ASSIGNMENT 4

Assumptions

1. Header is assumed to be everything above the first blank line.
2. Stop words present in the NLTK corpus are considered.

Pre-processing Steps

Removal of Header

(All the lines before the first blank line are removed)

Removal of Punctuation marks, comma, etc

(They are removed through regular expression)

Tokenization

(Tokens are formed using word_tokenize and special symbols are removed)

Removal of Stop Words

(Stop words are removed using NLTK stop words)

Normalization

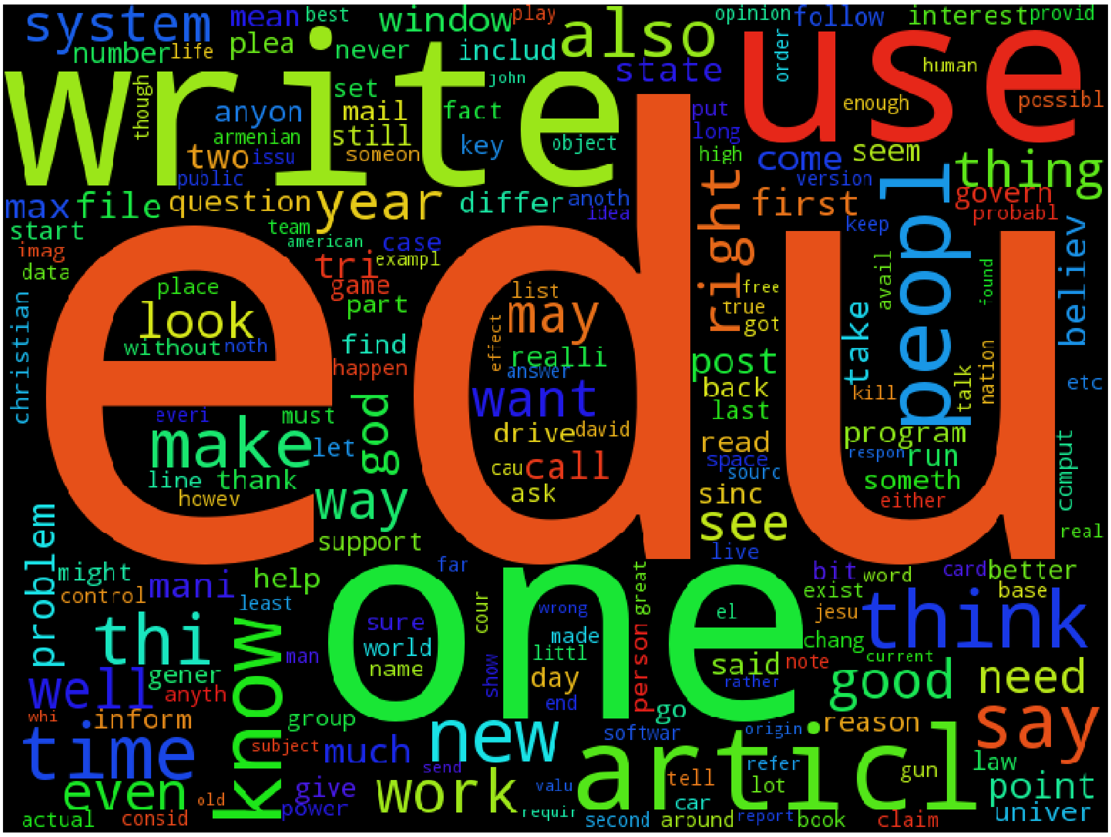
(All token are converted into lower case)

Stemming

(Stemming is performed using Porter algorithm to get the root word)

Number of Documents: 5000

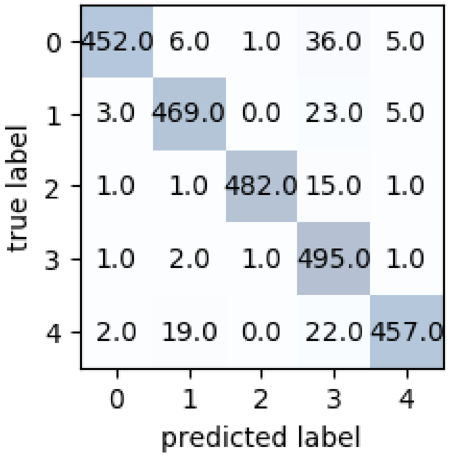
WordCloud



Rocchio Classification Algorithm

50:50 Train Test Split

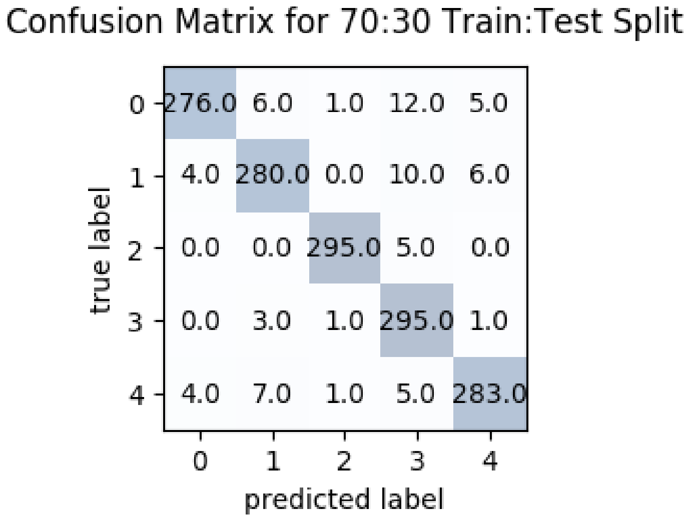
Confusion Matrix for 50:50 Train:Test Split



true label \ predicted label	0	1	2	3	4
0	452.0	6.0	1.0	36.0	5.0
1	3.0	469.0	0.0	23.0	5.0
2	1.0	1.0	482.0	15.0	1.0
3	1.0	2.0	1.0	495.0	1.0
4	2.0	19.0	0.0	22.0	457.0

