UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
TÓPICOS ESPECIAIS EM INTELIGÊNCIA ARTIFICIAL - ELE0606

Docente:

José Alfredo Ferreira Costa

Discente:

Gutembergue Ferreira da Silva

Pedro Artur Fernandes Varela de Lira

Detecção de SPAM em SMS com uso de Modelos de Machine Learning

Natal - RN

Dezembro de 2023

Resumo

Este trabalho aborda a importância da classificação de mensagens SMS entre Spam e Ham, explorando técnicas de Processamento de Linguagem Natural (NLP) e diversos algoritmos de Machine Learning. Iniciando com uma introdução sobre a relevância da classificação e da utilização de NLP, o estudo prosseguiu com a limpeza dos dados, análise exploratória para compreensão da distribuição e das principais palavras associadas a cada categoria, e a transformação dos textos por meio da remoção de stopwords e vetorização com Tfidf. Posteriormente, 10 algoritmos de classificação foram treinados e avaliados, destacando-se Naive Bayes (NB), Extra Trees Classifier (ETC) e Random Forest (RF) como os mais eficazes, obtendo altos níveis de acurácia e precisão, com o NB alcançando 97.3% de acurácia e 100% de precisão nos dados de teste. Esses resultados ilustram a eficiência dos modelos na distinção entre mensagens indesejadas e legítimas, demonstrando a viabilidade dessas abordagens para a classificação de SMS.

Os códigos utilizados nas análises estão na seguinte página: GitHub.

Sumário

Lista de Figuras

1 Introdução

A classificação de mensagens SMS como Spam (indesejado) ou Ham (legítimo) é uma aplicação crucial em um mundo onde a comunicação digital desempenha um papel central em nossas vidas. A capacidade de identificar e filtrar mensagens não solicitadas, como spam, é fundamental para garantir uma experiência de comunicação mais eficiente e segura para os usuários. A utilização de técnicas de Processamento de Linguagem Natural (NLP) e algoritmos de Machine Learning desempenha um papel vital nesse processo.

O Processamento de Linguagem Natural oferece as ferramentas necessárias para compreender e extrair informações significativas de textos não estruturados, como mensagens de texto. Isso envolve a limpeza, pré-processamento e transformação desses dados textuais em uma forma que os algoritmos de Machine Learning possam interpretar e aprender.

Algoritmos de Machine Learning, como Naive Bayes, Random Forest, e Extra Trees Classifier, entre outros, desempenham um papel crucial na classificação de texto. Eles podem aprender padrões complexos nos dados textuais e tomar decisões de classificação com base nesses padrões, permitindo a automação e a eficiência na identificação de mensagens indesejadas.

Este trabalho aborda a importância da classificação de SMS, explorando a aplicação de técnicas de Processamento de Linguagem Natural e a implementação de diversos algoritmos de Machine Learning para distinguir entre mensagens legítimas e indesejadas, fornecendo uma visão detalhada do processo de análise e classificação de mensagens SMS.

2 Desenvolvimento

2.1 A Base de Dados

A base de dados utilizada neste estudo, denominada SMS Spam Collection, consiste em 5.574 mensagens SMS em inglês categorizadas como legítimas (ham) ou spam. Essas mensagens foram coletadas de diversas fontes na internet:

- Um conjunto de 425 mensagens de spam foi manualmente extraído do site Grumbletext, um fórum no Reino Unido onde usuários de celulares compartilham reclamações sobre spam SMS.
- Um subconjunto de 3.375 mensagens legítimas foi selecionado aleatoriamente do Corpus SMS da NUS (NSC), um conjunto de dados de mensagens legítimas coletadas para pesquisa na Universidade Nacional de Singapura.
- Uma lista de 450 mensagens legítimas foi coletada da tese de doutorado de Caroline Tag.
- O SMS Spam Corpus v.0.1 Big, incorporado ao conjunto de dados, consiste em 1.002 mensagens legítimas e 322 mensagens de spam, disponíveis publicamente.

Essa diversidade de origens das mensagens na base de dados proporciona uma representação variada de diferentes tipos de mensagens SMS, permitindo uma análise abrangente e robusta para a classificação de spam e mensagens legítimas.

