UTILIZAÇÃO DE *MACHINE LEARNING* PARA CLASSIFICAÇÃO DE CRIMES DE MORTE NO ESTADO DE SÃO PAULO

Murilo Góes de Almeida[1], Edna Dias Canedo[1], Luan Borges dos Santos[1], Rodrigo Pereira Gomes[1] e Paula Miwa De Paiva Lima[2]

[1]*Universidade de Brasília*
Campus Universitário Darcy Ribeiro, Brasília, Distrito Federal, 70904-970, Brasil
[2]*Polícia Militar do Estado de São Paulo / Secretaria de Segurança Pública – CAP/SSP*
R. Líbero Badaró, 39 - Sé, São Paulo - SP, 01003-000, Brasil

RESUMO

Este trabalho tem o objetivo de automatizar a classificação dos crimes de homicídio no Estado de São Paulo – Brasil, utilizando uma abordagem de mineração de texto, por meio da análise do histórico dos boletins de ocorrência. Os crimes devem ser prevenidos pelas instituições de segurança pública, entretanto, o contexto do crime deve ser considerado, pois alguns deles estão fora do alcance dos órgãos de aplicação da lei, como por exemplo dentro de propriedade privada, entre membros da mesma família, ou seja, lugares em que a polícia não pode estar presente para interceder. O Estado de São Paulo possui um departamento específico incumbido de realizar a análise dos crimes e classificá-los. A atividade é feita com leitura manual de boletins de ocorrência e consome muito tempo e recursos humanos. Como a atividade manual é desenvolvida há anos, existe uma boa memória de dados para ser treinada como modelo supervisionado. Utilizando o processo CRISP-DM, estudamos e aplicamos uma forma de escolha de um modelo adequado a ser treinado, implantando e disponibilizando à Secretaria da Segurança Pública para automatizar a atividade de classificação de textos.

PALAVRAS-CHAVE

Mineração de Texto, Classificação de Texto, Segurança Pública, Polícia, Relatórios Policiais

1. INTRODUÇÃO

A segurança pública no Brasil é dever do Estado, garantido pela Constituição Federal, ou seja, cabe ao Estado a responsabilidade de preservar a ordem pública, a segurança das pessoas e do patrimônio. Existem órgãos de aplicação da lei criados pela Constituição Federal (BRASIL, 1988), cada um com uma atribuição específica no âmbito da segurança pública. O Brasil é uma República Federativa, dividida em 26 Estados mais o Distrito Federal. Os Estados têm seu próprio sistema de segurança pública, incumbido de prevenir e reprimir crimes.

A organização responsável pela prevenção criminal se chama Polícia Militar. Ela é responsável por estudar, entender e procurar prevenir os crimes pela atividade policial ostensiva, ou seja, a Polícia Militar deve estar presente em locais onde é mais provável que ocorra um crime, para que ele seja evitado. Além disso, a presença ostensiva da Polícia Militar ajuda as pessoas a terem uma maior sensação de segurança. É importante entender o contexto do crime, pois ele pode acontecer em situações que as instituições policiais não poderiam atuar, como em ambiente familiar ou em áreas privadas. No entanto, todos os homicídios são registrados, não importa o contexto, sendo contabilizados nas estatísticas. Este cenário pode ser injusto com as instituições policiais acerca da prevenção.

O Governo do Estado de São Paulo possui uma Secretaria denominada “Secretaria da Segurança Pública”, incumbida de administrar a força policial em todo o Estado (SÃO PAULO, 1906). Além das funções administrativas, possui um departamento específico para análise de crimes denominado CAP (Coordenadoria de Análise e Planejamento) (SÃO PAULO, 2022), ou seja, pesquisam dados de interesse policial e realizam estudos para prevenir e reprimir crimes.

Alguns tipos de crimes são mais importantes de serem prevenidos e solucionados, principalmente quando estão relacionados à perda de vidas. O Estado é muitas vezes cobrado pela sociedade, imprensa, órgãos fiscalizadores e outros tipos de organizações, que solicitam índices de criminalidade e exigem esclarecimentos, caso eles aumentem. A Agenda 2030 para o Desenvolvimento Sustentável das Nações Unidas (ONU) propõe como meta reduzir todas as formas de violência e as taxas de mortalidade relacionadas em todos os lugares (UN, 2022). Portanto, podemos concluir que a criminalidade, principalmente relacionada a mortes, é uma preocupação global, que deve ser estudada com atenção. A Secretaria da Segurança Pública de São Paulo possui um grupo específico composto por policiais que realizam a leitura de todos os boletins de ocorrência que envolvem crimes com resultado morte e os reclassificam, procurando entender se poderiam ter sido prevenidos pelas forças policiais. Esta tarefa permite melhorar a compreensão do crime, no entanto, a leitura manual de todos os relatórios pode ser muito onerosa e sobrecarregar os recursos humanos na maior parte do tempo, impedindo que sejam ser melhor aproveitados em outros trabalhos policiais. Esta atividade é desenvolvida desde 2015, portanto, existe um conjunto de dados significativo, que pode ser usado como modelo supervisionado para prever novos registros, ajudando a automatizar a tarefa de classificação.

