DATA SCIENCE NA LUTA CONTRA FAKE NEWS: UM ESTUDO DE CASO

Diego Santos
Gabriel Ribas
Matheus Bialuz
Matheus Freire Pessoa

RESUMO
A crescente disseminação de notícias falsas nas plataformas digitais tem despertado preocupações significativas, exigindo soluções para sua detecção. Este estudo propõe o desenvolvimento de um algoritmo para identificar notícias falsas, utilizando técnicas de aprendizado de máquina. A pesquisa utiliza o "WELFake_Dataset.csv" do Kaggle como base de dados. Inicialmente, realizou-se a limpeza e pré-processamento dos dados com a biblioteca Pandas, seguido da aplicação da técnica de vetorização TF-IDF e da implementação do algoritmo Naive Bayes Multinomial. Para avaliar a eficácia do modelo, utiliza-se métricas como a curva ROC, matriz de confusão e nuvem de palavras. Os resultados indicam que o algoritmo possui uma capacidade considerável de identificar notícias falsas com precisão. Dessa forma, a pesquisa contribui com as ferramentas que possam mitigar os impactos negativos das fake news.

Palavras-chave: 1. Detecção de fake News; 2. Aprendizado de máquina; 3. TF-IDF; 4. Naive Bayes; 5. Processamento de linguagem natural.

DATA SCIENCE IN THE COMBAT AGAINST FAKE NEWS: A CASE STUDY

ABSTRACT
The increasing spread of false news on digital platforms has raised significant concerns, necessitating effective solutions for its detection. This study proposes the development of a specific algorithm to identify false news using machine learning techniques. The research utilized the "WELFake_Dataset.csv" from Kaggle as the database. Initially, data cleaning and preprocessing were performed using the Pandas library, followed by the application of the TFIDF vectorization technique and the implementation of the Multinomial Naive Bayes algorithm. To evaluate the model's effectiveness, metrics such as the ROC curve, confusion matrix, and word cloud were used. The results indicate that the algorithm has a considerable capacity to accurately identify false news. Thus, this research contributes to the development of tools that can mitigate the negative impacts of fake news.

Key words: 1. Fake news detection; 2. Machine learning; 3. TF-IDF; 4. Naive Bayes; 5. Natural language processing.

1. INTRODUÇÃO

A disseminação de notícias falsas, ou fake news, torna-se um problema significativo na era digital, impactando a sociedade de diversas maneiras. Identificar e mitigar a disseminação de informações falsas é importante para manter a integridade da informação pública e a segurança social. Fake news pode influenciar eleições, criar pânico desnecessário e até mesmo afetar a saúde pública. Com a crescente quantidade de informações disponíveis online, tornar se cada vez mais difícil para os indivíduos discernirem entre notícias verdadeiras e falsas.

A identificação automática de fake news utilizando técnicas de aprendizado de máquina e Data Science surge como uma solução para este desafio. Data Science oferece ferramentas avançadas para a análise de grandes volumes de dados, permitindo a extração de padrões e insights não detectáveis de outra forma. Algoritmos de aprendizado de máquina são capazes de aprender com os dados e fazer previsões ou classificações baseadas em características previamente identificadas.

Neste estudo, foca-se na criação de um algoritmo que classifica notícias como falsas ou verdadeiras com alta precisão. Utiliza-se técnicas de processamento de linguagem natural (PLN) para analisar o conteúdo textual das notícias. O conjunto de dados utilizado é extraído do Kaggle, contendo notícias rotuladas como verdadeiras ou falsas. A preparação dos dados envolve várias etapas, incluindo a limpeza de dados, remoção de stop words e lematização, para garantir que o texto seja adequado para análise. Em seguida, aplica-se a técnica de vetorização TF-IDF (Term Frequency-Inverse Document Frequency) para transformar o texto em um formato numérico que pode ser processado pelos algoritmos de aprendizado de máquina.