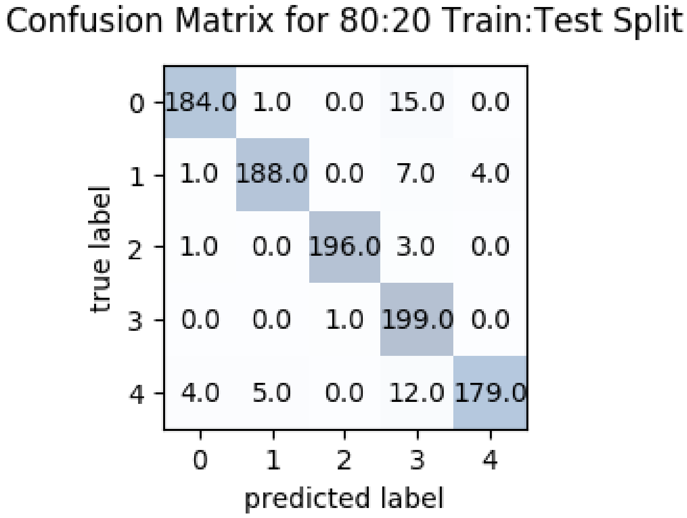
Accuracy = 94.2%

70:30 Train Test Split



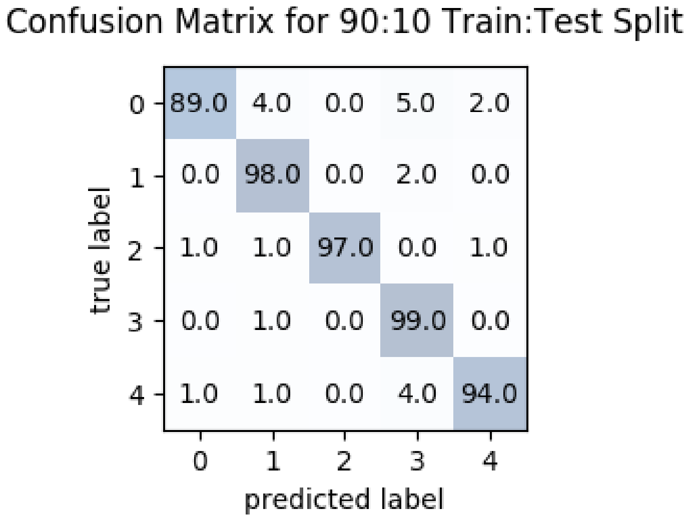
Accuracy= 95.2666666667%

80:20 Train Test Split

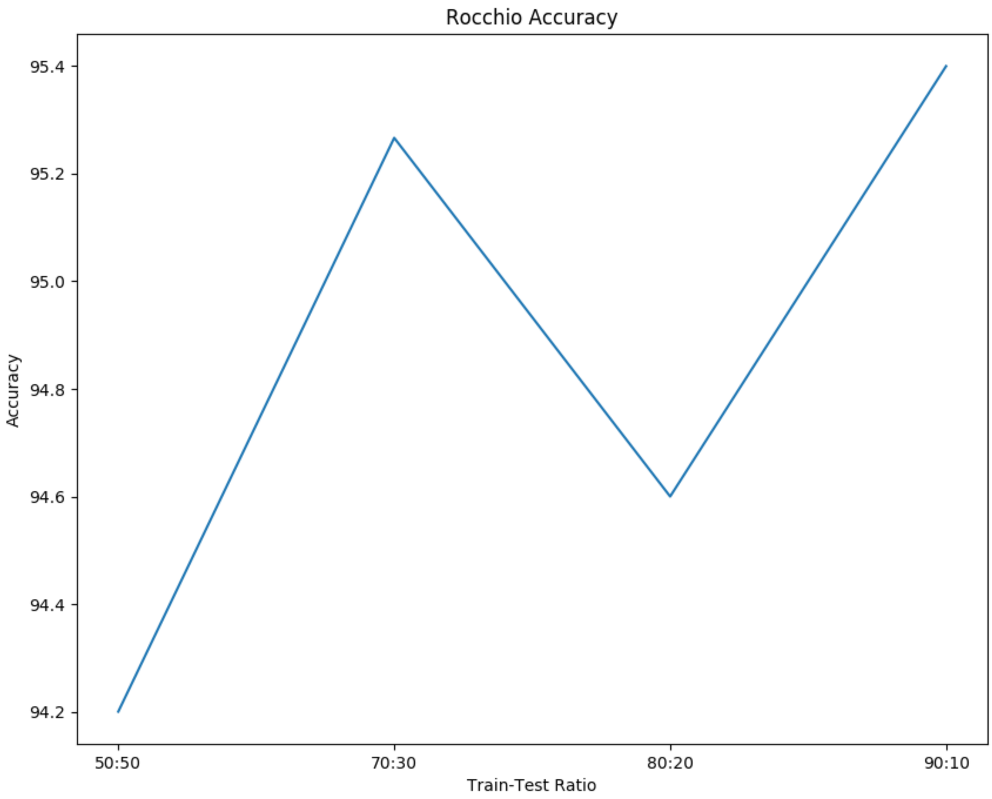


Accuracy = 94.6%

90:10 Train Test Split



Accuracy= 95.4%

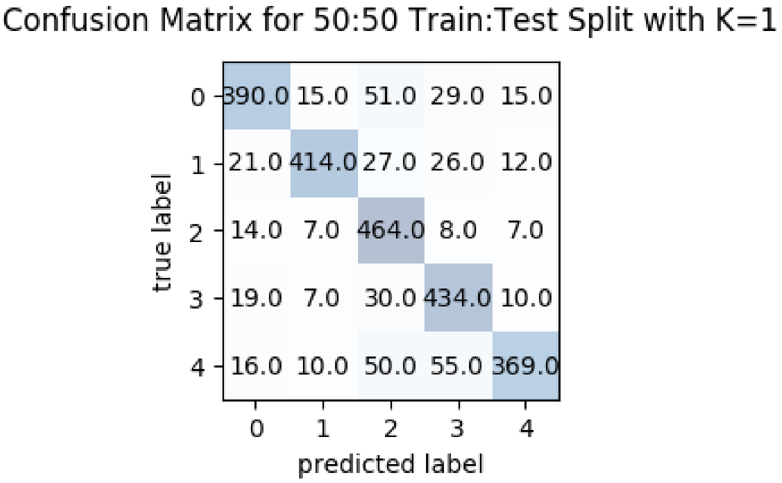


Accuracy vs Train-Test Ratio

From the graph, we can infer that the accuracy of the Rocchio classification algorithm increases with the increase in the Train Ratio (Train Data) (decrease in Test Ratio) (with a slight decrease in 80:20 ratio).

