Analysis File

Tools used:

1) NLTK
2) Pandas
3) Matplotlib
4) Pickle

Pre-Processing Steps:

1) Have used porter stemmer to perform stemming of the documents and the query.

2) Removed all the punctuations across all the docs and replaced it with ‘ ’ in order to handle cases for numbers like 5,000 .

3) Performed num2words() to convert the digits to numbers.

4) All the stop words were removed.

Part 1 Features Selection

A) Features Selection using TF-IDF

Methodology

- Divide Corpus into Test and Training set into the following splits :

Train	Test
80	20
70	30
50	50

- Obtain all unique terms from the training docs and preserve their respective classes.

- For each (Term,class) pair calculate logarithmic tf-idf scores.
- Divide all these pairs according to their respective class labels.
- Sort all these terms in the decreasing order of their tf-idf score in each class

- Select 50% terms from each class and obtain set union of all these terms of each class.

B) Features Selection using Mutual Information

<u>**Methodology**</u>

- Divide the corpus into Test and Train docs as performed earlier in TF-IDF feature selection. Obtain all unique terms from the training docs and preserve their respective classes.
- For each (term, class) pair calculate Mutual Information using the below formula:

$$\mathrm{I}(X;Y) = \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} p_{(X,Y)}(x,y) \log\left(\frac{p_{(X,Y)}(x,y)}{p_X(x)\,p_Y(y)}\right), \quad \text{(Eq.1)}$$

- Divide all these (term ,class)pairs according to the class labels and sort in decreasing order of MI value.
- Select 1/5 terms from each class and obtain set union of all these terms of each class.

Part 2 Text Classification

A) Naïve Byes

Training Phase

- Calculate Probability(term|Class) for all the terms selected after feature selection step.

Testing Phase

- Calculate Probability(term|Class) for all the terms of current testing document.
- Consider log likelihood and perform smoothing by adding 1 to numerator and |Vocab| to the denominator.
- Select the class for maximum sum of log likelihood values. (Please note that prior of each class is same therefore considering prior will have no effect on our result)

B) K Nearest Neighbors

Pre Processing Phase

- Obtain feature vectors for each test and train document. Arrange all testing document vectors into 2D array. Similarly obtain 2D array for training documents.
- Training Phase
- No specific training required.

Testing Phase

- Perform matrix multiplication of Testing and Training matrices. The resulting matrix will be of order (number of test docs X number of training docs). Each row represent a test document with column values as the similarity measure of(test,train) pair.

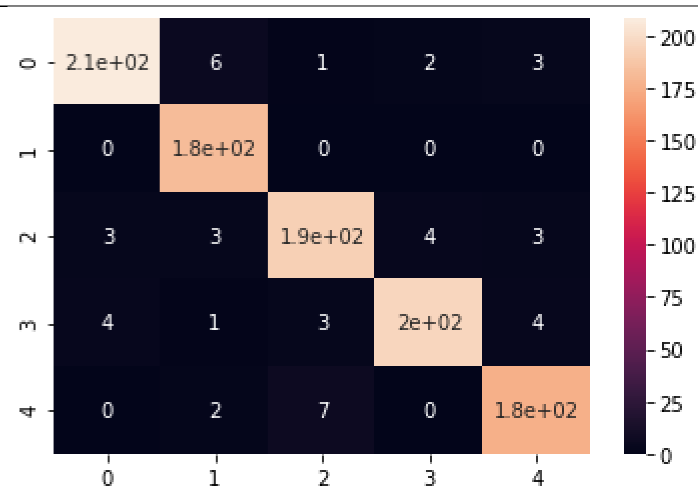
- Divide each row by magnitude of the respective test doc and each value of the column in a row by corresponding training doc's magnitude.

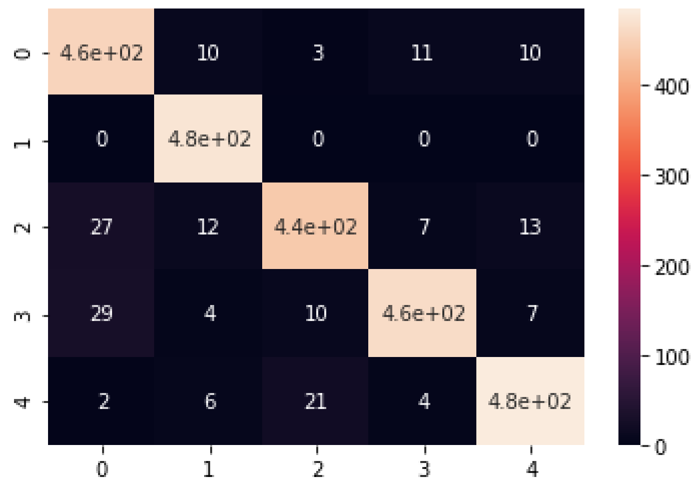
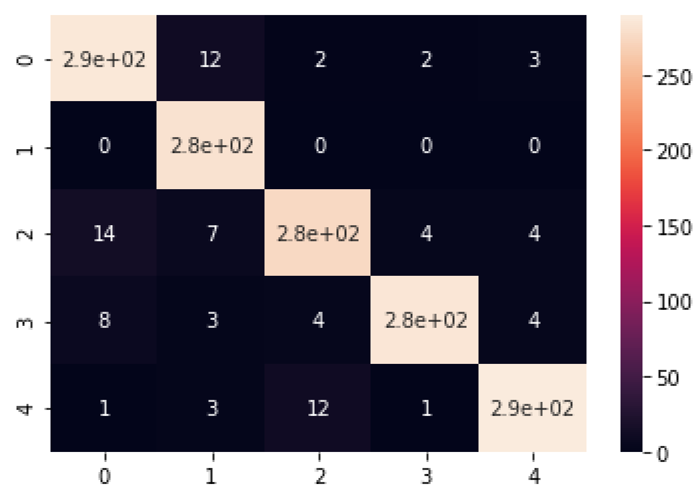
- Sort each row of resulting matrix in decreasing order.