Para a classificação das notícias, escolhe-se o algoritmo Naive Bayes Multinomial, baseado no Teorema de Bayes e adequado para a análise de texto devido à sua simplicidade. Utiliza-se várias métricas de desempenho, incluindo a curva ROC, a matriz de confusão e a geração de nuvens de palavras para visualizar os termos mais frequentes. Os resultados demonstram que o algoritmo desenvolvido é capaz de identificar fake news.

2. DESENVOLVIMENTO

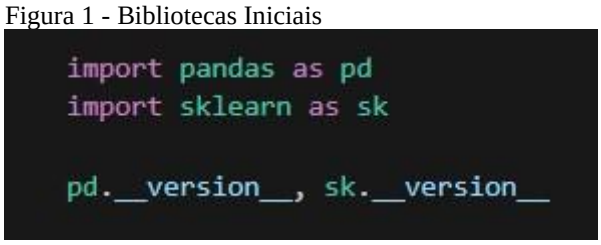
2.1 COLETA E PREPARAÇÃO DOS DADOS

Cabe destacar que, para a realização deste estudo, utiliza-se o arquivo

"WELFake_Dataset.csv" disponível no Kaggle, que é um conjunto de dados contendo uma variedade de notícias rotuladas como verdadeiras ou falsas. A seguir, detalhamos o processo de tratamento e preparação dos dados:

- **Carregamento dos Dados:** Utilizamos a biblioteca Pandas para carregar e visualizar os dados.

Figura 1 - Bibliotecas Iniciais



Fonte: Captura de tela realizada pela equipe (2024).

- **Limpeza dos Dados:** Removemos colunas e linhas duplicadas, além de eliminar valores nulos.

Figura 2 – Primeiro Tratamento

Fonte: Captura de tela realizada pela equipe (2024).

- **Análise Exploratória:** Analisamos a distribuição dos rótulos no conjunto de dados.

Figura 3 - Levantamento de Fake e Real

Fonte: Captura de tela realizada pela equipe (2024).

2.2 TRATAMENTO DOS DADOS TEXTUAIS

Para a preparação dos textos, utiliza-se técnicas de processamento de linguagem natural:

- **Remoção de Stop Words:** Utilizamos a biblioteca NLTK para remover palavras comuns que não contribuem para a detecção de fake news.

Figura 4 - Stopwords

```
import nltk
nltk.download('stopwords')

import nltk
from nltk.corpus import stopwords

stop_words = set(stopwords.words('english'))
df['news_no_stopwords'] = df['news'].apply(lambda x: ' '.join([word for word in x.split() if word not in stop_words]))

df['news_no_stopwords']
```

Fonte: Captura de tela realizada pela equipe (2024).

- **Lematização:** Aplicamos a lematização para reduzir as palavras às suas formas base.

Figura 5 - Lematização

```
import nltk
nltk.download('wordnet')

from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()
df['news_lemmatized'] = df['news_no_stopwords'].apply(lambda x: ' '.join([lemmatizer.lemmatize(word) for word in x.split()]))
```

Fonte: Captura de tela realizada pela equipe (2024).

2.3 VETORIZAÇÃO E MODELO DE CLASSIFICAÇÃO

Utiliza-se a técnica TF-IDF para transformar os textos em vetores numéricos:

Figura 6 - Vetores numéricos

```
tfidf_vectorizer = TfidfVectorizer()
train_features = tfidf_vectorizer.fit_transform(X_train)
test_features = tfidf_vectorizer.transform(X_test)
```

Fonte: Captura de tela realizada pela equipe (2024).

O algoritmo escolhido para a classificação foi o Naive Bayes Multinomial:

Figura 7 – Naive Bayes

```
from sklearn.naive_bayes import MultinomialNB

clf = MultinomialNB()
clf.fit(train_features, y_train)

train_accuracy = clf.score(train_features, y_train)
print("Acurácia no treino:", train_accuracy)

test_accuracy = clf.score(test_features, y_test)
print("Acurácia no teste:", test_accuracy)
```

Fonte: Captura de tela realizada pela equipe (2024).

2.4 AVALIAÇÃO DO MODELO

Para avaliar a performance do modelo, utiliza-se várias métricas e técnicas de visualização:

- **Curva ROC e AUC:** Avaliamos a capacidade discriminativa do modelo.