KNN Classification Algorithm

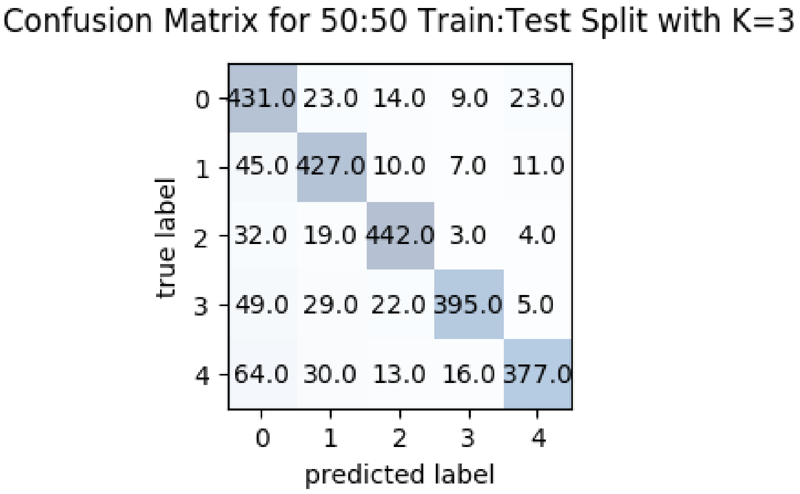
50:50 Train Test Split



Confusion Matrix for 50:50 Train:Test Split with K=1

	0	1	2	3	4
0	390.0	15.0	51.0	29.0	15.0
1	21.0	414.0	27.0	26.0	12.0
2	14.0	7.0	464.0	8.0	7.0
3	19.0	7.0	30.0	434.0	10.0
4	16.0	10.0	50.0	55.0	369.0

Accuracy= 82.84%

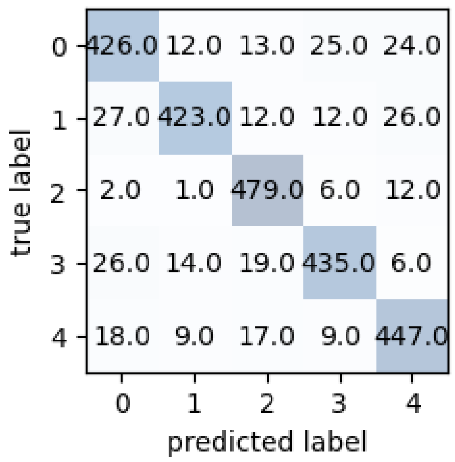


Confusion Matrix for 50:50 Train:Test Split with K=3

	0	1	2	3	4
0	431.0	23.0	14.0	9.0	23.0
1	45.0	427.0	10.0	7.0	11.0
2	32.0	19.0	442.0	3.0	4.0
3	49.0	29.0	22.0	395.0	5.0
4	64.0	30.0	13.0	16.0	377.0

