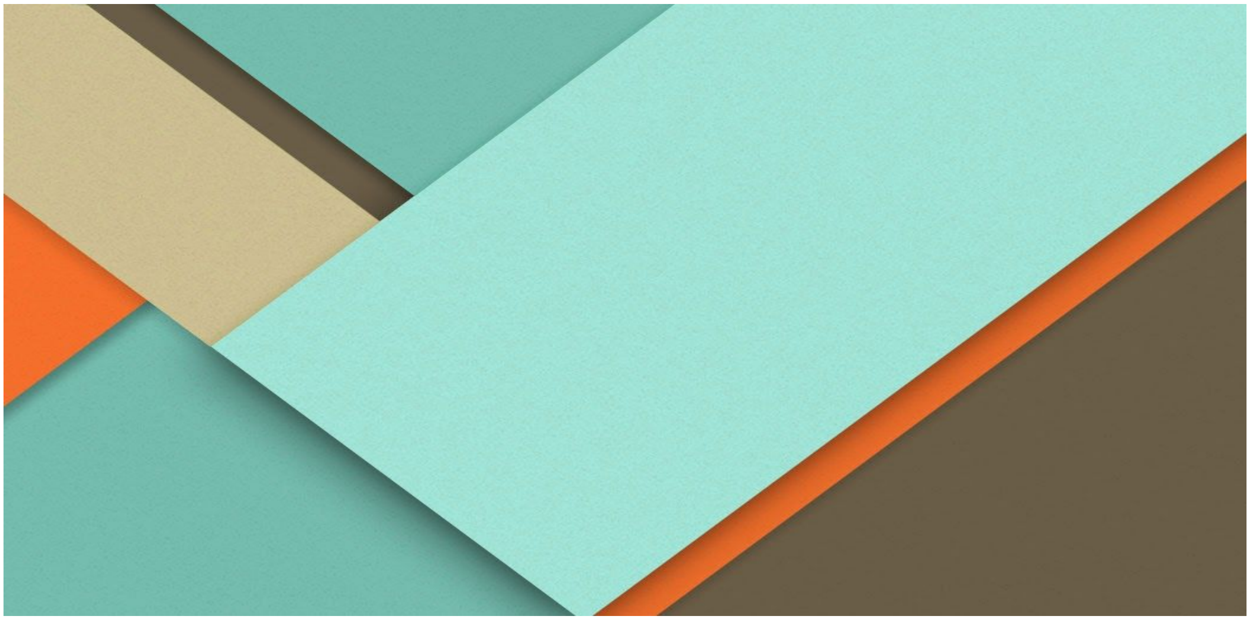


Sentiment Analysis with Intensity

CS 626

Team:

Piyush Onkar (160050012)

Vishal Pramanik (203050104)

Sandeep Parihar (203050083)

Instructor:

Prof. Pushpak Bhattacharya

Description of the Project

Our project is based on Sentiment Analysis of Amazon Reviews with Intensity. For this project we have used 2 different datasets, one with 2 labels (positive sentiment and negative sentiment) and one with 5 labels (labels describe the rating of the review given). We have used this Dataset for predicting the review as positive and negative and this Dataset for predicting label among 5 labels.

Data Preprocessing

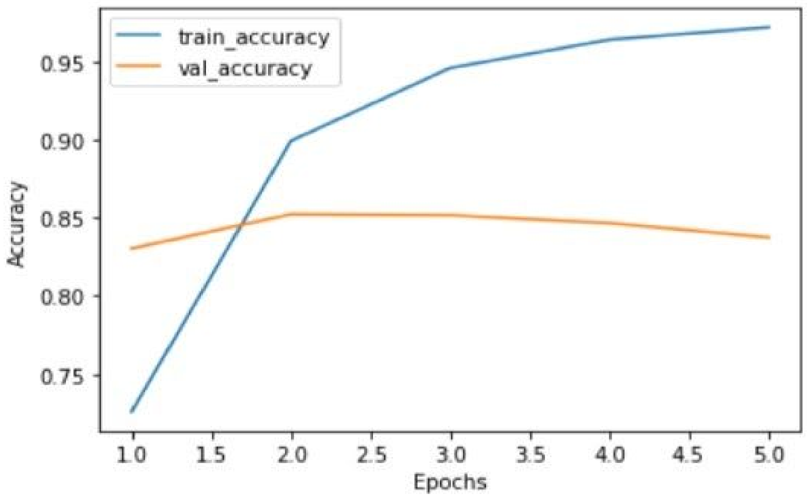
For data preprocessing we are removing all stop words from the sentence. Punctuations and special characters are also removed from dataset. Extra white spaces between are also removed. We are also checking whether the word is in valid English word or not and if it is not a valid English word, we are discarding it. Words in the dataset are converted to lower cases. Proper nouns, pronouns, determinants and verbs which has no contribution in the sentiment of the sentence are also removed.

