UTILIZAÇÃO DE NLP PARA PREDIÇÕES DE PRISÕES: Um estudo de caso com dados sobre os crimes de Chicago

USING NLP FOR PRISON PREDICTIONS: A case study with data regarding crime in Chicago

[1*]Gomes, Lucas Olian[a]. Yokota, Maria Angélica Pires[b]. Silva, Adriano Valério Santos da[c]

[a, b, c] Centro Universitário Facens - Sorocaba, SP, Brasil
lu.olian@hotmail.com
maria.a.yokota@gmail.com
adrianovss@gmail.com

Submetido em: 22 de Fevereiro de 2025

RESUMO

O Processamento de Linguagem Natural (PLN) é uma tecnologia que permite a interpretação da linguagem humana por máquinas, combinando Machine Learning, Deep Learning e linguística computacional. Este estudo utilizou o PLN para analisar relatos de crimes na cidade de Chicago, utilizando o conjunto de dados do sistema CLEAR *(Citizen Law Enforcement Analysis and Reporting*) do Departamento de Polícia de Chicago, referente ao ano de 2018. O objetivo foi identificar padrões textuais que pudessem auxiliar na previsão de resultados judiciais, como a probabilidade de prisão.
Durante a metodologia, foram aplicadas técnicas de pré-processamento, incluindo remoção de *stop words, stemming, lemmatization* e *tokenization*, além do uso da técnica TF-IDF (*Term Frequency - Inverse Document Frequency*) para conversão dos textos em representações numéricas. Para o balanceamento das classes, foi utilizado o SMOTE (*Synthetic Minority Over-sampling Technique*), que, no entanto, não apresentou eficácia significativa na melhoria do desempenho dos modelos. Diversos algoritmos de classificação foram testados, incluindo *Gaussian Naive Bayes, Adaboost Classifier, Random Forest, Decision Tree, Support Vector Machine* (SVM) e uma Rede Neural Artificial, todos avaliados também pela plataforma *PyCaret*, cujos ajustes de hiperparâmetros não trouxeram mudanças drásticas nos resultados.
Os principais achados indicam que, embora o SMOTE não tenha sido efetivo e os modelos otimizados pelo *PyCaret* não tenham mostrado avanços significativos, foi possível alcançar uma precisão (*precision*) relativamente alta na previsão de prisões, especialmente com o modelo SVM. Esses resultados reforçam o potencial do PLN na análise de dados criminais, contribuindo para o desenvolvimento de ferramentas preditivas que podem apoiar tanto a segurança pública quanto o sistema de justiça.

Palavras-chave: Processamento de Linguagem Natural, Machine Learning, Crimes em Chicago, Classificação de Texto, Inteligência Artificial.

ABSTRACT

Natural Language Processing (NLP) is a technology that allows machines to interpret human language, combining Machine Learning, Deep Learning and computational linguistics. This study used NLP to analyze crime reports in the city of Chicago, using the data set from the Chicago Police Department's CLEAR (Citizen Law Enforcement Analysis and Reporting) system for the year 2018. The aim was to identify textual patterns that could help predict judicial outcomes, such as the likelihood of arrest.

1

During the methodology, pre-processing techniques were applied, including the removal of stop words, stemming, lemmatization and tokenization, as well as the use of the TF-IDF (Term Frequency - Inverse Document Frequency) technique to convert the texts into numerical representations. SMOTE (Synthetic Minority Over-sampling Technique) was used to balance the classes, but it was not significantly effective in improving the performance of the models. Several classification algorithms were tested, including Gaussian Naive Bayes, Adaboost Classifier, Random Forest, Decision Tree, Support Vector Machine (SVM) and an Artificial Neural Network, all also evaluated using the PyCaret platform, whose hyperparameter adjustments did not bring about drastic changes in the results.
The main findings indicate that, although SMOTE wasn't effective and the models optimized by PyCaret did not show any meaningful advances, it was possible to achieve a high precision in the prison predicts, specially using the SVM model. These results strengthen the potential of NLP in criminal data analysis, contributing to the development of predictive tools that can support both the public safety and the justice system.

Keywords: Natural Language Processing, Machine Learning, Chicago Crimes, Text Classification, Artificial Intelligence.

1. INTRODUÇÃO

A cidade de Chicago, localizada no estado de Illinois, nos Estados Unidos da América, tem sido um ponto focal em discussões sobre crimes e segurança pública. Segundo a plataforma americana *NeighborhoodScout* (SCHILLER, 2025), a cidade, com uma taxa de crime de 40 a cada mil moradores, possui uma das taxas mais altas no país comparada com todas as outras comunidades, desde as menores até as maiores cidades.