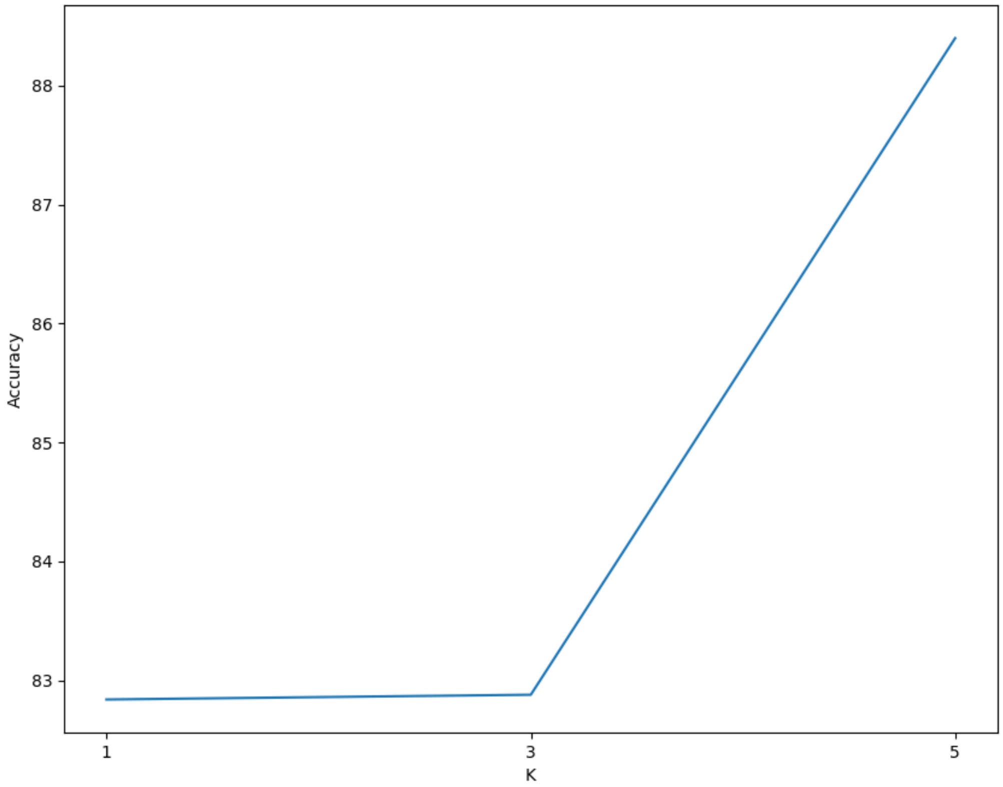
Accuracy= 82.88%

Confusion Matrix for 50:50 Train:Test Split with K=5



true label \ predicted label	0	1	2	3	4
0	426.0	12.0	13.0	25.0	24.0
1	27.0	423.0	12.0	12.0	26.0
2	2.0	1.0	479.0	6.0	12.0
3	26.0	14.0	19.0	435.0	6.0
4	18.0	9.0	17.0	9.0	447.0

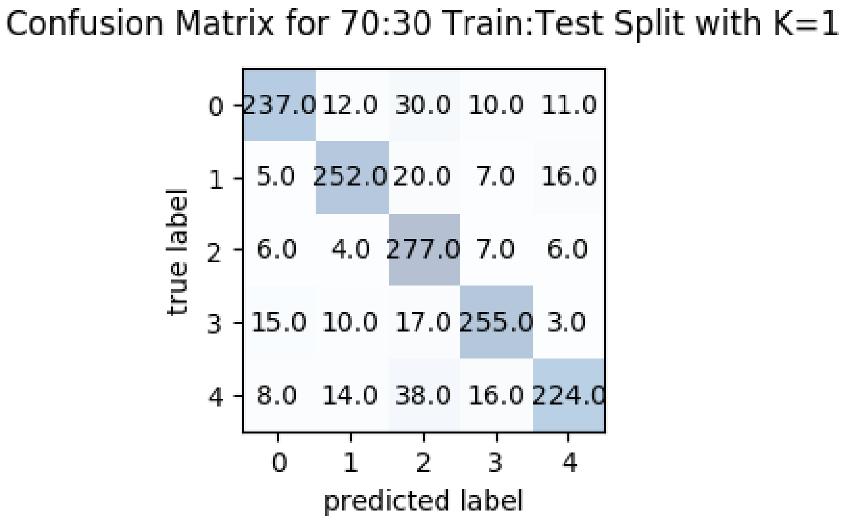
Accuracy= 88.4%



K vs Accuracy

From the graph, we can infer that the accuracy of the KNN increases with the increase in the K Value.