Por conta da situação supramencionada, o objetivo principal deste estudo é de desenvolver e implantar uma solução de *machine learning* capaz de automatizar a classificação de boletins de ocorrência relacionados a mortes, o que poderá gerar economia de recursos humanos e tempo, liberando o efetivo policial para a realização de outros tipos de atividades. Para tal, foi estudado um processo conhecido para Mineração de Dados.Aplicou-se o *Cross-Industry Standard Process for Data Mining* (CRISP-DM), onde as fases de Entendimento de Negócios, Entendimento de Dados, Preparação de Dados, Modelagem se encontram diluídas na Seção 3 deste trabalho, tendo as fases de Avaliação e Implementação seções dedicadas. Sua aplicação permitiu assim que fossem testados diversos tipos de algoritmos de classificação, sendo selecionado o que apresentasse um melhor resultado. Nosso principal achado é que o algoritmo *Random Forest*, aplicado com o uso de *TF-IDF* e a estratégia de *oversampling*, se mostrou o mais adequado para realizar a classificação.

2. TRABALHOS RELACIONADOS

Mineração de texto ou *Knowledge Discovery from Text* (KDT) — pela primeira vez mencionada por Feldman & Dagan (1995) foi baseada na apresentação da descoberta de conhecimento (*knowledge discovery*) (Frawley et al, 1992). A descoberta de conhecimento é definida como a extração não trivial de informações implícitas, previamente desconhecidas e potencialmente úteis encontradas nos dados fornecidos.

O campo interdisciplinar de mineração de texto é apresentado por Hotho & Paaß (2005). Este trabalho mostra a relação entre mineração de texto e as áreas de recuperação de informação, processamento de linguagem natural, extração de informação para abordar problemas de representação, classificação, agrupamento, extração de informação ou busca e modelagem de padrões de texto.

A mineração de texto está evoluindo e é usada para análise de sentimentos (Zambrano et al, 2020), análise de texto de redes sociais (Qudar & Mago, 2020), compreensão de linguagem (Devlin & Toutanova, 2018), classificação de texto (Suneera & Prakash, 2020) e vários outros campos. A classificação de texto é o campo de mineração de texto que é amplamente utilizada neste trabalho. É a tarefa de classificar um documento em uma categoria predefinida e requer um conjunto de dados inicial a ser utilizado (Ikonomakis & Tampakas, 2005).

Existem estudos disponíveis sobre o uso de mineração de texto para análise de relatórios policiais, como classificação de texto de registros policiais holandeses (Brandenburg, 2017), que procuram utilizar diferentes tipos de vetorização de texto como TF-IDF e algoritmos de classificação, como *Naive Bayes*, *Support Vector Machines*, *Random Forest* e *XGBoost*, que concluem que o tamanho do conjunto de treinamento e a capacidade computacional influenciam todas as abordagens, com vantagens e desvantagens. O algoritmo *Naive Bayes* é bem conhecido para análise e classificação automatizada de relatórios de crimes para governo eletrônico (Ku & Leroy, 2014). Existem também abordagens mais complexas que combinam relatórios criminais narrativos, bancos de dados policiais e informações de código aberto de banco de dados não estruturado, que visam a criação de um sistema automatizado para análise de relatórios policiais criminais (Carnaz et al, 2018).

3. PESQUISA E DESENVOLVIMENTO

O *Cross-Industry Standard Process for Data Mining* (CRISP-DM) propôs um modelo de processo abrangente para a realização de projetos de mineração de dados. Ele fornece um modelo genérico de referência para mineração de dados, que pode ser usado para cobrir o ciclo de vida de um projeto e contém as fases de um projeto, suas respectivas tarefas e suas saídas (Wirth & Hipp, 2000). As fases do CRISP-DM são: Entendimento de Negócios, Entendimento de Dados, Preparação de Dados, Modelagem, Avaliação e Implantação. Nesta seção abordamos todas as tarefas para resolver o problema de classificação de boletins de ocorrência de mortes. Ele é usado neste trabalho para o desenvolvimento da solução de classificação de texto.

