Senior Labs

Desafio de Data Science

13 de dezembro de 2021 / Eduardo de Paula Pazini

VISÃO GERAL

Para realização desse desafio foi escolhido a linguagem Python em conjunto às bibliotecas auxiliares Pandas (manipulação e análise dos dados), Matplotlib (plotagem dos gráficos), Numpy (operações matemáticas sobre estruturas de alto nível), Scikit-learn (aprendizado de máquina), Pytest (realização dos testes unitários) e Pylint (verificador de qualidade do código).

O código fonte utilizado para extração dos resultados pode ser encontrado no link: https://github.com/EduardoPazini/seniorlabs-challenge.

RESULTADOS - PRIMEIRA ETAPA

1. As palavras mais frequentes em toda a base de dados estão ordenadas da mais para a menos frequente na Figura 1, cada tupla contém a palavra em questão e sua respectiva quantidade de aparições. A palavra *'call'* é a mais utilizada com 581 aparições e as palavras *'getting'*, '*year'*, '*guaranteed'*, '*yet'*, '*people'*, '*thk'*, '*coming'* e '*mins*' são as menos utilizadas com 50 aparições. Os resultados de forma gráfica podem ser encontrados em: https://github.com/EduardoPazini/seniorlabs-challenge/blob/master/results/words_frequency_sorted.png.

```
[('call', 581), ('now', 479), ('can', 405), ('get', 390), ('will', 383), ('just', 368), ('dont', 292), ('free', 278), ('ltgt', 276), ('know', 257), ('like', 244), ('got', 240), ('ill', 239), ('good', 236), ('come', 229), ('day', 212), ('time', 208), ('love', 200), ('want', 193), ('send', 191), ('text', 189), ('going', 171), ('one', 171), ('need', 167), ('txt', 163), ('home', 162), ('lor', 160), ('see', 157), ('sorry', 156), ('stop', 155), ('still', 154), ('back', 152), ('reply', 144), ('today', 141), ('mobile', 139), ('tell', 137), ('new', 136), ('well', 135), ('later', 134), ('think', 132), ('please', 131), ('take', 126), ('phone', 126), ('cant', 125), ('week', 116), ('night', 115), ('claim', 113), ('much', 113), ('dear', 113), ('great', 111), ('hey', 111), ('pls', 109), ('happy', 107), ('hope', 104), ('give', 103), ('make', 101), ('way', 101), ('work', 100), ('thats', 99), ('wat', 96), ('number', 94), ('say', 92), ('prize', 92), ('right', 92), ('yes', 92), ('already', 90), ('tomorrow', 90), ('ask', 88), ('said', 87), ('really', 86), ('yeah', 86), ('amp', 84), ('message', 83), ('msg', 83), ('didnt', 81), ('miss', 79), ('life', 79), ('meet', 78), ('last', 78), ('morning', 77), ('babe', 77), ('thanks', 76), ('cos', 76), ('ive', 75), ('anything', 75), ('cash', 74), ('win', 73), ('won', 73), ('lol', 73), ('find', 73), ('every', 73), ('nokia', 72), ('sure', 71), ('pick', 71), ('also', 71), ('let', 70), ('something', 68), ('contact', 68), ('sent', 68), ('keep', 68), ('care', 68), ('urgent', 65), ('buy', 65), ('gud', 64), ('even', 63), ('next.', 62), ('feel', 62), ('first', 62), ('around', 61), ('went', 61), ('thing', 61), ('tonight', 60), ('someone', 60), ('per', 59), ('soon', 59), ('help', 59), ('wait', 59), ('place', 59), ('service', 59), ('many', 59), ('friends', 58), ('customer', 58), ('gonna', 58), ('always', 57), ('nice', 57), ('money', 57), ('chat', 57), ('wan', 57), ('wont', 56), ('late', 56), ('sleep', 56), ('dun', 55), ('leave', 55), ('youre', 54), ('waiting', 53), ('box', 53), ('things', 53), ('told', 53), ('wish', 52), ('name', 51), ('try', 51), ('getting', 50), ('year', 50), ('guaranteed', 50), ('yet', 50), ('people', 50), ('thk', 50), ('coming', 50), ('mins', 50)]
```

Figura 1. Relação de palavras por aparição nas mensagens

2. A quantidade de mensagem por tipo para cada mês está expresso na Figura 2, sendo:
    a. Janeiro: 1687 mensagens comuns e 266 spams;
    b. Fevereiro: 1512 mensagens comuns e 244 spams;
    c. Março: 1628 mensagens comuns e 237 spams.