- For each row (a test doc) taking voting of top K training vectors selected after sorting step for deciding predicted class label for that test doc.

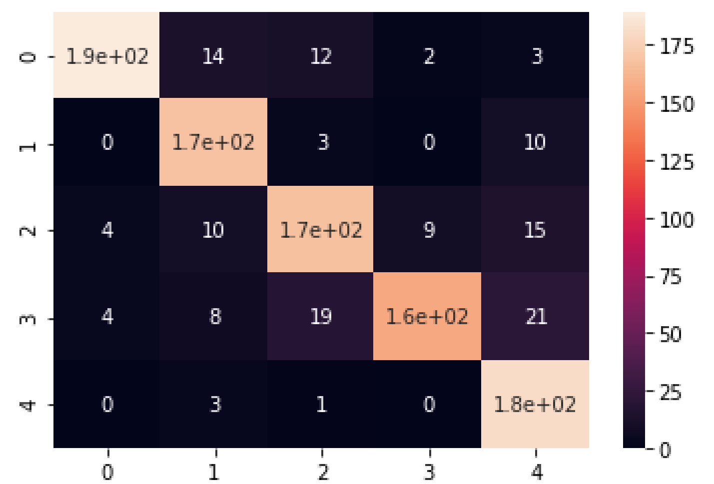
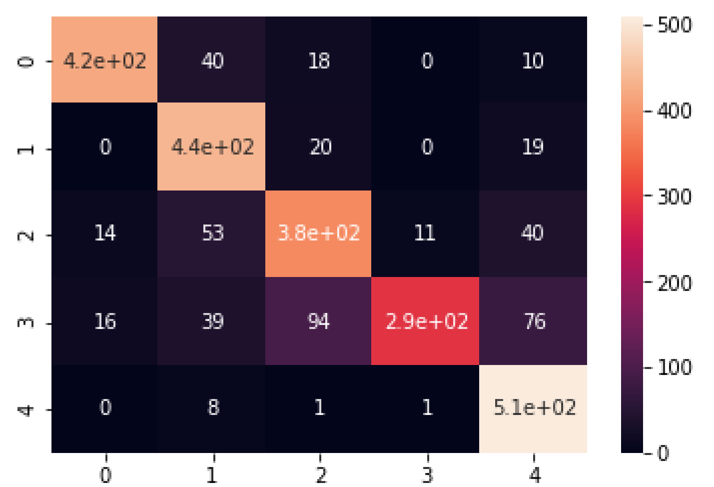
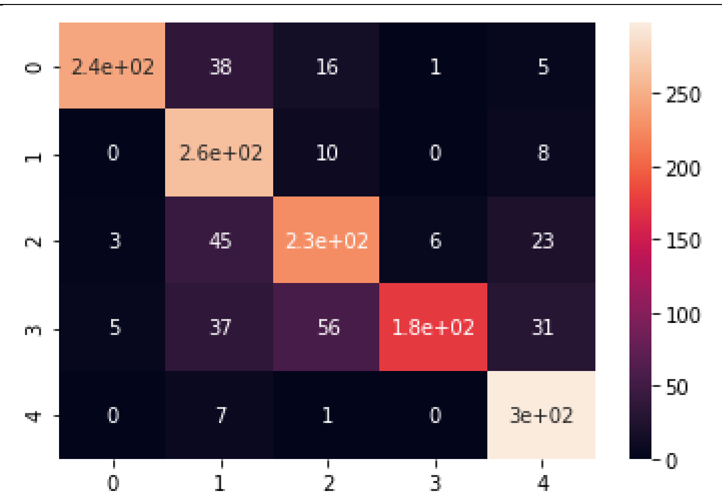
Results Summary

