NAME: Venkata Sai Siva Pruthvi. Ustepalle **S-ID: 800957126**

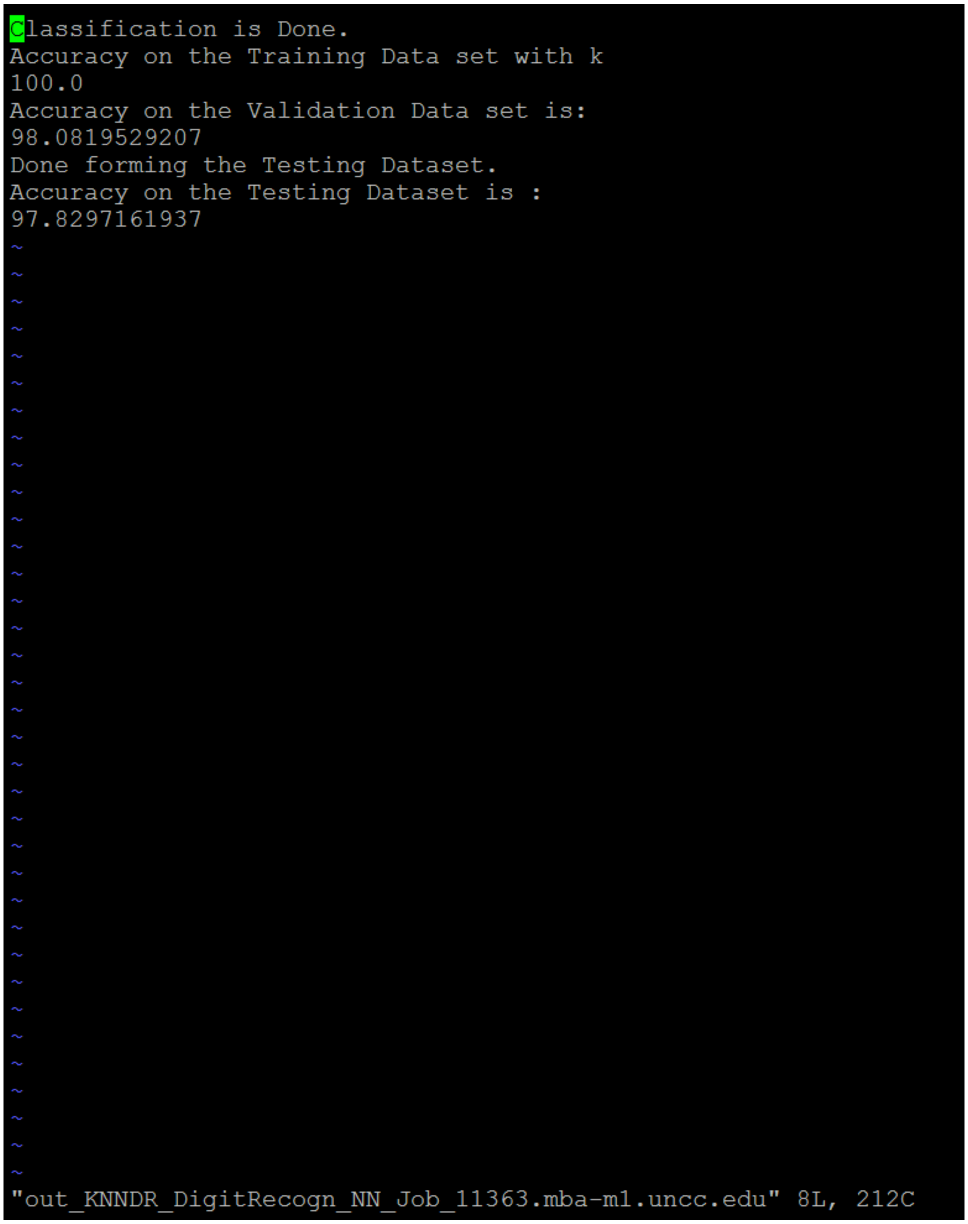
(Worked together with Chaitanya Sri Krishna Lolla)

Digit Recognition dataset

- RESULTS:

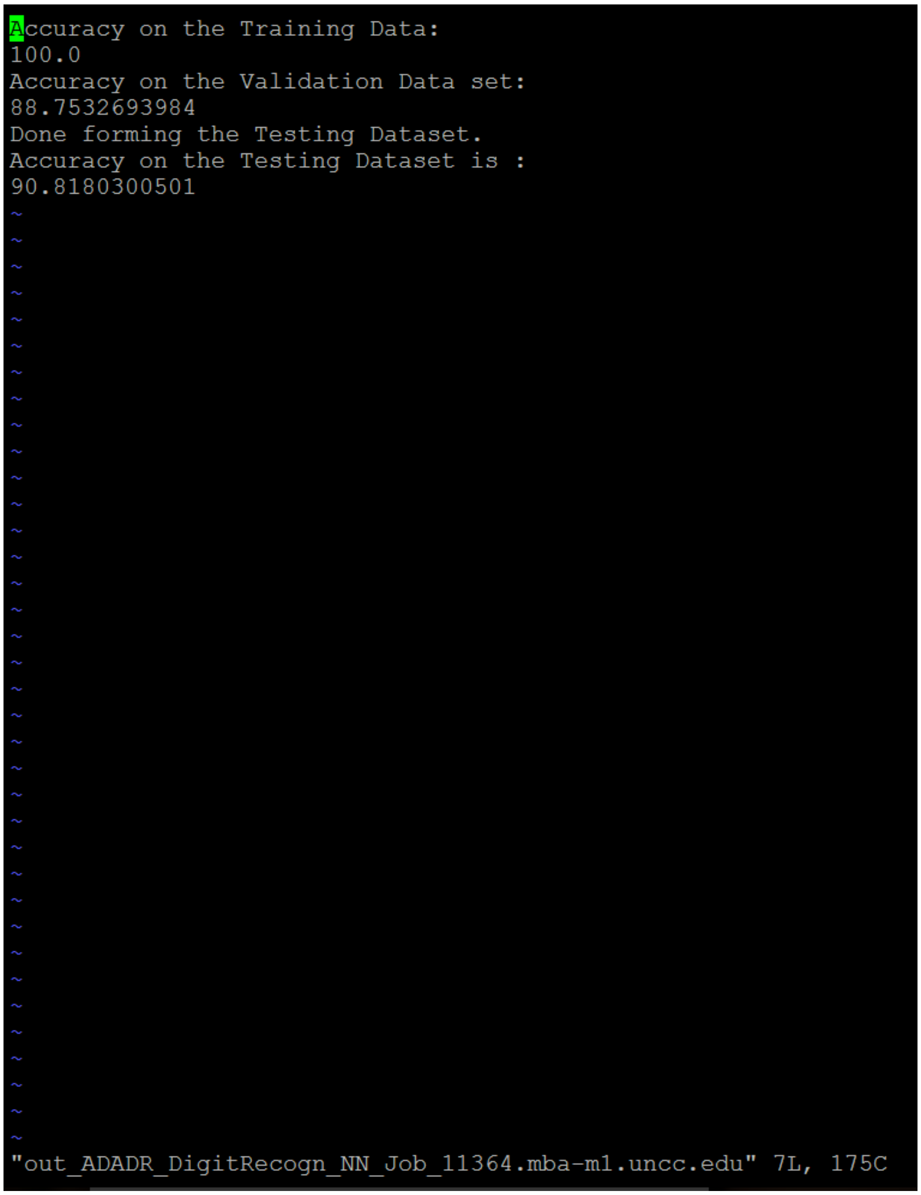
Splitting the given data into 2 parts to perform cross validation (as a step for pruning the data). Training on anything below 60% for cross validation which is resulting in a 100% accuracy on training set and 97% accuracy on the validation data. This split worked better for both KNN and Ada Boosting classifiers.

KNN Classifier



```
Classification is Done.
Accuracy on the Training Data set with k
100.0
Accuracy on the Validation Data set is:
98.0819529207
Done forming the Testing Dataset.
Accuracy on the Testing Dataset is :
97.8297161937
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"out_KNNDR_DigitRecogn_NN_Job_11363.mba-m1.uncc.edu" 8L, 212C
```

ADA Boost classifier



```
Accuracy on the Training Data:
100.0
Accuracy on the Validation Data set:
88.7532693984
Done forming the Testing Dataset.
Accuracy on the Testing Dataset is :
90.8180300501
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"out_ADADR_DigitRecogn_NN_Job_11364.mba-m1.uncc.edu" 7L, 175C
```

- Trying with various parameters did not give any better result than the default parameters, which are really working good for small data sets.
- Using different indexing structures like KD Tree didn't reflect any better classifier than the default one. It reduced the accuracy by 0.8 percent.
- For a small data set like Digit Recognition any higher amount of n_estimators are undermining the accuracy. And any tradeoff between n_estimators and learning rate is also not working good. With all the default values in the classifier and n_estimators=100 accuracy of the classifier is observed to be better than any other parameters.