Todos os crimes denunciados devem ser registrados pelos Órgãos Policiais, e serão objeto de investigações, visando elucidar o que realmente aconteceu e a respectiva identificação do(s) autor(es) do crime. Além disso, o registro correto dos crimes é muito importante para análise e entendimento da dinâmica dos crimes e geração de análises estatísticas. Proporcionará um melhor uso das forças policiais no futuro para que o crime seja devidamente prevenido. O registro do crime no Estado de São Paulo é feito em um sistema denominado RDO (Registro Digital de Ocorrências). O RDO contém dados relacionados ao crime, como data/hora, endereço, autor, vítima, objetos e o histórico da ocorrência, que descreve toda a narrativa do crime com base em depoimentos de pessoas e provas coletadas, além da conclusão formulada pela autoridade policial.

Após o registro do crime, uma equipe de policiais da Secretaria da Segurança Pública de São Paulo é incumbida de ler tais registros e reclassificá-los, dependendo do que entenderem nessa leitura. Por exemplo, ao realizar a leitura de um boletim de ocorrência de homicídio, eles devem procurar entender se a Polícia Militar poderia prevenir aquele crime ou não. De acordo com o contexto, então, o boletim de ocorrência será separado em diferentes classificações.

Hoje em dia, o processo é feito manualmente, portanto, custa muito esforço, tempo e recursos humanos. Por isso, surge a necessidade de uma forma de automatização que utilize um sistema capaz de ler e classificar os relatórios com pouca intervenção humana.

O processo de leitura e classificação dos boletins de ocorrência de forma manual vem sendo desenvolvido desde 2015, portanto, há uma grande quantidade de dados que podem ser utilizados como modelo supervisionado. No ano de 2021, o conjunto de dados possuía 21.212 registros corretamente classificados em contexto criminal e não criminal em crimes de resultado morte. Ele está basicamente separado em duas colunas, em que a primeira está relacionada com o boletim de ocorrência, contendo o histórico da ocorrência. A segunda coluna é sobre a classificação do contexto e classifica se o crime possui um “contexto policial” ou não, ou seja, separando ocorrências que, segundo a análise realizada, as forças policiais poderiam ou não evitar. Nesta base de dados há 13.741 relatórios classificados como “contexto policial” e 7.471 classificados como “contexto não policial”, conforme pode ser visualizado na Figura 1.

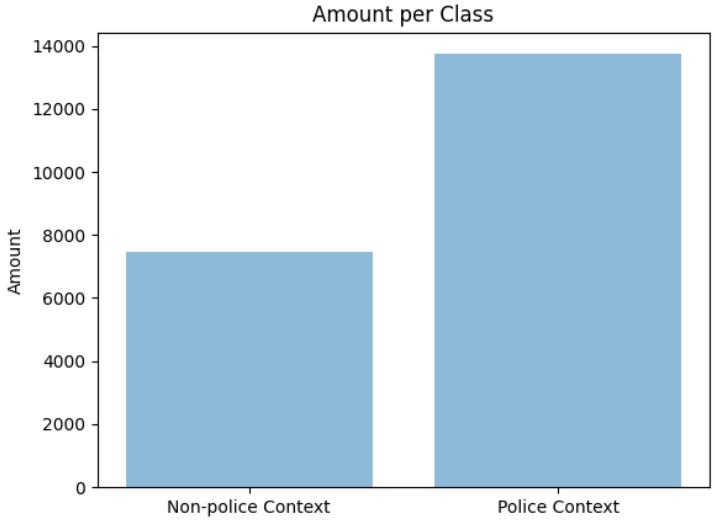

Figura 1. Quantitativo das Classes de Dados disponíveis

Os boletins de ocorrência com resultado morte são lavrados na delegacia de polícia, após o crime ser constatado. No Estado de São Paulo, existem muitas delegacias de polícia, portanto, os boletins podem ser escritos por vários policiais. Entretanto, há uma percepção que o estilo de escrita dos policiais costuma possuir uma certa similaridade, portanto, há uma padronização textual, como pode ser constatado na nuvem de palavras da Figura 2. Percebe-se que existem algumas palavras que são amplamente utilizadas, independentemente do local em que a ocorrência é registrada.

O banco de dados apresentado para ser treinado como modelo é muito bem-organizado, pois todo o conteúdo está com formatação adequada, sem necessidade de realizar ações complexas de ETL (Extrair, Transformar, Carregar). Apesar disso, é necessário que sejam realizadas algumas atividades de preparação de dados para mineração de texto. Tal atividade é muito importante para eliminar palavras e partes de textos que são irrelevantes para o treinamento e avaliação.