<u>TF-IDF and Naïve Byes Classifier</u>

Split (Train-Test)	**Accuracy**	**Confusion Matrix**
80 : 20	95.56%	

50:50	92.96%	
70:30	94.47	

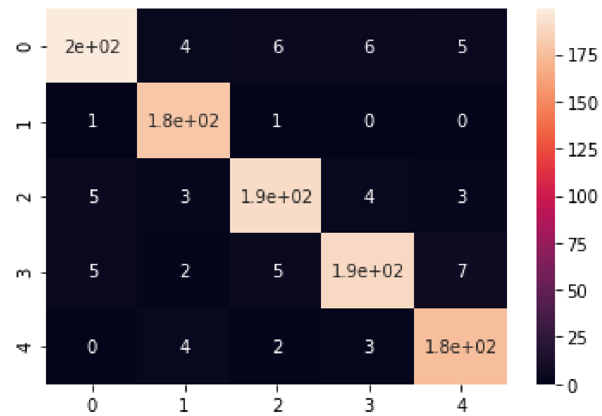
Mutual Information (MI) and Naïve Byes

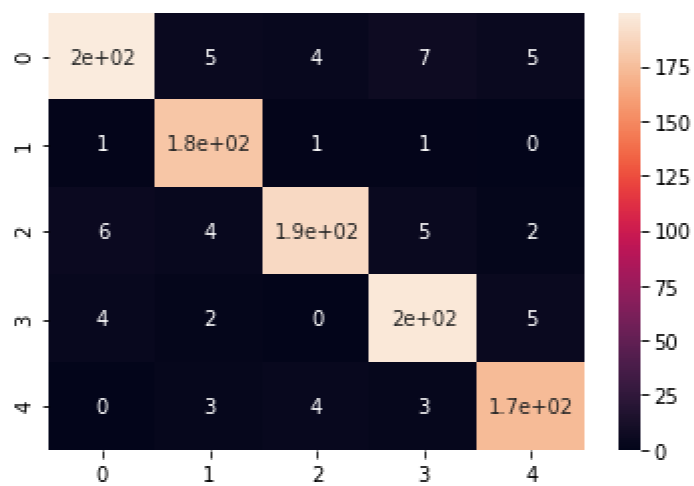
Split (Train: Test)	Accuracy	Confusion Matrix
80:20	86.3%	
50:50	81.6%	
70:30	80.54%	

TF-IDF Feature Selection and KNN

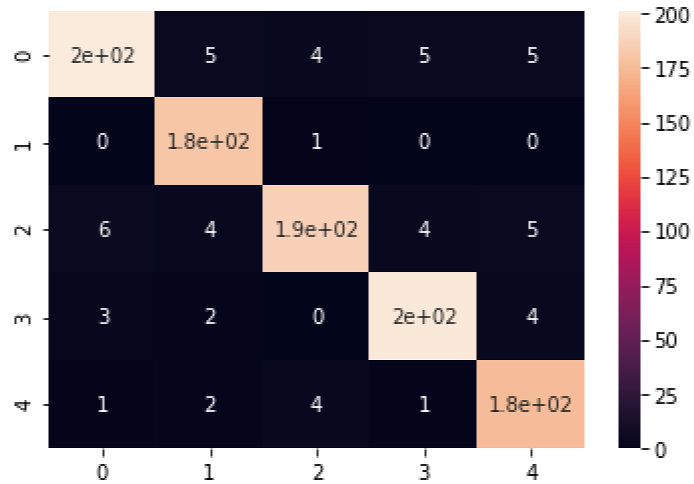
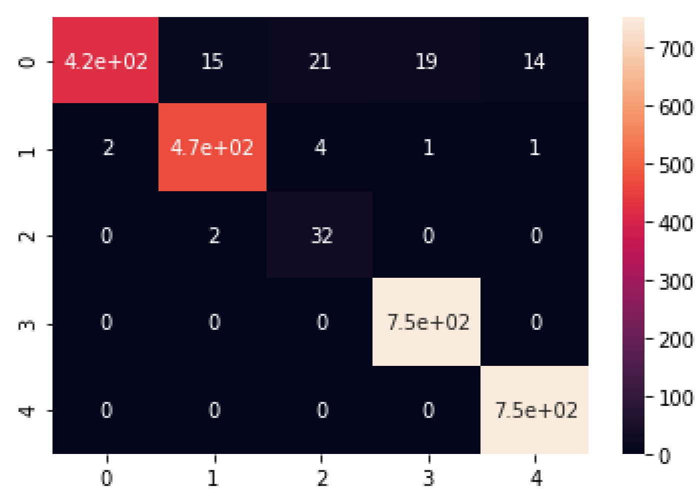
<table>
<tr><th>Split (Train:Test)</th><th>Accuracy</th><th>Confusion Matrix</th></tr>
<tr><td>80:20</td><td>K=1 93.4%