2.2 Limpeza dos Dados

A limpeza de dados desempenha um papel fundamental na preparação e no pré-processamento dos conjuntos de dados. A eliminação de dados duplicados é crucial para garantir a precisão e a confiabilidade das análises. Dados duplicados podem introduzir viés nos resultados, distorcendo a representação real do conjunto de dados e influenciando negativamente os modelos de Machine Learning. Ao remover duplicatas, garantimos que cada entrada seja única, evitando distorções nos resultados e melhorando a eficácia dos algoritmos de classificação.

Além disso, a presença de dados vazios ou nulos pode afetar negativamente a qualidade e a eficácia dos modelos. A ausência de informações em determinadas entradas pode levar a interpretações imprecisas ou até mesmo comprometer a capacidade dos algoritmos de aprender padrões significativos. Portanto, a eliminação ou preenchimento adequado desses dados ausentes é essencial para garantir a integridade e a confiabilidade dos resultados obtidos com os modelos de Machine Learning.

Ao lidar com problemas de classificação onde as labels são textuais, como é o caso da categorização de mensagens SMS em spam e legítimas (ham), é necessário converter essas

categorias em valores numéricos compreensíveis pelos algoritmos de Machine Learning. O uso do LabelEncoder do sklearn permite essa transformação, atribuindo um rótulo numérico único a cada categoria textual. Isso facilita o processo de treinamento dos modelos, garantindo que eles possam interpretar e aprender com essas categorias de maneira eficiente, proporcionando resultados mais precisos na classificação das mensagens.

2.3 Análise Exploratória dos Dados

A exploração dos dados de texto desempenha um papel crucial para compreender a natureza das mensagens SMS e sua distribuição dentro do conjunto de dados. Aqui estão alguns pontos importantes a considerar:

2.3.1 Compreensão da distribuição de Spam e Ham

É fundamental analisar a distribuição das categorias de spam e ham no conjunto de dados. O desequilíbrio entre as classes pode afetar a capacidade do modelo de aprender corretamente e resultar em viés na predição.

Na Figura 1, temos a proporção de ham e spam na base de dados.

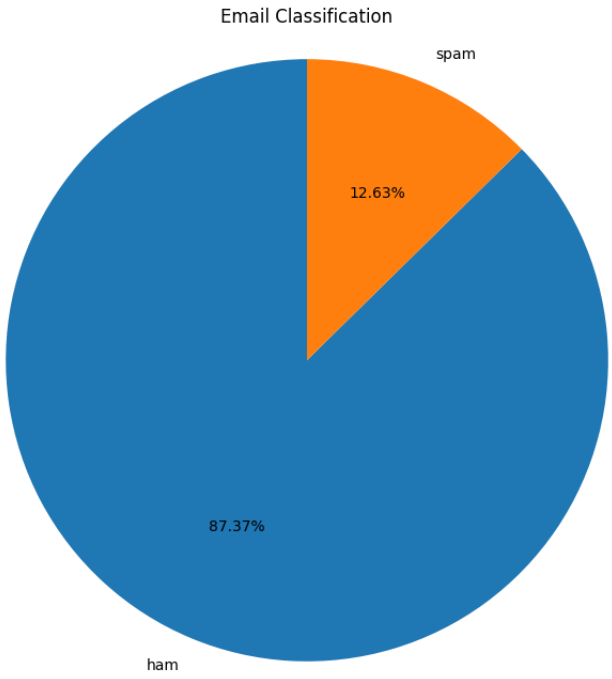

Figura 1: Dados por Label

Vemos que a base de dados está desbalanceada, há muito mais ham do que spam.

2.3.2 Distribuição do Número de Palavras e Caracteres

A análise da distribuição do número de palavras e caracteres por categoria (spam ou ham) oferece insights valiosos sobre as características dessas mensagens. Pode-se identificar diferenças significativas no tamanho médio das mensagens entre as classes. Essa análise ajuda na compreensão de padrões de uso de linguagem ou estilos de redação distintos entre mensagens legítimas e indesejadas, o que pode ser útil para o processo de classificação.

Nas Figuras 2 e 3 vemos, respectivamente, as distribuições dos números de caracteres e números de palavras por *label*, indicando que as spams são muito mais curtos que os emails válidos.

