INSTITUTO FEDERAL Piauí	**INSTITUTO FEDERAL DO PIAUÍ** CURSO: **TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS** DISCIPLINA: **TÓPICOS ESPECIAIS EM PROGRAMAÇÃO** PROFESSOR: **HÉLCIO DE ABREU SOARES** Teresina (PI) 10 de dezembro de 2025

ALUNOS:

- ***Marcos Araújo Silva***
- ***José Macêdo dos Santos Junior***
- ***Wadson Tardelle Dias de Lima***
- ***Leonardo Vitorio da Silva***
- ***Emanoel Sousa do Carmo***

Questões

1. Por que o pré-processamento textual é importante para o desempenho do modelo?

Porque textos brutos contêm ruídos (tipo pontuações, stopwords, maiúsculas/minúsculas, URLs, emojis, variações da mesma palavra). Se não forem tratados, esses ruídos viram features inúteis e prejudicam o modelo, gerando maior dimensionalidade desnecessária, maior chance de overfitting e dificuldade para o modelo identificar padrões. O pré-processamento reduz ruído, padroniza o texto e melhora a qualidade das features.

2. Qual a diferença fundamental entre BOW e TF-IDF? Dê um exemplo simples.

A diferença fundamental entre Bag-of-Words (BOW) e TF-IDF está na forma como cada método representa a importância das palavras. O BOW apenas conta quantas vezes cada termo aparece em um texto, sem considerar se essa palavra é comum ou rara no conjunto total de documentos. Já o TF-IDF leva em conta não só a frequência da palavra no texto, mas também o quão comum ela é no corpus inteiro, reduzindo o peso de termos que aparecem em muitos documentos e destacando palavras mais específicas. Por exemplo, nas frases "gostei muito do produto" e "não gostei", a palavra "gostei" aparece em ambas. No BOW, ela teria o mesmo peso nas duas frases, enquanto no TF-IDF seu peso seria reduzido por ser comum, dando mais relevância a termos como "produto", que ajudam mais a diferenciar os textos.

3. Em que situações TF-IDF tende a ser mais vantajoso do que BOW?

O TF-IDF tende a ser mais vantajoso do que o BOW quando há muitas palavras comuns que não contribuem para a classificação, quando queremos diferenciar textos longos de curtos, quando o modelo precisa valorizar termos característicos de cada classe ou quando o dataset é desequilibrado ou muito variado. Nessas situações, penalizar palavras frequentes melhora a capacidade discriminativa do modelo.

4. O que é um vetor esparso e por que isso é comum em PLN?

Em problemas de Processamento de Linguagem Natural, é comum trabalhar com vetores esparsos, ou seja, vetores em que a maioria dos valores é zero. Isso acontece porque o vocabulário total costuma ser muito grande, com milhares de palavras possíveis, enquanto cada documento individual utiliza apenas uma pequena fração delas. Assim, em um vocabulário de 10.000 palavras, um texto curto como um SMS pode usar apenas 10 ou 20, deixando quase todas as outras posições com valor zero.

5. Por que modelos lineares podem funcionar bem em textos com alta dimensionalidade?

Modelos lineares costumam funcionar muito bem nesse cenário de alta dimensionalidade porque cada palavra se torna uma feature relativamente independente, o que favorece a separação linear. Além disso, dados textuais geralmente apresentam boa separabilidade linear, como ocorre em problemas clássicos de spam versus ham, onde certas palavras são fortemente associadas a uma classe. A alta dimensionalidade ajuda na criação de hiperplanos de separação mais eficientes, e modelos lineares como Regressão Logística e SVM linear são rápidos, escaláveis e eficazes nesse tipo de tarefa.

6. Principais diferenças entre um pipeline de classificação tabular e um de textos

Ao comparar pipelines de classificação tabular e textual, a principal diferença está no tratamento dos dados. Em dados tabulares, o pré-processamento envolve normalização, imputação e codificação de variáveis, e as features já são numéricas ou facilmente convertidas. Em textos, é necessário limpar, tokenizar e transformar o conteúdo em vetores numéricos por meio de técnicas como BOW, TF-IDF ou embeddings. A dimensionalidade em textos é muito maior e o espaço de representação costuma ser esparso, enquanto dados tabulares geralmente possuem poucas colunas e um espaço denso.

7. O que significa interpretabilidade/explicabilidade de um modelo?

A interpretabilidade ou explicabilidade de um modelo se refere à capacidade de entender por que ele tomou determinada decisão. Em classificação de textos, isso significa saber quais palavras mais influenciaram uma previsão, como cada termo contribuiu para classificar uma mensagem como spam ou ham e se essa decisão faz sentido do ponto de vista humano.

