

ANÁLISE DE SENTIMENTOS EM AVALIAÇÕES DE NPS

Ricardo Silveira Martins de Mello

ricardosmdemello@gmail.com

Keras





Visão geral

- O presente trabalho tem como objetivo demonstrar e entender, através da análise de sentimentos se os clientes da empresa estavam satisfeitos ou insatisfeitos ao avaliar o produto, através dos comentários de avaliações dos dados de NPS coletados, para isso foram treinados dois algoritmos para a classificação, um algoritmo de machine learning MultinomialNB "scikit-learn" e uma rede neural recorrente LSTM "Keras"
- □ DADOS UTILIZADOS E ALGORITMOS: Foi utilizado uma base de dados *imdb pt-br* para o treinamento do algoritmo de machine learning e rede neural, e outra base NPS para a classificação do sentimento:
- □ Avaliações de NPS (Net Promoter Score) "Período avaliado": Avaliação de satisfação do usuário com a plataforma; Fonte: Safeweb Segurança da Informação LTDA.
- ☐ IMDB PT-BR: Dataset popularmente utilizado para treinar modelos de classificação de textos, entre "positivos" e "negativos" através de testes. Fonte: Kaggle
- ☐ MultinomialNB (Naive Bayes): Algoritmo de classificação probabilística, baseado no teorema de Bayes, que consiste em calcular um conjunto de probabilidades a quantidade de combinações de valores de um conjunto de dados e sua frequência.

Modelo MultinomialNB - scikit-learn



Modelo Keras

```
1 inputs = keras.Input(shape=(850,))
  2 embed = Embedding(20000, output_dim=300)(inputs)
  3 lstm = LSTM(256, return_sequences=True)(embed)
  4 pooling = GlobalMaxPool1D()(lstm)
 5 classification = Dense(2, activation='softmax')(pooling) # N x 2 # Said
  1 model = Model(inputs=inputs, outputs=classification)
 1 model.summary()
Model: "model_1"
 Layer (type)
                             Output Shape
                                                       Param #
 input_2 (InputLayer)
                             [(None, 850)]
 embedding_1 (Embedding)
                              (None, 850, 300)
 lstm_1 (LSTM)
                             (None, 850, 256)
                                                       570368
 global_max_pooling1d_1 (Glo (None, 256)
 balMaxPooling1D)
 dense_1 (Dense)
                                                       514
                             (None, 2)
Total params: 6,570,882
Trainable params: 6,570,882
Non-trainable params: 0
```

Resultado

☐ MultinomialNB obteve-se a assertividade da acurácia resultando <u>0.9143</u>, enquanto para a **rede** neural LSTM obteve-se a assertividade de <u>0.8088</u>. análise das acurácias apresentas, percebeu-se que o modelo de classificação mais eficiente para ser utilizado para este tipo de trabalho foi o modelo MultinomialNB, já que sua pontuação fscore (média harmônica ponderada da precisão e recordação, onde uma pontuação fscore atinge seu melhor valor em 1 e 0 para o pior valor.) obteve sua precisão em <u>0.9123</u>, enquanto para o modelo de rede neural recorrente LSTM obteve sua pontuação fscore em 0.4827, concluindo então que o modelo MultinomialNB seria o mais indicado, já que obteve a melhor assertividade em seus resultados durante a classificação dos dados de NPS.

Resultado das classificações

Promotores (9 e 10)	MultinomialNB	Keras	MultinomialNB/Keras
Positivo	7196	7294	7196
Negativo	2681	2583	1262
Total	9877	Diferença	2740
Passivos (7 e 8)	MultinomialNB	Keras	MultinomialNB/Keras
Positivo	101	128	77
Negativo	96	69	96
Total	197	Diferença	75
Detratores (0 a 6)	MultinomialNB	Keras	MultinomialNB/Keras
Positivo	51	99	34
Negativo	200	152	135
Total	251	Diferença	82
Total Registros	10325		