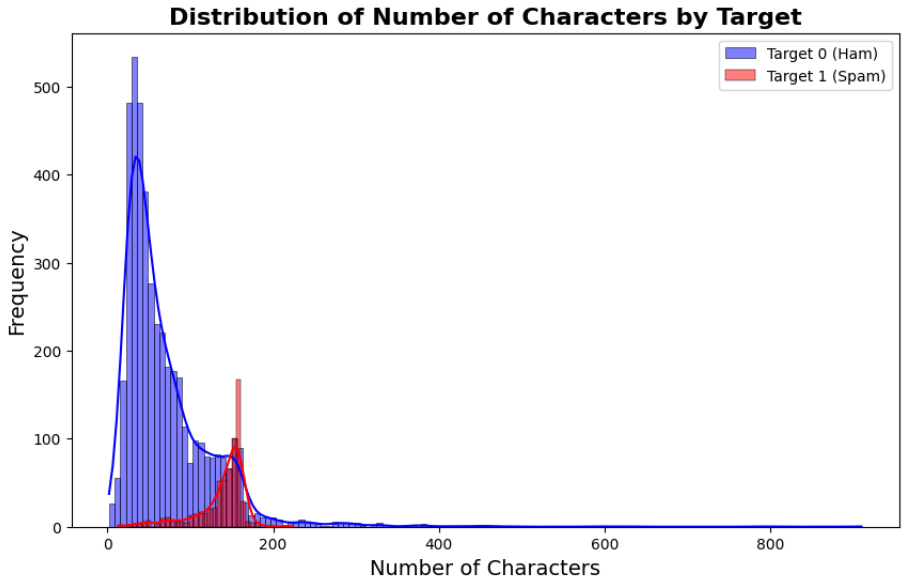

Figura 2: Distribuição da Quantidade de Caracteres por *label* (Ham ou Spam)

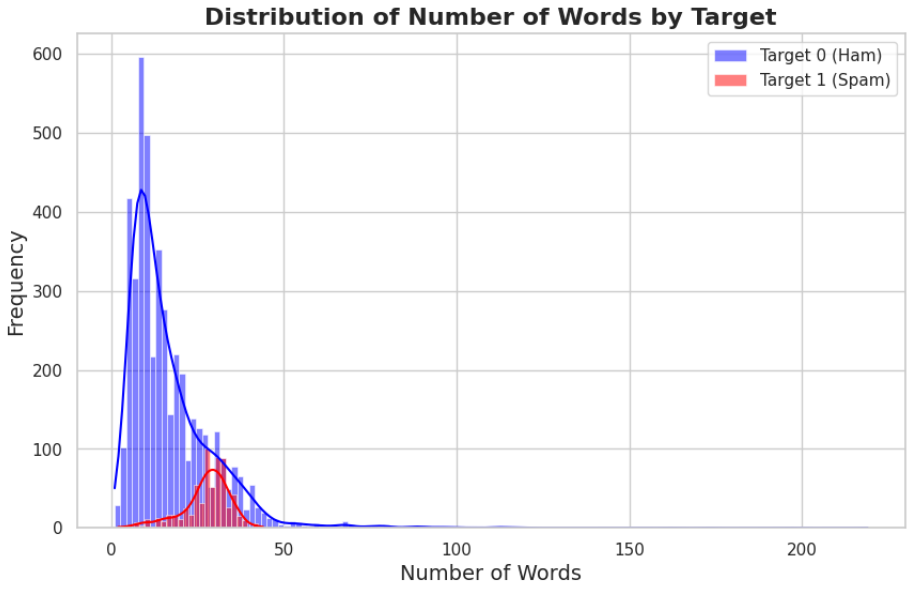

Figura 3: Distribuição da Quantidade de Palavras por *label* (Ham ou Spam)

2.3.3 Correlação

É importante analisar a correlação entre o número de sentenças, palavras e caracteres com relação à definição da *label* (spam ou ham).

Analisando o *heatmap* de correlação da Figura 4, vemos que há uma pequena correlação entre o número de caracteres, sentenças e palavras com relação à label, indicando que não seriam *features* úteis no processo de classificação.

