Respostas
Trabalho – Classificação Binária de Textos com LIME e SHAP

Grupo:
- Antoniel William
- Filipe Neiva
- Kleberson Vilela
- Lucas Coelho
- Lucas Farias

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

O pré-processamento remove ruídos e padroniza os textos, tornando-os mais adequados para análise. Operações como remoção de *stopwords*, normalização e limpeza de caracteres especiais reduzem a complexidade e evitam que o modelo aprenda padrões irrelevantes. Isso melhora a generalização e aumenta a eficiência do treinamento, resultando em modelos mais precisos e rápidos.

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

Bag of words (BOW): conta a frequência das palavras em um documento, sem considerar sua importância relativa, não opera com contextos, o que deixa sua análise enviesada por frequências.
TF-IDF: além da frequência, pondera a relevância da palavra considerando o corpus. Palavras muito comuns (como “o”, “de”) recebem peso baixo, enquanto palavras mais específicas recebem peso alto. A distribuição do cálculo entre os valores TF e IDF ajudam a chegar numa melhor avaliação para cada palavra.

Exemplo: Documento 1: “gato preto”; Documento 2: “gato branco”. No BOW, “gato” teria peso igual nos dois documentos. No TF-IDF, se “gato” aparece em quase todos os textos, seu peso será menor, enquanto “preto” ou “branco” terão maior relevância por distinguirem os documentos.

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

Quando queremos diferenciar documentos por termos mais específicos e informativos e em tarefas como classificação de textos, recuperação de informação e busca, pois o TF-IDF destaca termos que realmente caracterizam o conteúdo, enquanto o BOW pode dar peso excessivo a palavras comuns e pouco informativas.

4. O que é um vetor esparso e por que isso é comum em problemas de PLN?
Um vetor esparso é um vetor com muitos valores zero. A esparsidade surge porque cada texto usa apenas uma fração minúscula do vocabulário total, onde os zeros são as ausências em relação ao texto comparado com o vocabulário. Pela lógica um vocabulário é pra ser sempre bem maior que um texto, ao analisar o texto e criar o vetor, muitas palavras do vocabulário não estarão no texto analisado em questão, isso gera os zeros e um aumento de esparsidade.

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

Em textos, cada palavra do vocabulário vira uma dimensão do vetor, o que gera representações com milhares de dimensões. Mesmo assim, modelos lineares funcionam bem porque cada texto usa poucas dessas dimensões, e o modelo decide somando apenas as palavras presentes. Essa combinação de muitas dimensões com muitos zeros favorece modelos simples e lineares.

6. Quais são as principais diferenças entre um pipeline de classificação com dados tabulares e um de textos?

Na origem das features, nas questões de dimensionalidade, esparsidade, pré-processamento. Classificação em textos dependem da extração das features, a dimensionalidade dos vetores é bem alta, assim como a esparsidade, e são feitas limpezas, tokenização para ajudar melhor na análise. Nos dados tabulares, as features já vem estruturadas, o que reduz nos outros aspectos, necessitando de menos dimensionalidade, gera vetores com menor esparsidade, e o pré-processamento usa escala, encoding.

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

Interpretabilidade está relacionada ao funcionamento do modelo, que tipos de recursos ele faz uso para alcançar seus objetivos, entender o que o modelo usa. Explicabilidade é mais voltada para explicação de como o modelo chega em suas conclusões, de como os resultados foram operados e obtidos.

8. Qual a diferença entre explicação local e explicação global?

A explicação local busca mostrar quais *features* influenciaram a previsão de uma instância específica, explicando por que o modelo tomou aquela decisão pontual em algum documento. A explicação global resume o comportamento do modelo como um todo, mostrando a importância média das *features* ao longo de todo o dataset e como elas influenciam as predições de forma geral.

9. Por que precisamos de LIME e SHAP mesmo usando modelos lineares?

Modelos lineares oferecem pesos globais que não necessariamente devem ser válidos para todo o corpo do dataset, seus dados não são capazes de determinar a importância da apalvra num texto e no seu nível de decisão, pois falta contexto. Esses contextos são oferecidos pelo LIME e SHAP, através de seus métodos de testagem nas palavras, conseguem dar melhor contexto de classificação em como a palavra contribui para uma classificação.

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

Criando perturbações no documento, retirando palavras do seu corpo de documento, para verificar se sua ausência pode influenciar numa mudança de predição, isso determina o valor daquele palavra pro contexto local e para a tomada de decisão no documento em específico.

11. Como o SHAP calcula a importância das palavras? Qual é a ideia básica por trás dos valores de Shapley?

O SHAP calcula a importância das palavras usando os valores de Shapley, que vêm da Teoria dos Jogos.
A ideia central é avaliar quanto cada feature contribui para a diferença entre a previsão atual e a previsão média, considerando todas as combinações possíveis de features entrando ou saindo do modelo.
Assim, cada palavra recebe um valor SHAP que indica sua contribuição individual e justa para a predição.

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

O summary plot segue essa estrutura: tem o eixo vertical com as features (palavras). No meio do gráfico, tem uma linha que vai fazer esse limite entre as classes. Para cada feature, tem uma barra colorida de uma determinada cor e intensidade, indicando a frequência daquela palavra pelo dataset o quanto ela contribui na classificação, com cores mais quentes para indicar essa frequência e maior valor. As features ficam no eixo vertical e os valores SHAP ficam no eixo horizontal.

Features que estão do lado direito da linha, classificam para a classe positiva. As que estão à esquerda, geralmente classificam para negativa.

13. Em que situações as explicações do LIME e do SHAP concordaram no experimento de vocês ?

Eles concordaram nas explicações locais, quando as principais palavras destacadas por ambos os métodos mostravam importância semelhante e apontavam para a mesma direção na classificação da instância analisada.

14. Em que situações elas discordaram? Qual hipótese vocês têm para essa diferença ?

Houve discordância em alguns casos individuais, como no exemplo em que a palavra *"play"* teve maior contribuição no SHAP, enquanto *"ac"* apareceu mais relevante no LIME. Essa diferença pode ocorrer porque os métodos usam critérios distintos:

O LIME depende das perturbações locais e do modelo linear aproximado.

O SHAP usa valores de Shapley, considerando todas as combinações possíveis de features.

Apesar dar palavras serem o objeto de estudo desses explicadores, eles as manipulam de formas diferentes, então acontecerão eventuais discordâncias e até palavras diferentes num mesmo documento, assim como oposições, como “tncs” ter sido classificada como não spam no LIME e como spam no SHAP.

15. Houve alguma palavra considerada importante pelo modelo que parecia não fazer

sentido para vocês? O que isso pode indicar ?

Sim, a palavra *"ac"* no SHAP local do exemplo 10.

Isso pode indicar: ruído no texto, correlações espúrias aprendidas pelo modelo, algum padrão do dataset que não é intuitivo para humanos ou até problemas no pré-processamento dos dados.

Como algumas regras não foram criadas no pré-processamento, alguns ruídos ficaram, principalmente por serem textos em inglês, que gostam de unir palavras como símbolos, como em casos como ‘it is’, que flexionado fica it’s e no corte de caracteres especiais, ficou o “s” solto como palavra no dataset.

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

O LIME, por si só, é um método local. Para obter uma explicação global com ele, é preciso agregar várias explicações locais, por exemplo: gerar explicações para muitas instâncias e somar/contar a frequência das palavras mais importantes.

No experimento, porém, a explicação global foi feita usando o SHAP, que já fornece nativamente summary plots e importâncias globais, enquanto o LIME ficou responsável pelas explicações locais.