

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

Methodology

- 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:

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

- 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.

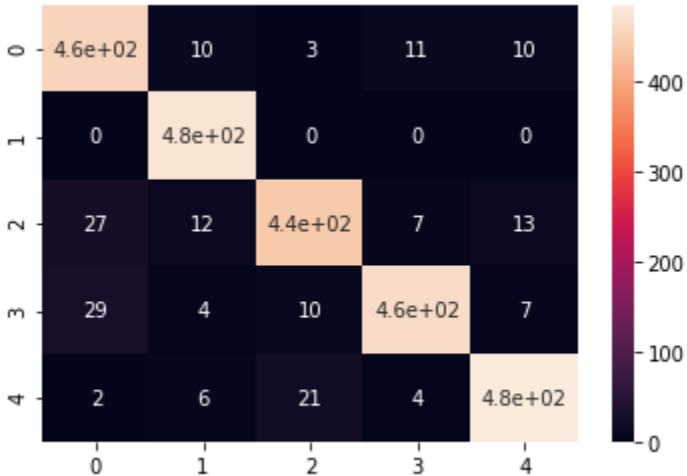
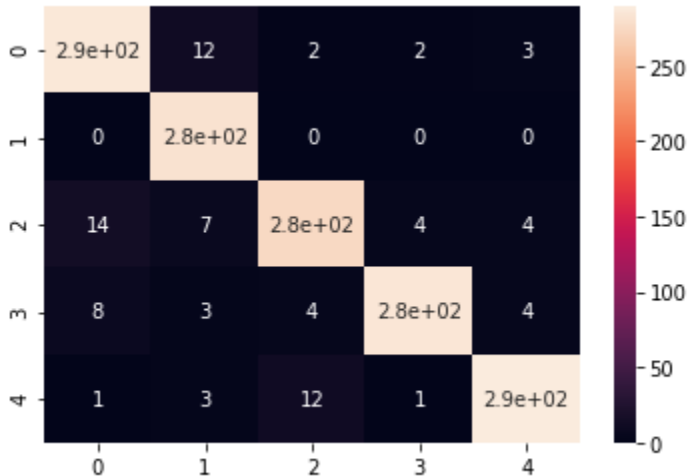
Testing Phase

- 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

TF-IDF and Naïve Byes Classifier

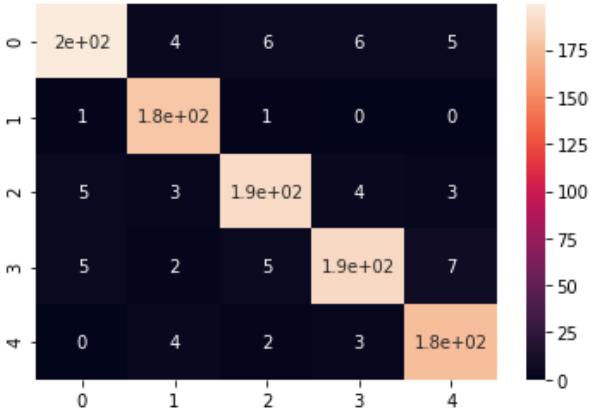
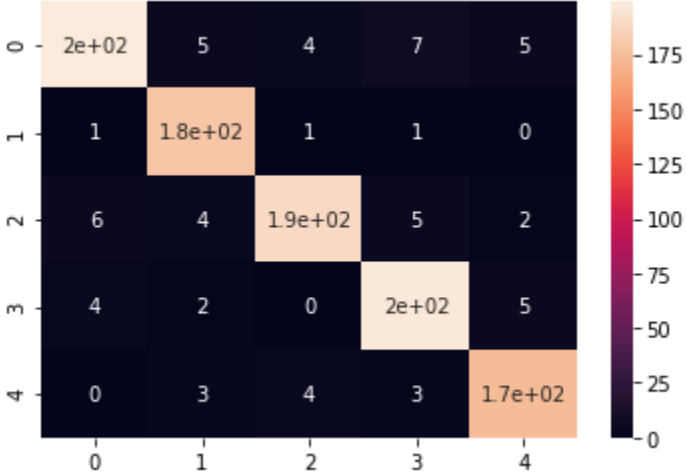
Split (Train-Test)	Accuracy	Confusion Matrix																																				
80 : 20	95.56%	<div><table><tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th></tr><tr><th>0</th><td>2.1e+02</td><td>6</td><td>1</td><td>2</td><td>3</td></tr><tr><th>1</th><td>0</td><td>1.8e+02</td><td>0</td><td>0</td><td>0</td></tr><tr><th>2</th><td>3</td><td>3</td><td>1.9e+02</td><td>4</td><td>3</td></tr><tr><th>3</th><td>4</td><td>1</td><td>3</td><td>2e+02</td><td>4</td></tr><tr><th>4</th><td>0</td><td>2</td><td>7</td><td>0</td><td>1.8e+02</td></tr></table></div>		0	1	2	3	4	0	2.1e+02	6	1	2	3	1	0	1.8e+02	0	0	0	2	3	3	1.9e+02	4	3	3	4	1	3	2e+02	4	4	0	2	7	0	1.8e+02
	0	1	2	3	4																																	
0	2.1e+02	6	1	2	3																																	
1	0	1.8e+02	0	0	0																																	
2	3	3	1.9e+02	4	3																																	
3	4	1	3	2e+02	4																																	
4	0	2	7	0	1.8e+02																																	