8. Diferença entre explicação local e explicação global

As explicações podem ser locais ou globais. A explicação local busca entender uma previsão específica, como o motivo de um SMS isolado ter sido classificado como spam. Já a explicação global analisa o comportamento geral do modelo, mostrando quais palavras, em média, são mais importantes ao longo de todo o conjunto de dados.

***9. Por que precisa*mos de LIME e SHAP mesmo usando modelos lineares?**

Mesmo quando usamos modelos lineares, ainda precisamos de métodos como LIME e SHAP porque o pipeline de PLN envolve etapas complexas de pré-processamento e vetorização, e o modelo pode ter milhares de features. Os coeficientes globais não explicam bem decisões individuais, enquanto LIME e SHAP conseguem mostrar contribuições palavra a palavra em uma previsão específica, oferecendo explicações locais mais intuitivas.

10. Como o LIME gera explicações locais? Qual o papel das perturbações da instância?

O LIME gera explicações locais a partir de perturbações da instância original. Ele cria várias versões do texto removendo ou alterando palavras, observa como a previsão do modelo muda e, com base nisso, treina um modelo simples e local que aproxima o comportamento do modelo original naquela vizinhança. As perturbações permitem identificar quais palavras têm maior impacto na decisão.

11. Como o SHAP calcula a importância das palavras?

O SHAP, por sua vez, calcula a importância das palavras usando valores de Shapley, da teoria dos jogos. Cada palavra é vista como um "jogador" que contribui para o resultado final, e o SHAP mede a contribuição média de cada termo considerando diferentes combinações possíveis de palavras. Assim, ele fornece uma medida mais consistente e teoricamente fundamentada da importância das features.

12. Como interpretar um summary plot do SHAP em um problema de texto?

Em um summary plot do SHAP para textos, é possível ver quais palavras mais influenciam o modelo, em qual direção elas empurram a previsão (por exemplo, para spam ou ham) e o quão forte é essa influência. Valores positivos geralmente indicam contribuição para uma classe, enquanto valores negativos indicam contribuição para a classe oposta, e a magnitude mostra a intensidade desse efeito.

13. Em que situações as explicações do LIME e do SHAP concordaram?

As explicações do LIME e do SHAP sempre concordam quando aparecem palavras muito fortes e características, quando os textos são curtos ou quando o modelo é linear, pois o comportamento é mais simples e menos ambíguo. Por outro lado, podem discordar em textos longos ou complexos, já que o LIME depende das perturbações locais e pode dar peso excessivo a palavras neutras, enquanto o SHAP segue uma lógica mais global e consistente com a distribuição do dataset.

14. Em que situações elas discordaram? Qual hipótese para isso?

As discordâncias ocorreram com mais frequência em textos longos ou mais complexos, onde muitas palavras contribuem simultaneamente para a previsão. Nesses casos, o LIME, por ser um método local baseado em perturbações, pode acabar atribuindo importância exagerada a palavras neutras que mudam durante as perturbações. Já o SHAP segue uma lógica mais global e consistente com a distribuição do dataset, o que pode levar a diferenças nos pesos atribuídos. A principal hipótese é que o LIME foca muito na vizinhança específica da instância analisada, enquanto o SHAP distribui a importância de forma mais estável ao longo do modelo.

15. Houve palavra importante que não fazia sentido? O que isso indica?

Sim, em alguns casos surgiram palavras consideradas importantes que não faziam sentido do ponto de vista semântico. Isso geralmente indica a presença de ruído no dataset, falhas no pré-processamento — como tokens estranhos, repetições ou símbolos —, overfitting do modelo a padrões irrelevantes ou desequilíbrio entre as classes. Nessas situações, a palavra não é realmente significativa, mas acabou acionando o modelo por coincidência estatística, e não por carregar significado real.

16. Como vocês construíram uma explicação global a partir de um método local como o LIME?

Embora o LIME seja um método local, foi possível construir uma explicação global agregando várias explicações individuais. Para isso, selecionamos um conjunto representativo de textos do conjunto de teste, abrangendo ambas as classes. Em seguida, geramos explicações locais com o LIME para cada instância e extraímos as palavras mais importantes, seus pesos e a direção da

influência. Esses resultados foram armazenados e agregados, contabilizando a frequência com que cada palavra aparecia nas explicações e o peso médio associado a ela. A partir dessa agregação, foi possível identificar quais termos mais influenciavam o modelo de forma geral, quais caracterizavam cada classe e se o comportamento do modelo se mantinha consistente ao longo do dataset.