Detecção de Spam em E-mails

1. Introdução

Spam é a prática de enviar e-mails não solicitados, com o objetivo de divulgar produtos e serviços. Sua principal característica é o envio dessas mensagens para um grande número de pessoas em um curto período de tempo, com baixo custo, para atrair o maior número de pessoas. O principal problema para quem recebe estas mensagens é o inconveniente de abrir estas mensagens, além do perigo, pois algumas mensagens podem conter vírus. Alguns servidores possuem listas com IPs de servidores que enviam spam,compartilhando estas listas com outros servidores. Assim que a mensagem é recebida de um IP existente na lista ele já é classificado como spam. O objetivo deste trabalho é apresentar um algoritmo de classificação instantânea de e-mails de spam, através de aprendizado de máquina.

2. Análise exploratória

Com o objetivo de entender melhor os dados foi feita uma análise exploratória. Na figura a seguir são mostradas as palavras mais frequentes em todos os e-mails.

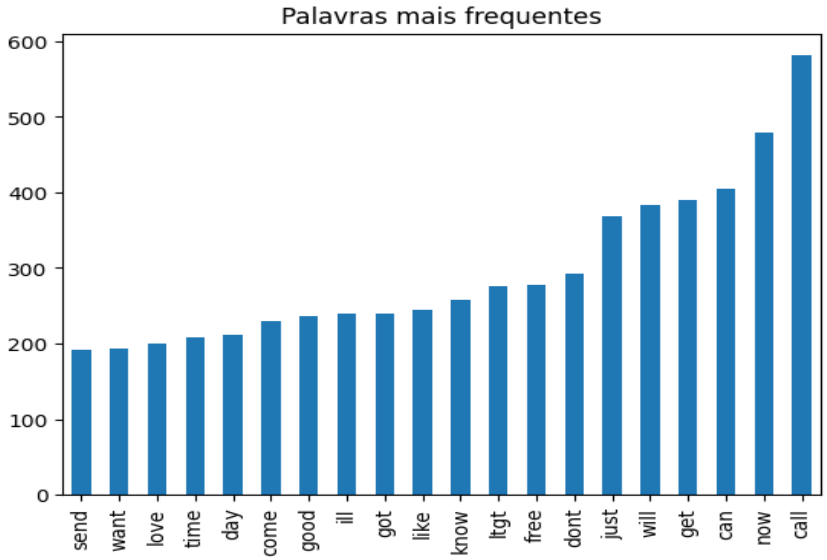

Fig1. Ocorrência de palavras

A seguir é mostrada a quantidade de e-mails comuns e e-mails spam.

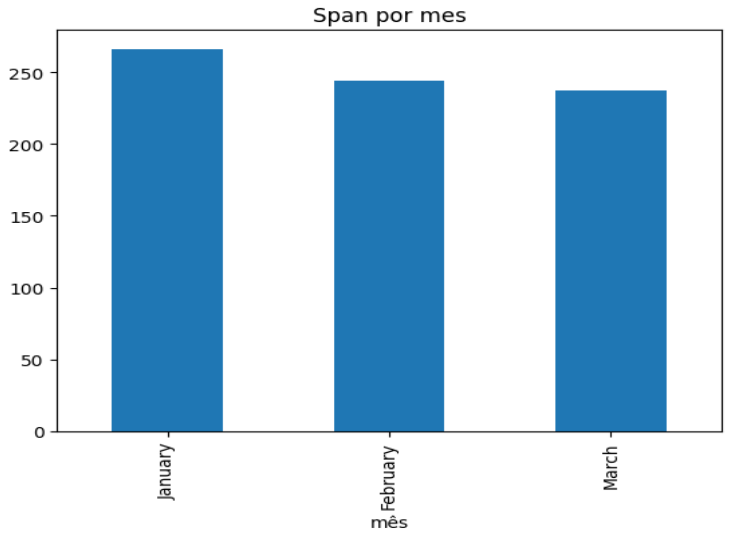

Fig2. Ocorrência de spam

Aqui é mostrada a quantidade de e-mails comuns recebidos.

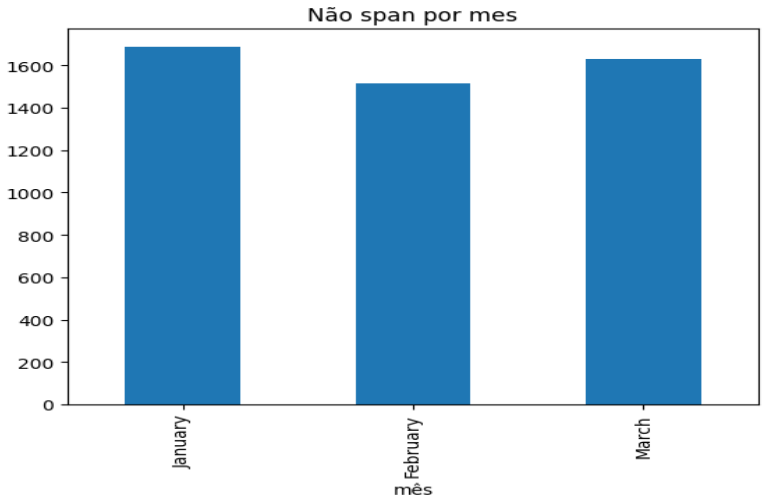

Fig3. Ocorrência de e-mails comuns

É possível observar que os e-mails spam representam aproximadamente 13% do total de e-mails recebidos por mês. A tabela abaixo mostra para cada mês as estatísticas de número de palavras nos e-mails, o número mínimo,máximo,a média,desvio padrão e variância.

mês	min	max	média	var	desvio
Janeiro	2	190	16.3	157.6	12.5
Fevereiro	2	100	16.0	121.9	11.0
Março	2	115	16.2	134.0	11.5

Tab1. Estatísticas de número de palavras

Observamos que a média,variância e desvio padrão do número de palavras não varia muito nos meses.