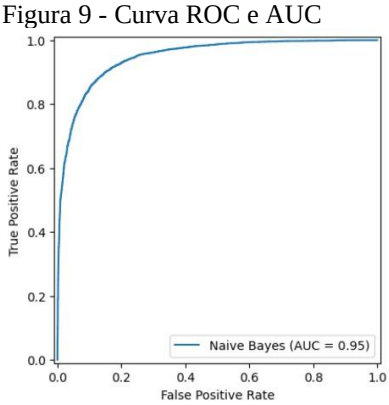
Figura 8 - Teste com curva ROC

```
from sklearn.metrics import roc_curve, auc, RocCurveDisplay

y_probs = clf.predict_proba(test_features)[:, 1]
fpr, tpr, thresholds = roc_curve(y_test, y_probs)
roc_auc = auc(fpr, tpr)

RocCurveDisplay(fpr=fpr, tpr=tpr, roc_auc=roc_auc, estimator_name='Naive Bayes').plot()
plt.show()
```

Fonte: Captura de tela realizada pela equipe (2024).

A figura 9 abaixo fornece a Curva ROC (Receiver Operating Characteristic) do modelo Naive Bayes com o conjunto de dados de notícias falsas. A Curva ROC é um gráfico de um classificador binário, onde a relação entre a taxa de verdadeiros positivos (True Positive Rate) e a taxa de falsos positivos (False Positive Rate) é representada em várias situações de corte.

Figura 9 - Curva ROC e AUC



Fonte: Captura de tela realizada pela equipe (2024).

Explicação da figura 9:

Eixo X (FPR - False Positive Rate): É a taxa de classificações erradas entre os exemplos negativos (notícias genuínas classificadas incorretamente).

Eixo Y (TPR - True Positive Rate): É a taxa de classificações genuínas entre os exemplos positivos (notícias falsas).

AUC (Área Sob a Curva): O valor de AUC (Área Under the Curve) é 0,95, mostrando que o modelo funciona na separação de notícias falsas de verdadeiras. O valor de AUC próximo de 1 indica eficácia.

A curva ROC ilustrada evidencia que o modelo Naive Bayes adotado tem uma capacidade de discriminar entre notícias verdadeiras e falsas, com uma taxa de verdadeiros positivos e uma baixa taxa de falsos positivos, com o resultado em identificação de notícias falsas.

• **Matriz de Confusão:** Analisamos os erros de classificação.

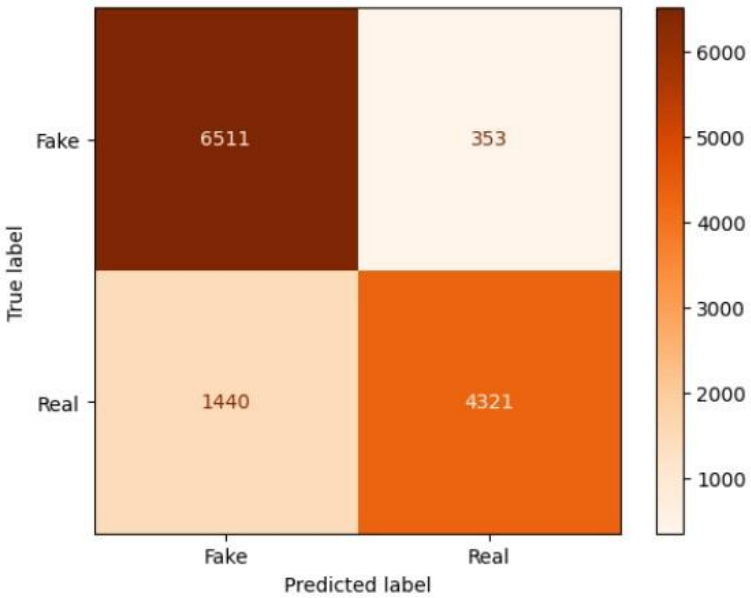
Figura 10 - Criação da Matriz

```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=['Fake', 'Real'])
disp.plot(cmap=plt.cm.Oranges)
plt.show()
```

Fonte: Captura de tela realizada pela equipe (2024).