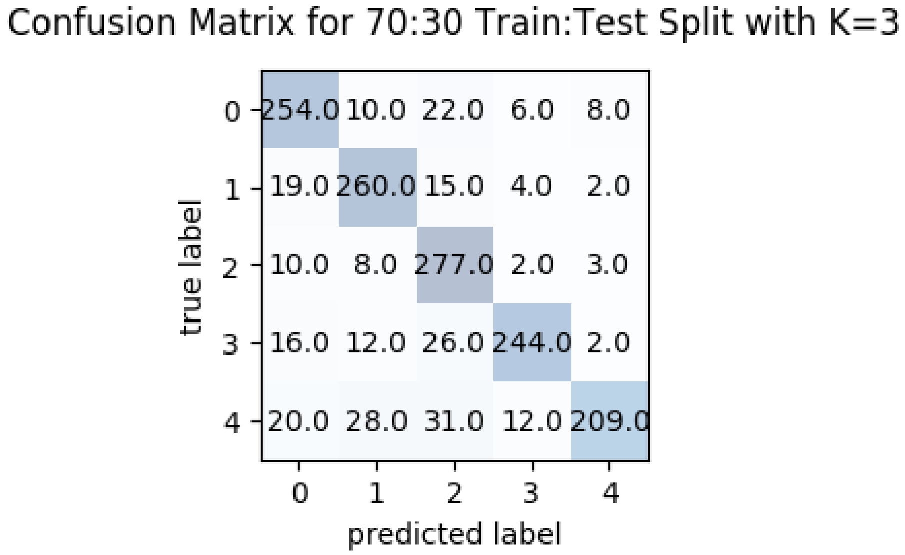
70:30 Train Test Split



Confusion Matrix for 70:30 Train:Test Split with K=1

	0	1	2	3	4
0	237.0	12.0	30.0	10.0	11.0
1	5.0	252.0	20.0	7.0	16.0
2	6.0	4.0	277.0	7.0	6.0
3	15.0	10.0	17.0	255.0	3.0
4	8.0	14.0	38.0	16.0	224.0

Accuracy= 83.00%

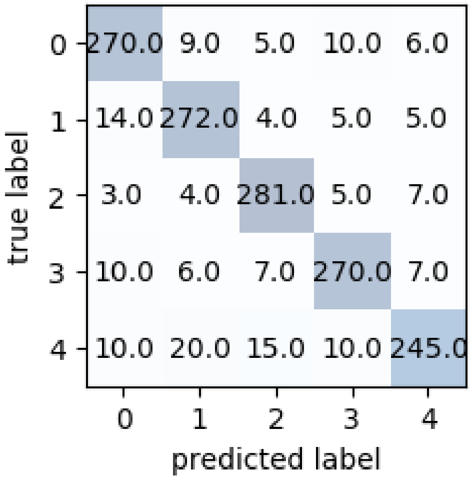


Confusion Matrix for 70:30 Train:Test Split with K=3

	0	1	2	3	4
0	254.0	10.0	22.0	6.0	8.0
1	19.0	260.0	15.0	4.0	2.0
2	10.0	8.0	277.0	2.0	3.0
3	16.0	12.0	26.0	244.0	2.0
4	20.0	28.0	31.0	12.0	209.0

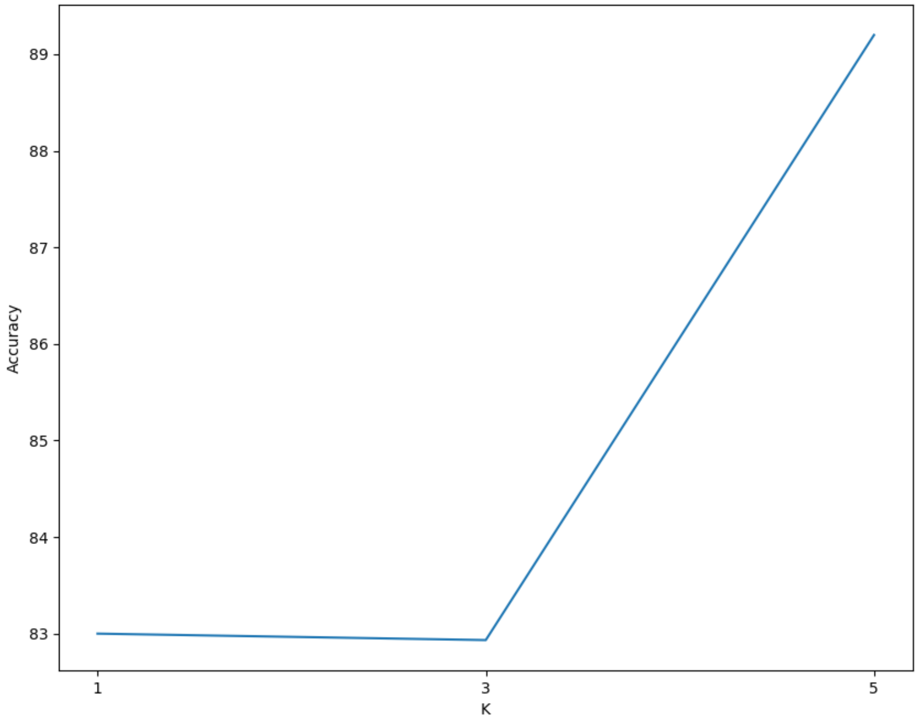
Accuracy= 82.9333333%

Confusion Matrix for 70:30 Train:Test Split with K=5



true label \ predicted label	0	1	2	3	4
0	270.0	9.0	5.0	10.0	6.0
1	14.0	272.0	4.0	5.0	5.0
2	3.0	4.0	281.0	5.0	7.0
3	10.0	6.0	7.0	270.0	7.0
4	10.0	20.0	15.0	10.0	245.0

