DESAFIO DAS FAKE NEWS

Participantes:

Daniel Nobre de Queiroz Pereira

José Ben Hur Nascimento

Júlio César da Silva Filho

Uma-análise por meio de aprendizado de máquina para detectar notícias reais e falsas

Abstract. *This paper presents a machine learning model to distinguish between real and fake news. We use a database composed of real and fake news to train the artificial intelligence and obtain a model capable of evaluating the veracity of external news. The use of machine learning techniques to identify fake news has been widely studied and can contribute to the promotion of the dissemination of accurate and reliable information.*

Resumo. *Este trabalho apresenta um modelo de aprendizado de máquina para distinguir entre notícias verdadeiras e falsas. Utilizamos uma base de dados composta por notícias reais e falsas para treinar a inteligência-artificial e obter um modelo capaz de avaliar a veracidade de notícias externas. O uso de técnicas de aprendizado de máquina para identificar notícias falsas têm sido amplamente-estudado e pode contribuir para a promoção da disseminação de informações precisas e confiáveis.*

Contextualização

A disseminação de fake news, ou notícias falsas, tem se tornado uma preocupação crescente na era da informação digital. Com o rápido avanço da tecnologia e o aumento da conectividade, é cada vez mais fácil para indivíduos mal-intencionados espalhar informações falsas com objetivo de influenciar a opinião pública, causar pânico ou obter ganhos pessoais. A propagação dessas notícias falsas pode ter consequências graves, minando a confiança na mídia, distorcendo os fatos e prejudicando a tomada de decisões informadas.

Diante desse cenário, a detecção de fake news torna-se uma tarefa desafiadora, exigindo abordagens sofisticadas e eficazes. Nesse contexto, nosso objetivo foi desenvolver um modelo de detecção de fake news utilizando técnicas de aprendizado de máquina. Através da aplicação de algoritmos de classificação, busca-se identificar padrões e características nas notícias que possam indicar a presença de informações falsas.

Dataset

O dataset utilizado neste projeto consiste em dois conjuntos de dados: "Fake.csv" e "True.csv". Cada arquivo possui quatro colunas: "title" (título da notícia), "text" (corpo do texto da notícia), "subject" (assunto da notícia) e "date" (data da notícia). O dataset "Fake.csv" contém exemplos de notícias falsas, enquanto o dataset "True.csv" contém exemplos de notícias verdadeiras.

No código, os datasets "Fake.csv" e "True.csv" são lidos e armazenados em dois dataframes separados: "data_fake" e "data_true". Em seguida, é adicionada uma nova coluna chamada "class" em ambos os dataframes, atribuindo o valor 0 para as notícias falsas e o valor 1 para as notícias verdadeiras. Essa nova coluna "class" é utilizada como a variável alvo do modelo.

Para combinar os dois dataframes em um único conjunto de dados, utilizamos a função "pd.concat()" para concatenar os dataframes "data_fake" e "data _true" ao longo do

eixo 0, resultando no dataframe "data_merge". Esse dataframe consolidado é utilizado para treinar e avaliar o modelo de detecção de fake news.

Metodologia

O desenvolvimento do modelo de detecção de fake news seguiu a seguinte metodologia:

Pré-processamento dos Dados

Antes de construir os modelos, realizamos o pré-processamento dos dados. Foram removidas as colunas "title", "subject" e "date" do dataframe "data_merge", mantendo apenas a coluna "text" que contém o corpo do texto da notícia. Essa etapa de pré-processamento visa padronizar e limpar o texto, removendo informações irrelevantes e facilitando a análise subsequente, e por isso, a função wordopt é aplicada ao texto de cada notícia, realizando as seguintes transformações:

a) Conversão para letras minúsculas: todas as letras do texto são convertidas para minúsculas, o que ajuda a padronizar o texto e evitar duplicações de palavras devido a diferenças de capitalização.

b) Remoção de caracteres especiais: caracteres especiais, como colchetes, são removidos do texto. Esses caracteres muitas vezes não possuem relevância para a detecção de fake news e podem interferir na análise posterior.

c) Eliminação de URLs: links para páginas da web são removidos do texto. Esses URLs geralmente não contêm informações relevantes para a classificação das notícias como verdadeiras ou falsas.

d) Remoção de pontuações: sinais de pontuação, como vírgulas e pontos, são removidos do texto. Esses sinais não contribuem diretamente para a detecção de fake news e podem ser considerados ruídos desnecessários.

e) Eliminação de quebras de linha: quebras de linha são removidas do texto para garantir uma representação contínua e coesa das informações.

f) Exclusão de dígitos e números: números e dígitos presentes no texto são eliminados, pois não são relevantes para a detecção de fake news.