Em primeiro lugar, removemos todos os caracteres especiais, pontuação e números, utilizando recursos da própria linguagem *Python* e expressões regulares. Depois disso, foi realizada a remoção das palavras chamadas de "*stop words*". É uma boa prática remover alguns termos no texto que possuem função gramatical, mas sem relevância no conteúdo, como por exemplo os termos "o", "mas", etc (Wilbur & Sirotkin, 1992). Para a remoção deste tipo de palavras, utilizamos uma funcionalidade da biblioteca NLTK, que busca todas as *stop words* do texto e as remove, neste caso, utilizando o idioma português do Brasil. Foi ainda utilizado um algoritmo de *stemming* usando o mesmo NLTK. *Stemming* é um procedimento computacional que reduz todas as palavras com a mesma raiz (Lovins, 1968).

Figura 2. Nuvem de palavras extraídas dos relatórios policiais

Após todo o processo de preparação de dados, os textos estão prontos para processamento. Nesta tarefa, escolhemos o *Scikit-learn*, uma biblioteca de aprendizado de máquina *Python* que fornece diversas ferramentas úteis, como vetorização, algoritmos de classificação e testes, sendo bastante útil para a execução desta atividade. Dentre as etapas executadas, destacam-se as seguintes tarefas:

3.1 Vetorização de Texto

De acordo com a documentação oficial do *Scikit-learn*, não é possível usar os dados do texto bruto para alimentar os algoritmos de classificação. Portanto, é necessário que se extraiam as características numéricas do conteúdo do texto, aplicando-se um processo de vetorização. O *Scikit* possui algumas maneiras de realizar tal tarefa, transformando os dados de texto em vetores de recursos numéricos, como um vetor TF-IDF. O TF-IDF, que é uma abreviação de "*Term Frequency-Inverse Document Frequency*", considerada uma forma simples e eficiente para realizar a correspondência das palavras em uma consulta com documentos, apontando aquelas que são mais relevantes (Ramos, 2003). Neste estudo, o TF-IDF está configurado com min df=5 e max df=0,8, ou seja, o processo de vetorização eliminou as palavras que apareceram em menos de 5 documentos e em 80% do total de documentos. É uma tentativa de reduzir a quantidade de palavras sem atrapalhar o modelo. Outro algoritmo muito importante para classificação de textos é o BM25 (Robertson & Zaragoza, 2009), que utiliza função de pontuação probabilística. BM é uma abreviação de "*best matching*" (melhor correspondência), que fornece uma pontuação entre a consulta e os documentos, com base nos termos comuns. O *Scikit Learn* não fornece uma implementação do BM25, por isso usamos um algoritmo adaptado fornecido pela comunidade.

3.2 Reamostragem (*Resampling*)

De acordo com observado Fig. 1, verifica-se que os dados estão, de certa forma, desequilibrados, já que há uma incidência maior das classificações de contexto policial em relação a não policial. Modelos de previsão construídos a partir de conjuntos de dados desequilibrados costumam ser tendenciosos para a maioria (Santos et al, 2018), portanto, é importante saber como resolver esse problema caso os testes do nosso conjunto de dados pareçam tendenciosos. Felizmente, existem algumas maneiras de balancear o conjunto de dados sem a necessidade de fazer manualmente. Para isso, utilizamos a biblioteca "*Imbalanced Learn*", que funciona muito bem junto com o *scikit*. A biblioteca fornece técnicas de reamostragem como *Oversampling*, que aumenta o menor lado e o *Undersampling*, que diminui o lado maior do conjunto de dados.

3.3 Algoritmos de Classificação

Com os documentos corretamente vetorizados usando as referidas abordagens, há a necessidade de treinar o modelo, mediante um algoritmo de classificação apropriado. Os algoritmos escolhidos foram *Random Forest, NaiveBayes, SVM (LinearSVC e SVC), SGDClassifier e MLPClassifier*. O critério de escolha foi a disponibilidade dentro da ferramenta *scikit learn* juntamente com a frequência em que são citados nos estudos da seção de trabalhos relacionados. Cada algoritmo foi devidamente testado em cenários distintos, para que fosse possível comparar os resultados, o que possibilitou verificar qual deles é o mais adequado para resolver o problema em tela.

As especificações de hardware utilizado nesta experiência são: CPU: Intel(R) Xeon(R) CPU @ 2.30GHz; RAM: 12.6 GB; Disco 33 GB.