50:50	92.96%	 <p>Heatmap visualization for the 50:50 ratio. The matrix is 5x5, with rows and columns indexed 0 to 4. The color scale ranges from 0 (dark purple) to 400 (light orange). The diagonal elements are significantly higher than the off-diagonal elements, indicating a strong self-similarity or high values for the same categories.</p> <table><tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th></tr><tr><th>0</th><td>4.6e+02</td><td>10</td><td>3</td><td>11</td><td>10</td></tr><tr><th>1</th><td>0</td><td>4.8e+02</td><td>0</td><td>0</td><td>0</td></tr><tr><th>2</th><td>27</td><td>12</td><td>4.4e+02</td><td>7</td><td>13</td></tr><tr><th>3</th><td>29</td><td>4</td><td>10</td><td>4.6e+02</td><td>7</td></tr><tr><th>4</th><td>2</td><td>6</td><td>21</td><td>4</td><td>4.8e+02</td></tr></table>		0	1	2	3	4	0	4.6e+02	10	3	11	10	1	0	4.8e+02	0	0	0	2	27	12	4.4e+02	7	13	3	29	4	10	4.6e+02	7	4	2	6	21	4	4.8e+02
	0	1	2	3	4																																	
0	4.6e+02	10	3	11	10																																	
1	0	4.8e+02	0	0	0																																	
2	27	12	4.4e+02	7	13																																	
3	29	4	10	4.6e+02	7																																	
4	2	6	21	4	4.8e+02																																	
70:30	94.47	 <p>Heatmap visualization for the 70:30 ratio. The matrix is 5x5, with rows and columns indexed 0 to 4. The color scale ranges from 0 (dark purple) to 250 (light orange). The diagonal elements are significantly higher than the off-diagonal elements, indicating a strong self-similarity or high values for the same categories.</p> <table><tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th></tr><tr><th>0</th><td>2.9e+02</td><td>12</td><td>2</td><td>2</td><td>3</td></tr><tr><th>1</th><td>0</td><td>2.8e+02</td><td>0</td><td>0</td><td>0</td></tr><tr><th>2</th><td>14</td><td>7</td><td>2.8e+02</td><td>4</td><td>4</td></tr><tr><th>3</th><td>8</td><td>3</td><td>4</td><td>2.8e+02</td><td>4</td></tr><tr><th>4</th><td>1</td><td>3</td><td>12</td><td>1</td><td>2.9e+02</td></tr></table>		0	1	2	3	4	0	2.9e+02	12	2	2	3	1	0	2.8e+02	0	0	0	2	14	7	2.8e+02	4	4	3	8	3	4	2.8e+02	4	4	1	3	12	1	2.9e+02
	0	1	2	3	4																																	
0	2.9e+02	12	2	2	3																																	
1	0	2.8e+02	0	0	0																																	
2	14	7	2.8e+02	4	4																																	
3	8	3	4	2.8e+02	4																																	
4	1	3	12	1	2.9e+02																																	