Com uma população diversa e com uma paisagem urbana complexa, a cidade passa por desafios quando o assunto é prevenção de crimes. Nos anos mais recentes, a análise de dados se tornou uma ferramenta muito forte para alguns padrões de crime começarem a ser entendidos. Por esse meio, pesquisadores e autoridades podem estudar algumas causas, distribuições geográficas e tendências das atividades criminais para melhor entendimento.

O estudo deste artigo trará como principal referência a base de dados oficial de Crimes em Chicago, encontrada no *Chicago Data Portal* e utilizará técnicas de Processamento de Linguagem Natural (NLP) e análise exploratória para extrair informações relevantes a fim de gerar *insights.* O objetivo é ser capaz de prever, a partir da descrição de um crime, se a pessoa seria presa ou não.

Essa abordagem inovadora oferece um potencial significativo para melhorar a compreensão dos fatores que influenciam as decisões judiciais em Chicago, possibilitando, assim, uma análise mais profunda sobre como as autoridades aplicam as punições, além de auxiliar no aprimoramento de

estratégias de prevenção e intervenção.

2. REFERENCIAL TEÓRICO

a. *Natural Language Processing* (NLP)

A cada dia, a internet e as redes sociais estão carregadas de informações geradas por usuários, muitas das quais não foram pensadas para serem interpretadas por máquinas. O Processamento de Linguagem Natural (ou *Natural Language Processing*, NLP) é um campo que envolve uma série de técnicas que permitem que os computadores analisem e compreendam a linguagem humana (CAMBRIA; WHITE, 2014).

O NLP é um campo da Inteligência Artificial que se concentra na interação entre computadores e a linguagem humana. Seu objetivo principal é permitir que as máquinas compreendam, interpretem e manipulem a linguagem natural de forma útil e eficiente (JURAFSKY; MARTIN, 2021).

Outro aspecto importante do NLP é a capacidade de lidar com ambiguidades linguísticas. A linguagem humana é repleta de nuances, gírias, metáforas e contextos variáveis, o que torna a interpretação automatizada um grande desafio. Técnicas como o reconhecimento de entidades nomeadas (NER, na sigla em inglês), a análise de dependência sintática e a desambiguação de palavras têm sido amplamente empregadas para melhorar a precisão das interpretações (NADEAU; SEKINE, 2007).

b. TF-IDF

TF-IDF (*Term Frequency - Inverse Document Frequency*) é uma das técnicas mais amplamente utilizadas no campo do Processamento de Linguagem Natural (NLP) para representar e avaliar a importância das palavras dentro de um conjunto de documentos. O objetivo principal do TF-IDF é destacar termos que são mais relevantes para um determinado documento dentro de uma coleção e, ao mesmo tempo, diminuir a importância de palavras que aparecem frequentemente em todo o conjunto de dados, como palavras de parada (*stop words*) (RAMOS, 2003).

c. Classificadores

Os classificadores são modelos de aprendizado de máquina utilizados

para prever categorias ou rótulos a partir de dados de entrada, com aplicações em análise de sentimentos, diagnóstico médico, reconhecimento de imagens e, no contexto do PLN (Processamento de Linguagem Natural), na categorização de documentos e previsão de penalidades associadas a crimes (BISHOP, 2006).

Entre os principais classificadores, destaca-se o Naive Bayes, um modelo probabilístico baseado no Teorema de Bayes, conhecido por sua simplicidade e eficiência em tarefas de classificação de textos, mesmo assumindo a independência entre as características (RISH, 2001). O *Random Fores*t, por sua vez, utiliza um conjunto de árvores de decisão para aumentar a precisão e a robustez, realizando previsões por meio de votação majoritária entre as árvores (BREIMAN, 2001).

O *Support Vector Machine* (SVM) é um algoritmo de *Machine Learning* amplamente utilizado em tarefas de classificação e regressão, especialmente eficaz em problemas de classificação binária. Seu principal objetivo é encontrar um hiperplano que melhor separa as classes no espaço de dados, maximizando a margem entre elas. Para lidar com dados não linearmente separáveis, o SVM utiliza o *kernel trick*, projetando os dados em um espaço de maior dimensão (CORTES; VAPNIK, 1995).