Preparação dos Dados

Após o pré-processamento, dividimos os dados em duas variáveis: "x" (características do texto) e "y" (classes - 0 para notícias falsas e 1 para notícias verdadeiras). Em seguida, realizamos a vetorização do texto usando a classe "TfidfVectorizer", que atribui pesos às palavras com base na frequência do termo e na inversa da frequência do documento. Isso transforma o texto em uma representação numérica adequada para os algoritmos de aprendizado de máquina.

Construção dos Modelos

Para construir o modelo de detecção de fake news, foram utilizados os algoritmos de Regressão Logística, Árvore de Decisão, Floresta Aleatória e um ensemble dos dois primeiros. Cada um desses algoritmos têm suas próprias características e abordagens para lidar com a tarefa de classificação.

A Regressão Logística é um modelo linear amplamente utilizado para problemas de classificação, onde, por exemplo, é necessário determinar se uma notícia é verdadeira ou falsa. A regressão logística estima a probabilidade de pertencer a uma classe específica usando uma função logística (sigmoid), que mapeia a soma ponderada das características (ou variáveis independentes) para um valor entre 0 e 1. Se a probabilidade estimada for maior do que um determinado limite, a notícia é classificada como verdadeira; caso contrário, é classificada como falsa.

A Árvore de Decisão é um modelo não linear que divide o espaço de características em regiões, utilizando uma estrutura de árvore. Cada nó interno da árvore representa um teste

em uma característica específica, enquanto os nós folha representam as classes ou valores de saída. A árvore de decisão é construída através de uma abordagem de aprendizado supervisionado, onde a árvore é treinada usando um conjunto de dados rotulados, como notícias verdadeiras ou falsas. Durante o processo de treinamento, a árvore de decisão aprende a fazer divisões nas características com base nos rótulos dos dados de treinamento, buscando maximizar a pureza das regiões resultantes. Uma vez treinada, a árvore de decisão pode ser usada para classificar novas notícias com base nas características observadas.

A Floresta Aleatória é um conjunto de árvores de decisão que trabalham em conjunto para realizar a classificação. Ao contrário de uma única árvore de decisão, a floresta aleatória utiliza técnicas de amostragem e aleatorização para criar várias árvores de decisão independentes. Durante o treinamento, cada árvore é treinada em um subconjunto aleatório dos dados de treinamento, e cada divisão em cada árvore é feita considerando apenas um subconjunto aleatório das características. Essa abordagem reduz o risco de overfitting e ajuda a capturar diferentes aspectos do conjunto de dados. A classificação final de uma nova notícia é obtida por meio de votação, onde cada árvore na floresta contribui com seu voto. A classe com o maior número de votos é escolhida como a classificação final.

O ensemble (conjunto) construído nesse caso combina as previsões da Regressão Logística e da Floresta Aleatória por meio de votação. Essa abordagem busca aproveitar as diferentes vantagens de cada modelo. A Regressão Logística é eficaz na modelagem da relação linear entre as características e a classe, enquanto a Floresta Aleatória é capaz de capturar relações não lineares e interações mais complexas. Ao combinar as previsões dos dois modelos, o ensemble pode melhorar seu desempenho geral e a robustez.