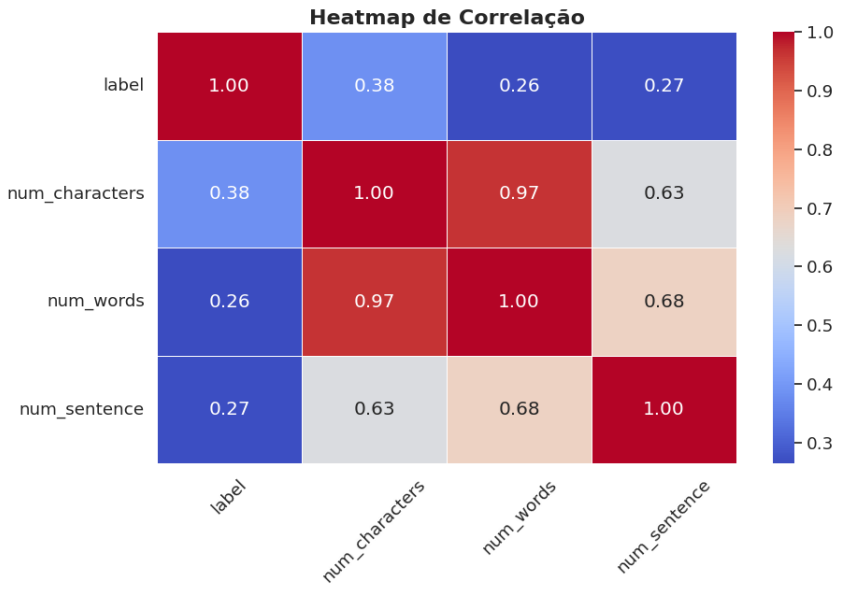

Figura 4: *Heatmap* de Correlação entre as Variáveis

2.4 Pré-Processamento

No processo de pré-processamento, foram realizadas a transformação dos dados levando em consideração a remoção aspectos irrelevantes dos textos de sms brutos e a realização de *Stemming* e Tokenização. Depois de realizados o pré-processamento dos dados, foram geradas *WordClouds* para os dados spam (Figura 5) e ham (Figura 6), a fim de analisar as palavras que mais aparecem para cada *label*.

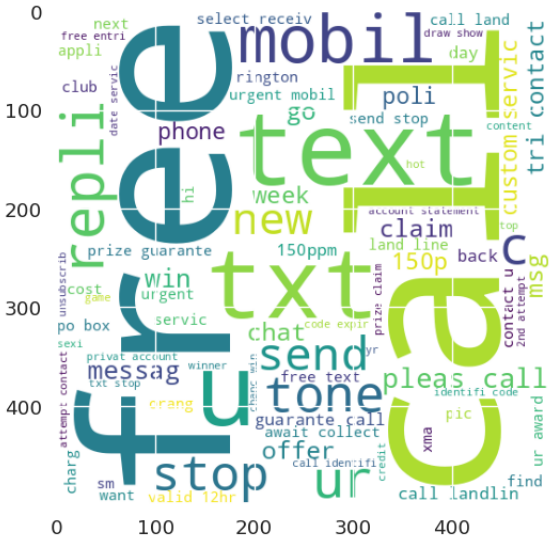

Figura 5: WordCloud dos SMS Spam

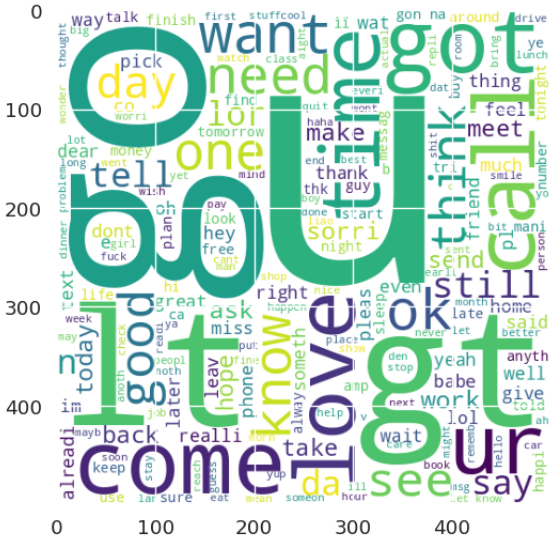

Figura 6: WordCloud dos SMS Ham

2.4.1 Remoção de Caracteres Especiais, Pontuações e Stopwords

A eliminação de caracteres especiais, pontuações e *stopwords* (palavras comuns que geralmente não contribuem significativamente para o sentido de um texto, como "o", "a",