Por fim, o *Decision Tree* é utilizado tanto para classificação quanto para regressão, sendo estruturado na forma de uma árvore de decisão. Cada nó interno representa uma condição baseada em uma característica dos dados, os ramos indicam os possíveis resultados dessa condição e os nós folha fornecem a classificação final ou o valor previsto (QUINLAN, 1986).

d. Ferramentas para análise exploratória

A análise exploratória de dados é uma etapa fundamental no processo de análise de dados que envolve a inspeção e a visualização dos dados para entender sua estrutura, identificar padrões e detectar anomalias. O objetivo principal da análise exploratória é obter entendimentos iniciais que possam orientar as etapas subsequentes de modelagem e análise. Essa abordagem envolve a aplicação de várias ferramentas e técnicas para explorar as características dos dados antes de realizar uma análise mais aprofundada ou construir modelos preditivos.

Para analisar, podemos utilizar ferramentas de análise estatística descritiva (tais como média, mediana, desvio padrão e variância), visualização de dados (como histogramas e *boxplots*), bem como as descritas a seguir.

i. Nuvem de palavras

A nuvem de palavras (ou *word cloud*) é uma ferramenta visual que representa a frequência de palavras em um determinado conjunto de texto de maneira gráfica. As palavras que aparecem mais frequentemente são exibidas em tamanho maior, enquanto as palavras menos frequentes são exibidas em menor tamanho. Essa técnica de visualização é útil para identificar rapidamente os termos mais relevantes ou predominantes em um conjunto de dados textuais, proporcionando uma compreensão intuitiva do conteúdo sem a necessidade de análises complexas.

ii. *Stemming, Lemmatization e Tokenization*

O *stemming* é o processo de remover afixos de palavras para reduzir a palavra a um radical ou forma base, que nem sempre é uma palavra real. O algoritmo de *stemming* trabalha com regras simples e rápidas para cortar partes das palavras de forma direta, com o objetivo de reduzir as palavras ao seu radical, independentemente de seu significado gramatical.

Por sua vez, *lemmatization* (ou lematização, em português) é uma abordagem mais sofisticada, que visa reduzir uma palavra à sua forma base ou lema. O lema é a forma dicionarizada de uma palavra, ou seja, a forma que você encontraria em um dicionário. Ao contrário do *stemming*, a *lemmatization* leva em consideração o contexto gramatical da palavra, como o tempo verbal, o número e o gênero, para garantir que a forma reduzida seja linguística e semanticamente correta.

A tokenização, de outra maneira, é um dos passos desta transformação durante o processamento de linguagem natural. A tokenização significa dividir o texto de entrada, que para um computador é apenas uma longa sequência de caracteres, em subunidades, chamadas *tokens*. Estes *tokens* são depois introduzidos em etapas subsequentes do processamento da linguagem natural, como a análise morfológica, a etiquetagem de classes de palavras e a análise (GREFENSTETTE, 1999).

3. METODOLOGIA

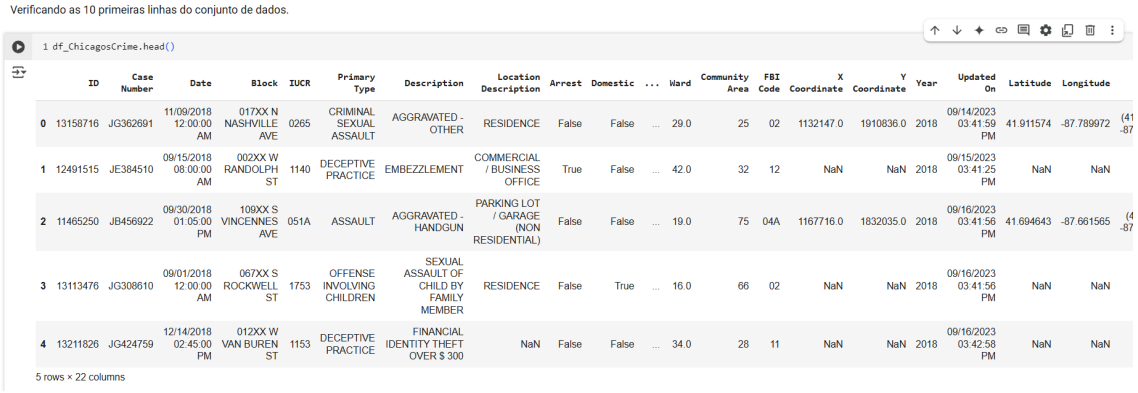
a. Conjunto de dados

O conjunto de dados escolhido para este projeto consiste em incidentes de crimes relatados na cidade de Chicago de 2018. Os dados são extraídos do sistema CLEAR (Citizen Law Enforcement Analysis and Reporting) do Departamento de Polícia de Chicago, no qual é possível filtrar pelo ano desejado de 2001 a 2019. Esse é um dos mais ricos conjuntos de dados na área de crimes, fornecendo informações detalhadas sobre a data, tipo, descrição e localização dos crimes, entre outros detalhes relevantes para nossa análise.