4. AVALIAÇÃO

O processo de avaliação é muito importante e necessita de uma atenção especial, pois uma abordagem incorreta nesta tarefa poderá sobreajustar o modelo, também conhecido como *overfit*. Caso os dados de treinamento atinjam um resultado muito alto, corremos o risco de encontrar uma regra preditiva geral (Dietterich, 1995). O processo de avaliação deve usar todo o conjunto de dados disponível ou até mesmo novos registros possíveis. Os resultados da avaliação devem ser bastante semelhantes ao que é demonstrado nos resultados do treinamento. Caso contrário, se os dados treinados tiverem resultados melhores do que a avaliação, o modelo será superajustado (*overfit*). Durante esse processo, é necessário gerar a quantidade adequada de métricas para comprovar que o modelo é adequado para ser implementado na Secretaria da Segurança Pública.

Existem muitas possibilidades para realizar a avaliação, e neste estudo são utilizadas duas técnicas. Primeiramente, dividindo os dados para treinamento e testes (*Split training and test data*). Posteriormente, usamos uma técnica de validação cruzada (*cross-validation*).

4.1 *Split training and test data*

O conjunto de dados foi dividido aleatoriamente em duas partes: 80% para treinamento e 20% para teste. Depois disso, o modelo foi treinado para tentar prever o restante dos dados que não foram usados no treinamento. Como é um modelo supervisionado, podemos descobrir se ele possui uma boa precisão. Durante o teste, utilizamos as seguintes métricas: *Accuracy, Precision, Recall* e f1. *Accuracy* indica um desempenho geral do modelo; *Precision* mostra quantos positivos estão classificados corretamente; *Recall* é a frequência com que cada classe é classificada; finalmente, F1 combina *Accuracy* e *recall*, sendo uma média harmônica entre eles. Os resultados foram ainda plotados em uma Matriz de Confusão, para uma melhor observação deste resultado.

Usando as métricas acima, os algoritmos foram treinados e testados com todas as alternativas possíveis, combinando vetorização de texto, *resampling* e algoritmo de classificação. Por exemplo, escolhendo o TF-IDF sem reamostragem e algoritmo *RandomForest*, como é possível ver na Tabela I. Conforme mostrado nessa tabela, o TF-IDF e a Vetorização BM-25 apresentaram resultados muito semelhantes, embora o BM25 tenha demorado mais. Dados desequilibrados e subamostrados apresentaram resultados mais baixos em precisão e o

restante das métricas parecia tendencioso. Os melhores resultados na maioria dos algoritmos são demonstrados quando é utilizado o processo *Oversampler*, principalmente no algoritmo *Random Forest* utilizando TF-IDF ou mesmo BM25. O bom resultado dessa combinação pode ser ilustrado em uma Matriz de Confusão plotada na Figura 3.

O uso da vetorização TF-IDF com algoritmo *Random Forest* e estratégia *OverSampling* pareceu muito apropriado, porém, é prudente confirmá-lo usando mais alternativas de avaliação, neste caso, a *Cross-Validation.*

4.2 Validação Cruzada (*Cross-Validation*)

A validação cruzada é um método de reamostragem de dados usado para estimar o verdadeiro erro de previsão dos modelos e ajustar os parâmetros do modelo (Berrar, 2019). Neste teste, nossos dados foram divididos em k dobras, que gerarão k modelos diferentes, usando dados diferentes para treinamento em cada caso. Além disso, é muito importante prestar atenção nas classes, no nosso caso, "contexto policial" e "contexto não policial", que precisam ser balanceados em cada dobra. Caso contrário, corre-se o risco de dobras específicas serem treinadas apenas usando uma classe. Assim, utilizamos a estratificação, que reduz a possibilidade de modelos tendenciosos (Berrar, 2019).

Tabela 1. Comparação entre diferentes combinações para geração do modelo.