K=3 93.8%

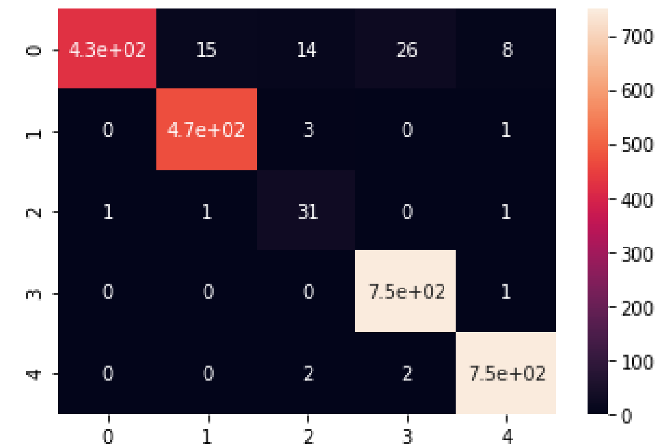
K=5 94.4%</td><td>K=1


K=3


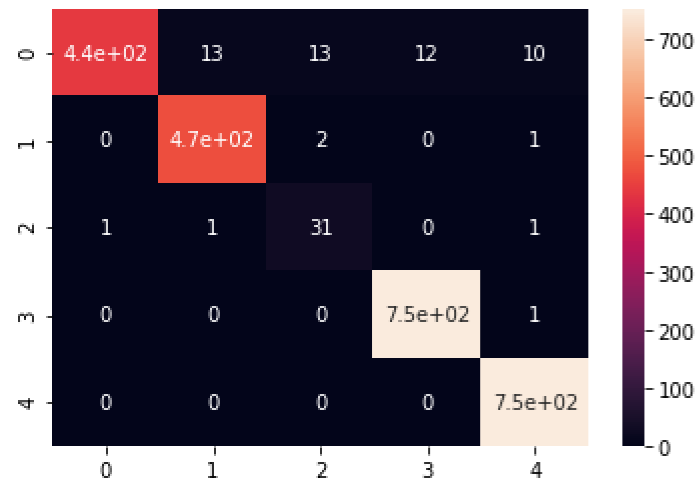
</td></tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td>K=5

</td>
</tr>
<tr>
<td>50:50</td>
<td>K=1 92.1%
K=3 93.04%
K=5 94.6%</td>
<td>K=1


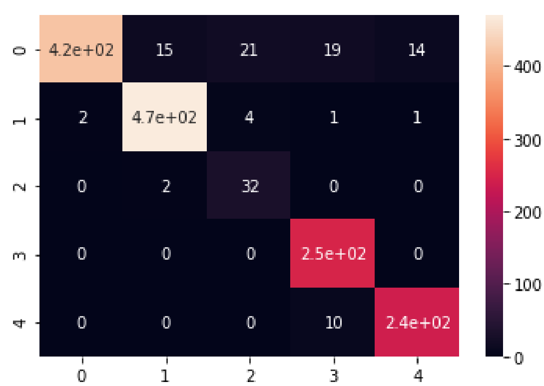
K=3</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td>


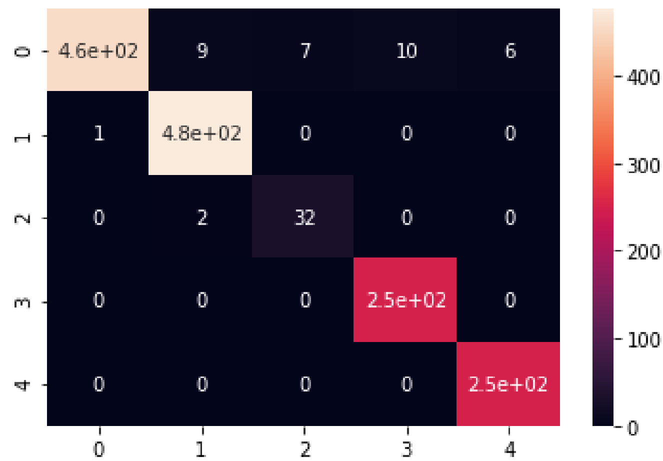
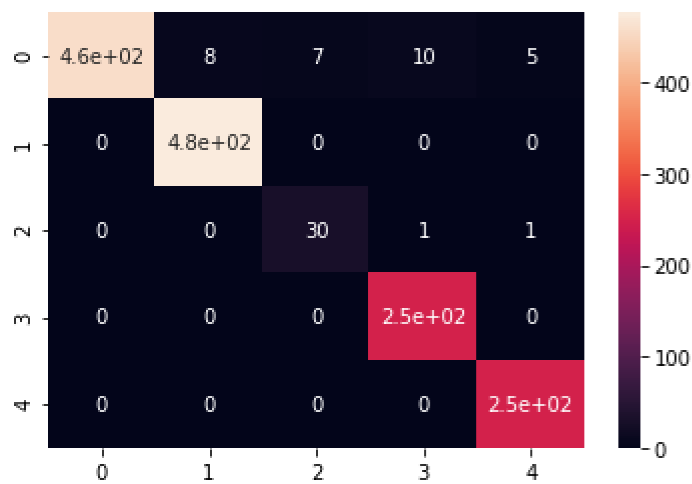
K=5