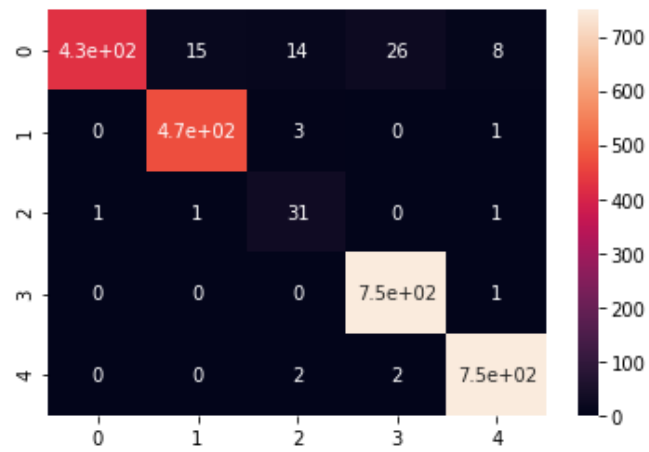
Mutual Information (MI) and Naïve Byes

Split (Train: Test)	Accuracy	Confusion Matrix																																				
80:20	86.3%	<table><tr><td>0</td><td>1.9e+02</td><td>14</td><td>12</td><td>2</td><td>3</td></tr><tr><td>1</td><td>0</td><td>1.7e+02</td><td>3</td><td>0</td><td>10</td></tr><tr><td>2</td><td>4</td><td>10</td><td>1.7e+02</td><td>9</td><td>15</td></tr><tr><td>3</td><td>4</td><td>8</td><td>19</td><td>1.6e+02</td><td>21</td></tr><tr><td>4</td><td>0</td><td>3</td><td>1</td><td>0</td><td>1.8e+02</td></tr><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr></table>	0	1.9e+02	14	12	2	3	1	0	1.7e+02	3	0	10	2	4	10	1.7e+02	9	15	3	4	8	19	1.6e+02	21	4	0	3	1	0	1.8e+02		0	1	2	3	4
0	1.9e+02	14	12	2	3																																	
1	0	1.7e+02	3	0	10																																	
2	4	10	1.7e+02	9	15																																	
3	4	8	19	1.6e+02	21																																	
4	0	3	1	0	1.8e+02																																	
	0	1	2	3	4																																	
50:50	81.6%	<table><tr><td>0</td><td>4.2e+02</td><td>40</td><td>18</td><td>0</td><td>10</td></tr><tr><td>1</td><td>0</td><td>4.4e+02</td><td>20</td><td>0</td><td>19</td></tr><tr><td>2</td><td>14</td><td>53</td><td>3.8e+02</td><td>11</td><td>40</td></tr><tr><td>3</td><td>16</td><td>39</td><td>94</td><td>2.9e+02</td><td>76</td></tr><tr><td>4</td><td>0</td><td>8</td><td>1</td><td>1</td><td>5.1e+02</td></tr><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr></table>	0	4.2e+02	40	18	0	10	1	0	4.4e+02	20	0	19	2	14	53	3.8e+02	11	40	3	16	39	94	2.9e+02	76	4	0	8	1	1	5.1e+02		0	1	2	3	4
0	4.2e+02	40	18	0	10																																	
1	0	4.4e+02	20	0	19																																	
2	14	53	3.8e+02	11	40																																	
3	16	39	94	2.9e+02	76																																	
4	0	8	1	1	5.1e+02																																	
	0	1	2	3	4																																	
70:30	80.54%	<table><tr><td>0</td><td>2.4e+02</td><td>38</td><td>16</td><td>1</td><td>5</td></tr><tr><td>1</td><td>0</td><td>2.6e+02</td><td>10</td><td>0</td><td>8</td></tr><tr><td>2</td><td>3</td><td>45</td><td>2.3e+02</td><td>6</td><td>23</td></tr><tr><td>3</td><td>5</td><td>37</td><td>56</td><td>1.8e+02</td><td>31</td></tr><tr><td>4</td><td>0</td><td>7</td><td>1</td><td>0</td><td>3e+02</td></tr><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr></table>	0	2.4e+02	38	16	1	5	1	0	2.6e+02	10	0	8	2	3	45	2.3e+02	6	23	3	5	37	56	1.8e+02	31	4	0	7	1	0	3e+02		0	1	2	3	4
0	2.4e+02	38	16	1	5																																	
1	0	2.6e+02	10	0	8																																	
2	3	45	2.3e+02	6	23																																	
3	5	37	56	1.8e+02	31																																	
4	0	7	1	0	3e+02																																	
	0	1	2	3	4																																	

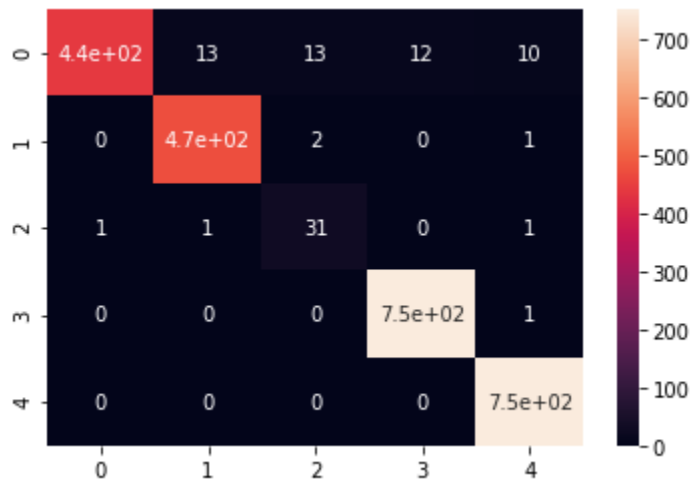
TF-IDF Feature Selection and KNN

Split (Train:Test)	Accuracy	Confusion Matrix
80:20	K=1 93.4%	<div>K=1</div>  <div>K=3</div> 
	K=3 93.8%	
	K=5 94.4%	

