Avaliação de modelos para detecção de *spams* em mensagens de texto

Amanda Fagundes de Paula

Dezembro 5, 2021

1 Introdução

Com o avanço da tecnologia e a crescente adesão ao uso de *smartphones*, a troca de mensagens de texto cresceu significativamente nos últimos anos. Com isso, as empresas passaram a enxergar o envio de mensagens promocionais via *e-mail*, SMS e até mesmo *Whatsapp* como uma oportunidade de crescimento e divulgação de seus produtos. No entanto, é comum que tenhamos de lidar diariamente com o recebimento de conteúdos não solicitados, o spam, o que acaba se tornando um inconveniente no dia a dia. Diante disso, soluções têm sido desenvolvidas para identificação e filtragem desse tipo de mensagem, como forma de poupar os usuários de terem que lidar com conteúdo indesejado.

Normalmente, esse tipo de conteúdo apresenta uma linguagem específica, que costuma ser mais apelativa que as mensagens tradicionais. Essa premissa é utilizada em modelos de classificação para identificação de *spams*, um método de aprendizagem supervisionado, onde o algoritmo é treinado a partir de um conjunto de dados rotulados, de forma a identificar padrões que normalmente passariam despercebidos sob uma análise humana.

Diante disso, neste artigo, será descrita a metodologia adotada para a construção de um modelo de classificação para classificação de mensagens SMS como "*spams*" ou "não *spams*".

2 Dados

O *dataset* utilizado é composto por 5.574 mensagens descritas por 5 colunas que as caracterizam e 149 que contabilizam a ocorrência dos termos mais frequentes em cada uma. Do total de mensagens, 747 (aproximadamente 13%) são *spams* e 4.827 são mensagens de texto comuns.

O *dataset* contém mensagens que foram enviadas nos meses de Janeiro, Fevereiro e Março de 2017 e na Figura 1 é possível perceber que não existe uma grande variação na quantidade de mensagens (*spams* ou não) enviadas mensalmente.

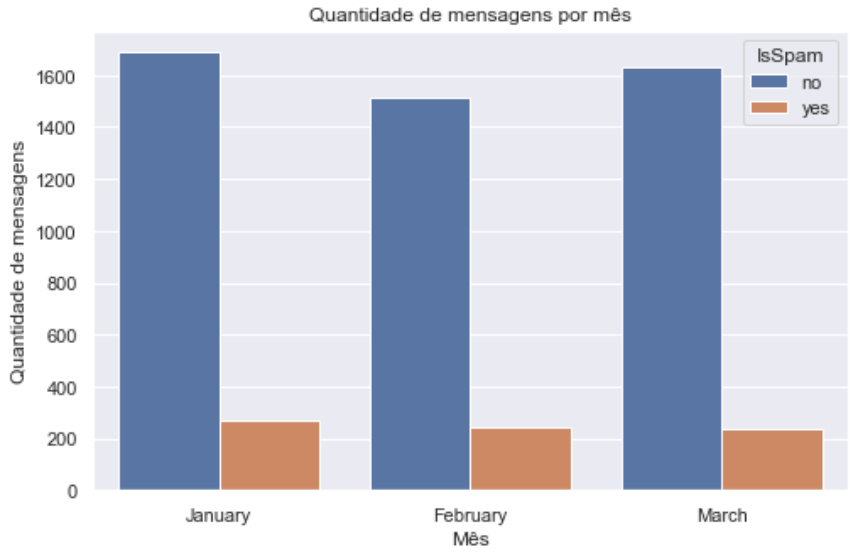

Figura 1: Quantidade de mensagens por classe

3 Metodologia

3.1 Pré-processamento

Neste primeiro momento, os dados são submetidos a um pipeline de tratamento no qual o conteúdo das mensagens é padronizado e termos irrelevantes são removidos. O pré-processamento de dados textuais é essencial para tarefas de classificação de conteúdo e, neste trabalho, adotou-se os processos descritos abaixo:

- *Lower-casing*: o conteúdo é convertido em caixa baixa.
- Remoção de *stopwords*: remove palavras que não agregam valor semântico ao conteúdo por serem muito comuns dentro de um idioma.
- Remoção de símbolos: remoção de caracteres especiais.
- Remoção de caracteres: remoção de caracteres únicos.
- Lematização: reduz os termos a sua forma base.