</td>
</tr>
<tr>
<td>70:30</td>
<td>K=1 92.6%
K=3 93.4%
K=5 94.0%</td>
<td>
K=1



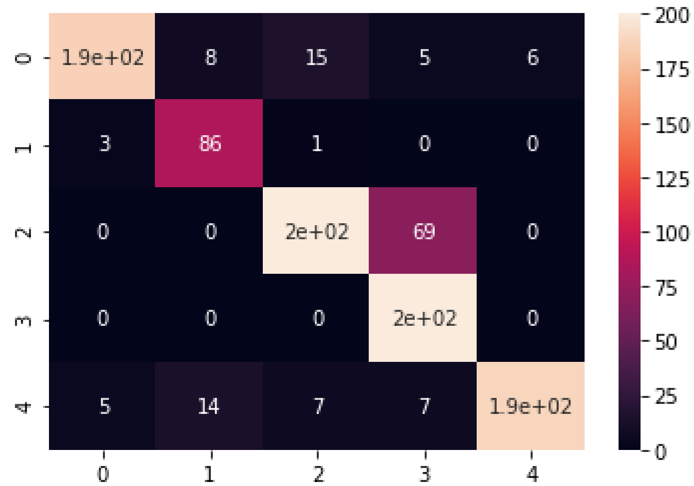
</td>
</tr>
</table>

		K=3  K=5 

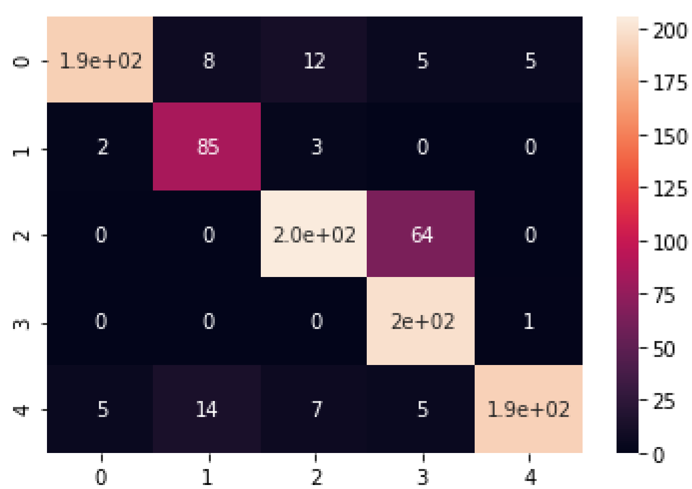
Mutual Information and KNN Classifier

Split (Train:Test)	Accuracy	Confusion Matrix
80:20	K=1 87.75%	K=1

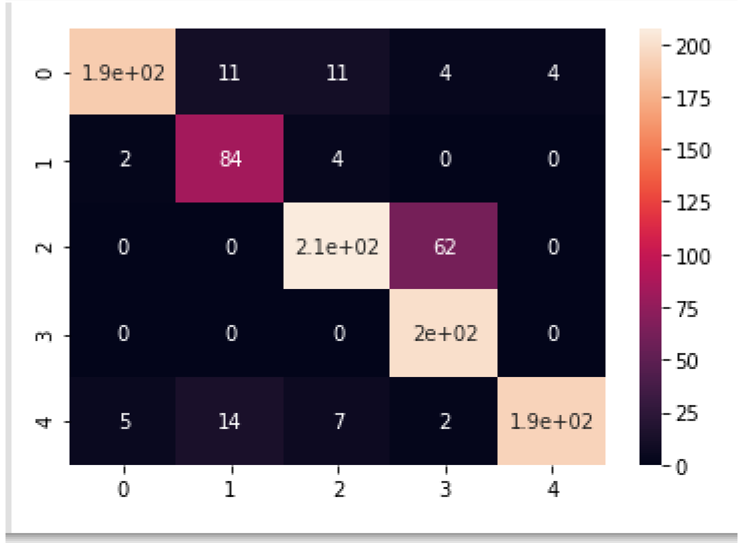
<table>
<tr>
<td></td>
<td>K=3 88.6%

K=5 88.4%</td>
<td>


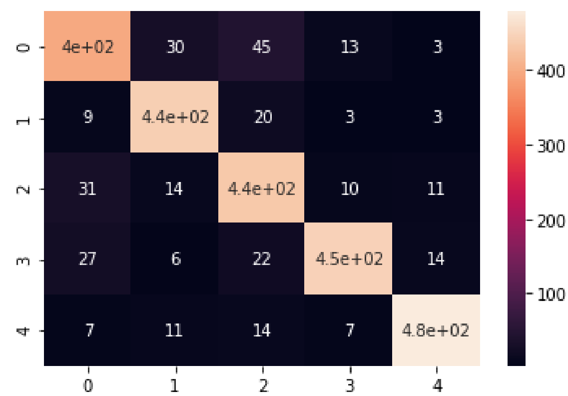
K=3