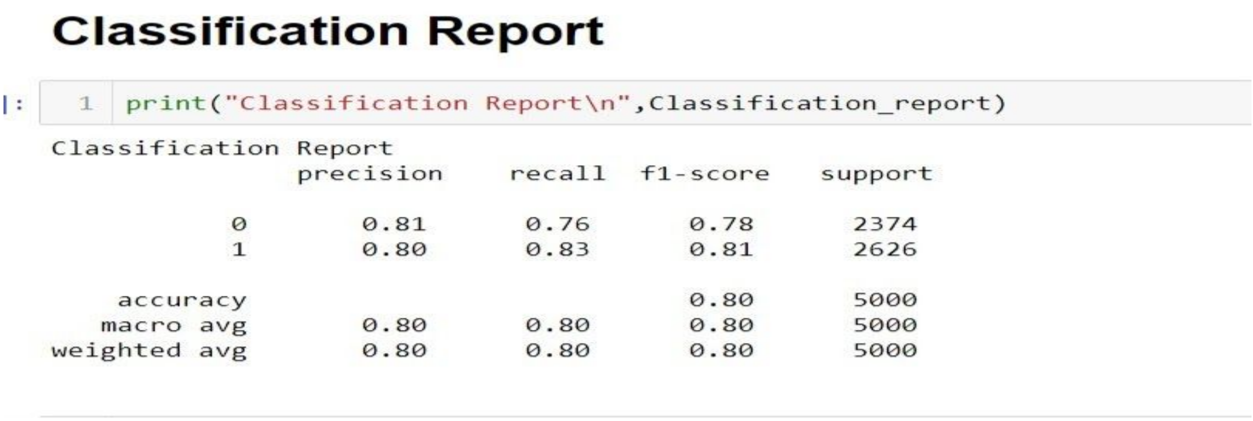
Bi-LSTM Model for Dataset with 2 Labels

In this model, we are first cleansing the dataset using data preprocessing step. After this we are padding sentences so that all sentences have equal length.

The description of model is as follows:
Embedding Layer (64,input_length=Length of sentence) -> Dropout with rate of 0.5 -> Bi-Directional LSTM Layer with 128 neurons -> Dense Layer with 64 neurons having 'relu' activation -> Dense layer with 1 neuron giving output between 0 and 1 with 'sigmoid' activation.
We are using 'Adam' optimizer for optimizing the model.

Observations of the model :
The model gives train accuracy of 95% and test accuracy around 80%. We are using a total of 10,000 sentences. It takes time around 5 hours to test the model. It is difficult to make iterations in the model .

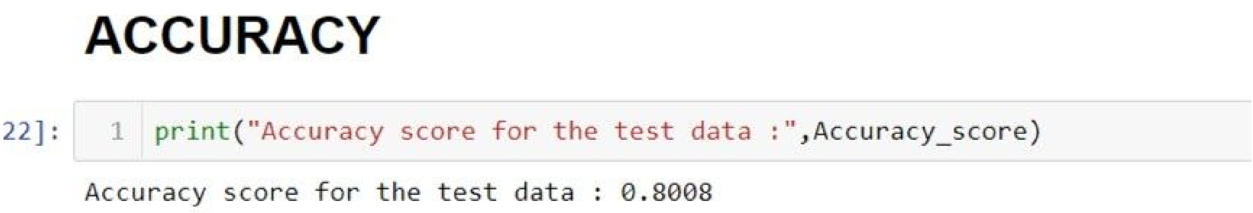




Classification Report

```
print("Classification Report\n",Classification_report)
```