"de", "em", etc.) é crucial. Caracteres especiais e pontuações não contribuem diretamente para a compreensão do significado das mensagens e podem interferir na análise. A remoção de *stopwords* reduz o ruído nos dados, focando nas palavras mais informativas para a classificação. Isso ajuda a melhorar a eficácia dos modelos ao destacar as palavras-chave mais relevantes para a tarefa de classificação.

2.4.2 Stemming e Tokenização

O processo de *stemming* envolve a redução de palavras às suas raízes ou formas básicas. Isso é feito para agrupar palavras relacionadas que possuem a mesma raiz, como "correr", "correndo"e "correu"todos derivam da palavra raiz "correr". Os algoritmos de *stemming* (stemmers) são utilizados para simplificar as palavras, o que pode reduzir a dimensionalidade dos dados, melhorar a generalização do modelo e ajudar na identificação correta de palavras-chave.

Já a tokenização envolve a divisão do texto em partes menores, como frases ou palavras individuais (tokens). Isso facilita o processamento e a análise dos dados textuais, permitindo que os modelos de Machine Learning compreendam e trabalhem com cada elemento textual de maneira mais eficaz.

2.5 Construção do Modelo

Os dados transformados foram divididos em 20% para teste 80% para treinamento e foram treinados e avaliados com 11 algoritmos diferentes, foram eles:

- **SVC (Support Vector Classifier):** Um classificador que busca encontrar o hiperplano de separação que maximiza a margem entre classes em um espaço de alta dimensionalidade.
- **KNN (K-Nearest Neighbors):** Um algoritmo de classificação que determina a classe de um ponto baseado na maioria das classes de seus vizinhos mais próximos (com base em uma medida de distância).
- **NB (Naive Bayes):** Um classificador probabilístico que aplica o Teorema de Bayes com a suposição "ingênua"de independência entre os recursos, sendo útil especialmente para dados textuais.
- **DT (Decision Tree):** Um modelo que divide os dados em subconjuntos menores com base nos atributos para tomar decisões de classificação.
- **LR (Logistic Regression):** Apesar do nome, é um modelo de classificação usado para prever a probabilidade de uma variável pertencer a uma classe ou outra, utilizando a função logística.

- **RF (Random Forest):** Um conjunto de árvores de decisão que combina várias árvores para aumentar a precisão e controlar o overfitting.

- **Adaboost:** Um algoritmo de ensemble learning que constrói um modelo forte a partir de vários modelos mais fracos, dando mais peso às instâncias classificadas erroneamente.

- **Bgc (Bagging Classifier):** Um método de ensemble que combina múltiplos classificadores de um mesmo tipo, geralmente melhorando a precisão e reduzindo a variância.

- **ETC (Extra Trees Classifier):** Similar ao Random Forest, porém, utiliza uma estratégia de construção de árvores mais aleatória para diminuir a variância.

- **GBDT (Gradient Boosting Decision Tree):** Um método de boosting que constrói árvores de decisão sequencialmente, melhorando o modelo em cada etapa com base nos erros anteriores.

- **XGBoost:** Uma implementação otimizada e escalável do algoritmo Gradient Boosting, conhecida por sua eficiência e desempenho em competições de machine learning.

Mas antes de cada classificador treinar e serem fornecidas validações, o textos de sms transformados (*features*) devem ser vetorizados.

2.5.1 Vetorização dos Textos

A técnica de vetorização usando o critério TF-IDF (Term Frequency-Inverse Document Frequency) desempenha um papel fundamental na representação dos dados textuais para os modelos de Machine Learning.