		Accuracy		Precision		Recall		F1		Time Elapsed (sec)	
		TF-IDF	*BM25*	*TF-IDF*	*BM25*	*TF-IDF*	*BM25*	*TF-IDF*	*BM25*	*TF-IDF*	*BM25*
Random Forest	Imbalanced	0,823	0,825	[0.84 0.82]	[0.85 0.82]	[0.61 0.94]	[0.61 0.94]	[0.71 0.87]	[0.71 0.87]	40.8	53
	Oversample	0,904	0,892	[0.89 0.92]	[0.89 0.9]	[0.92 0.89]	[0.9 0.89]	[0.91 0.9]	[0.89 0.89]	55	74
	Undersample	0,820	0,824	[0.83 0.82]	[0.82 0.83]	[0.82 0.82]	[0.84 0.81]	[0.82 0.82]	[0.83 0.82]	24	36
Naive Bayes	Imbalanced	0,825	0,827	[0.8 0.84]	[0.81 0.85]	[0.67 0.91]	[0.67 0.91]	[0.73 0.87]	[0.73 0.87]	38	50
	Oversample	0,817	0,816	[0.8 0.83]	[0.78 0.86]	[0.84 0.79]	[0.87 0.76]	[0.82 0.81]	[0.83 0.81]	1.2	1.38
	Undersample	0,804	0,798	[0.78 0.83]	[0.77 0.83]	[0.83 0.78]	[0.85 0.75]	[0.81 0.8]	[0.81 0.79]	19	31
Linear SVC	Imbalanced	0,834	0,831	[0.79 0.86]	[0.78 0.86]	[0.73 0.89]	[0.72 0.89]	[0.75 0.87]	[0.75 0.87]	36	1.60
	Oversample	0,870	0,872	[0.85 0.9]	[0.84 0.91]	[0.9 0.84]	[0.91 0.83]	[0.87 0.87]	[0.88 0.87]	1.7	75
	Undersample	0,811	0,771	[0.8 0.82]	[0.78 0.76]	[0.83 0.79]	[0.77 0.77]	[0.82 0.81]	[0.78 0.76]	16	28
SGD Classifier	Imbalanced	0,840	0,835	[0.8 0.86]	[0.84 0.84]	[0.72 0.9]	[0.68 0.92]	[0.76 0.88]	[0.75 0.88]	33.87	1.78
	Oversample	0,856	0,865	[0.85 0.86]	[0.83 0.91]	[0.86 0.85]	[0.92 0.81]	[0.86 0.85]	[0.87 0.86]	1.2	74
	Undersample	0,823	0,785	[0.82 0.82]	[0.75 0.83]	[0.82 0.83]	[0.85 0.73]	[0.82 0.83]	[0.8 0.78]	13	26
MLP Classifier	Imbalanced	0,786	0,804	[0.7 0.83]	[0.73 0.84]	[0.69 0.84]	[0.7 0.86]	[0.7 0.84]	[0.72 0.85]	379	781

	Oversample	0,863	0,880	[0.83 0.9]	[0.85 0.92]	[0.91 0.82]	[0.92 0.84]	[0.87 0.86]	[0.88 0.88]	393	805
	Undersample	0,774	0,792	[0.77 0.78]	[0.78 0.8]	[0.78 0.77]	[0.8 0.78]	[0.78 0.77]	[0.79 0.79]	228	436
SVC	Imbalanced	0,834	0,839	[0.8 0.85]	[0.81 0.85]	[0.71 0.9]	[0.72 0.91]	[0.75 0.88]	[0.76 0.88]	672	1179
	Oversample	0,855	0,875	[0.85 0.87]	[0.84 0.91]	[0.87 0.84]	[0.92 0.83]	[0.86 0.85]	[0.88 0.87]	956	2889
	Undersample	0,826	0,788	[0.82 0.83]	[0.78 0.8]	[0.84 0.82]	[0.8 0.78]	[0.83 0.82]	[0.79 0.79]	382	621

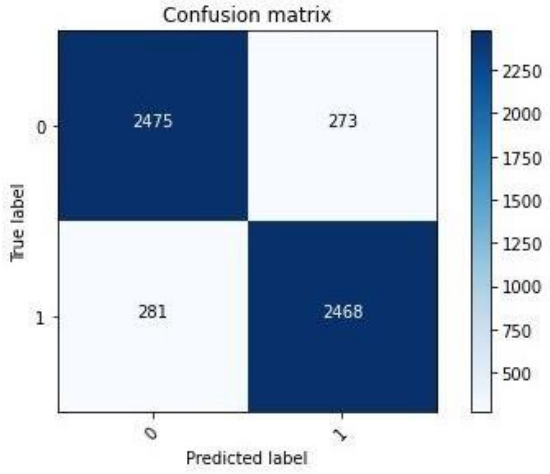

Figura 3. Matrix de Confusão utilizando *Random Forest*, TF-IDF e *OverSampling*

Conforme demonstrado na Tabela II, uma estratégia de validação cruzada estratificada com 10 dobras em um algoritmo *Random Forest* parece bem executada, confirmando a primeira avaliação que divide o conjunto de dados. Nesse caso, os dados usados não são sobreamostrados, o que reduz o potencial de sobreajuste. A média deste processo de validação cruzada representa o resultado em 0,824 de acurácia, 0,819 de precisão, 0,935 de recall e 0,873 de f1, o que confirma que esta combinação pode ser adequada para ser utilizada em futuras previsões.