A Figura 11 abaixo é a matriz de confusão resultante ao executar o modelo Naive Bayes treinado pelos dados de notícias falsas. Uma matriz de confusão é uma das informações que normalmente se usa para traçar o desempenho visual de um algoritmo de classificação, de modo que se possa examinar minuciosamente as previsões corretas e erradas de um modelo.

Figura 11 - Matriz de Confusão



Fonte: Captura de tela realizada pela equipe (2024).

Explicação da figura 11:

- **Eixo Y (Rótulo Real)**: Representa os rótulos reais das amostras no conjunto de dados (Fake e Real);
- **Eixo X (Rótulo Previsto)**: Representa os rótulos previstos pelo modelo (Fake e Real); Quadrantes:
- **Verdadeiros Positivos (6511)**: Número de notícias falsas preditas corretamente;
- **Falsos Negativos (353)**: Número de notícias falsas que são preditas como verdadeiras;
- **Falsos Positivos (1440)**: Número de notícias verdadeiras que são preditas como falsas;
- **Verdadeiros Negativos (4321)**: Número de notícias verdadeiras preditas corretamente.

A matriz de confusão revela que o classificador Naive Bayes é preciso, especialmente na tarefa de identificar notícias falsas, com 6511 previsões corretas. No entanto, também revela um número de falsos positivos (1440), indicando que o classificador erra ao rotular algumas notícias verdadeiras como falsas. Portanto, um olhar mais atento para os erros será bastante instrutivo ao tentar ajustar o classificador.

- **Wordcloud:** Visualizamos as palavras mais frequentes nas notícias.

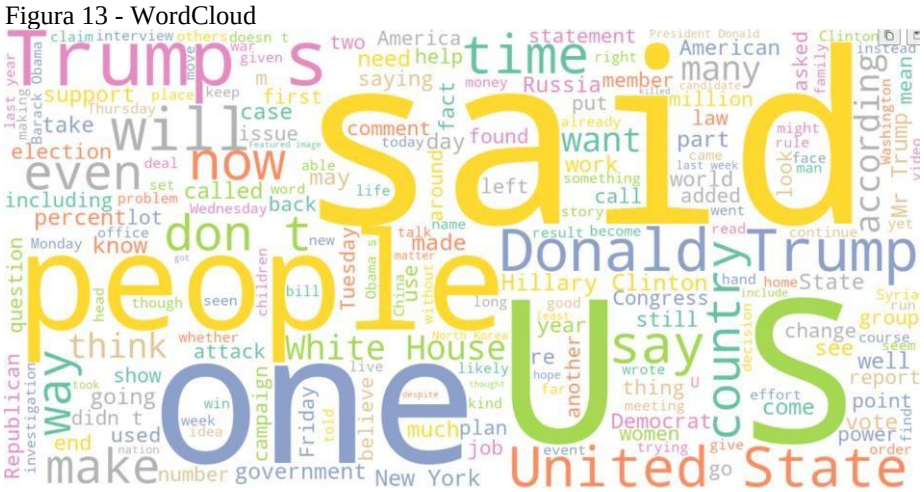
Figura 12 - Criação da WordCloud

```
from wordcloud import WordCloud, STOPWORDS
import pandas as pd
import matplotlib.pyplot as plt

stopwords = set(STOPWORDS)

data_file = pd.read_csv('WELFake_Dataset.csv')
data_file['text'] = data_file['text'].astype(str).fillna('')
text = ' '.join(data_file['text'])

wordcloud = WordCloud(stopwords=stopwords, width=1600, height=800, background_color="white", colormap="Set2").generate(text)

plt.figure(figsize=(20, 10), facecolor='k')
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.tight_layout(pad=0)
plt.show()
```

Fonte: Captura de tela realizada pela equipe (2024).

A figura 13 abaixo é uma nuvem de palavras feita usando o conjunto de dados anotado de notícias utilizado no projeto. A nuvem de palavras é um tipo de diagrama que mostra de maneira visual as palavras mais dominantes usadas em uma determinada extensão de texto para fins de análise. Cada palavra na nuvem é de tamanho relativo dependente do número de vezes que é repetida no conjunto de dados.

