Desafio Mensagens de Spam

Evandro Matheus Schmitz
evandroschmitz2010@hotmail.com

Introdução

O desafio proposto para este artigo consiste em ler um arquivo csv que contém 4827 mensagens comuns e 747 mensagens de spam, extrair algumas características sobre esta base de dados e apresentar um método para tentar classificar automaticamente as mensagens. Os dados estão divididos da seguinte forma:

- 1 coluna com a mensagem original;
- 149 colunas com valores inteiros que indicam a frequência de uma certa palavra;
- 1 coluna contendo a quantidade de palavras frequentes na mensagem;
- 1 coluna com a quantidade total de palavras;
- 1 coluna com a data e hora de mensagem;
- 1 coluna que identifica se a mensagem é spam ou não

Metodologia

Para resolver o desafio foi escolhida a linguagem de programação Python, junto com o uso das bibliotecas pandas, matplotlib e scikit-learn. O pandas é uma ferramenta para manipulação e análise de dados (PANDAS, 2020). O matplotlib é usado para gerar visualização de dados em Python (MATPLOTLIB, 2020). Já o scikit-learn é uma biblioteca de aprendizado de máquina (SCIKIT-LEARN, 2020a).

Na primeira etapa do desafio é necessário ler a base de dados do arquivo csv e extrair características dela. Para este fim foi usado o pandas para leitura e manipulação da base de dados e o matplotlib para criar a visualização de gráficos. A codificação do arquivo é `latin1`. Para facilitar a manipulação e a extração de certas características a coluna de data foi convertida para uma objeto de data do Python além de terem sido criadas 2 novas colunas, uma para armazenar o mês e o ano da mensagem e outra para armazenar somente a data da mensagem. Depois disto feito foi possível gerar gráficos e estatísticas sobre frequências de palavras, quantidades de mensagens por mês e o dia do mês com a maior quantidade de mensagens.

Para a segunda etapa do desafio foi necessário usar um método de classificação para tentar detectar automaticamente se uma mensagem era spam ou não. Foram usados o pandas e o scikit-learn. O algoritmo escolhido foi o vizinho mais próximo. Este algoritmo simples resolve problemas de classificação com base na distância entre um novo ponto e dos pontos já conhecidos para um problema (SCIKIT-LEARN, 2020b). O novo ponto terá sua classe definida com base nas classes dos seus k vizinhos mais próximos (*k-nearest neighbors* ou KNN em inglês) (SCIKIT-LEARN, 2020b). A biblioteca scikit-learn tem uma implementação própria para um classificador usando esta técnica: `KNeighborsClassifier`. Para realizar o treinamento e o teste do método escolhido a base foi dividida em duas partes, uma de treino que representa 80% do total de casos e outra de teste que representa 20% dos casos.

Resultados

Para a primeira etapa do desafio o quadro 1 apresenta algumas estatísticas extraídas sobre a quantidade máxima de palavras das mensagens recebidas por mês.

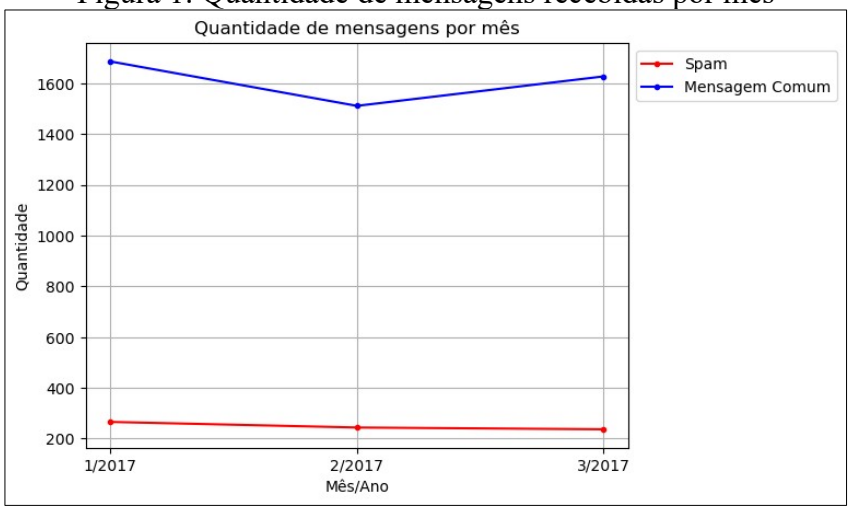
Quadro 1: Estatísticas do total de palavras das mensagens por mês

	Mínimo	Máximo	Média	Mediana	Desvio Padrão	Variância
01/2017	2	190	~16.34	13	~12.56	~157.68
02/2017	2	100	~16.03	13	~11.04	~121.94
03/2017	2	115	~16.29	12	~11.58	~134.01

Fonte: Autor

Já a figura 1 apresenta um dos gráficos gerados, apresentando o número de mensagens de recebidos por mês.

Figura 1: Quantidade de mensagens recebidas por mês



Fonte: Autor

Na segunda etapa do desafio a configuração do `KNeighborsClassifier` para a quantidade de vizinhos foi definida como três. Aplicando o modelo sobre os 20% da base de dados que foram reservados para testes a taxa de pontuação chegava a 95%. Esta taxa pode sofrer variações dependendo da quantidade de vizinhos escolhidos e também de como os dados de teste e treino são divididos.

Conclusão

O resultado apresentado pelo classificador KNN parece ser número muito bom e significativo, 95%, mas deve-se olhar pra ele com cuidado. É necessário avaliar a quantidade real de dados de cada classe que existe na base. São 4827 mensagens comuns contra 747 mensagens de spam, uma diferença bem significativa. E isto pode gerar uma tendência para classificar os dados mais em uma direção do que na outra (ROCCA, 2019). Para se ter uma ideia melhor se o classificador está funcionando de forma correta seria necessário uma base mais equilibrada ou verificar se o comportamento como ele está é aceitável (ROCCA, 2019).

Referências

MATPLOTLIB. **Matplotlib**. Disponível em: <https://matplotlib.org/index.html>. Acesso em: 03 maio 2020.

PANDAS. **Package overview**. Disponível em: <https://pandas.pydata.org/docs/getting_started/overview.html>. Acesso em: 03 maio 2020.

ROCCA, Baptiste. **Handling imbalanced datasets in machine learning**. 2019. Disponível em: <https://towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28>. Acesso em: 03 maio 2020.

SCIKIT-LEARN. **Getting Started**. Disponível em: <https://scikit-learn.org/stable/getting_started.html>. Acesso em: 03 maio 2020a.

SCIKIT-LEARN. **1.6. Nearest Neighbors**. Disponível em: <https://scikit-learn.org/stable/modules/neighbors.html>. Acesso em: 03 maio 2020b.