		<div><p>K=5</p><table><tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th></tr><tr><th>0</th><td>2e+02</td><td>5</td><td>4</td><td>5</td><td>5</td></tr><tr><th>1</th><td>0</td><td>1.8e+02</td><td>1</td><td>0</td><td>0</td></tr><tr><th>2</th><td>6</td><td>4</td><td>1.9e+02</td><td>4</td><td>5</td></tr><tr><th>3</th><td>3</td><td>2</td><td>0</td><td>2e+02</td><td>4</td></tr><tr><th>4</th><td>1</td><td>2</td><td>4</td><td>1</td><td>1.8e+02</td></tr></table></div>		0	1	2	3	4	0	2e+02	5	4	5	5	1	0	1.8e+02	1	0	0	2	6	4	1.9e+02	4	5	3	3	2	0	2e+02	4	4	1	2	4	1	1.8e+02
	0	1	2	3	4																																	
0	2e+02	5	4	5	5																																	
1	0	1.8e+02	1	0	0																																	
2	6	4	1.9e+02	4	5																																	
3	3	2	0	2e+02	4																																	
4	1	2	4	1	1.8e+02																																	
50:50	<div><p>K=1 92.1%</p><p>K=3 93.04%</p><p>K=5 94.6%</p></div>	<div><p>K=1</p><table><tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th></tr><tr><th>0</th><td>4.2e+02</td><td>15</td><td>21</td><td>19</td><td>14</td></tr><tr><th>1</th><td>2</td><td>4.7e+02</td><td>4</td><td>1</td><td>1</td></tr><tr><th>2</th><td>0</td><td>2</td><td>32</td><td>0</td><td>0</td></tr><tr><th>3</th><td>0</td><td>0</td><td>0</td><td>7.5e+02</td><td>0</td></tr><tr><th>4</th><td>0</td><td>0</td><td>0</td><td>0</td><td>7.5e+02</td></tr></table></div> <div><p>K=3</p></div>		0	1	2	3	4	0	4.2e+02	15	21	19	14	1	2	4.7e+02	4	1	1	2	0	2	32	0	0	3	0	0	0	7.5e+02	0	4	0	0	0	0	7.5e+02
	0	1	2	3	4																																	
0	4.2e+02	15	21	19	14																																	
1	2	4.7e+02	4	1	1																																	
2	0	2	32	0	0																																	
3	0	0	0	7.5e+02	0																																	
4	0	0	0	0	7.5e+02																																	