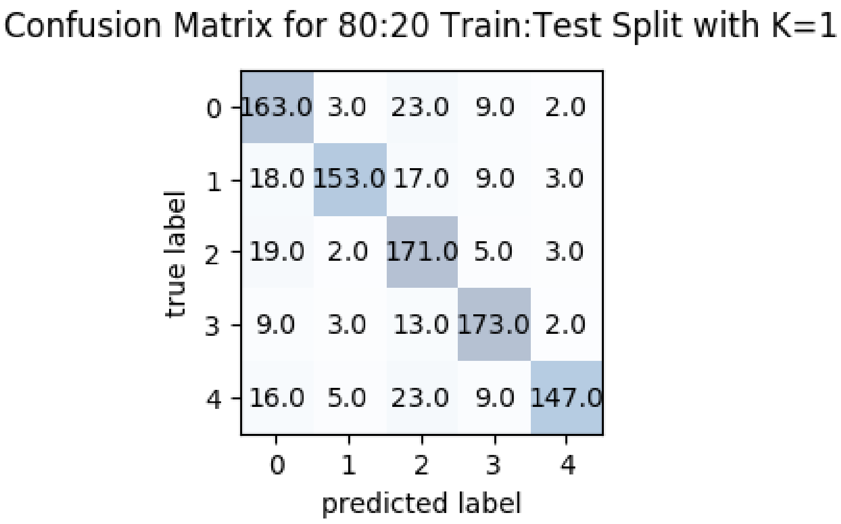
Accuracy= 89.2%



K vs Accuracy

From the graph, we can infer that the accuracy of the KNN increases with the increase in the K Value.

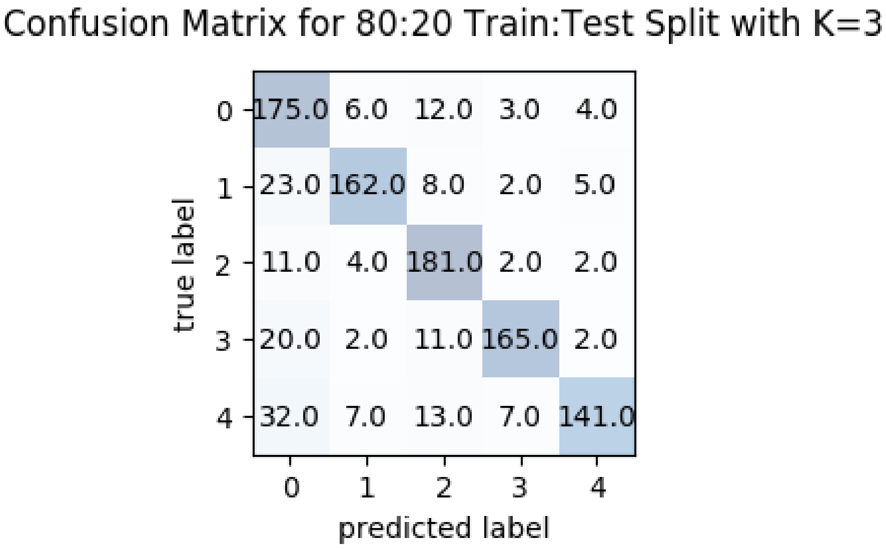
80:20 Train Test Split



Confusion Matrix for 80:20 Train:Test Split with K=1

	0	1	2	3	4
0	163.0	3.0	23.0	9.0	2.0
1	18.0	153.0	17.0	9.0	3.0
2	19.0	2.0	171.0	5.0	3.0
3	9.0	3.0	13.0	173.0	2.0
4	16.0	5.0	23.0	9.0	147.0

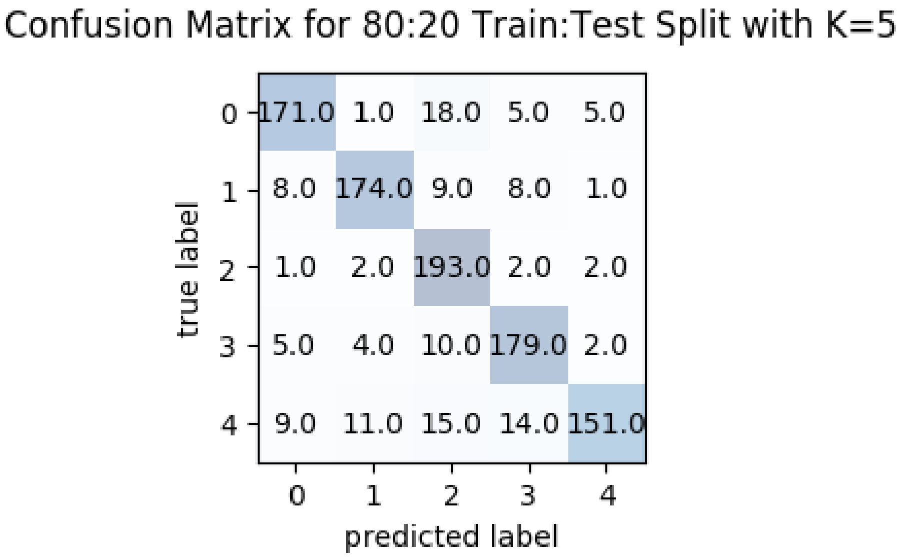
Accuracy= 80.7%



Confusion Matrix for 80:20 Train:Test Split with K=3

	0	1	2	3	4
0	175.0	6.0	12.0	3.0	4.0
1	23.0	162.0	8.0	2.0	5.0
2	11.0	4.0	181.0	2.0	2.0
3	20.0	2.0	11.0	165.0	2.0
4	32.0	7.0	13.0	7.0	141.0

