTELS	Número de meios de hospedagem	2020-03	-	http://dados.turismo.gov.br/cadastur
BEDS	Número de leitos de hospedagem	2020-03	-	http://dados.turismo.gov.br/cadastur
Pr_Agencies	Número de agências de bancos privados	2019-02	-	https://www.bcb.gov.br/estatisticas/estatisticabancariamunicipios
Pu_Agencies	Número de agências de bancos públicos	2019-02	-	https://www.bcb.gov.br/estatisticas/estatisticabancariamunicipios
Pr_Bank	Número de bancos privados	2019-02	-	https://www.bcb.gov.br/estatisticas/estatisticabancariamunicipios
Pu_Bank	Número de bancos públicos	2019-02	-	https://www.bcb.gov.br/estatisticas/estatisticabancariamunicipios
Pr_Assets	Ativos totais de bancos privados	2019-02	R$	https://www.bcb.gov.br/estatisticas/estatisticabancariamunicipios
Pu_Assets	Ativos totais de bancos públicos	2019-02	R$	https://www.bcb.gov.br/estatisticas/estatisticabancariamunicipios
CARS_RATE	Taxa de número de carros por habitante	2019-01 (veículos), 2018-07 (população)	-	https://www.denatran.gov.br/estatistica/639-frota-2019 & https://www.ibge.gov.br/estatisticas/sociais/populacao/9103-estimativas-de-populacao.html?=&t=o-que-e

Campo	Descrição	Período de Referência	Unidade de Medida	Fonte(s)
MOTORCYCLES_RATE	Taxa de número de motocicletas por habitante	2019-01 (veículos), 2018-07 (população)	-	https://www.denatran.gov.br/estatistica/639-frota-2019 & https://www.ibge.gov.br/estatisticas/sociais/populacao/9103-estimativas-de-populacao.html?=&t=o-que-e
UBER	1 se o município tiver serviços da Uber	2019-05	-	https://www.uber.com/en-BR/cities/
MCDONALDS	Número total de lojas da rede McDonald's	2018-11	-	https://www.mcdonalds.com.br/enderecos
WALMART	Número total de lojas da rede Walmart	2018-12	-	https://tabloide.walmartbrasil.com.br/
POST_OFFICES_RATE	Número total de agências dos Correios a cada 100.000 habitantes	2019-05	-	http://www2.correios.com.br/sistemas/agencias/ & https://www.ibge.gov.br/estatisticas/sociais/populacao/9103-estimativas-de-populacao.html?=&t=o-que-e
IBGE_POP_10+_2010	Percentual da população com 10 anos de idade ou mais	2010	-	https://sidra.ibge.gov.br/tabela/3540 & https://sidra.ibge.gov.br/tabela/1497
IBGE_EDU_0_PERC	Percentual da população com 10 anos de idade ou mais - sem escolaridade ou ensino fundamental incompleto	2010	-	https://sidra.ibge.gov.br/tabela/3540
IBGE_EDU_1_PERC	Percentual da população com 10 anos de idade ou mais - ensino fundamental completo ou ensino médio incompleto	2010	-	https://sidra.ibge.gov.br/tabela/3540
IBGE_EDU_2_PERC	Percentual da população com 10 anos de idade ou mais - ensino médio completo ou ensino superior incompleto	2010	-	https://sidra.ibge.gov.br/tabela/3540
IBGE_EDU_3_PERC	Percentual da população com 10 anos de idade ou mais - ensino superior completo	2010	-	https://sidra.ibge.gov.br/tabela/3540

Campo	Descrição	Período de Referência	Unidade de Medida	Fonte(s)
IBGE_EDU_NA_PERC	Percentual da população com 10 anos de idade ou mais - informações de escolaridade indisponíveis	2010	-	https://sidra.ibge.gov.br/tabela/3540
IBGE_ESTIMATED_POP_2017	População estimada	2017	-	http://www.ipea.gov.br/atlasviolencia/arquivos/downloads/8099-tabelamunicipiostodossite.pdf
HOMICIDES	Número de homicídios registrados	2017	-	http://www.ipea.gov.br/atlasviolencia/arquivos/downloads/8099-tabelamunicipiostodossite.pdf
HIDDEN_HOMICIDES	Número de homicídios ocultos	2017	-	http://www.ipea.gov.br/atlasviolencia/arquivos/downloads/8099-tabelamunicipiostodossite.pdf
GINI_INDEX_2000	Índice de Gini	2000	-	http://tabnet.datasus.gov.br/cgi/ibge/censo/cnv/ginibr.def
GINI_INDEX_2010	Índice de Gini	2010	-	http://tabnet.datasus.gov.br/cgi/ibge/censo/cnv/ginibr.def
ILLITERACY_RATE_10+_2010	Percentual de analfabetismo na população com 10 anos ou mais	2010	-	https://sidra.ibge.gov.br/tabela/1390
UNEMPLOYMENT_RATE_10+_2010	Percentual de desocupação na população com 10 anos ou mais	2010	-	https://sidra.ibge.gov.br/tabela/1381