Trabalho Final - Ciência de Dados 3

Universidade Federal do Ceará

Aluno: Matheus Viana - 492959

1. Conjunto de Dados Utilizado:

Usei o conjunto 'twitter_training.csv', que tem tweets classificados em sentimentos como Positive, Negative, Neutral e Irrelevant. Como o trabalho pedia classificação binária, filtrei só os tweets Positive e Negative pra fazer uma análise de sentimento polarizada. Depois, balanceei as classes pegando 3000 amostras de cada. (testei com outros valores tb)

2. Sumário dos Resultados Obtidos para Cada Experimento:

Aqui estão os números que consegui depois de rodar os 8 experimentos pedidos. Usei acurácia, precisão e recall pra avaliar como os modelos se saíram no conjunto de teste (20% dos dados):

- Bag of Words + Regressão Logística: Acurácia: 0.4745, Precision: 0.4771, Recall: 0.4745
- Bag of Words + Naive Bayes: Acurácia: 0.2905, Precision: 0.4061, Recall: 0.2905
- Bag of Words + KNN: Acurácia: 0.5937, Precision: 0.6166, Recall: 0.5937
- Bag of Words + Rede Neural: Acurácia: 0.5227, Precision: 0.5230, Recall: 0.5227
- TF-IDF + Regressão Logística: Acurácia: 0.4869, Precision: 0.4882, Recall: 0.4869
- TF-IDF + Naive Bayes: Acurácia: 0.3107, Precision: 0.4042, Recall: 0.3107
- TF-IDF + KNN: Acurácia: 0.5784, Precision: 0.5994, Recall: 0.5784
- TF-IDF + Rede Neural: Acurácia: 0.5276, Precision: 0.5301, Recall: 0.5276

O KNN com Bag of Words foi o que mandou melhor, com quase 60% de acurácia. O NB, coitado, ficou lá embaixo, mal passou dos 30%. Acho que os tweets têm muito ruído ou sarcasmo que ele não pegou.

3. Melhor Hiperparâmetro Obtido em Cada Experimento:

- Bag of Words + Regressão Logística: C = 0.1
- Bag of Words + Naive Bayes: Nenhum hiperparâmetro ajustado.
- Bag of Words + KNN: n_neighbors = 4

- Bag of Words + Rede Neural: hidden_layer_sizes = (32, 32, 32)
- TF-IDF + Regressão Logística: C = 1.0
- TF-IDF + Naive Bayes: Nenhum hiperparâmetro ajustado
- TF-IDF + KNN: n_neighbors = 8
- TF-IDF + Rede Neural: hidden_layer_sizes = (16, 16, 16)

A Regressão Logística ficou melhor com C mais baixo no Bag of Words, mas subiu um pouco com TF-IDF. O KNN variou o k dependendo da representação, e a Rede Neural gostou de camadas mais robustas com 16 ou 32 neurônios.

OBS: Utilizei PCA(n_components=100), pois n_components=0.9 não funcionou.