K=5



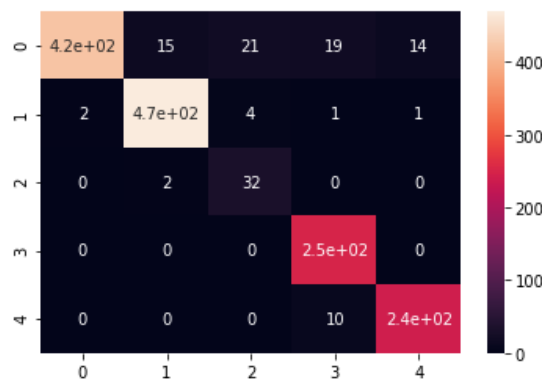
70:30

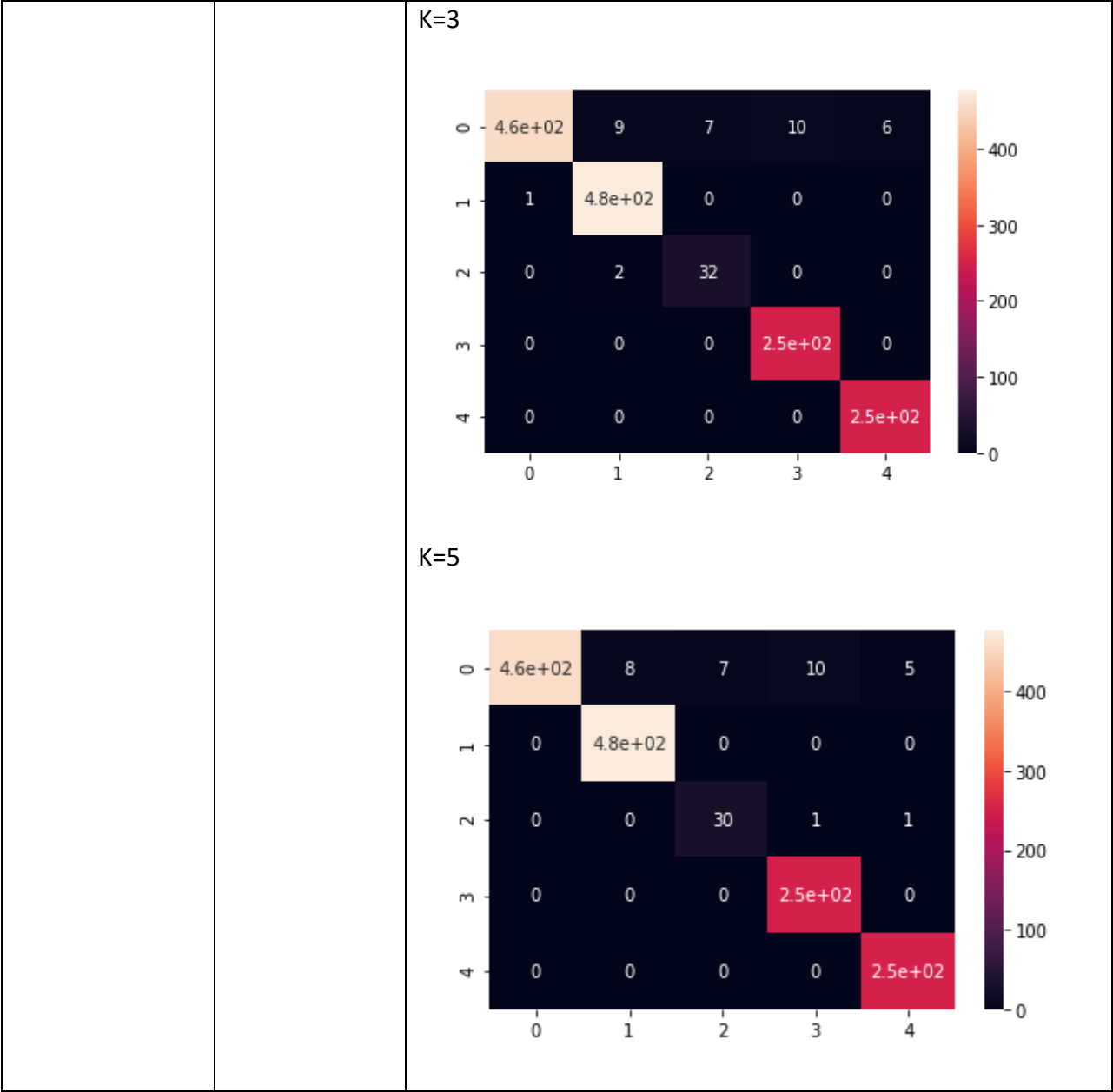
K=1 92.6%

K=3 93.4%

K=5 94.0%

K=1



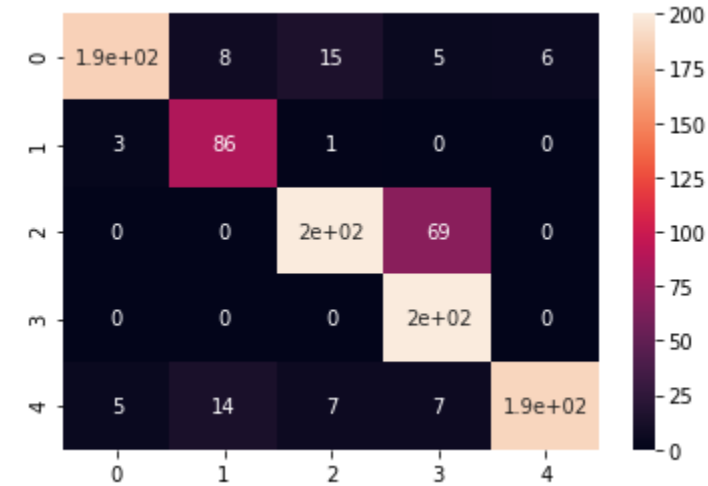


Mutual Information and KNN Classifier

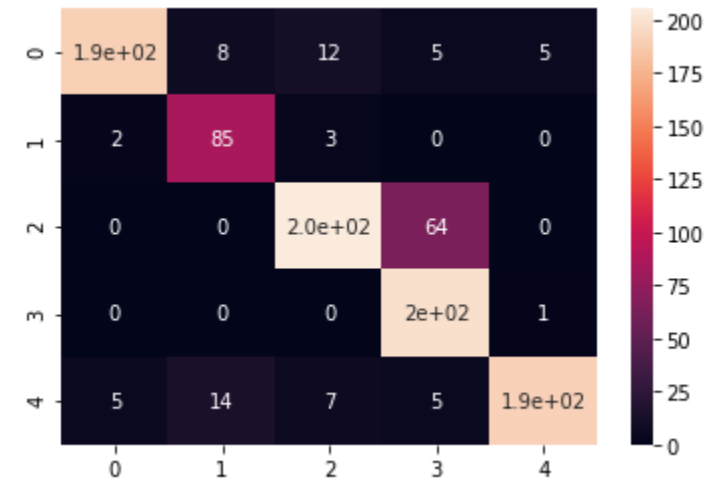
Split (Train:Test)	Accuracy	Confusion Matrix
80:20	K=1 87.75%	K=1