Figura 13 - WordCloud



Fonte: Captura de tela realizada pela equipe (2024).

Explicação da figura 13:

- **Palavras mais destacadas**: As palavras mais destacadas na nuvem, como "pessoa", "disse", "Trump", "um", "vai", "EUA" e "país", são as que têm maior frequência de repetição nas notícias coletadas. Isso revela que essas palavras são as mais prevalentes no conteúdo das notícias, independentemente do

assunto;

- **Análise de Contexto**: Palavras como "Trump", "people", e "US" indicam tópicos relacionados à política e a notícias nacionais, que são fortemente o epicentro da desinformação. O uso das palavras "said" e "will" indica citações e previsões, o que é outra marca das fake news.

A nuvem de palavras é uma representação visual direta das palavras mais comuns usadas em um contexto de notícias e, portanto, pode ser usada para inferir o tópico do qual mais se fala.

3. CONSIDERAÇÕES FINAIS

Neste projeto, ilustramos como utilizar técnicas de ciência de dados e aprendizado de máquina para detectar notícias falsas. Utilizamos um grande volume de dados, técnicas de processamento de linguagem natural e vetorização TF-IDF, junto com o classificador Multinomial Naive Bayes, para desenvolver um modelo capaz de identificar notícias com alta precisão. As análises ROC, matriz de confusão e nuvem de palavras mostram que o modelo é capaz de diferenciar entre notícias falsas e reais.

A aplicação prática deste projeto é significativa. A ferramenta criada pode ser usada em plataformas de notícias e mídias sociais para limitar a disseminação de notícias falsas. Este método é flexível e escalável, permitindo a adição contínua de novos dados e técnicas, tornando-se uma base sólida para pesquisas futuras. O uso de técnicas de aprendizado de máquina e ciência de dados se mostra uma abordagem na erradicação da desinformação, reforçando a importância de soluções tecnológicas para problemas sociais críticos.

REFERÊNCIAS

Kaggle. WELFake Dataset. Disponível em: https://www.kaggle.com/datasets/saurabhshahane/fake-news-classification Acesso em: 06 de maio de 2024.

UNIVERSO DISCRETO, canal. Detecção de Fake News usando Bag-of-Words - Machine Learning 25. YouTube, há 1 ano. 1h 06min 05s. Disponível em: https://www.youtube.com/watch?v=W7aORTUCAqQ&t=1601s Acesso em: 13 de maio de 2024.

CODING IS FUN, canal. How To Create A Word Cloud In Python | Tutorial [EASY]. YouTube, há 3 anos. 04 min 33s. Disponível em: https://www.youtube.com/watch?v=HcKUU5nNmrs Acesso em: 14 de maio de 2024.

DATA PROFESSOR, canal. How to Plot an ROC Curve in Python | Machine Learning in Python. YouTube, há 4 anos. 07 min 38s. Disponível em: https://www.youtube.com/watch?v=uVJXPPrWRJ0&t=332s Acesso em: 20 de maio de 2024.

HASHTAG PROGRAMAÇÃO, canal. Como Trabalhar com Arquivos CSV no Python. YouTube, há 2 anos. 22 min 12s. Disponível em: https://www.youtube.com/watch?v=AnJPtKLtc7o&pp=ygUfcHJvamV0byBjb20gYXJxdWl2byBjc3YuIHBhbmRhcw%3D%3D
Acesso em: 25 de maio de 2024.

DATA VIKING, canal. Curso Machine Learning - Floresta Aleatória - Python [#08 Avaliando a Matriz de Confusão]. YouTube, há 3 anos. 09 min 08s. Disponível em: https://www.youtube.com/watch?v=4CoDN2PwoGM&t=176s&pp=ygUSbWF0cml4IGRlIGNvbmZ1c2Fv Acesso em: 27 de maio de 2024.

[Recebido/Received: Dezembro 18 2024; Aceito/Accepted: Janeiro 29, 2025]