O TF-IDF é uma abordagem que avalia a importância de uma palavra em um documento em relação a um conjunto de documentos (corpus). Ele leva em consideração dois fatores principais:

- Frequência da Palavra (TF - Term Frequency): Refere-se à frequência com que uma palavra específica aparece em um determinado documento. Quanto mais vezes uma palavra aparece em um texto específico, maior é sua importância para aquele texto.

- Frequência Inversa no Documento (IDF - Inverse Document Frequency): Mede a raridade ou a importância relativa de uma palavra em relação ao corpus. Palavras que aparecem em muitos documentos têm um IDF menor, enquanto aquelas que aparecem em poucos documentos têm um IDF maior. Isso permite dar mais peso a palavras raras e menos peso a palavras comuns em todo o corpus.

Ao multiplicar o TF pelo IDF, obtemos o valor TF-IDF para cada palavra em cada documento. Esses valores representam a relevância das palavras no contexto do documento específico e do corpus geral. A vetorização com base no TF-IDF resulta em representações numéricas dos textos, onde cada documento é representado por um vetor, e cada palavra é atribuída a uma dimensão no vetor, com valores TF-IDF correspondentes.

2.6 Resultados

A tabela abaixo, mostra a acurácia e precisão (precisão é a razão $\frac{t_p}{t_p + f_p}$ onde t_p é o número de verdadeiros positivos e f_p o número de falsos positivos, é intuitivamente a capacidade do classificador de não rotular como positiva uma amostra negativa) nos os dados de teste para cada classificador.

Classificador	**Precisão**	**Acurácia**
SVC	0.9667	0.9749
KNN	1.0000	0.9052
NB	1.0000	0.9729
DT	0.8351	0.9294
LR	0.9519	0.9574
RF	0.9739	0.9720
Adaboost	0.9316	0.9642
Bgc	0.8527	0.9545
ETC	0.9832	0.9778
GBDT	0.9293	0.9487
xgb	0.9417	0.9691

Tabela 1: Resultados dos Classificadores: Precisão e Acurácia

3 Considerações Finais

A análise e classificação de mensagens SMS como spam ou ham são essenciais para filtrar e identificar com precisão mensagens indesejadas em nossas caixas de entrada. Neste estudo, exploramos a eficácia de diversos algoritmos de Machine Learning na tarefa de classificação de mensagens, utilizando métricas como precisão e acurácia para avaliar o desempenho dos modelos.

Observamos que os classificadores apresentaram desempenhos distintos na classificação das mensagens. Destacam-se o Extra Trees Classifier (ETC), Naive Bayes (NB) e Random Forest (RF) como os modelos mais eficazes, atingindo altos níveis de acurácia e precisão. O ETC se destacou com uma precisão de 98.32% e acurácia de 97.78%, seguido pelo NB com 100% de precisão e acurácia de 97.29%, e o RF com 97.39% de precisão e acurácia de 97.20% nos dados de teste.

Esses resultados demonstram a viabilidade e eficácia desses modelos na distinção entre mensagens de spam e ham. No entanto, é importante considerar que a escolha do classificador ideal pode variar de acordo com as necessidades específicas do contexto e a natureza dos dados.

A análise revelou a capacidade desses algoritmos em identificar padrões nos textos e destacar as palavras-chave relevantes para a classificação. A vetorização utilizando TF-IDF e o pré-processamento adequado dos dados contribuíram significativamente para a qualidade dos resultados obtidos.

Em resumo, os resultados obtidos destacam a importância do uso de técnicas de Processamento de Linguagem Natural (NLP) e algoritmos de Machine Learning na identificação de mensagens indesejadas, fornecendo um ambiente mais seguro e eficiente de comunicação para os usuários.

Referencial Teórico

GeeksforGeeks. (s/d). Stemming Words with NLTK in Python. Recuperado de geekforgeeks.

GeeksforGeeks. (s/d). Tokenize Text using NLTK in Python. Recuperado de geekforgeeks.

GeeksforGeeks. (s/d). Understanding TF-IDF (Term Frequency-Inverse Document Frequency). Recuperado de geekforgeeks.

GeeksforGeeks. (s/d). ML — Label Encoding of Datasets in Python. Recuperado de geekforgeeks.

Kaggle. (s/d). SMS Spam Collection Dataset. Recuperado de kaggle.