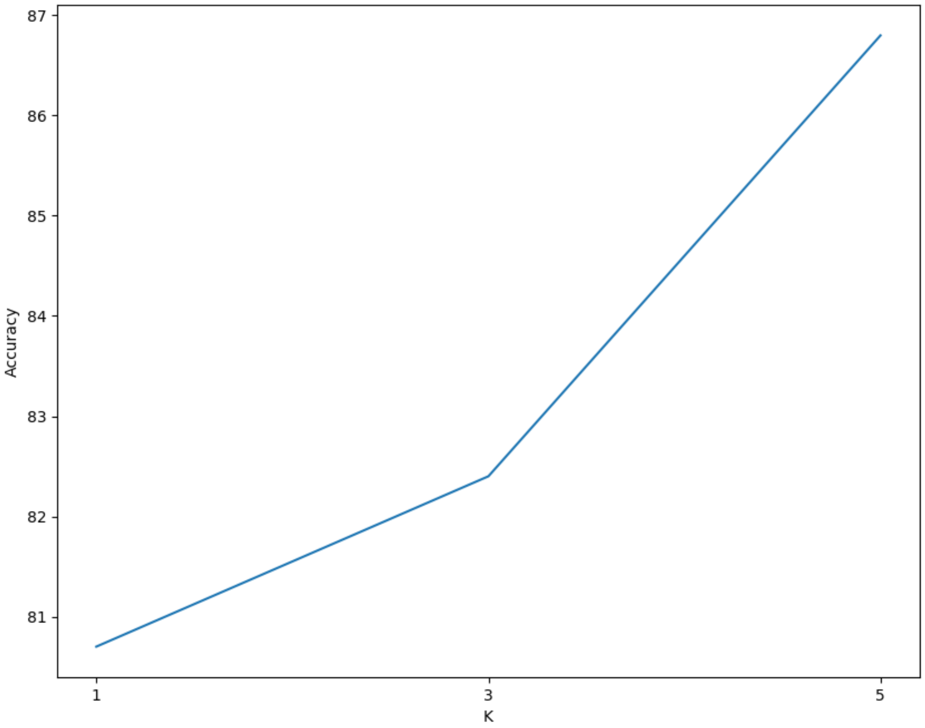
Accuracy= 82.4%



Confusion Matrix for 80:20 Train:Test Split with K=5

true label \ predicted label	0	1	2	3	4
0	171.0	1.0	18.0	5.0	5.0
1	8.0	174.0	9.0	8.0	1.0
2	1.0	2.0	193.0	2.0	2.0
3	5.0	4.0	10.0	179.0	2.0
4	9.0	11.0	15.0	14.0	151.0

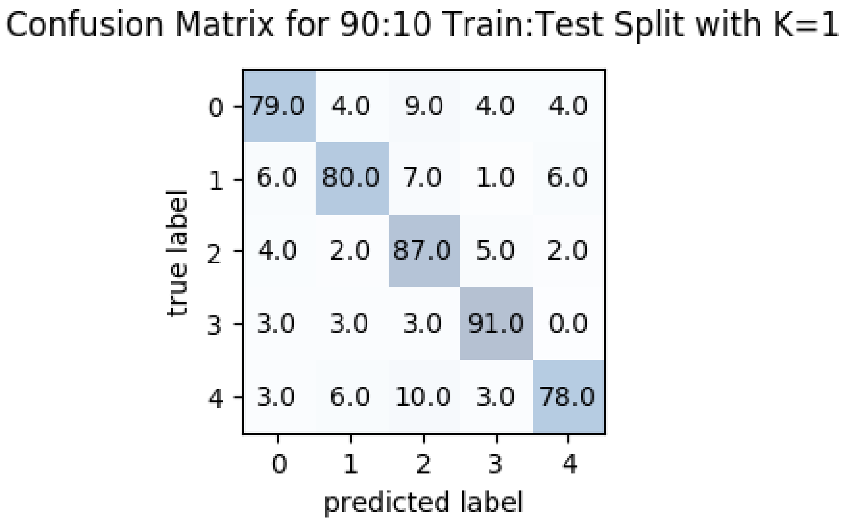
Accuracy= 86.8%



K vs Accuracy

From the graph, we can infer that the accuracy of the KNN increases with the increase in the K Value.

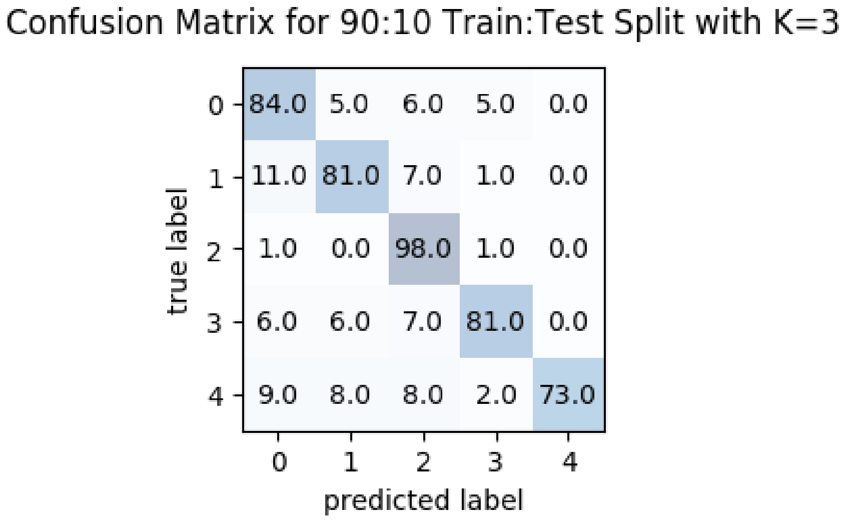
90:10 Train Test Split



Confusion Matrix for 90:10 Train:Test Split with K=1

true label \ predicted label	0	1	2	3	4
0	79.0	4.0	9.0	4.0	4.0
1	6.0	80.0	7.0	1.0	6.0
2	4.0	2.0	87.0	5.0	2.0
3	3.0	3.0	3.0	91.0	0.0
4	3.0	6.0	10.0	3.0	78.0

