Utilizando machine learning para classificação de mensagens de spam

Felipe Eduardo Gomes

Desafio Data Science - Senior Labs – Blumenau/SC – Brasil

felipe.edug@gmail.com

Resumo. *As mensagens de spam são um grande problema para as pessoas e empresas. Identificar as mensagens de spam torna-se uma ação preventiva para evitar a aplicação de golpes, disseminação de boatos e uso de softwares maliciosos. Diante disto, este trabalho apresenta a implementação de três modelos de Machine Learning, sendo eles Decision Tree, Random Forest e Nayves Bayes, que, a partir do conteúdo da mensagem, consigam predizer se a mensagem é ou não um spam. Foram utilizados dados de dataset disponibilizado pelo Senior Labs para treinar e validar, aplicando as métricas de acurácia, precisão, sensibilidade (recall) e a Area Under the Curve (AUC). Os três modelos obtiveram performance semelhante, com Decision Tree e Random Forest obtendo melhor acurácia de 90% e Nayves Bayes obtendo o melhor indicador da área sob a curva (AUC) de 0.92.*

Palavras-chave: Spam. Machine Learning. Decision Tree. Random Forest. Nayves Bayes

1. Introdução

A prática de SPAM consiste em utilizar meios eletrônicos para enviar mensagens que não foram solicitadas, em geral com o objetivo de fazer propaganda de produtos e serviços, mas também para aplicar golpes, disseminar boatos e espalhar softwares maliciosos (ou malware). Para empresas e pessoas a inconveniência é o dano principal, pois o SPAM o faz perder tempo abrindo e excluindo mensagens desnecessárias. É possível utilizar métodos estatísticos e de aprendizado de máquina (Machine Learning - ML) com o intuito de auxiliar na identificação de mensagens de spam. Com a aplicação dos modelos, é possível realizar ações preventivas de forma que a empresa diminua os riscos envolvidos. Diante deste cenário, este artigo apresenta o desenvolvimento de três modelos de Machine Learning, sendo eles, Decision Tree, Random Forest e Nayves Bayes, para resolução dos problemas propostos no desafio do Senior Labs. O trabalho foi dividido em duas etapas: 1) Extração de estatísticas da base de dados que foi disponibilizada: a) Exibir gráfico as palavras mais frequentes em toda a base de dados (Ex.: gráfico de barras, nuvem de palavras, etc); b) Exibir gráfico com as quantidades de mensagens comuns e spams para cada mês; c) Calcular o máximo, o mínimo, a média, a mediana, o desvio padrão e a variância da quantidade total de palavras (Word_Count) para cada mês; (d) Exibir o dia de cada mês que possui a maior sequência de mensagens comuns (não spam)

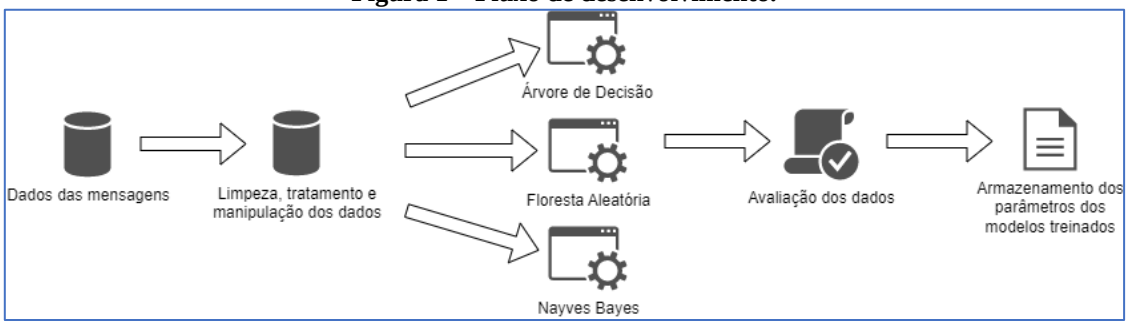
(2) Aplicação de um método capaz de classificar automaticamente as mensagens como “comum” e “spam”, justificando os resultados encontrados.