Tabela 2. Cross-Validation com *Random Forest*, TF-IDF e *OverSampling*

Fold	Accuracy	Precision	Recall	F1
0	0.838831	0.826996	0.949782	0.884146
1	0.825636	0.817035	0.941818	0.875000
2	0.823668	0.819693	0.933042	0.872703
3	0.826025	0.821909	0.933770	0.874276
4	0.819425	0.816613	0.930131	0.869684
5	0.821782	0.814394	0.938865	0.872211
6	0.822254	0.824756	0.921397	0.870402
7	0.817539	0.810573	0.937409	0.869389
8	0.810467	0.810742	0.922853	0.863172
9	0.836398	0.828115	0.943231	0.881933

5. RESULTADOS E IMPLANTAÇÃO

Conforme demonstrado nas últimas subseções, um modelo treinado usando vetorização TF-IDF com algoritmo de classificação de *Randon Forest* e estratégia de sobreamostragem pode ser a melhor opção a ser usada para prever dados futuros neste cenário estudado. Assim, ele será implantado como modelo usado para prever novos dados. A linguagem *Python* fornece um módulo chamado "*pickle*" que implementa protocolos binários para serializar e desserializar uma estrutura de objeto. Armazenar o modelo treinado em um arquivo *pickle* pode ser útil, pois não há necessidade de treinar o mesmo modelo toda vez que for necessário prever novos dados.

O código fonte utilizado neste trabalho foi disponibilizado em um repositório público[1] *git* contendo todos os *scripts* e o modelo treinado, estando disponível o uso pela Secretaria da Segurança Pública de São Paulo ou outra instituição pública que possa ter problemas semelhantes. Além disso, o repositório fornece uma *API REST* e um cliente *frontend* preparado para realizar a tarefa de classificação. Para prever novos dados, basta carregar no *frontend* um arquivo .CSV contendo uma única coluna com todos os boletins de ocorrência em cada linha. Então, o sistema retornará o arquivo preenchido identificando em uma nova coluna se o crime tem ou não contexto policial. Há também outra opção, que insere um histórico de boletim de ocorrência individual em um campo de texto do próprio *frontend*, com classificação automática como "contexto policial" ou não. O desenvolvimento de um *frontend* intuitivo pode ajudar os usuários que não possuem conhecimento de linguagem de programação, portanto, não há necessidade de interagir diretamente com os scripts *Python*.

6. CONCLUSÕES E TRABALHOS FUTURO

Demonstramos neste artigo um problema real de classificação de texto enfrentado pelo setor de segurança pública no Brasil, que pode ser resolvido usando técnicas de mineração de texto e *machine learning*. Existe uma vasta bibliografia disponível sobre classificação de textos, no entanto, foram verificadas poucas referências sobre sua utilização em boletins de ocorrência. Apesar disso, o uso correto dos processos CRISP-DM ajuda a entender o negócio e os dados, ao aplicar as abordagens e tecnologias corretas para preparar e processar os dados, implantando um modelo apropriado que ajude a resolver o problema principal. Existem muitas possibilidades para solucionar a questão de mineração de texto, e a execução de várias combinações deles ajudou a comparar entre si e encontrar uma solução mais adequada.

É importante monitorar a qualidade da previsão relacionada aos novos dados de boletins de ocorrência que serão imputados no futuro, assim, os especialistas em atividades de classificação precisam eventualmente verificar se o modelo cataloga de forma semelhante se comparado com as métricas apresentadas neste trabalho, a fim de proporcionar um melhor ajuste no modelo.

Para trabalhos futuros, a classificação do crime poderia ser dividida em classes mais específicas. Por exemplo, além de classificar o contexto, descobrir se o crime de morte ocorreu em ambiente familiar, área privada, contra criança ou mulher, ou ainda tentar identificar o objeto usado no crime, como arma branca ou arma de fogo. Além disso, parece um trabalho interessante utilizar as técnicas aqui apresentadas para relacionar outros crimes além do homicídio, como roubos, estupros ou qualquer tipo de violência contra pessoas vulneráveis. Tal ação ajudará a compreender mais os diferentes crimes e propor melhores políticas de prevenção desenvolvidas pelas instituições policiais.