Além disso, o conjunto de dados inclui informações sobre a hora do dia em que os crimes ocorreram, o distrito policial responsável pela área, o status do caso (se foi resolvido ou não) e a descrição das armas usadas, se aplicável. Essas informações adicionais permitem uma análise mais aprofundada dos padrões de criminalidade, ajudando a identificar tendências e áreas de alta incidência de crimes.

A base de dados possui 22 colunas e 269052 linhas, entre elas o "*Arrest",* que é a informação que indica se uma prisão foi feita. A Figura 1 apresenta visualmente os dados utilizados.

Figura 1: Cinco primeiras linhas da Base de Dados.



Verificando as 10 primeiras linhas do conjunto de dados.

```
1 df_ChicagosCrime.head()
```

	ID	Case Number	Date	Block	IUCR	Primary Type	Description	Location Description	Arrest	Domestic	...	Ward	Community Area	FBI Code	X Coordinate	Y Coordinate	Year	Updated On	Latitude	Longitude	
0	13158716	JG362691	11/09/2018 12:00:00 AM	017XX N NASHVILLE AVE	0265	CRIMINAL SEXUAL ASSAULT	AGGRAVATED - OTHER	RESIDENCE	False	False	...	29.0	25	02	1132147.0	1910836.0	2018	09/14/2023 03:41:59 PM	41.911574	-87.789972	(41 -87
1	12491515	JE384510	09/15/2018 08:00:00 AM	002XX W RANDOLPH ST	1140	DECEPTIVE PRACTICE	EMBEZZLEMENT	COMMERCIAL / BUSINESS OFFICE	True	False	...	42.0	32	12	NaN	NaN	2018	09/15/2023 03:41:25 PM	NaN	NaN	
2	11465250	JB456922	09/30/2018 01:05:00 PM	109XX S VINCENNES AVE	051A	ASSAULT	AGGRAVATED - HANDGUN	PARKING LOT / GARAGE (NON RESIDENTIAL)	False	False	...	19.0	75	04A	1167716.0	1832035.0	2018	09/16/2023 03:41:56 PM	41.694643	-87.661565	(4 -87
3	13113476	JG308610	09/01/2018 12:00:00 AM	067XX S ROCKWELL ST	1753	OFFENSE INVOLVING CHILDREN	SEXUAL ASSAULT OF CHILD BY FAMILY MEMBER	RESIDENCE	False	True	...	16.0	66	02	NaN	NaN	2018	09/16/2023 03:41:56 PM	NaN	NaN	
4	13211826	JG424759	12/14/2018 02:45:00 PM	012XX W VAN BUREN ST	1153	DECEPTIVE PRACTICE	FINANCIAL IDENTITY THEFT OVER $ 300	NaN	False	False	...	34.0	28	11	NaN	NaN	2018	09/16/2023 03:42:58 PM	NaN	NaN	

5 rows × 22 columns

Fonte: CLEAR.

b. Pré-processamento e Análise Exploratória

Inicialmente, foi realizada uma análise exploratória para compreender a distribuição e os padrões dos crimes. Como parte desse processo, nuvens de palavras foram geradas antes e depois do pré-processamento, permitindo a visualização dos termos mais frequentes nos relatos.

Após essa etapa, diversas técnicas de pré-processamento foram aplicadas para limpeza e padronização dos dados textuais. Caracteres especiais, números e pontuação foram removidos, uma vez que não agregam valor à análise. Em seguida, os textos foram convertidos para letras minúsculas, garantindo uniformidade.

A figura abaixo apresenta duas versões de uma *word cloud* gerada a partir das descrições dos crimes. A primeira exibe as palavras mais frequentes nos textos originais, sem qualquer processamento adicional. A segunda exibe essas palavras mais frequentes já com as *stop words* removidas. É possível observar que não houve uma mudança significativa.

Figura 2: *Word Cloud* comparativo

Fonte: próprio autor.

Para reduzir a dimensionalidade e aprimorar a qualidade das representações textuais, foram empregadas técnicas de remoção de *stop words*, *stemming* e *lemmatization*. Além disso, foi realizada a tokenização, transformando os textos em unidades menores para processamento.

c. Balanceamento de Dados com SMOTE

A base de dados original apresentava um desbalanceamento entre as classes da variável *Arrest* (prisão), onde crimes com e sem prisões possuíam

distribuições diferentes. Para lidar com esse problema e evitar que o modelo de *machine learning* fosse enviesado para a classe majoritária, foi utilizada a técnica SMOTE (*Synthetic Minority Over-sampling Technique*).

