ESCOLA SUPERIOR DE

TECNOLOGIA DA INFORMAÇÃO

INSTITUTO INFNET

LEONARDO DA CONCEIÇÃO MUNIZ

Projeto de Bloco:
Inteligência Artificial e Machine Learning

TP 2

Docente: Tiago Cariolano de Souza Xavier

Rio de Janeiro

16/06/2025

LINK DO NOTEBOOK DO GOOGLE COLAB REFERENTE AO TP 2 DE PB DE IA:
https://colab.research.google.com/drive/1TKprrskUR12WvIbbm6TTuACgxUk8_1Np

Objetivo: Este projeto tem como objetivo proporcionar aos alunos experiência prática em técnicas de Processamento de Linguagem Natural (NLP) e Machine Learning, aplicadas a conjuntos de dados textuais.

Base de dados: As duas bases de dados, **Fake.csv** e **True.csv**, representam conjuntos de notícias classificadas de acordo com sua veracidade.

- **Fake.csv**: Esta base de dados contém notícias que foram **verificadas e confirmadas como falsas**. Ou seja, são exemplos de "fake news".
- **True.csv**: Esta base de dados, por outro lado, é composta por notícias que foram **verificadas e confirmadas como verdadeiras**. Elas servem como exemplos de notícias autênticas e confiáveis.

Descrição das Atividades:

1. **Criação das features**: Computar o Term Frequency-Inverse Document Frequency (TF-IDF) para representar a importância das palavras em um conjunto de documentos.
2. **Modelagem de K-Nearest Neighbors (KNN):** Criar modelos simples de classificação utilizando a base de dados codificada por TF-IDF.
    a. Explore diferentes valores para o parâmetro K do KNN e analise seu impacto nos resultados obtidos (através da acurácia do modelo para os dados de validação).
3. **Avaliação de Modelos**: Aplicar técnicas de validação cruzada para estimar a eficiência dos modelos desenvolvidos.
4. **Avaliação de Classificadores Binários**: Utilizar figuras de mérito como Curva ROC, precisão, recall, f1-score, sensibilidade e especificidade para avaliar os modelos.
5. **Baseado nos valores encontrados para as diferentes figuras de mérito, interprete os resultados e disserte sobre a eficiência do classificador criado.**

A Transformação do Texto em Dados para o Computador

Quando trabalhamos com texto, o computador não o compreende da mesma forma que nós. Ele precisa que as palavras sejam convertidas em números. É nesse ponto que entra a criação de features, que é basicamente o processo de transformar o texto em algo que o computador possa contar e analisar. Uma das técnicas mais comuns e importantes para fazer isso é o **TF-IDF**.

O que é o TF-IDF?

O TF-IDF, que significa **Term Frequency-Inverse Document Frequency** (ou Frequência do Termo - Frequência Inversa do Documento), é uma forma de atribuir um peso a cada palavra em um conjunto de documentos, como um grupo de notícias.

- **Frequência do Termo (TF - Term Frequency)**: Imagine que a palavra "gato" aparece 10 vezes em uma notícia específica. Isso indica que ela é importante para aquele documento. O TF mede quantas vezes uma palavra aparece em um documento.
- **Frequência Inversa do Documento (IDF - Inverse Document Frequency)**: Agora, considere a palavra "de". Ela aparece em quase todas as notícias, tornando-a muito comum e, portanto, menos útil para diferenciar uma notícia da outra. O IDF faz o oposto: ele diminui o peso de palavras que são muito comuns em todos os documentos e aumenta o peso de palavras que são mais raras, mas que aparecem com frequência em documentos específicos.

O TF-IDF combina essas duas métricas: ele atribui um peso alto a palavras que aparecem com muita frequência em um documento específico (TF alto), mas que são raras no conjunto geral de documentos (IDF alto).

Qual a Importância do TF-IDF?