3. Algoritmo escolhido

O algoritmo Naïve Bayes tem sido muito usado em aprendizado de máquina para classificação de texto, por este motivo ele foi escolhido neste problema. Naïve Bayes é uma coleção de algoritmos de classificação que se baseiam na regra de Bayes, que considera que as variáveis do problema a ser resolvido são independentes de cada uma. O primeiro passo é dividir o conjunto de dados em duas partes: variáveis de entrada e variável resposta(saída).

4. Teorema de Bayes

O Teorema de Bayes calcula a probabilidade de um evento ocorrer dada a probabilidade de outro evento que já ocorreu. Esta é sua fórmula:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

onde:

A e B são chamados de eventos.

P(A | B) é a probabilidade do evento A, dado que o evento B é verdadeiro (ocorreu). O evento B também é denominado como evidência.

P(A) é o priori de A (a probabilidade independente anterior, ou seja, a probabilidade do evento antes que a evidência seja vista).

P(B | A) é a probabilidade de B dado o evento A, ou seja, a probabilidade do evento B depois que a evidência A é vista.

Assim podemos aplicar o Teorema de Bayes no nosso problema da seguinte forma, dada a matriz X de variáveis e o vetor y de resposta:

$$P(y \mid X) = \frac{P(X \mid y)P(y)}{P(X)}$$

P(y|X) é a probabilidade de observar a classe y dada a amostra X com X = (x1,x2,x2,xd)

Como assume-se que as probabilidades são mutuamente independentes:

$$P(y \mid x_1, \ldots, x_d) \propto P(y) \prod_{i=1}^{d} P(x_i \mid y)$$

5. Tratamento de texto

Como o texto não pode ser diretamente colocado como entrada do algoritmo é necessário transformá-lo em um vetor de variáveis numéricas. Vetorização e contagem é um método que transforma texto em um vetor com o número de vezes em que cada palavra apareceu no texto. A biblioteca scikit learn possui o método CountVectorizer, capaz de realizar esta operação de forma simples.

É necessário também remover as palavras mais comuns, chamadas em inglês de ‘stopwords’, que não adicionam muito significado para as frases. Para isto o método ContVectorizer possui a opção stop_words, onde é selecionado o idioma do nosso texto, no caso o inglês.

6. Treinamento do modelo

Nesta etapa foi feito o treinamento do modelo, separando a matriz X e o vetor y em 80% para ser usado como treino e em 20% para teste da performance do modelo na identificação de spam. Como verificou-se através da análise dos dados recebidos que 86% dos e-mails são comuns e apenas 14% são fraude, optou-se também por usar validação cruzada, treinando e testando o modelo 10 vezes com diferentes dados de treino e teste. Esta é uma prática usada frequente em problemas de classificação com classes desbalanceadas, pois reduz a possibilidade de sobretreino.

7. Resultados do modelo

As métricas analisadas são: Acurácia, Precisão, Revocação. Estas métricas variam do nível mais baixo até o mais alto: 0 - 1

Acurácia: É a quantidade de previsões que o modelo acertou para as classes e-mail spam e e-mail comum. No caso de conjuntos de dados com classes desbalanceadas tende a ser alta.

Precisão: É a quantidade de previsões corretas para a classe positiva, no caso e-mail spam.

Revocação: É a capacidade do modelo de identificar a classe positiva(e-mail spam)

A tabela abaixo mostra os resultados obtidos:

Acurácia	0.98
Precisão	0.92
Revocação	0.96
Acurácia Média	0.98

A acurácia é bastante alta, já que as classes são desbalanceadas. A precisão de 0.92 de acerto para fraude é bastante satisfatória, mostrando que o modelo acerta muito quando indica que um e-mail é spam. Já a revocação de 0.96, também elevada, nos mostra que o modelo tem alta capacidade de identificar e-mails que são spam. A acurácia média foi obtida através do treinamento com validação cruzada, se mostrou igual à acurácia do modelo treinado da forma simples.

8. Conclusão

Os resultados obtidos foram bastante satisfatórios, indicando que o algoritmo poderia ser usado com bastante sucesso na resolução do problema proposto.É preciso notar também que o código em python foi criado com bastante facilidade, resultando em uma solução em poucas linhas e de fácil manutenção. Algumas sugestões que poderiam melhorar a performance do modelo:

- Otimização de hiperparâmetros do algoritmo.
- Utilização do horário das mensagens, já que a análise exploratório mostrou um número maior de spam em alguns horários.

9. Referências

1. Scikit Learn- **https://scikit-learn.org/stable/modules/naive_bayes.html**