K=3 88.6%

K=5 88.4%



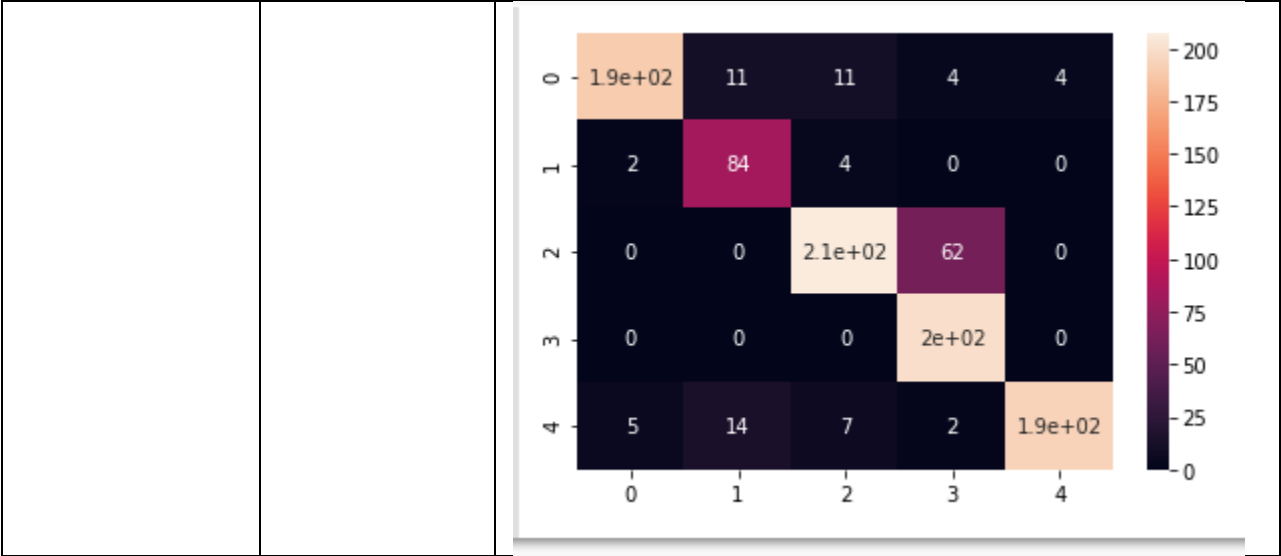
K=3



K=5

		<table><tr><td>0</td><td>1.9e+02</td><td>11</td><td>11</td><td>4</td><td>4</td></tr><tr><td>1</td><td>2</td><td>84</td><td>4</td><td>0</td><td>0</td></tr><tr><td>2</td><td>0</td><td>0</td><td>2.1e+02</td><td>62</td><td>0</td></tr><tr><td>3</td><td>0</td><td>0</td><td>0</td><td>2e+02</td><td>0</td></tr><tr><td>4</td><td>5</td><td>14</td><td>7</td><td>2</td><td>1.9e+02</td></tr></table>	0	1.9e+02	11	11	4	4	1	2	84	4	0	0	2	0	0	2.1e+02	62	0	3	0	0	0	2e+02	0	4	5	14	7	2	1.9e+02																														
0	1.9e+02	11	11	4	4																																																									
1	2	84	4	0	0																																																									
2	0	0	2.1e+02	62	0																																																									
3	0	0	0	2e+02	0																																																									
4	5	14	7	2	1.9e+02																																																									
50:50	<p>K=1 88%</p> <p>K=3 87.8%</p> <p>K=5 87.5%</p>	<p>K=1</p> <table><tr><td>0</td><td>4e+02</td><td>30</td><td>45</td><td>13</td><td>3</td></tr><tr><td>1</td><td>9</td><td>4.4e+02</td><td>20</td><td>3</td><td>3</td></tr><tr><td>2</td><td>31</td><td>14</td><td>4.4e+02</td><td>10</td><td>11</td></tr><tr><td>3</td><td>27</td><td>6</td><td>22</td><td>4.5e+02</td><td>14</td></tr><tr><td>4</td><td>7</td><td>11</td><td>14</td><td>7</td><td>4.8e+02</td></tr></table> <p>K=3</p> <table><tr><td>0</td><td>4.1e+02</td><td>29</td><td>36</td><td>13</td><td>4</td></tr><tr><td>1</td><td>16</td><td>4.3e+02</td><td>21</td><td>0</td><td>7</td></tr><tr><td>2</td><td>37</td><td>19</td><td>4.1e+02</td><td>15</td><td>17</td></tr><tr><td>3</td><td>29</td><td>5</td><td>19</td><td>4.4e+02</td><td>18</td></tr><tr><td>4</td><td>6</td><td>13</td><td>12</td><td>7</td><td>4.8e+02</td></tr></table> <p>K=5</p>	0	4e+02	30	45	13	3	1	9	4.4e+02	20	3	3	2	31	14	4.4e+02	10	11	3	27	6	22	4.5e+02	14	4	7	11	14	7	4.8e+02	0	4.1e+02	29	36	13	4	1	16	4.3e+02	21	0	7	2	37	19	4.1e+02	15	17	3	29	5	19	4.4e+02	18	4	6	13	12	7	4.8e+02
0	4e+02	30	45	13	3																																																									
1	9	4.4e+02	20	3	3																																																									
2	31	14	4.4e+02	10	11																																																									
3	27	6	22	4.5e+02	14																																																									
4	7	11	14	7	4.8e+02																																																									
0	4.1e+02	29	36	13	4																																																									
1	16	4.3e+02	21	0	7																																																									
2	37	19	4.1e+02	15	17																																																									
3	29	5	19	4.4e+02	18																																																									
4	6	13	12	7	4.8e+02																																																									