A importância do TF-IDF é significativa, pois nos auxilia a:

1. **Identificar palavras-chave**: Ele destaca as palavras que realmente carregam significado em um texto e que ajudam a diferenciá-lo dos outros. Por exemplo, em uma notícia falsa, palavras como "exclusivo", "urgente" ou nomes de teorias da conspiração podem ter um TF-IDF alto. Já em uma notícia verdadeira sobre política, termos como "congresso", "lei" ou "ministro" seriam mais relevantes.
2. **Preparar o texto para a máquina**: Ele converte as palavras em números, criando um **vetor** (uma lista de números) para cada notícia. Esse vetor é a **feature** (característica) que o modelo de Machine Learning (como o KNN, que foi escolhido) utilizará para aprender e classificar.
3. **Filtrar ruído**: Palavras muito comuns e com pouco significado (como "o", "a", "de", "e" – artigos, preposições, conjunções, etc.), que chamamos de **stop words**, geralmente recebem um TF-IDF muito baixo e acabam sendo menos relevantes na análise, o que é excelente para focar no que realmente importa.

Quais Palavras Importantes Configuramos no TF-IDF?

Em nosso código, as palavras importantes que configuramos para o TF-IDF são definidas pelos seguintes parâmetros:

- **max_features=5000**: Isso indica que o TF-IDF considerará apenas as 5000 palavras mais relevantes (aquelas com o maior peso TF-IDF) em todo o nosso conjunto de notícias. É como focar nos 5000 termos que mais contribuem para distinguir as notícias entre si. Essa configuração é crucial para não sobrecarregar o computador e garantir que a análise se concentre nas informações mais relevantes.
- **stop_words='english'**: Com esta configuração, solicitamos ao TF-IDF que ignore as palavras muito comuns da língua inglesa (como "the", "a", "is", "and"). Elas não agregam muito valor para diferenciar uma notícia da outra, portanto, as removemos para reduzir o ruído e focar nas palavras mais significativas.
- **min_df=5**: Essa configuração instrui o TF-IDF a ignorar palavras que aparecem em menos de 5 documentos (notícias). Por que isso é importante? Palavras que aparecem em apenas 1 ou 2 notícias podem ser erros de digitação, termos muito específicos que não generalizam bem, ou simplesmente não são relevantes o suficiente para serem consideradas features importantes em nosso conjunto de dados. Isso ajuda a limpar o vocabulário e a focar em termos mais representativos.

Outras Informações Relevantes

- **Dimensionalidade**: Após o processo de TF-IDF, cada notícia é transformada em um vetor de 5000 números (devido ao max_features=5000). Isso é o que chamamos de dimensionalidade. É como se cada notícia fosse um ponto em um espaço de 5000 dimensões, e o KNN calculará a distância entre esses pontos para encontrar os vizinhos mais próximos.
- **Limitações**: Embora o TF-IDF seja uma ferramenta poderosa, ele não compreende o significado real das palavras ou o contexto. Por exemplo, ele não identificaria que "cachorro" e "cão" têm o mesmo significado, ou que a ordem das palavras em uma frase pode alterar completamente o sentido ("Não sou eu" versus "Eu sou não"). Para isso, existem outras técnicas mais avançadas, mas para começar, o TF-IDF é um excelente ponto de partida.

Em resumo, a criação de features com TF-IDF é uma etapa essencial que transforma o texto bruto em dados numéricos significativos, permitindo que algoritmos de Machine Learning como o KNN possam ler e classificar as notícias de forma eficaz.

A Modelagem com K-Nearest Neighbors (KNN)
Depois de transformar nosso texto em números com o TF-IDF (nossas features), o próximo passo é utilizar um algoritmo de Machine Learning para, de fato, classificar as notícias como verdadeiras ou falsas. Para isso, escolhemos o **K-Nearest Neighbors**, ou simplesmente **KNN**.