Resultados

Durante o desenvolvimento do projeto, encontramos algumas dificuldades relacionadas ao desempenho dos modelos. Inicialmente, utilizamos a Árvore de Decisão, mas, após testes, observamos que o modelo estava sofrendo com o overfitting. O que significa que o modelo estava se ajustando muito bem aos dados de treinamento, mas não generaliza bem

para novos dados. Tentamos ajustar a proporção dos dados para treinamento e para teste, iniciamos com 75-25 e realizamos testes com 70-30, 65-35, mas sem mudanças significativas. Na tentativa de solucionar esse problema, decidimos aplicar o cross-validation (validação cruzada) para avaliar o desempenho dos modelos de forma mais robusta e evitar o overfitting.

```
# Cross-validation
LR = LogisticRegression()
RF = RandomForestClassifier(random_state=0)
DT = DecisionTreeClassifier()

models = [('Logistic Regression', LR), ('Decision Tree', DT), ('Random Forest', RF)]

for model_name, model in models:
    scores = cross_val_score(model, xv, y, cv=5)
    print(f"{model_name} Cross-Validation Accuracy: {np.mean(scores)}")
```

```
Logistic Regression Cross-Validation Accuracy: 0.9782618785580113
Decision Tree Cross-Validation Accuracy: 0.992226821752612
Random Forest Cross-Validation Accuracy: 0.9826494882516021
```

Figura 1 - Teste com validação cruzada

Após a aplicação do cross-validation, observamos que a precisão da Regressão Logística e da Floresta Aleatória diminuiu aproximadamente 1%. Além disso, notamos que a Regressão Logística e a Floresta Aleatória passaram a concordar mais entre si em termos de desempenho quando testados com nossos inputs. No entanto, a Árvore de Decisão continuou consistentemente apresentando resultados diferentes do esperado, de forma que, já não estava contribuindo de forma eficaz para a detecção de fake news.

```
Logistic Regression Evaluation:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      5876
           1       0.98      0.99      0.99      5349

    accuracy                           0.99     11225
   macro avg       0.99      0.99      0.99     11225
weighted avg       0.99      0.99      0.99     11225


Decision Tree Evaluation:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      5876
           1       1.00      0.99      1.00      5349

    accuracy                           1.00     11225
   macro avg       1.00      1.00      1.00     11225
weighted avg       1.00      1.00      1.00     11225


Random Forest Evaluation:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      5876
           1       0.99      0.98      0.99      5349

    accuracy                           0.99     11225
   macro avg       0.99      0.99      0.99     11225
weighted avg       0.99      0.99      0.99     11225
```

Figura 2 - Resultados individuais dos modelos

Tentamos utilizar um ensemble da Regressão Logística com a Floresta Aleatória, juntamente com o cross-validation, mas não houveram mudanças significativas nos resultados dos testes quando comparado aos modelos utilizados de forma isolada, logo, a Regressão Logística, a Floresta Aleatória e o ensemble apresentaram desempenhos semelhantes na detecção de fake news e a medidas de desempenho dos modelos não apresentaram mudanças substanciais ao utilizar o ensemble, reforçando que a combinação dos dois algoritmos não trouxe benefícios significativos.

```
[ ]  # Cross-validation
     scores = cross_val_score(ensemble_model, xv, y, cv=5)
     print(f"Ensemble Cross-Validation Accuracy: {np.mean(scores)}")

     Ensemble Cross-Validation Accuracy: 0.9808903551493945
```

Figura 3 - Ensemble com validação cruzada

Conclusão

Após utilizarmos a Regressão Logística, Árvore de Decisão e a Floresta Aleatória, para construir modelos capazes de identificar notícias falsas, chegamos a conclusão de que a aplicação do pré-processamento dos dados e da vetorização do texto contribuiu para melhorar a representação das informações, maximizando o desempenho dos modelo.

Contudo, ao realizarmos testes com valores externos ao dataset, notamos que a performance não era a esperada, chegando a conclusão de que houve um overfit.

Apesar dos desafios enfrentados, o uso do cross-validation ajudou a amenizar minimamente o overfitting e a aumentar a capacidade de generalização dos modelos, fazendo com que houvesse resultados mais condizentes com a precisão apresentada.

Os resultados mostraram que a Regressão Logística e Floresta Aleatória tiveram desempenhos semelhantes na detecção de fake news, mas não apresentaram melhorias significativas ao utilizar o ensemble, e que a Árvore de Decisão, diferentemente dos outros modelos, não apresentou mudanças em relação ao overfitting após o cross-validation. Logo, embora os valores apresentados sejam superficialmente muito bons, quando testados com inputs do usuário, os modelos não se comportam completamente como esperado do seu desempenho apresentado.

Referências

Collaboratory: Aprendizado de Máquina.ipynb