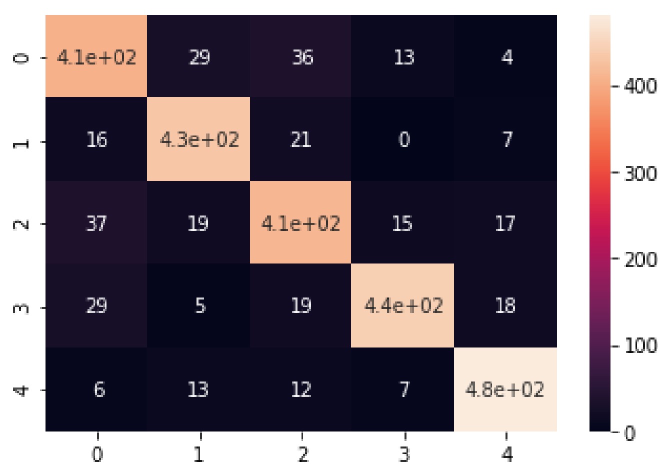


K=5
</td>
</tr>
</table>

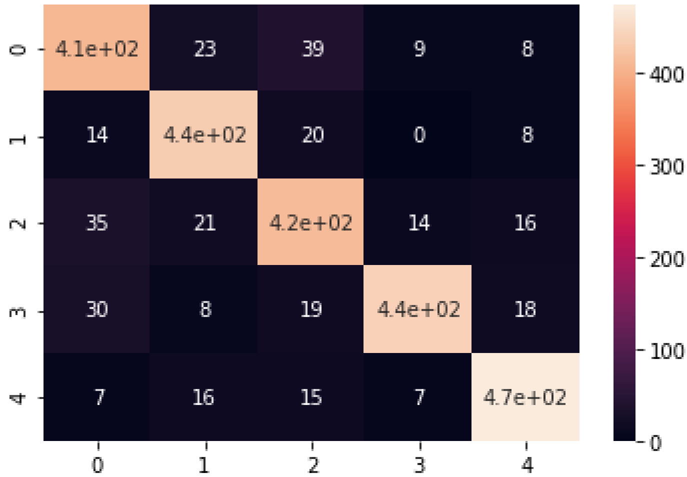
<table>
<tr>
<td></td>
<td></td>
<td>
</td>
</tr>
<tr>
<td>50:50</td>
<td>K=1 88%

K=3 87.8%

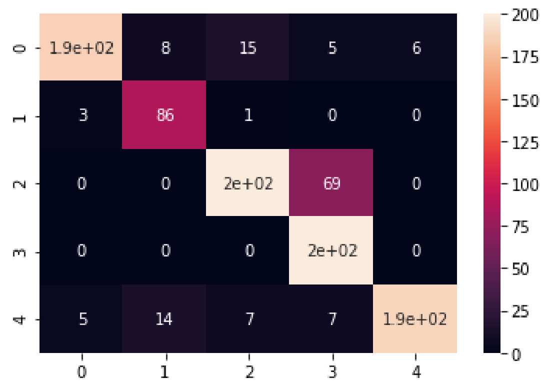
K=5 87.5%</td>
<td>K=1


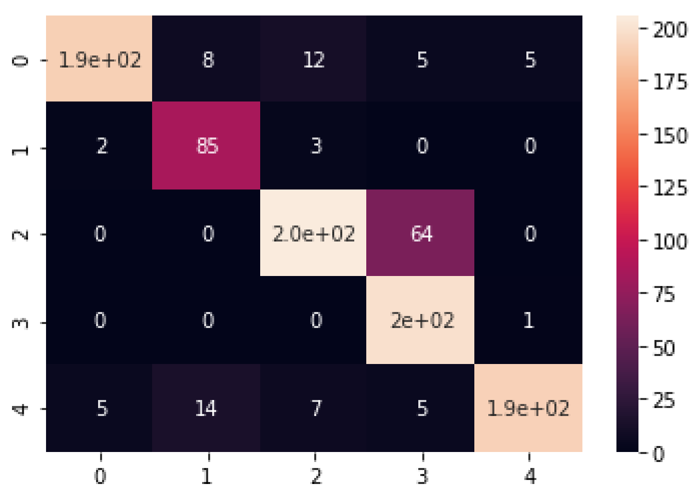
K=3


K=5</td>
</tr>
</table>

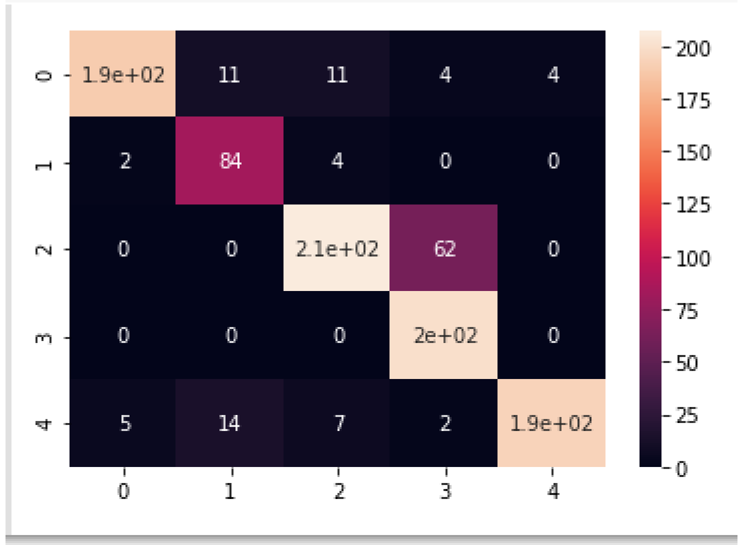
<table>
<tr><td></td><td></td><td>
</td></tr>
<tr><td>70:30</td><td>K=1 87.57%