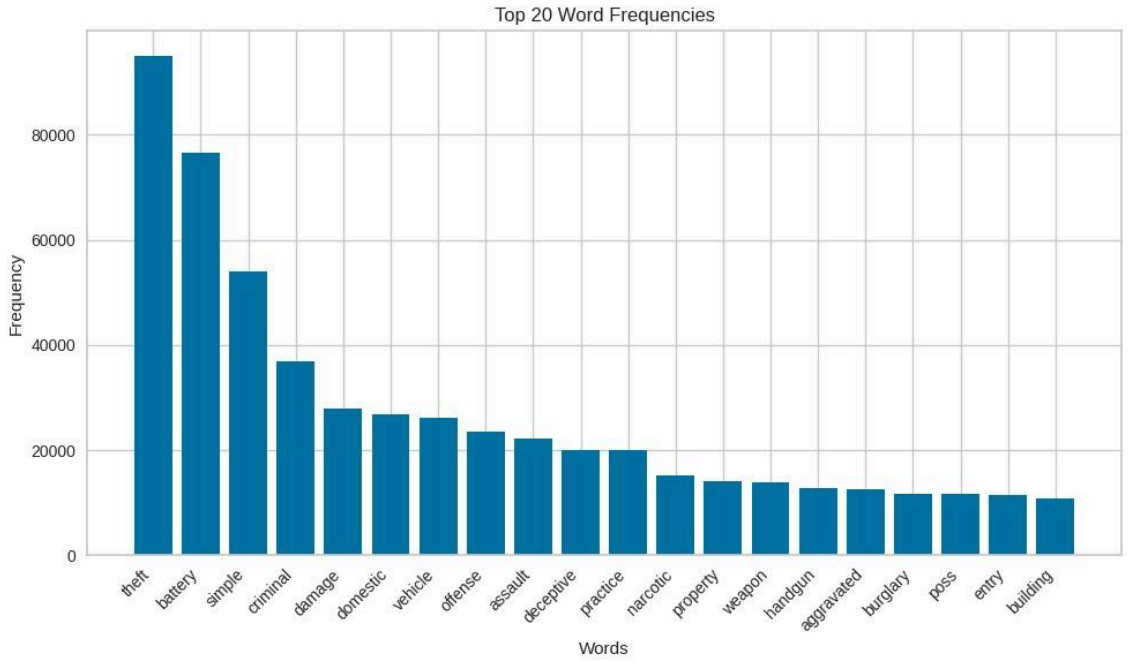
d. Representação Vetorial com TF-IDF

A técnica TF-IDF (*Term Frequency - Inverse Document Frequency*) foi utilizada para converter os textos em representações numéricas, atribuindo pesos às palavras com base em sua relevância dentro do conjunto de dados.

O resultado final foi uma matriz esparsa representando os textos em um espaço vetorial, onde cada linha corresponde a um relato de crime e cada coluna representa um termo do vocabulário aprendido. Essa matriz foi, então, utilizada como entrada para os modelos de classificação.

Ao concatenar todas as palavras das colunas *Primary Type* e *Description*, tokenizar o texto, calculando a frequência das palavras, e converter para um *DataFrame* para melhor visualização, foi possível gerar um gráfico com as 20 palavras mais frequentes, que pode ser observado abaixo.

Figura 3: Gráfico com as 20 palavras mais frequentes.



Fonte: próprio autor.

e. Modelo

Testamos diversos algoritmos para prever informações sobre os crimes com base nas descrições textuais. Os modelos utilizados foram:

- Gaussian Naïve Bayes
- AdaBoost Classifier
- Random Forest
- Decision Tree
- Support Vector Machine (SVM)
- Rede Neural Artificial

Para facilitar a comparação e otimização dos modelos, utilizamos a biblioteca *PyCaret*, que permite o ajuste automatizado de hiperparâmetros e avaliação do desempenho dos classificadores.

4. RESULTADOS E DISCUSSÕES

Com base nos resultados dos modelos de classificação, obtivemos resultados satisfatórios. Conforme a Figura 4 observamos que a métrica Precisão (*Precision)* apresentou um valor relativamente alto. No total, testamos seis modelos de classificação e um modelo de regressão. Também realizamos testes em dois conjuntos de dados distintos: um utilizando o filtro de *stopwords* e o pré-processamento do TF-IDF, e outro apenas com a aplicação do balanceamento *SMOTE*, para avaliarmos a diferença na aplicação de ambos e seus resultados. Após vários testes, o modelo com melhor desempenho, validado tanto por nossos modelos prévios quanto pelo uso do *Pycaret*, foi o classificador *Decision Tree Classifier*.