3.2 Representação vetorial

Para que os algoritmos sejam capazes de extrair valor das mensagens, é necessário que seu conteúdo seja transformado de uma maneira que seja possível inferir suas características. Neste trabalho, optou-se pela representação vetorial utilizando o TF-IDF (*term frequency–inverse document frequency*) que calcula, para cada termo presente em um documento, a sua relevância em relação à coleção de documentos no qual está inserido (Equação 1).

$$w_{i,j} = tf_{i,j} \times \log \frac{N}{df_i} \tag{1}$$

Essa estratégia se diferencia do *bag-of-words* por medir a importância de um termo considerando o número de vezes em que ocorre em todos os documentos da coleção. Em um contexto de identificação de *spams*, essa diferença pode ser fundamental para o desempenho do algoritmo.

3.3 Modelagem

Para classificação do conteúdo foram implementados modelos probabilísticos utilizando os algoritmos *Random Forest*, Regressão Logística e SVM (*Support Vector Machines*) cujos parâmetros foram definidos através de um tunning, no qual diferentes valores foram experimentados a fim de se obter uma combinação que trouxesse um melhor desempenho para os modelos.

A fim de se medir a qualidade do resultado obtido, foram calculadas as métricas de Acurácia, Precisão, Revocação e F1-Score, cujo objetivo é medir a performance geral dos modelos em termos da quantidade acertos e evidenciar o quão bem lidou com as duas classes de dados.

4 Resultado

Na Tabela 4 estão sumarizados os melhores resultados obtidos para cada modelo após o *tunning* de seus parâmetros. É possível perceber que a medida de Revocação é inferior a de Precisão para todos os modelos, o que indica que o algoritmo tem um desempenho inferior na classificação de mensagens que sejam *spams*.

Modelo	**Acurácia**	**Precisão**	**Revocação**	**F1-Score**
Regressão Linear	0.9505	0.9217	0.6790	0.7811
SVM	0.9828	0.9742	0.8917	**0.9310**
Random Forest	0.9710	0.9927	0.7834	0.8753

Tabela 1: Resultado

A matriz de confusão ilustrada na Figura 2 exibe a taxa de acerto do modelo utilizando o algoritmo *SVM*, cujo resultado foi superior aos demais. Analisando as taxas de acerto, percebe-se que todas as mensagens da categoria "Comum" foram classificadas corretamente e que 9% das mensagens da categoria "Spam" foram classificadas de maneira incorreta.

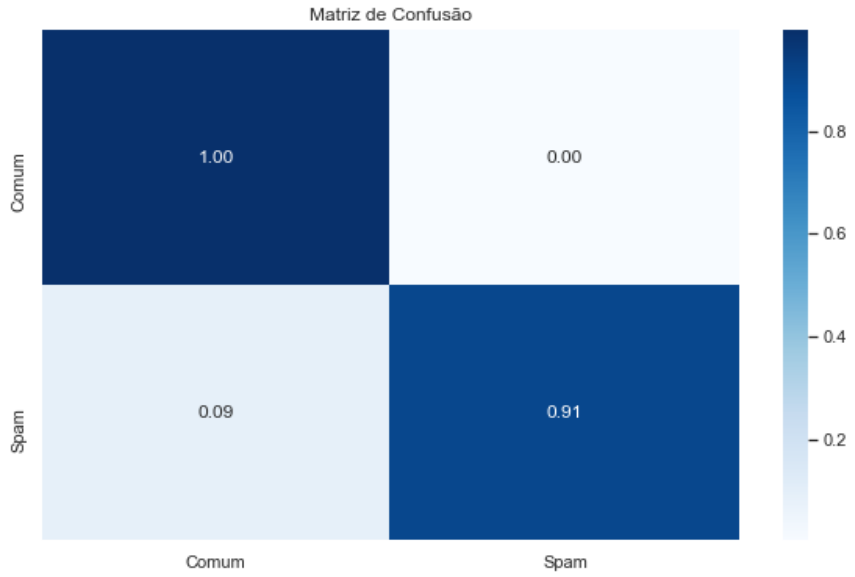

Figura 2: Matriz de confusão

5 Conclusão

O resultado obtido utilizando o modelo de SVM apresentou uma performance interessante ao lidar com a tarefa de classificação de textos no contexto estudado neste trabalho. A taxa de revocação de 89%, sugere que cerca de 11% dos conteúdos do tipo "Spam" são classificados incorretamente como mensagens comuns. Em contrapartida, o modelo apresenta uma Precisão de 97% e uma taxa de acerto de 100% para as mensagens da categoria "Comum". Considerando que a perda de uma mensagem importante que poderia ser filtrada pelo algoritmo, seja mais prejudicial do que um spam não identificado, pode-se concluir que a solução desenvolvida cumpre o seu papel.