K=3 88.66%

K=5 88.91%</td><td>K=1


K=3


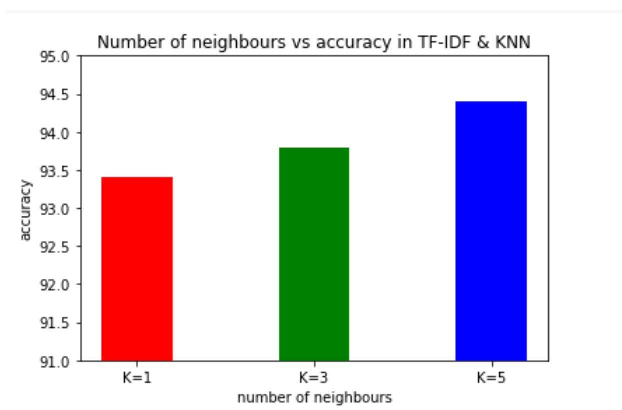
K=5</td></tr>
</table>

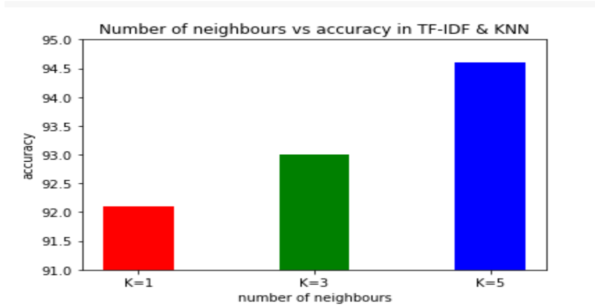
Analysis Performed

<u>Accuracy vs Variation in k of KNN classifier using TF-IDF Feature selection</u>

