Assignment 3

Using Classical Machine Learning for an NLP Task

Name: Islam Mostafa Abdelaziz Aboulkhair Gaber

ID: 13

Name: Omar Youseef Abdelsattar

ID: 43

Problem statement:

You are given the IMDB movie review dataset, which is a dataset for binary sentiment classification.

The IMDB dataset was first proposed by Maas et al. [1] as a benchmark for sentiment analysis.

The core dataset contains 50,000 reviews split evenly into 25k training and 25k testing sets.

The overall distribution of labels is balanced in both the training and testing sets (25k positive and 25k negative).

There are additional 50,000 unlabeled reviews that may be used for unsupervised learning.

The dataset can be found at: https://ai.stanford.edu/~amaas/data/sentiment/
You are required to apply any required text preprocessing techniques on the dataset.

Then, you are required to construct different classification models using different approaches, tune the hyper-parameters of these models and compare the performance of the models under multiple factors.

Observations:

- First, we downloaded the data in colab.
- Then we check if it has nulls or not , so it doesn't contain nulls.

- Then we made the text preprocessing using nltk library as it has operations such that: tokenization, stop words removal, lemmatization.

- After preprocessing, we convert the text of each review into a vector form to construct the data matrix.
- We used **mainly** sklearn feature vectorizer to create the required feature vector to construct the data matrix.

- Also, we used Gensim library for obtaining fasttext word embedding.

- Then we applied different classifiers such as: knn , naïve bayes, Adaboost, Random forests, Logistic regression.
- Also, we made the linear svm classifier but it takes very long time.

- Then we made the hyperparameter tuning.

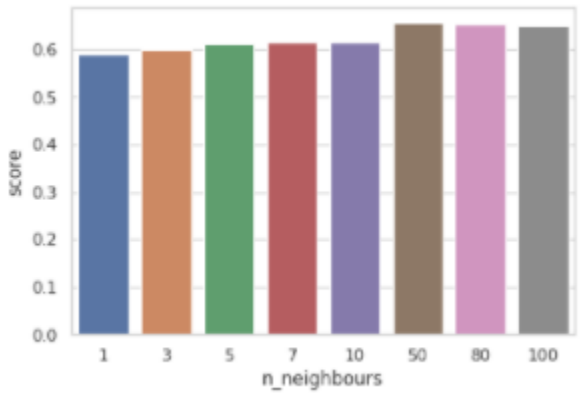
- Then we made evaluation for every classifier by varying the parameter and indicating the accuracy according to the parameter.

- And in the end we made evaluation by comparing all the classifiers together and their accuracies.

Evaluation results:

- For the **knn classifier** by varying number of neighbours:

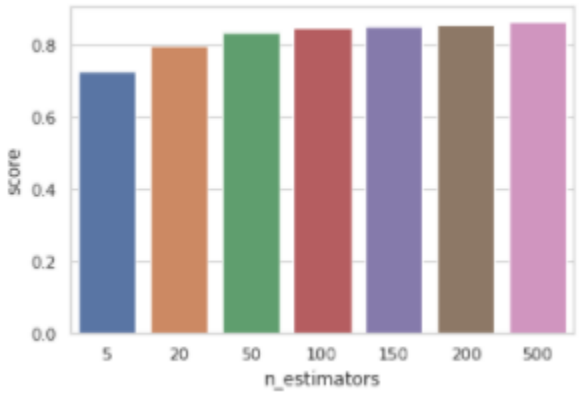
	n_neighbours	score
0	1	0.589000
1	3	0.600400
2	5	0.611160
3	7	0.614640
4	10	0.614520
5	50	0.655840
6	80	0.654480
7	100	0.651720



➢ The best accuracy is at number of neighbours: **50**

- For the **Random forest classifier** by varying number of estimators:

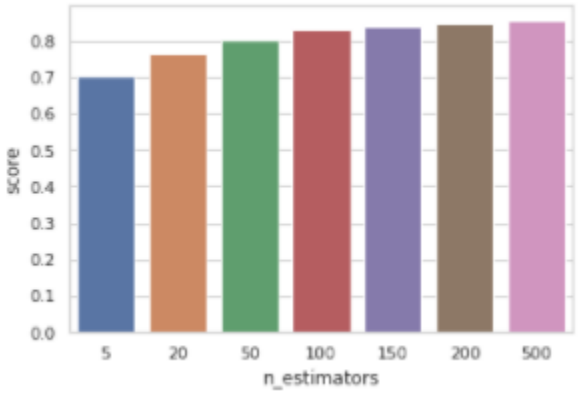
	n_estimators	score
0	5	0.725000
1	20	0.798560
2	50	0.831960
3	100	0.847480
4	150	0.851400
5	200	0.856240
6	500	0.862280



➢ The best accuracy is at number of estimators: **500**

- For the **Adaboost classifier** by varying number of estimators:

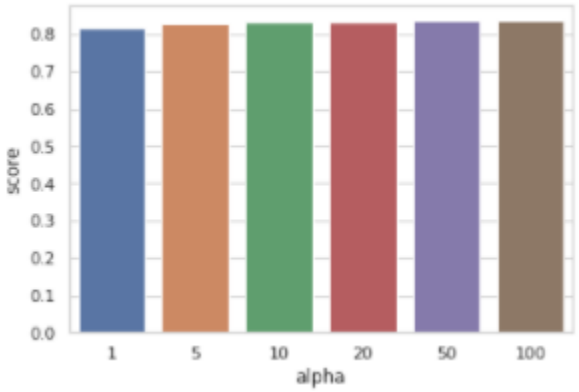
	n_estimators	score
0	5	0.703480
1	20	0.765600
2	50	0.800640
3	100	0.829520
4	150	0.840880
5	200	0.846560
6	500	0.855240



➢ The best accuracy is at number of estimators: **500**

- For the **naïve bayes classifier** by varying alpha:

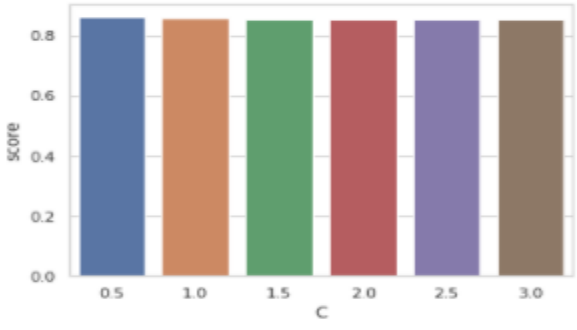
	alpha	score
0	1	0.817840
1	5	0.827600
2	10	0.831120
3	20	0.834200
4	50	0.835880
5	100	0.835760



➢ The best accuracy is at alpha: **50**

- For the **logistic regression classifier** by varying C:

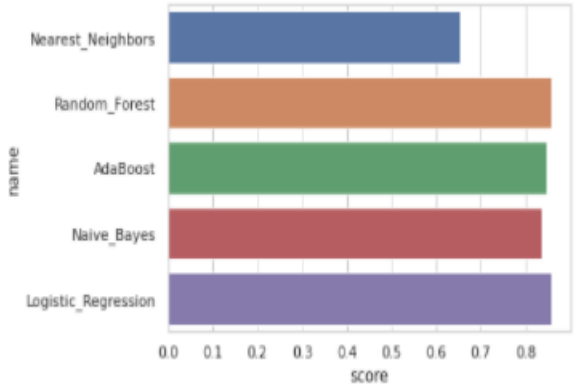
	C	score
0	0.500000	0.861840
1	1.000000	0.857040
2	1.500000	0.855520
3	2.000000	0.853760
4	2.500000	0.853720
5	3.000000	0.852960



➢ The best accuracy is at C: **0.5**

- Finally for all classifiers at the following parameters:
  - Knn classifier by number of neighbours = 80.
  - Random forest classifier by number of estimators = 200.
  - Adaboost classifier by number of estimators = 200.
  - Naïve bayes classifier by alpha = 50.
  - Logistic regression by C = 1.

	name	score
0	Nearest_Neighbors	0.654480
1	Random_Forest	0.856240
2	AdaBoost	0.846560
3	Naive_Bayes	0.835880
4	Logistic_Regression	0.857040



  - The best accuracy is at logistic regression and random forest.
  - The worst accuracy is at knn.