Além disso, a aplicação apenas do *SMOTE* não se mostrou eficaz, tendo muitas vezes um baixo desempenho nos resultados obtidos. Dessa forma, optamos pelo uso de *stop words* e TF-IDF para alguns modelos que exigiam mais tempo de processamento e até para resultados de comparação.

Figura 4: Relatório de resultados - *Decision Tree Classifier*

```
Decision Tree Classifier - SMOTE:
              precision    recall  f1-score   support

           0       0.79      0.82      0.80     43032
           1       0.81      0.78      0.79     43032

    accuracy                           0.80     86064
   macro avg       0.80      0.80      0.80     86064
weighted avg       0.80      0.80      0.80     86064


Decision Tree Classifier - TF-IDF:
              precision    recall  f1-score   support

           0       0.89      0.99      0.93     43032
           1       0.90      0.50      0.65     10779

    accuracy                           0.89     53811
   macro avg       0.89      0.74      0.79     53811
weighted avg       0.89      0.89      0.88     53811
```

Fonte: próprio autor.

Os hiperparâmetros do *Pycaret* não surtiram mudanças drásticas nos resultados de maior pontuação em comparação com os resultados do modelo padrão (sem modificações) do modelo *Decision Tree Classifier*, conforme a figura a seguir.

Figura 5: Árvore de Decisão - *Pycaret*

```
[ ]  1 dt_classifier_tfidf = DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
     2                       max_depth=None, max_features=None, max_leaf_nodes=None,
     3                       min_impurity_decrease=0.0, min_samples_leaf=1,
     4                       min_samples_split=2, min_weight_fraction_leaf=0.0,
     5                       random_state=42, splitter='best')
     6
     7 dt_classifier_tfidf.fit(X_train_tfidf, y_train_tfidf)
     8 y_pred_dtPyCaret_tfidf = dt_classifier_tfidf.predict(X_test_tfidf)
     9
    10 print("\nDecision Tree Classifier - TF-IDF:")
    11 print(classification_report(y_test_tfidf, y_pred_dtPyCaret_tfidf))
    12
```

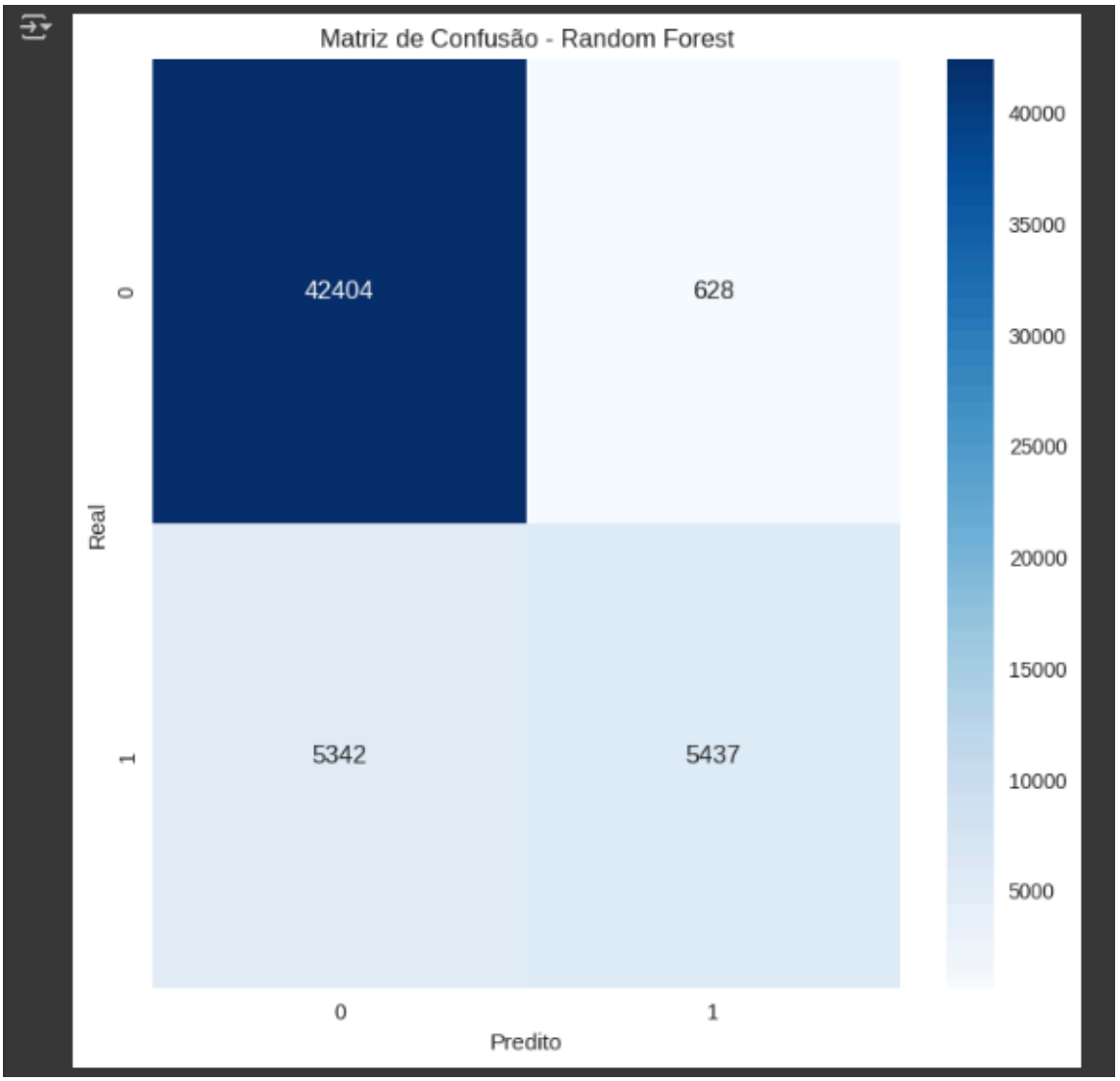
```
Decision Tree Classifier - TF-IDF:
              precision    recall  f1-score   support

           0       0.89      0.99      0.93     43032
           1       0.90      0.50      0.65     10779

    accuracy                           0.89     53811
   macro avg       0.89      0.74      0.79     53811
weighted avg       0.89      0.89      0.88     53811
```