2. Desenvolvimento

Para o desenvolvimento do trabalho foi empregado o método Cross Industry Standard Process for *Data Mining* (CRISP-DM). Este modelo, na sua versão 1.0, propõe um ciclo de vida para projetos de mineração de dados, composto de seis etapas: entendimento do

negócio, entendimento dos dados, preparação dos dados, modelagem, avaliação e implantação [1]. A Figura 1 apresenta o fluxo utilizado para implementar a classificação das mensagens de spam. Para o desenvolvimento do trabalho utilizou-se a linguagem Python e bibliotecas de algoritmos e análise de dados, como scikit-learn, NumPy e Pandas.

Figura 1 – Fluxo de desenvolvimento.



Fonte: elaborado pelo autor.

2.1 Primeira Etapa

Como base de dados, utilizou-se o dataset de mensagens disponibilizado pelo Senior Labs, que possui 154 atributos e 5574 amostras. O dataset contém informações das mensagens (conteúdo, frequência de cada palavra, palavras mais utilizadas, entre outras). Dentre as variáveis utilizadas no dataset, todas estavam totalmente preenchidas, sem valores ausentes (missing). Na etapa de pré-processamento e transformação dos dados, foram criadas 3 variáveis adicionais, que são: (1) IsSpam_numeric; (2) Month; (3) Month_Description.

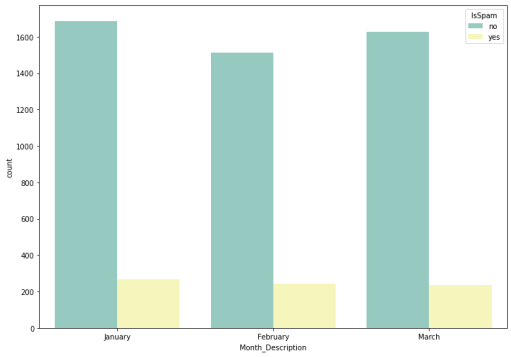
A figura 2 mostra a nuvem e o gráfico de barras com as palavras mais frequentes do dataset, sendo as cinco primeiras: "call", "now", "can", "get" e "will".

Figura 2 – Nuvem de palavras e gráfico de barras.

Fonte: elaborado pelo autor.

Na figura 3 é exibido gráfico com as quantidades de mensagens comuns e spams para cada mês, sendo possível observar que o mês de Janeiro teve mais mensagens de spam que os outros meses.

Figura 3 – Quantidade de mensagens comuns e spams por mês.



Fonte: elaborado pelo autor.

Podemos observar na tabela 1 a coleta das informações do máximo, o mínimo, a média, a mediana, o desvio padrão e a variância da quantidade total de palavras (Word_Count) para cada mês. A média de palavras entre os meses é 16 e a mediana entre 12 e 13. A maior mensagem foi recebida no mês de Janeiro com 190 palavras e a menor mensagem entre todos os meses foi de 2 palavras.

Tabela 1 – Estatística descritiva dos meses com base na quantidade de palavras.

	Fevereiro	Janeiro	Março
Máximo	100	190	115
Mínimo	2	2	2
Média	16,02	16,33	16,28
Mediana	13	13	12
Desvio Padrão	11,04	12,55	11,57
Variância	121,93	157,68	134

Fonte: elaborado pelo autor

Na tabela 2, é demonstrado os dias de cada mês que mais tiveram mensagens comuns (não spam). O mês de fevereiro foi o que teve mais mensagens em um dia, sendo 72. É possível observar também que os dias que mais possuem mensagem, fazem parte da primeira quinzena do mês.

Tabela 2 – Dias de cada mês que mais tiveram mensagens comuns (não spam).

