Análise Crítica do Trabalho de Classificação de Spam em SMS

Desenvolvido por: Peterson Alves, Gian Stuan, Gabriel Yuji, Bruno Henrique

Data da Análise: 22/04/25

1. Introdução

Este documento apresenta uma análise crítica do dataset de Spam SMS. O objetivo do referido notebook é a construção e avaliação de modelos de aprendizado de máquina para a classificação de mensagens SMS como *spam* ou *ham* (legítimas), utilizando o dataset público "SMS Spam Collection". A presente análise visa avaliar a robustez da metodologia empregada, a validade e interpretação dos resultados apresentados, bem como identificar potenciais limitações e áreas para aprimoramento técnico e metodológico.

2. Análise Metodológica

O desenvolvimento seguiu um fluxo de trabalho padrão para tarefas de classificação de texto, englobando etapas desde a exploração de dados até a avaliação de modelos.

- **2.1. Aspectos Positivos:**
    - **Estrutura e Documentação:** O notebook apresenta uma estrutura lógica e clara, com seções bem definidas e comentários que facilitam a compreensão do processo.
    - **Análise Exploratória de Dados (EDA):** Foi realizada uma EDA pertinente, investigando a distribuição de classes (identificando o desbalanceamento) e as características das mensagens (como o comprimento), fornecendo insights relevantes sobre o dataset.
    - **Seleção de Técnicas:** Foram empregadas técnicas padrão e relevantes para Processamento de Linguagem Natural (PLN) e classificação, incluindo vetorização por TF-IDF e *Bag of Words*, e algoritmos como SVM Linear, Naive Bayes Multinomial, Random Forest e Regressão Logística.
    - **Métrica de Avaliação:** A escolha da Pontuação F1 como métrica principal para comparação de modelos é metodologicamente correta, dada a natureza desbalanceada do dataset, pois busca um compromisso entre precisão e recall.
    - **Tratamento de Desbalanceamento (Tentativa):** A técnica SMOTE (Synthetic Minority Over-sampling Technique) foi corretamente introduzida como estratégia para mitigar o desbalanceamento, sendo conceitualmente aplicada apenas ao conjunto de treinamento para evitar vazamento de dados (*data leakage*).
- **2.2. Limitações Metodológicas:**
    - **Aplicação Inconsistente do SMOTE:** Uma limitação crítica reside na desconexão entre a demonstração do SMOTE e o treinamento dos modelos

finais. Os modelos comparados (SVM, NB, RF, LR) foram treinados utilizando o conjunto de dados original desbalanceado (`X_train`, `y_train`), e não os dados sinteticamente balanceados (`X_train_smote`, `y_train_smote`). Isso impede a avaliação do impacto real do SMOTE no desempenho dos classificadores.

- **Inconsistência no Pré-processamento:** Embora técnicas como remoção de *stopwords* e *stemming* tenham sido exploradas na seção 3.2, a vetorização final (seções 4.2 e 4.3) utilizou apenas um pré-processamento mais básico (função `preprocess_text`). Falta uma justificativa para essa escolha e uma análise comparativa do impacto das diferentes abordagens de pré-processamento nos resultados.
- **Ausência de Otimização de Hiperparâmetros:** Os modelos foram avaliados utilizando suas configurações padrão. A ausência de um processo de otimização de hiperparâmetros (e.g., via *GridSearchCV* ou *RandomizedSearchCV*) limita a capacidade de aferir o desempenho ótimo de cada algoritmo neste dataset específico.
- **Falta de Estratificação na Divisão dos Dados:** A função `train_test_split` não foi explicitamente configurada com `stratify=y`. Em datasets com significativo desbalanceamento de classes, a estratificação é crucial para garantir que as proporções das classes sejam mantidas nos conjuntos de treino e teste.
- **Parâmetros de Vetorização Não Justificados:** A definição de `max_features=5000` para os vetorizadores TF-IDF e CountVectorizer parece arbitrária, carecendo de experimentação ou justificativa para essa escolha específica. A exploração de outros parâmetros (`min_df`, `max_df`, `ngram_range`) poderia levar a representações mais eficazes.
- **Comparação Limitada de Técnicas de Vetorização:** Embora o *Bag of Words* (CountVectorizer) tenha sido implementado, a comparação final de modelos concentrou-se exclusivamente nos dados vetorizados por TF-IDF, sem apresentar uma avaliação comparativa direta entre as duas técnicas de vetorização.

3. Análise dos Resultados

Os resultados apresentados indicam um desempenho geral elevado em termos de acurácia (acima de 95% para a maioria dos modelos), métrica que pode ser inflada e enganosa em cenários desbalanceados.

- O SVM Linear foi identificado como o modelo de melhor desempenho com base na Pontuação F1 (~0.94), exibindo excelente precisão (~0.99), mas um recall (~0.91) que sugere que uma fração não desprezível de mensagens de *spam* não foi detectada.
- Os modelos Naive Bayes e Random Forest apresentaram valores de recall consideravelmente inferiores para a classe *spam*, potencialmente devido à maior sensibilidade ao desbalanceamento e à falta de otimização.
- A validade dos resultados como indicadores do desempenho ótimo ou mesmo da superioridade definitiva do SVM Linear é comprometida pelas limitações

metodológicas, especialmente a não aplicação efetiva do SMOTE nos modelos finais e a ausência de otimização de hiperparâmetros.

4. Discussão e Recomendações

O trabalho constitui uma aplicação inicial competente das técnicas de classificação de texto. No entanto, para aumentar a robustez e a confiabilidade dos achados, recomenda-se:

1. **Aplicação Efetiva do SMOTE:** Treinar e avaliar todos os modelos utilizando os dados balanceados (`X_train_smote`, `y_train_smote`) e comparar rigorosamente os resultados com aqueles obtidos nos dados desbalanceados.
2. **Otimização de Hiperparâmetros:** Implementar um processo sistemático de otimização de hiperparâmetros para cada modelo avaliado.
3. **Estratificação:** Garantir o uso de `stratify=y` na divisão treino-teste.
4. **Experimentação com Vetorização:** Avaliar o impacto de diferentes configurações dos vetorizadores (incluindo `ngram_range`) e realizar uma comparação direta entre TF-IDF e *Bag of Words*.
5. **Análise do Pré-processamento:** Investigar e justificar o impacto das diferentes etapas de pré-processamento (e.g., *stemming* vs. *lematização*, remoção de *stopwords*) no desempenho final.
6. **Expandir a Seção de Melhorias Futuras:** Detalhar os pontos acima e outras possíveis extensões (e.g., uso de *word embeddings*, modelos de *deep learning* como LSTMs ou Transformers) na seção correspondente do notebook.

5. Conclusão

O notebook demonstra uma compreensão fundamental do processo de construção de um classificador de spam em SMS. No entanto, inconsistências metodológicas significativas, como a não utilização efetiva dos dados balanceados pelo SMOTE e a ausência de otimização de hiperparâmetros, limitam a generalização e a confiança nas conclusões sobre o desempenho relativo e ótimo dos modelos. A implementação das recomendações propostas poderá fortalecer substancialmente a qualidade e a validade do estudo.