Fonte: próprio autor.

Também obtivemos resultados semelhantes para os outros modelos utilizando a métrica de Precisão. Por termos vários modelos de classificação

que apresentam um alto valor de Precisão, isso geralmente significa que esses modelos têm um baixo número de falsos positivos. Em outras palavras, os modelos são bons em prever corretamente os casos positivos e são menos propensos a classificar incorretamente os exemplos negativos como positivos. Para validarmos essa afirmativa, foi também verificada a matriz de confusão de um dos modelos, abaixo na Figura 6. Essas métricas indicam que o modelo está se saindo bem, especialmente no que diz respeito à sensibilidade (*recall*), mostrando que ele consegue identificar a maioria dos verdadeiros positivos.

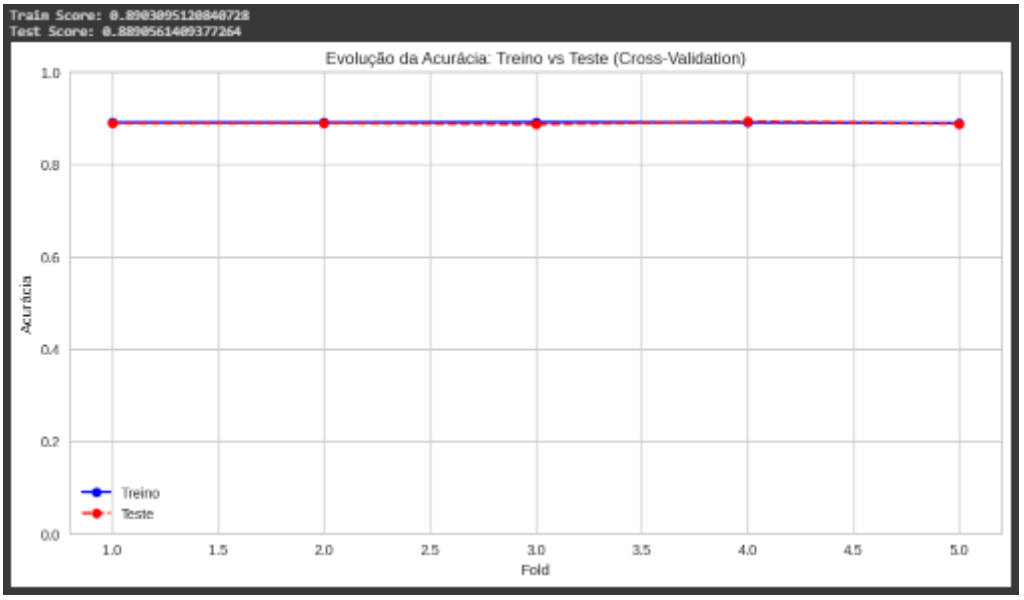
Figura 6: "Matriz de confusão - *Random Forest*"



Fonte: próprio autor.