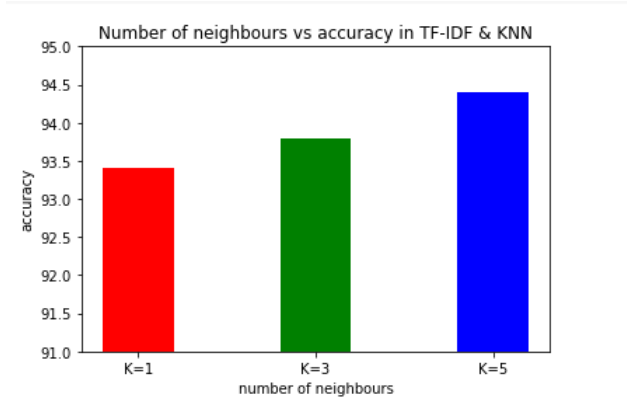
70:30	<div>K=1 87.57%</div> <div>K=3 88.66%</div> <div>K=5 88.91%</div>	<div>K=1</div> <div>K=3</div> <div>K=5</div>



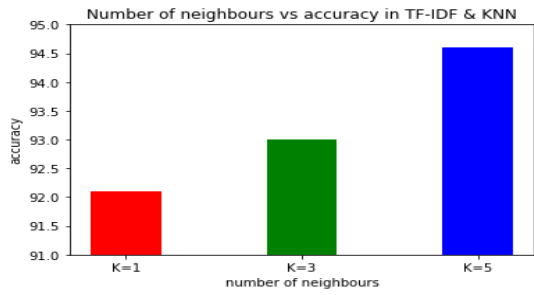
Analysis Performed

Accuracy vs Variation in k of KNN classifier using TF-IDF Feature selection

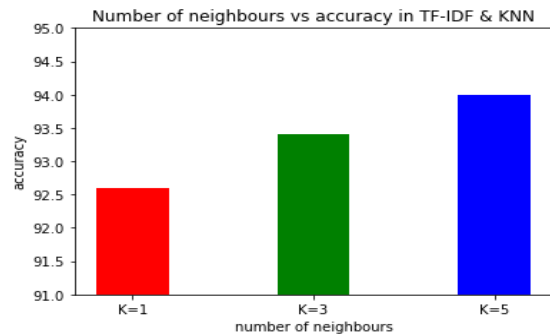
80:20 Split



50:50 Split

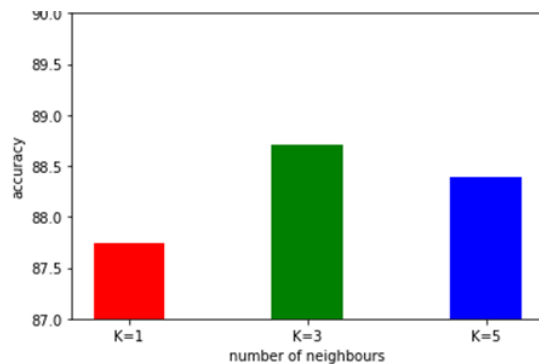


70:30 Split

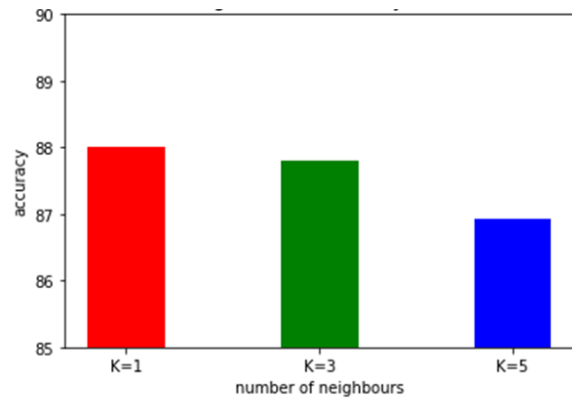


Accuracy vs Variation in k of KNN classifier using MI Feature selection

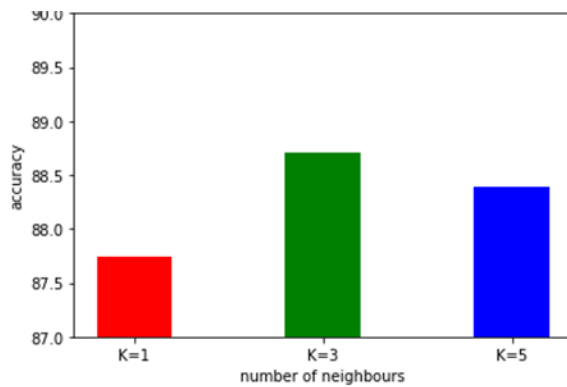
80:20 Split



50:50 Split

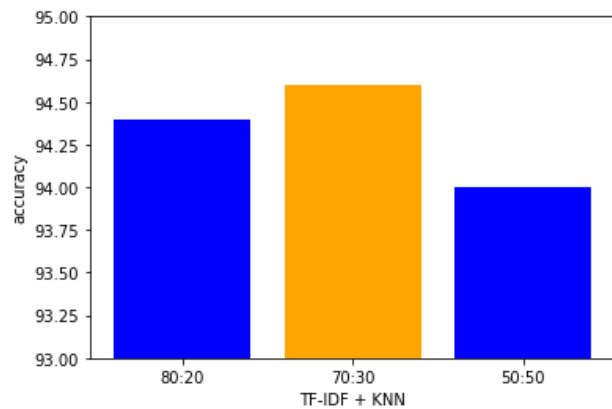