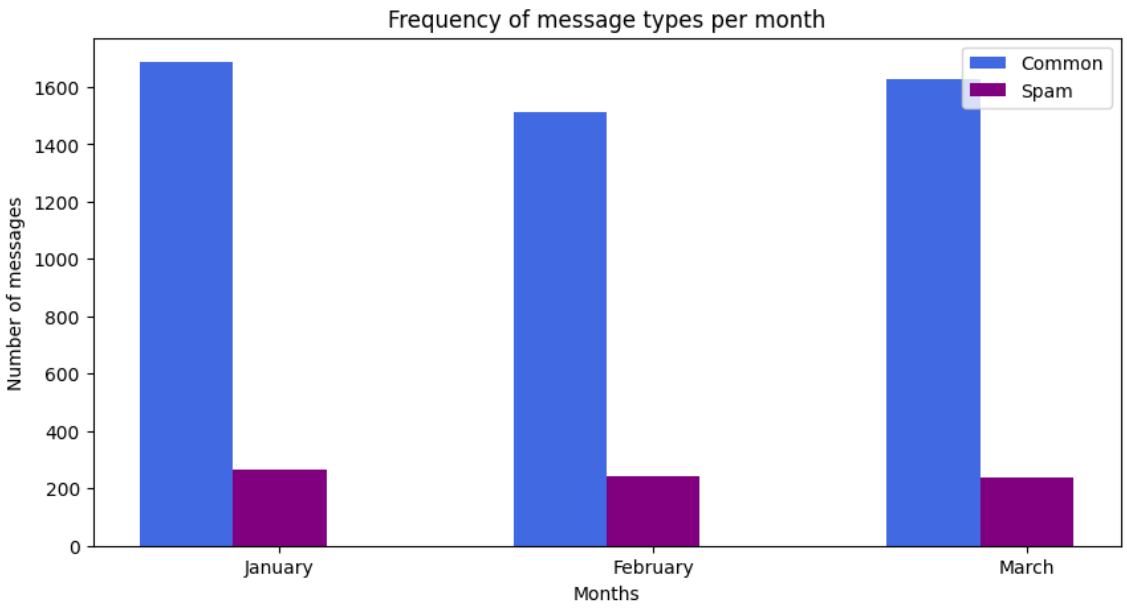

Figura 2. Quantidade de mensagens por tipo para cada mês

3. As estatísticas sobre a quantidade total de palavras utilizadas para cada mês foram:
    a. Janeiro:
        i. Máximo: 190
        ii. Mínimo: 2
        iii. Média: 16.337
        iv. Mediana: 13
        v. Desvio padrão: 12.557
        vi. Variância: 157.682
    b. Fevereiro:
        i. Máximo: 100
        ii. Mínimo: 2
        iii. Média: 16.029
        iv. Mediana: 13
        v. Desvio padrão: 11.042
        vi. Variância: 121.936
    c. Março:
        i. Máximo: 115
        ii. Mínimo: 2
        iii. Média: 16.285
        iv. Mediana: 12
        v. Desvio padrão: 11.576
        vi. Variância: 134.009

4. O dia de cada mês com a maior sequência de mensagens não spam foram:
    a. Janeiro: 2017-01-26 com 31 mensagens comuns em sequência;
    b. Fevereiro: 2017-02-04 com 39 mensagens comuns em sequência;
    c. Março: 2017-03-31 com 46 mensagens comuns em sequência.

RESULTADOS - SEGUNDA ETAPA

Para realizar a classificação automática das mensagens como “comum” ou “spam” foi utilizado uma técnica de *machine learning* chamada *train-test*, essa é utilizada para avaliar o desempenho de um algoritmo de aprendizado de máquina supervisionado. O procedimento envolve pegar um conjunto de dados e dividi-lo em dois conjuntos de dados separados. O primeiro conjunto de dados é usado para ajustar o modelo e é conhecido como conjunto de dados de treinamento. O segundo conjunto de dados, o conjunto de dados de teste, é usado para avaliar o ajuste do modelo de aprendizado de máquina. Finalmente, fazemos previsões, comparando-as com a saída real.

Para esse processo, nossa amostra foi delimitada em apenas duas colunas de informações, a primeira contendo os textos das mensagens e a segunda com suas respectivas classificações como sendo spam ou não. Nessa aplicação, o conjunto de teste representa 20% da amostragem total.

Para o treinamento da máquina atribuiu-se aleatoriamente um número para cada palavra. O método da biblioteca Scikit-learn conta o número de ocorrências de cada palavra e as salva em uma variável. Dessa forma, o modelo de aprendizado de máquina será capaz de prever mensagens de spam com base no número de ocorrências de certas palavras que são comuns nesse tipo de mensagem.

O modelo foi construído sobre o algoritmo SVM (*support vector machine*), esse que é um modelo linear para classificação e regressão, a ideia do algoritmo é simples, ele cria um linha/hiperplano que separa os dados em classes. Em seguida, ele verifica a previsão e ajusta os parâmetros até atingir a maior precisão possível.

Por fim, quando os testes foram aplicados os resultados se apresentaram bem satisfatórios, o modelo pontua uma acurácia na detecção das mensagens, para todas as execuções conseguiu-se classificar os spams com mais de 97% de precisão.