Ao final, foi verificado a qualidade do treino *vs.* teste do melhor modelo em um gráfico, conforme Figura 7.

Figura 7: “Treino e teste de validação”.



Fonte: próprio autor.

5. CONCLUSÃO

Esse estudo explorou a aplicação do Processamento de Linguagem Natural (NLP) e do Aprendizado em Máquina (*Machine Learning*) na análise de crimes em Chicago. Utilizando técnicas como SMOTE para balanceamento de dados e TF-IDF para a representação textual, foi possível identificar padrões relevantes nas descrições dos crimes. Embora a utilização de SMOTE não tenha mostrado grande eficácia no balanceamento das classes e a aplicação de hiperparâmetros no *PyCaret* não tenha causado mudanças drásticas nos resultados, a análise ainda foi capaz de obter valores de Precisão relativamente altos, indicando que os modelos conseguiram prever com boa acurácia os resultados relacionados aos crimes.

Através da comparação de diversos modelos de classificação, como *Random Forest, Decision Tree e* SVM, foi possível observar que, apesar das limitações, a metodologia aplicada demonstrou o potencial do NLP na análise de dados criminais. Esse trabalho destaca, assim, a importância de aprimorar os modelos e o pré-processamento de dados para obter resultados mais precisos, contribuindo para o uso da inteligência artificial na segurança pública e na previsão de comportamentos criminosos.

A partir dos resultados obtidos, futuras investigações podem explorar melhorias no balanceamento de classes, como o uso de outras técnicas de reamostragem. Além disso, ajustes nos hiperparâmetros podem ser realizados de forma mais aprofundada, utilizando abordagens de otimização automática para melhorar a performance dos modelos.

REFERÊNCIAS

NADEAU, David; SEKINE, Satoshi. **A survey of named entity recognition and classification**. Nova Iorque: National Research Council Canada, 2007. 20 p.

RISH, Irina. **An Empirical Study of the Naive Bayes Classifier**. Montreal: Ijcai, 2001. 46 p.

DRAGONI, Mauro; RIZZI, Williams; VILLATA, Serena; GOVERNATORI, Guido. **Combining NLP Approaches for Rule Extraction from Legal Documents.** Hal: Hal, 2016.

ZHONG, Haoxi; XIAO, Chaojun; TU, Cunchao; ZHANG, Tianyang; LIU, Zhiyuan; SUN, Maosong. **How Does NLP Benefit Legal System: A Summary of Legal Artificial Intelligence.** Pequim: Department Of Computer Science And Technology Institute For Artificial Intelligence, 2020.

QUINLAN, John Ross. **Induction of decision trees**. Sydney: Kluwer Academic Publisher, 1986.

FABRO, Marcos Didonet del. **Iudicium Textum Dataset: uma base de textos jurídicos para NLP**. Curitiba: Universidade Federal do Paraná, 2019. 11 p.

CAMBRIA, Erik; WHITE, Bebo. **Jumping NLP Curves: a review of natural language processing research**. Stanford: Ieee Computational Intelligence Magazine, 2014.

SCHILLER, Andrew. **NeighborhoodScout**. Disponível em: https://www.neighborhoodscout.com/. Acesso em: 14 jan. 2025.

BISHOP, Christopher Michael. **Pattern Recognition and Machine Learning**. Birmingham: Springer, 2005.

SCHEPERS, Iris; MEDVEDEVA, Masha; BRUIJN, Michelle; WIELING, Martijn; VOLS, Michel. **Predicting citations in Dutch case law with natural language processing**. Groningen: Springer, 2023. 32 p.

BREIMAN, Leo. **Random Forests**. Berkeley: Kluwer Academic, 2001. 28 p.

REDDY, Poreddy Sasi Vardhan. **Review and Approaches to Develop Legal Assistance for Lawyers and Legal Professionals using Artificial Intelligence and Machine Learning**. Bangalore: T John Institute Of Technology, 2022.

JURAFSKY, Dan; MARTIN, James H.. **Speech and Language Processing: an introduction to natural language processing, computational linguistics, and speech recognition with language models**. 3. ed. Stanford: Pearson, 2025.

CORTES, Corinna; VAPNIK, Vladimir. **Support-vector networks**. Nova Jersey: Kluwer Academic Publishers, 1995. 25 p.

GREFENSTETTE, Gregory. Tokenization. In: VAN HALTEREN, Hans. **Syntactic Wordclass Tagging**. Nijmegen: Springer, 1999. Cap. 9. p. 117-118.

RAMOS, Juan. **Using Tf-idf to Determine Word Relevance in Document Queries**. Piscataway: Department Of Computer Science, Rutgers University, 2003.