70:30 Split

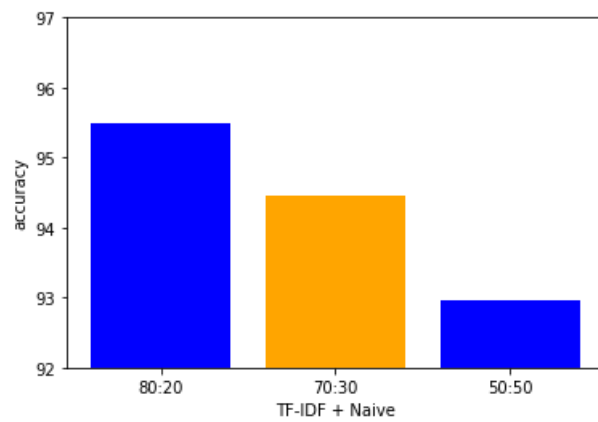


Performance of (classifier + feature selection method) vs Accuracy

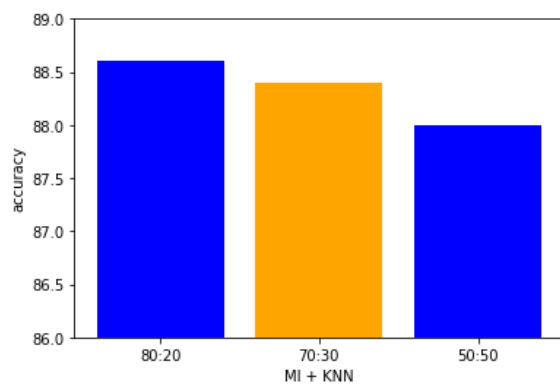
TF-IDF + KNN



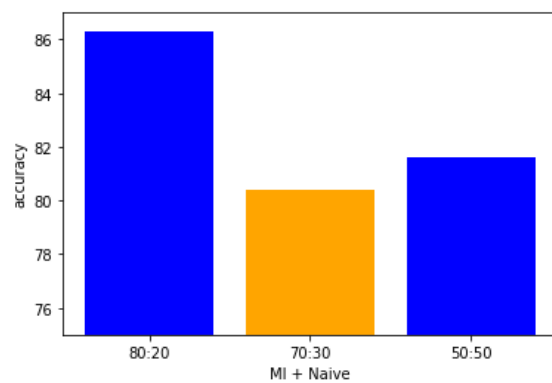
TF_IDF + Naïve



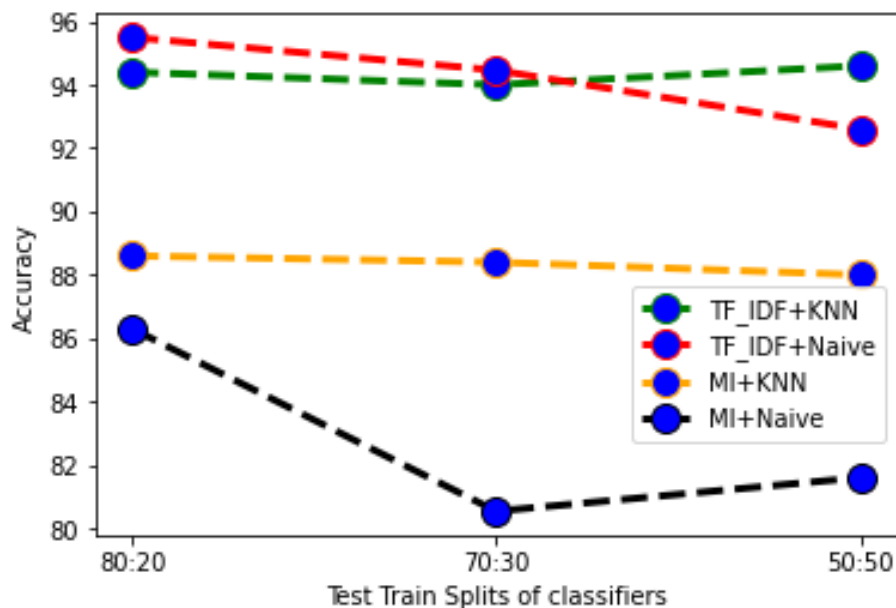
MI +KNN



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 + Naïve 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 + Naïve 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 $\frac{1}{5}$ of total number of terms as the selected features gives optimal results whereas TF_IDF with $\frac{1}{2}$ 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.