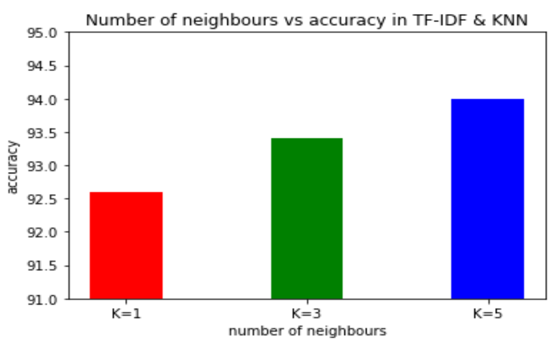
80:20 Split



50:50 Split

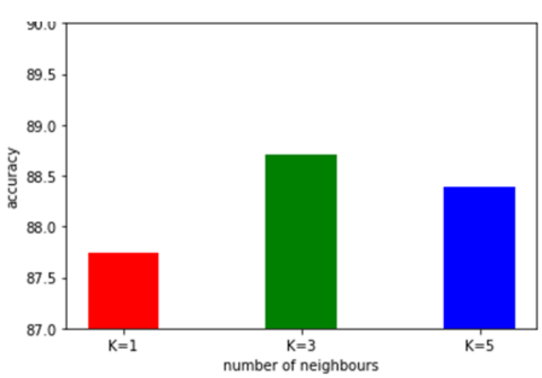


70:30 Split

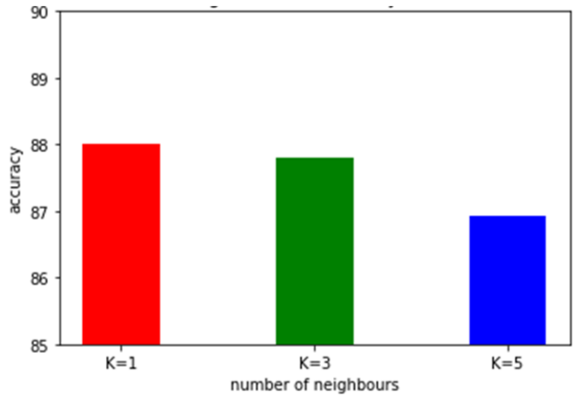


<u>Accuracy vs Variation in k of KNN classifier using MI Feature selection</u>

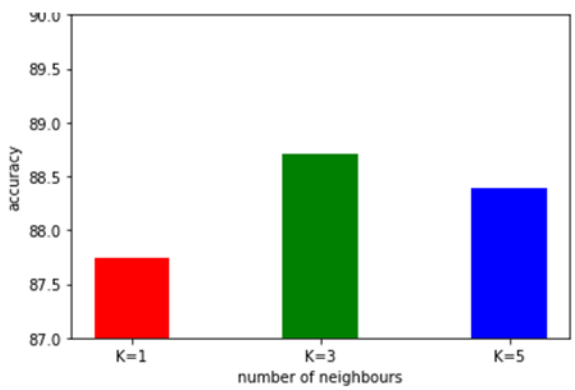
80:20 Split



50:50 Split

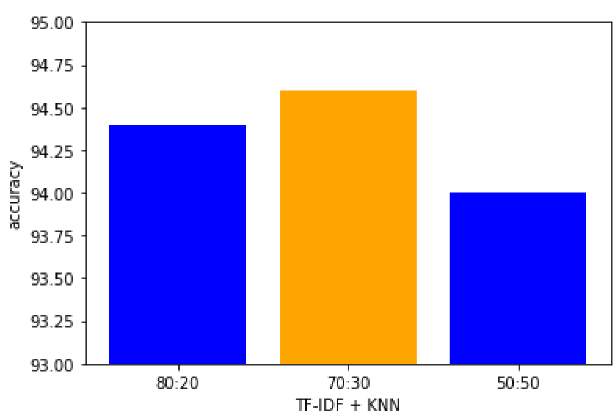


70:30 Split

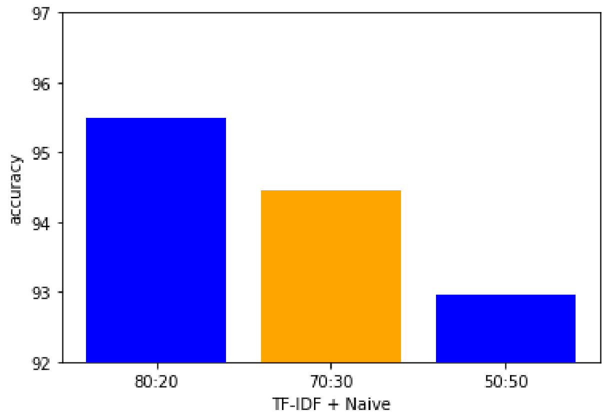


<u>Performance of (classifier + feature selection method) vs Accuracy</u>

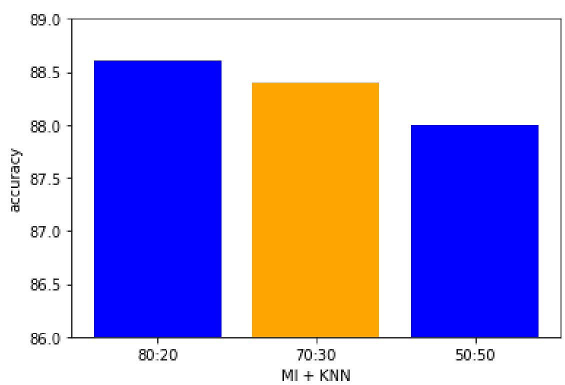
<u>TF_IDF + KNN</u>



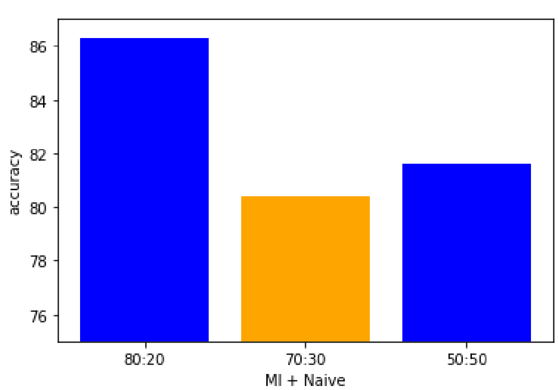
TF_IDF + Naïve



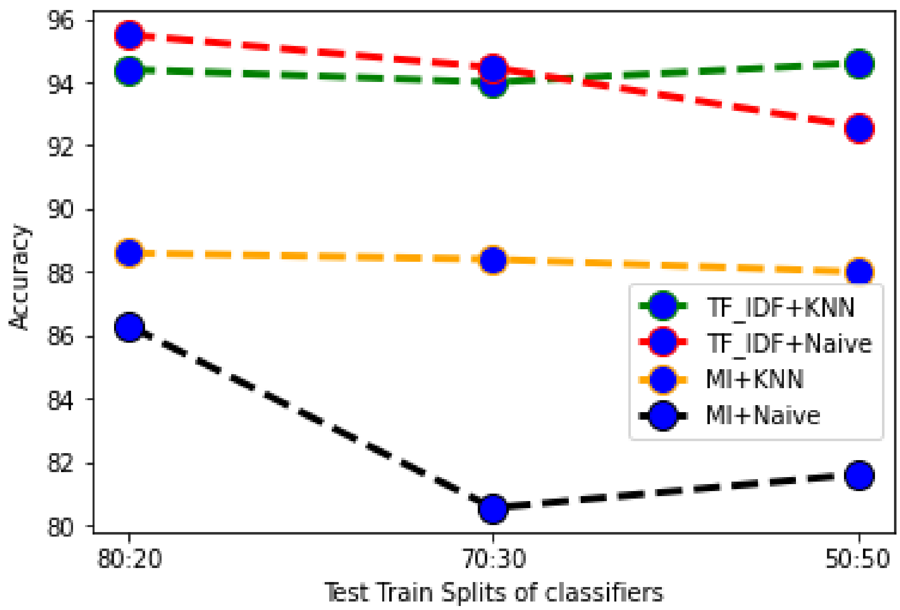
MI +KNN



MI + Naïve



Evaluating Performance of classifiers



Inferences Drawn :

- TF_IDF + Naïve byes outperforms every other combination of features selector and classifier combination at 80:20 and 70:30 splits. However TF_IDF + KNN proved to give highest accuracy at 50:50 split.
- Since the Data is shuffled before splitting into test and train therefore minor variation from the above inference might vary. But TF_IDF + Naive and TF_IDF + KNN are the best combination among four.
- Increase in the number of features selected using MI/TF_IDF increase the processing time.
- Selecting more number of features from each class increase the chances of noisy features getting selected for that class.
- Mutual information features selection technique is introducing more noisy features as compared to TF_IDF feature selection. Therefore MI + Naïve or MI+KNN is giving lesser accuracy than TF_IDF + Naive and TF_IDF + KNN.
- When the splitting is more towards the training docs , the accuracy of any of the four combinations is more.

- MI with 1/5 of total number of terms as the selected features gives optimal results whereas TF_IDF with ½ of total number of terms as the selected features gives optimal results.
- Increase in the number of neighbors in KNN may Increase/decrease the accuracy.

Other Inferences :

- Increase/decrease in accuracy is not very certain with variation of Test Train Splits. This may be due to some noisy features.
- Run time increases with increase in the number of training docs.