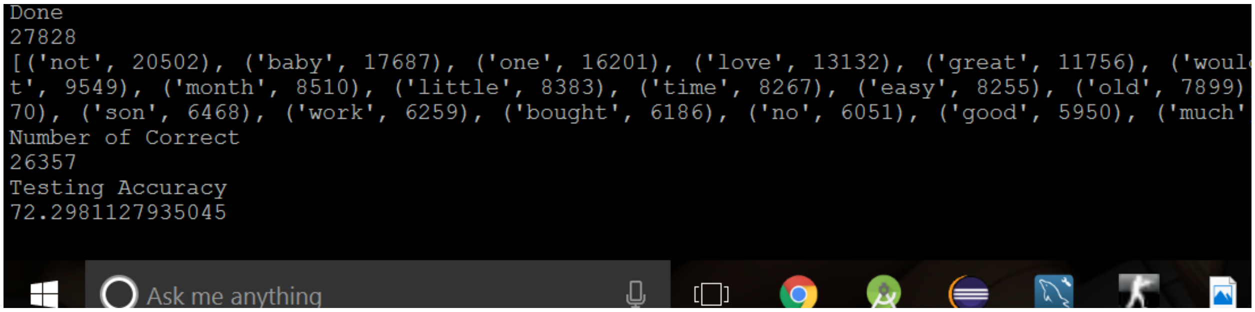
Amazon Review dataset

Result :

Using nltk library services to lemmatize, tokenize the reviews along with removing the punctuation from the review texts as a part of preprocessing steps as in done in the previous assignment. Pre-processing the text and making it into two sets of positive and negative reviews. Picking top 5000 of the most commonly occurring words and assigning them numerical values to pass as parameters for the decision tree classifier method. Used TF-IDF to build a matrix of numerical representation of the words in each sentence. And the same procedure is repeated on the test dataset and the accuracy is measured.

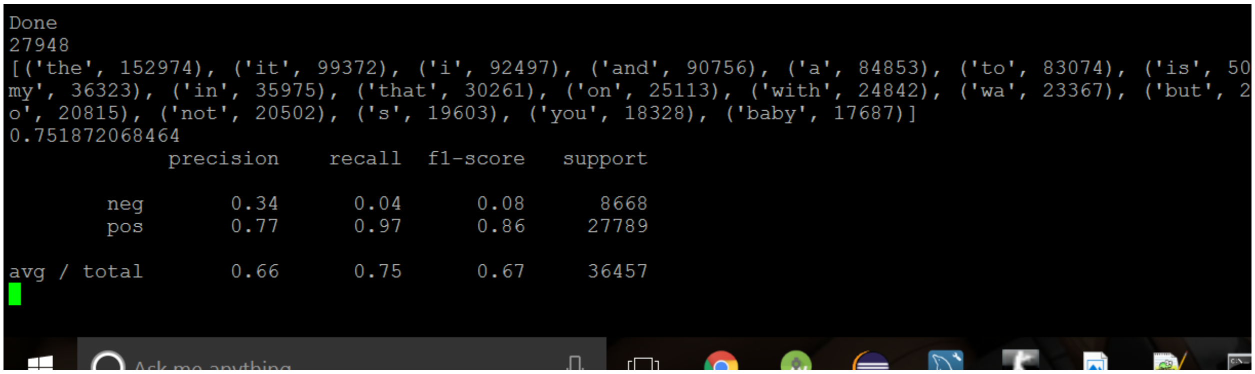
- As no combination of parameters passed to the classifiers is resulting in a better model I used default parameters for both models.

KNN Classifier



```
Done
27828
[('not', 20502), ('baby', 17687), ('one', 16201), ('love', 13132), ('great', 11756), ('woul
t', 9549), ('month', 8510), ('little', 8383), ('time', 8267), ('easy', 8255), ('old', 7899)
70), ('son', 6468), ('work', 6259), ('bought', 6186), ('no', 6051), ('good', 5950), ('much'
Number of Correct
26357
Testing Accuracy
72.2981127935045
```

ADA Boost



```
Done
27948
[('the', 152974), ('it', 99372), ('i', 92497), ('and', 90756), ('a', 84853), ('to', 83074), ('is', 50
my', 36323), ('in', 35975), ('that', 30261), ('on', 25113), ('with', 24842), ('wa', 23367), ('but', 2
o', 20815), ('not', 20502), ('s', 19603), ('you', 18328), ('baby', 17687)]
0.751872068464
             precision    recall  f1-score   support

        neg       0.34      0.04      0.08      8668
        pos       0.77      0.97      0.86     27789

avg / total       0.66      0.75      0.67     36457
```