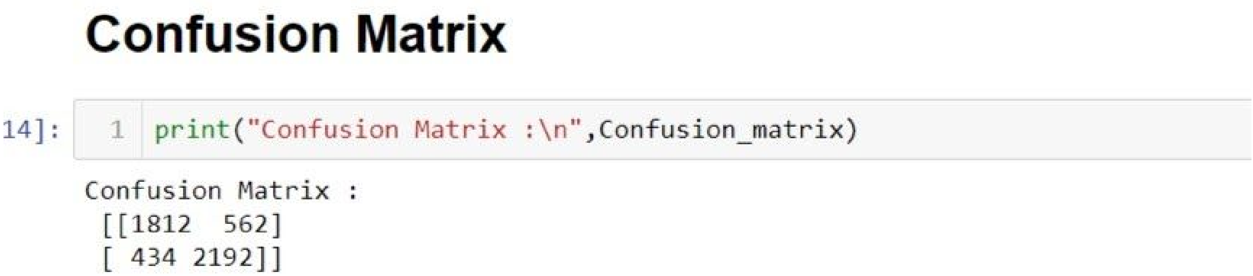
```
Classification Report
               precision    recall  f1-score   support

           0       0.81      0.76      0.78      2374
           1       0.80      0.83      0.81      2626

    accuracy                           0.80      5000
   macro avg       0.80      0.80      0.80      5000
weighted avg       0.80      0.80      0.80      5000
```



ACCURACY

```
print("Accuracy score for the test data :",Accuracy_score)
```

```
Accuracy score for the test data : 0.8008
```



Confusion Matrix

```
print("Confusion Matrix :\n",Confusion_matrix)
```

```
Confusion Matrix :
 [[1812  562]
 [ 434 2192]]
```

Error Analysis :

The model gave correct answers for sentences like:

- This is a nice car.
- The toy is good.
- This is not a nice movie.

The model gave incorrect answers for sentences:

- This is hardly a good movie. -> The overall sentiment should be negative but it is predicting as positive sentiment
- This is not a good movie. -> The overall sentiment should be negative but it is predicting as positive sentiment

After analysing the output of sentences we found out that the model we have developed is not deep enough. Also we need to train the model on more number of sentences than we already trained.

Naive Bayes Model for Dataset with 2 Labels

In this model, we are first cleansing the dataset using data preprocessing step.

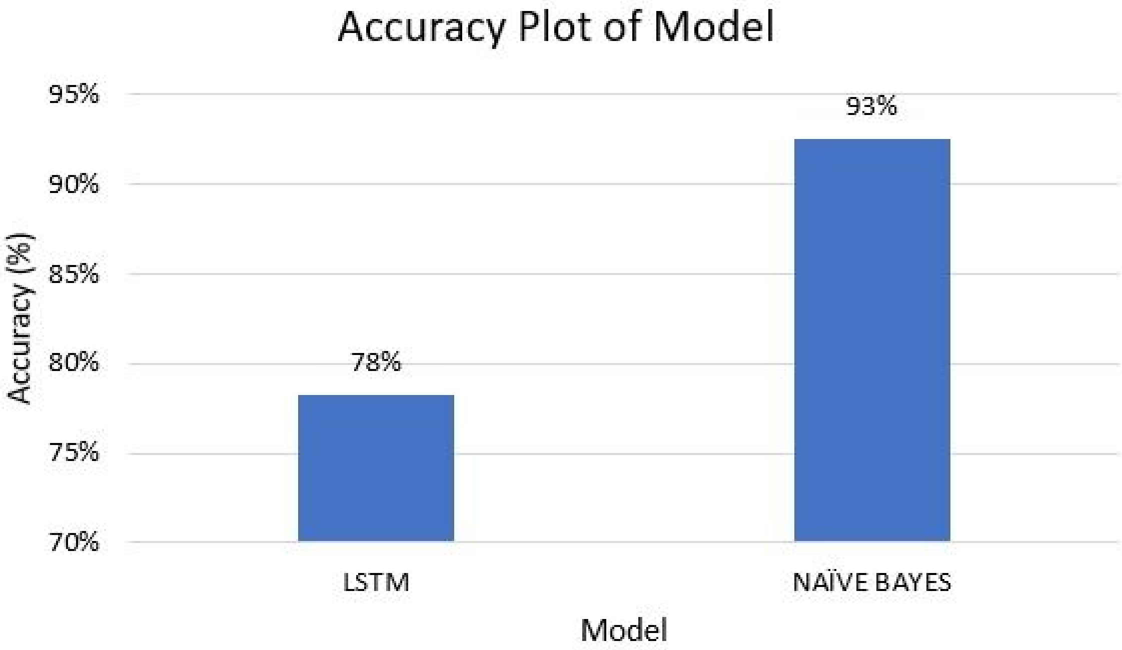
The description of model is as follows:

The words in the sentences are converted to vectors. After that the vectors are passed to the Naive Bayes model of the sklearn.

Observations of the model :

The train accuracy of the model is 95% and test accuracy of the model is 93%. The Naive Bayes model is better than the Bi-LSTM model in this case. This is maybe because the Naive Bayes classifier takes less number of features into account.

	POSITIVE	NEGATIVE
TRUE	2024	340
FALSE	437	2198



Error Analysis :

The model gave correct answers for sentences like:

- Amsterdam is a beautiful city.
- This is hardly a good movie.

The model gave incorrect answers for sentences:

- This is cheap product but the performance is not best -> The overall sentiment should be positive but it is predicting as negative sentiment

  This is the good accuracy for the model.

Bi-LSTM Model for Dataset with 5 Labels

In this model, we are first cleansing the dataset using data preprocessing step. After this we are padding sentences so that all sentences have equal length.

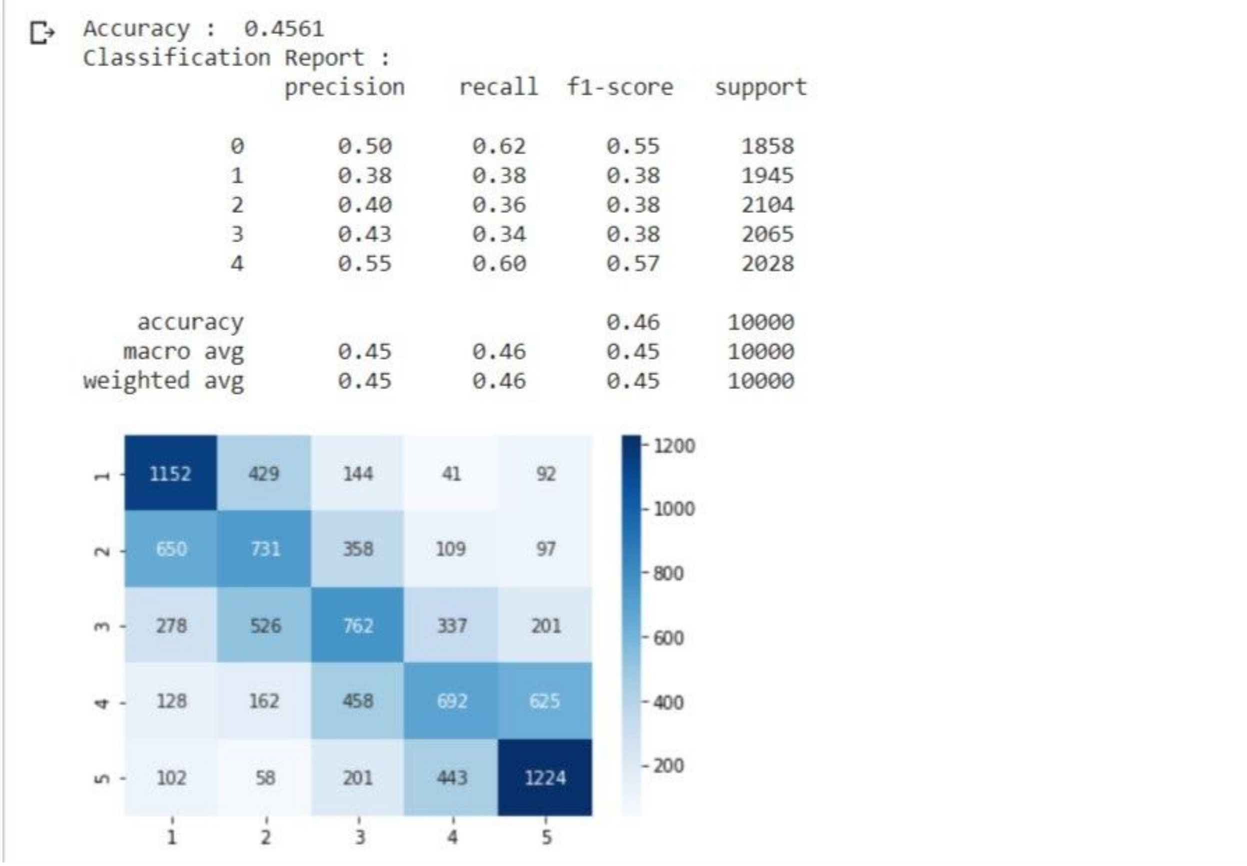
The description of model is as follows:

Embedding Layer (64,input_length=Length of sentence) -> Dropout with rate of 0.5 -> Bi-Directional LSTM Layer with 128 neurons -> Dense Layer with 64 neurons having 'relu' activation -> Dense layer with 5 neuron with 'softmax" activation.

We are using 'Adam' optimizer for optimizing the model.

Observations of the model :

The model gives train accuracy of 55% and test accuracy around 45%. We are using a total of 30,000 sentences. It takes time around 5 hours to test the model. It is difficult to make iterations in the model .



```
Accuracy :  0.4561
Classification Report :
              precision    recall  f1-score   support

           0       0.50      0.62      0.55      1858
           1       0.38      0.38      0.38      1945
           2       0.40      0.36      0.38      2104
           3       0.43      0.34      0.38      2065
           4       0.55      0.60      0.57      2028

    accuracy                           0.46     10000
   macro avg       0.45      0.46      0.45     10000
weighted avg       0.45      0.46      0.45     10000
```

	1	2	3	4	5
1	1152	429	144	41	92
2	650	731	358	109	97
3	278	526	762	337	201
4	128	162	458	692	625
5	102	58	201	443	1224

Error Analysis :

The accuracy of the model is very low because the sentences which are labelled as 4 stars are predicted with 3 , 4 or 5 stars. If the sentence contains all words having positive sentiment then it would predict 5. If the sentence contains all words having negative sentiment then it would predict 1. It is difficult for human to label sentence from 1 to 5 stars after reading it. Therefore we are getting less accuracy for the model.

Intensity (Fine Grained Sentiment Analysis)

In this model, we are first cleansing the dataset using data preprocessing step. We have implemented the model using the paper published by the Sir. This is an algorithmic model and does not learn from past experience.

The description of model is as follows:

For fine-grained intensity analysis we are using a dataset which contains reviews which are divided into 5 stars where the 1 star shows highly negative sentiment and the 5 star shows high positive sentiment.We are storing these sentiments in 5 lists which are sup_pos_sen(stores 5 star rating reviews), pos_sen(stores 4 star reviews), neu_sen(stores 3 star reviews), neg_sen(stores 2 star reviews) and sup_neg_sent(which stores 1 star rating reviews).Now the words present in the processed test data is matched with each of the words present in the 5 lists and their count is stored in the 5 count variables.These count variables are used to calculate the intensity of the sentiment.

Observations and Error Analysis :

Review	Intensity
The product is amazing and surprising	**1.0**
The product is nice overall but not that comfortable	**0.60**
The kindle is great and makes my reading comfortable	**0.00**
The product is definitely is not that good	**-0.09**
I want my money back, it's not original	**-1.00**

For positive sentences the intensity should be greater than 0. For negative sentences the intensity should be less than 0.

The intensity of the sentence “The kindle is great and makes my reading comfortable” should be more than 0 but it is equal to 0. Many sentences give satisfactory intensity output.

References :

1) https://www.researchgate.net/publication/261453417_Sentiment_Analysis
2) https://monkeylearn.com/sentiment-analysis/
3) https://medium.com/analytics-vidhya/simplifying-social-media-sentiment-analysis-using-vader-in-python-f9e6ec6fc52f
4) https://www.aclweb.org/anthology/D15-1300.pdf