Por fim, todas as bibliotecas utilizadas nestes estudos são de código aberto e não requer recursos computacionais caros para serem processadas. Assim, se os objetivos forem cumpridos, o uso deste algoritmo justificará o dinheiro do contribuinte e os recursos humanos, otimizando significativamente um importante serviço público como segurança pública.

AGRADECIMENTOS

Nós gostaríamos de agradecer à Universidade de Brasília, à Secretaria da Segurança Pública e à Polícia Militar do Estado de São Paulo, por colaborarem com a pesquisa.

REFERÊNCIAS

Berrar, D. (2019). *Cross-Validation*. Encyclopedia of bioinformatics and computational biology, Tokyo, Japan.

Brandenburg, M. (2017). *Text Classification of Dutch police records.* Utrecht University.

Brazil (1988) *Constitution of Brazil*, http://www.planalto.gov.br/ccivil_03/constituicao/constituicao.htm, accessed 22 July 2022.

Carnaz, G., Beires Nogueira, V., Antunes, M., & Ferreira, N. (2018). *An automated system for criminal police reports analysis. In International Conference on Soft Computing and Pattern Recognition* (pp. 360-369). Springer, Cham.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). *Bert: Pre-training of deep bidirectional transformers for language understanding*. arXiv preprint arXiv:1810.04805.

Dietterich, T. (1995). *Overfitting and undercomputing in machine learning*. ACM computing surveys (CSUR), 27(3), 326-327.

Feldman, R., & Dagan, I. (1995). *Knowledge discovery in textual databases (KDT)*. In proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95), Montreal, Canada.

Frawley, W. J., Piatetsky-Shapiro, G., & Matheus, C. J. (1992). Knowledge discovery in databases: An overview. AI magazine, 13(3), 57-57.

Hotho, A., Nürnberger, A., & Paaß, G. (2005). *A brief survey of text mining*. In Ldv Forum (Vol. 20, No. 1, pp. 19-62).

Ikonomakis, M., Kotsiantis, S., & Tampakas, V. (2005). *Text classification using machine learning techniques*. WSEAS transactions on computers, 4(8), 966-974.

Ku, C. H., & Leroy, G. (2014). *A decision support system: Automated crime report analysis and classification for e-government*. Government Information Quarterly, 31(4), 534-544.

Lovins, J. B. (1968). *Development of a stemming algorithm*. Mech. Transl. Comput. Linguistics, 11(1-2), 22-31.

Qudar, M. M. A., & Mago, V. (2020). *Tweetbert: a pretrained language representation model for twitter text analysis*. arXiv preprint arXiv:2010.11091.

Ramos, J. (2003, December). *Using tf-idf to determine word relevance in document queries*. In Proceedings of the first instructional conference on machine learning (Vol. 242, No. 1, pp. 29-48).

Robertson, S., & Zaragoza, H. (2009). *The probabilistic relevance framework: BM25 and beyond.* Foundations and Trends in Information Retrieval, 3(4), 333-389.

Santos, M. S., Soares, J. P., Abreu, P. H., Araujo, H., & Santos, J. (2018). *Cross-validation for imbalanced datasets: avoiding overoptimistic and overfitting approaches*. IEEE Computational intelligence magazine, 13(4), 59-76.

Sao Paulo (1906) *Law 1006. Creation of Sao Paulo Department of Public Safety*, https://www.al.sp.gov.br/repositorio/legislacao/lei/1906/lei-1006-17.09.1906.html, accessed 22 July 2022.

Sao Paulo (2022) *Indicators of crime in the State of São Paulo*, https://www.al.sp.gov.br/repositorio/legislacao/lei/1906/lei-1006-17.09.1906.html, accessed 22 July 2022.

Suneera, C. M., & Prakash, J. (2020, December). *Performance analysis of machine learning and deep learning models for text classification*. In 2020 IEEE 17th India Council International Conference (INDICON) (pp. 1-6). IEEE.

UN (2022) *Transforming our world: the 2030 Agenda for Sustainable Development*, https://sdgs.un.org/2030agenda, accessed 22 July 2022.

Wilbur, W. J., & Sirotkin, K. (1992). *The automatic identification of stop words*. Journal of information science, 18(1), 45-55.

Wirth, R., & Hipp, J. (2000). *CRISP-DM: Towards a standard process model for data mining*. In Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining (Vol. 1, pp. 29-39).

Zambrano, F. J. R., Flores, B. F. V., Mendoza, W. J. P., Zambrano, R. A. R., & Rivadeneira, S. M. C. (2020). *Aplicación de minería de texto para el análisis de sentimientos del servicio de telefonía móvil en el ecuador*. Holos, 7, 1-16.