Mês	Dia	Quantidade
Janeiro	01/01/2017	69
Fevereiro	13/02/2017	72
Março	08/03/2017	69

Fonte: elaborado pelo autor

2.2 Segunda Etapa

Na etapa de modelagem, optou-se pelos algoritmos Decision Tree, Random Forest e Nayves Bayes como métodos de classificação na construção do modelo preditivo. A Decision Tree representa uma função que toma como entrada um vetor de valores de atributos e retorna uma “decisão” — um valor de saída único [2]. A Decision Tree é um algoritmo muito utilizado para aprendizado de máquina supervisionado, pois utiliza uma abordagem de fácil visualização. A Decision Tree consiste em utilizar o “nó folha” que corresponde a uma classe que é identificada através de um “nó de decisão”, que realiza um teste sobre algum atributo. Para cada resultado do teste existe uma aresta para uma subárvore, sendo que cada subárvore tem a mesma estrutura da árvore [3]. O algoritmo de Random Forest é um agrupamento de árvores de decisão, que utiliza a técnica de bagging para melhorar a precisão da classificação, reduzindo a variância e evitando o

overfitting. A Random Forest constrói a sua decisão por meio da contagem de votos de cada árvore em cada classe e seleciona a classe vencedora com mais votos [3]. Os classificadores Naive Bayes são uma família de classificadores bastante semelhantes aos modelos lineares. Os modelos Naive Bayes são tão eficientes que aprendem os parâmetros observando cada recurso individualmente e coletando estatísticas simples por classe de cada recurso [4]. Existem três tipos de classificadores Nayves Bayes, sendo que neste trabalho será implementado o MultinomialNB, que é o mais utilizado para classificadores de texto. Na parte mineração dos dados, foi utilizada a técnica de validação cruzada para dividir os dados em conjuntos de treino e teste. A técnica utilizada foi a stratified k-fold, sendo dividida em 10 partições. A vantagem deste método é que todos os dados são usados para treinamento [4]. Para encontrar a melhor configuração dos parâmetros utilizados para cada modelo foi a aplicada a estratégia GridSearch. Essa técnica consiste em testar todas as combinações possíveis para encontrar o melhor conjunto de configurações para os modelos [4]. Foi realizado tratamento do conteúdo das mensagens, realizando tokenização (divisão das frases em suas palavras), remoção de stopwords, remoção de pontuações, todas as palavras foram passadas para minúsculo e stemming (remoção da parte final das palavras).

Na Tabela 3 são apresentados os resultados obtidos nos modelos, usando as medidas de acurácia, precisão, sensibilidade e AUC. A partir da Tabela 3, pode-se observar o desempenho de cada algoritmo. Os três modelos obtiveram performance semelhante, com Decision Tree e Random Forest obtendo melhor acurácia de 90% e Nayves Bayes obtendo o melhor indicador da área sob a curva (AUC) de 0.92.

Tabela 3 – Resultado da classificação dos modelos Decision Tree e Random Forest utilizando dados de teste

	Acurácia	Precisão	Sensibilidade	AUC
Decision Tree	0.90	0.94	0.85	0.90
Random Forest	0.90	0.95	0.85	0.89
Nayves Bayes	0.89	0.89	0.90	0.92

Fonte: elaborado pelo autor.

3. Conclusão

Este trabalho apresentou a construção de um modelo preditivo para auxiliar no processo de classificação de mensagens de spam. A partir de um conjunto de dados com amostragem de 5574 mensagens, onde 13% dessas mensagens são spam, foi possível construir um modelo de aprendizado de máquina capaz de classificar se a mensagem é spam ou não. Os três modelos obtiveram performance semelhante, com Decision Tree e Random Forest obtendo melhor acurácia de 90% e Nayves Bayes obtendo o melhor indicador da área sob a curva (AUC) de 0.92, podendo classificar e identificar através do conteúdo das mensagens se é spam ou não.

Referências

[1] CHAPMAN, P. et al. **CRISP-DM 1.0**: step-by-step data mining guide. [S.l.], 2000. Disponível em: http://www.statoo.com/CRISP-DM.pdf. Acesso em: 21 jan. 2023.

[2] NORVIG, P; RUSSELL, S. **Inteligência artificial.** 3. ed. Rio de Janeiro: Elsevier Editora, 2013. Disponível em: https://www.cin.ufpe.br/~gtsa/Periodo/PDF/4P/SI.pdf. Acesso em: 21 jan. 2023.

[3] BREIMAN, L.; CUTLER, A. **Random forests**. Berkeley, 2001. Disponível em: https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#intro. Acesso em: 21 jan. 2023.

[4] MULLER, A.; GUIDO, S. **Introduction to machine learning with Python**: a guide for data scientists. Sebastopol: O'Reilly Media, 2017.