O que é o KNN?
Podemos entender o KNN como um classificador que opera por votação da vizinhança. Imagine que cada notícia (agora representada por um conjunto de números graças ao TF-IDF) é um ponto em um grande espaço. Quando uma nova notícia precisa ser classificada como falsa ou verdadeira, o KNN executa os seguintes passos:

1. **Encontra os vizinhos mais próximos**: Ele busca os K pontos (notícias) que já conhecemos (nossos dados de treino) que são mais semelhantes à nova notícia. A similaridade é determinada pela distância entre os pontos nesse espaço numérico (quanto menor a distância, mais parecidos eles são).
2. **Conta os votos**: Entre esses K vizinhos mais próximos, ele verifica qual é a classificação mais comum. Por exemplo, se 7 dos 10 vizinhos mais próximos são notícias falsas, a nova notícia será classificada como falsa.
3. **Classifica a nova notícia**: A nova notícia recebe a classificação da maioria de seus vizinhos mais próximos.

É por isso que ele é chamado de **"K-Nearest Neighbors"** (K Vizinhos Mais Próximos): o "K" representa a quantidade de vizinhos que o algoritmo considerará para tomar a decisão.

Explorando o Parâmetro K e seu Impacto
O valor de K é crucial no KNN, e não existe um número ideal que funcione para todas as situações. Ele é um **hiperparâmetro**, o que significa que precisamos ajustá-lo para otimizar o desempenho do modelo para o problema específico que estamos abordando.

- **Impacto do K Pequeno (ex: K=1 ou K=3):** Se você escolher um K pequeno, o modelo baseia sua decisão em pouquíssimos vizinhos. Isso pode torná-lo muito sensível a ruídos ou dados atípicos. Pense que ele pode ser excessivamente influenciado por um único vizinho que talvez não seja tão representativo. A acurácia pode ser boa nos dados de treino, mas talvez não generalize bem (**"overfitting"**).
- **Impacto do K Grande (ex: K=900, como vimos):** Se você escolher um K maior, o modelo considerará a opinião de mais vizinhos. Isso o torna mais suave e menos sensível a pontos isolados. Ele busca um consenso mais amplo na vizinhança. Geralmente, isso ajuda o modelo a generalizar melhor para dados novos, como observado nos nossos resultados, onde valores de K maiores (como 900) resultaram em melhor acurácia. No entanto, um K grande demais pode começar a misturar classes ou ignorar padrões importantes, levando a um desempenho inferior (**"underfitting"**) e também aumentando o tempo de cálculo.

Para descobrir o melhor K, realizamos um processo chamado **validação cruzada**. É como testar o modelo várias vezes com diferentes partes dos nossos dados de treino.

- **Dividimos os dados de treino:** Pegamos nosso conjunto de dados de treino e o dividimos em algumas partes (por exemplo, 5 partes, ou "folds").

- **Testamos e treinamos repetidamente:** Em cada rodada, utilizamos 4 dessas partes para treinar o modelo e a 1 parte restante para validar (testar) o modelo. Repetimos isso 5 vezes, garantindo que cada parte seja usada uma vez como validação.
- **Calculamos a acurácia média:** Ao final, somamos a acurácia de cada uma dessas 5 rodadas e calculamos uma média. Essa acurácia média da validação cruzada nos oferece uma estimativa muito mais confiável de quão bem nosso modelo, com aquele K específico, se comportará com dados que ele nunca viu.

Ao explorar diferentes valores de K e analisar a acurácia média da validação cruzada (e o desvio padrão), conseguimos identificar qual K oferece o melhor equilíbrio, evitando que o modelo seja bom apenas nos dados que ele "decorou". Como vimos nos resultados, a acurácia da validação cruzada nos direcionou para valores de K mais altos, que se mostraram mais eficazes.

Analisando Nossos Resultados do KNN

Acabamos de realizar alguns testes importantes para avaliar o desempenho do nosso modelo de detecção de notícias falsas, e os resultados são extremamente promissores! Utilizamos um método super confiável para escolher o melhor K para o nosso KNN: a validação cruzada.

Lembre-se que a validação cruzada divide nossos dados de treino em várias partes. O modelo é então treinado e testado nessas partes múltiplas vezes. A média desses resultados nos proporciona uma confiança muito maior. E, para nossa satisfação, a validação cruzada indicou que nosso **melhor K é 900**! Isso significa que, para o tipo de dados e notícias com os quais estamos trabalhando, considerar os 900 vizinhos mais próximos é o que torna o modelo mais estável e eficaz.

Agora, vamos analisar como o modelo, já treinado com este K=900, se comportou com as notícias que ele nunca tinha visto antes (nosso conjunto de teste). Afinal, o que realmente importa mesmo é o desempenho dele no mundo real.

Desempenho do Nosso Modelo Final (K=900) no Teste

- **Acurácia: 90.95%** Um índice de **90.95% de acertos** é excelente! Significa que, de cada 10 notícias fornecidas ao modelo, ele classificou corretamente aproximadamente 9. Isso demonstra uma capacidade notável de distinguir entre notícias verdadeiras e falsas.
- **Precisão: 87.94%** Aqui, avaliamos as notícias que o modelo classificou como verdadeiras. Desses casos, **87.94% eram realmente verdadeiras**. Esse resultado é ótimo, pois indica que o modelo raramente erra ao liberar uma notícia como verdadeira. A probabilidade de uma fake news passar despercebida como se fosse verdadeira é muito baixa, o que é crucial para evitar a disseminação de desinformação.
- **Recall (Sensibilidade): 94.85%** Esta métrica se destacou! Ela mostra que, das notícias que eram verdadeiras, nosso modelo conseguiu identificar **94.85%** delas. Isso é sensacional, pois ele é muito eficiente em não confundir uma notícia verdadeira com uma falsa. Ou seja, minimizamos o risco de acusar uma notícia real de ser falsa.
- **F1-Score: 0.9127** O F1-Score pode ser visto como uma pontuação que equilibra a Precisão e o Recall. Nosso F1-Score de **0.9127** é muito alto, o que indica que o modelo é bem balanceado. Ele não é apenas bom em um aspecto e deficiente em outro; ele consegue ser preciso sem perder a capacidade de identificar o que é necessário. Isso é o ideal quando o objetivo é evitar tanto a liberação de fake news quanto a censura de notícias verdadeiras.
- **Especificidade: 87.08%** E das notícias que eram realmente falsas, o modelo acertou **87.08%** delas. Isso prova que ele também é muito eficaz em identificar as fake news, impedindo que elas permaneçam sem detecção.
- **Área Sob a Curva ROC (AUC-ROC): 0.9754** Este é o ponto alto! Um AUC-ROC de **0.9754** é **quase perfeito**! É como se o modelo tivesse uma habilidade excepcional para separar o que é

relevante do que não é. Quanto mais próximo de 1.0, melhor. Esse valor demonstra que, independentemente de como ajustarmos o filtro, o modelo é incrivelmente bom em distinguir notícias verdadeiras de falsas.

Conclusão Final: Acertamos em Cheio!
Analisando todos esses números, podemos afirmar com segurança: desenvolvemos um classificador KNN extremamente eficaz para identificar notícias falsas! A escolha do **K=900**, que validamos de forma rigorosa, fez toda a diferença.

Nosso modelo:
- É **extremamente preciso** em sua avaliação geral (quase 91% de acurácia).
- Atinge um **alto índice de acertos** em suas previsões.
- Consegue **encontrar a maioria** das **notícias verdadeiras** existentes.
- É **habilidoso em identificar** as **notícias falsas**.
- E, para finalizar com chave de ouro, possui uma capacidade **notável de separar** o que é real do que não é (aquele AUC-ROC próximo da perfeição!).

É um resultado para se orgulhar!