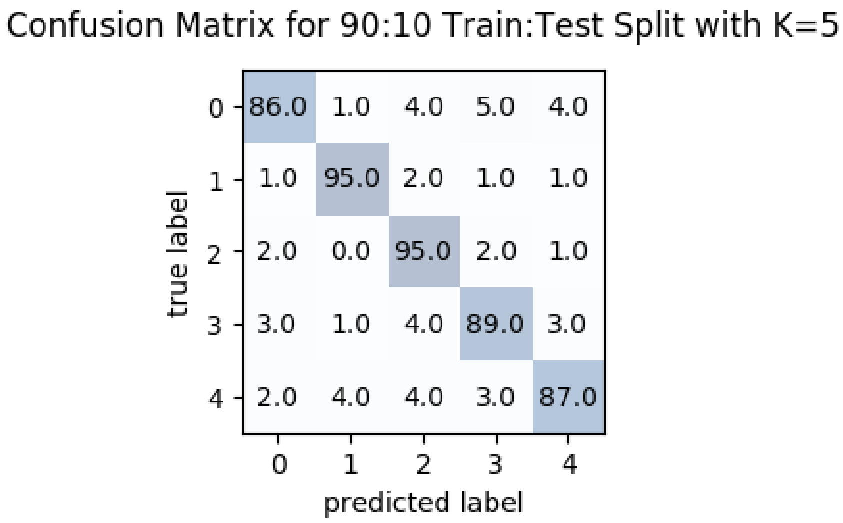
Accuracy= 83.0%



Confusion Matrix for 90:10 Train:Test Split with K=3

true label \ predicted label	0	1	2	3	4
0	84.0	5.0	6.0	5.0	0.0
1	11.0	81.0	7.0	1.0	0.0
2	1.0	0.0	98.0	1.0	0.0
3	6.0	6.0	7.0	81.0	0.0
4	9.0	8.0	8.0	2.0	73.0

Accuracy= 83.4%

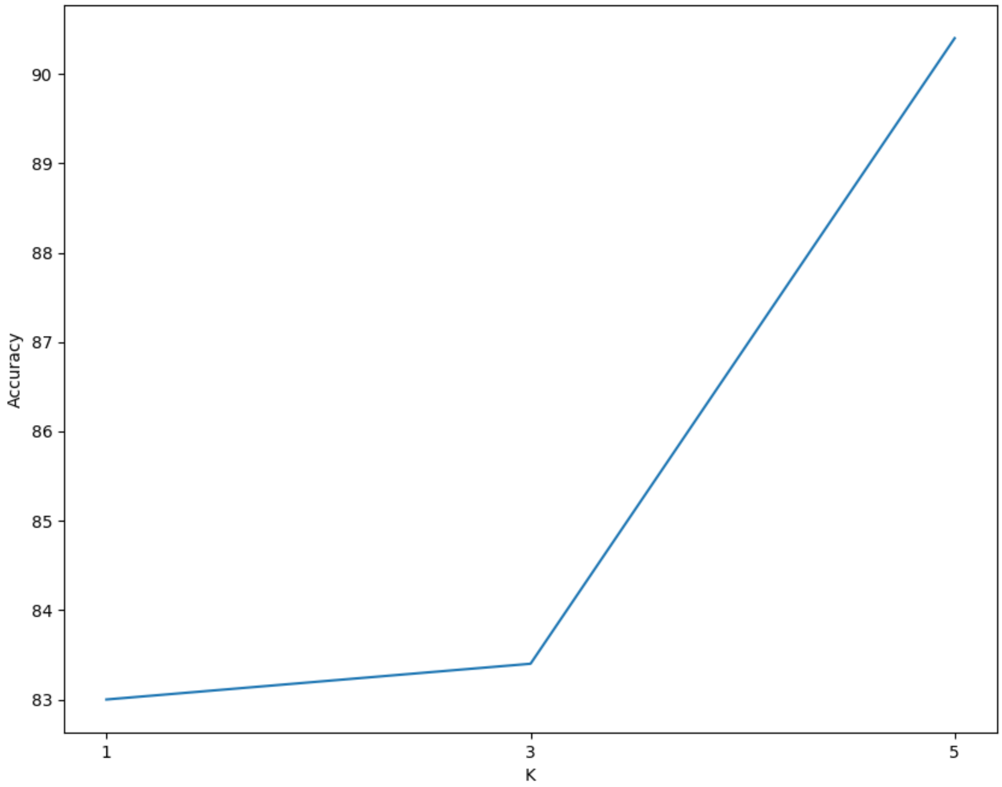


Confusion Matrix for 90:10 Train:Test Split with K=5

	0	1	2	3	4
0	86.0	1.0	4.0	5.0	4.0
1	1.0	95.0	2.0	1.0	1.0
2	2.0	0.0	95.0	2.0	1.0
3	3.0	1.0	4.0	89.0	3.0
4	2.0	4.0	4.0	3.0	87.0

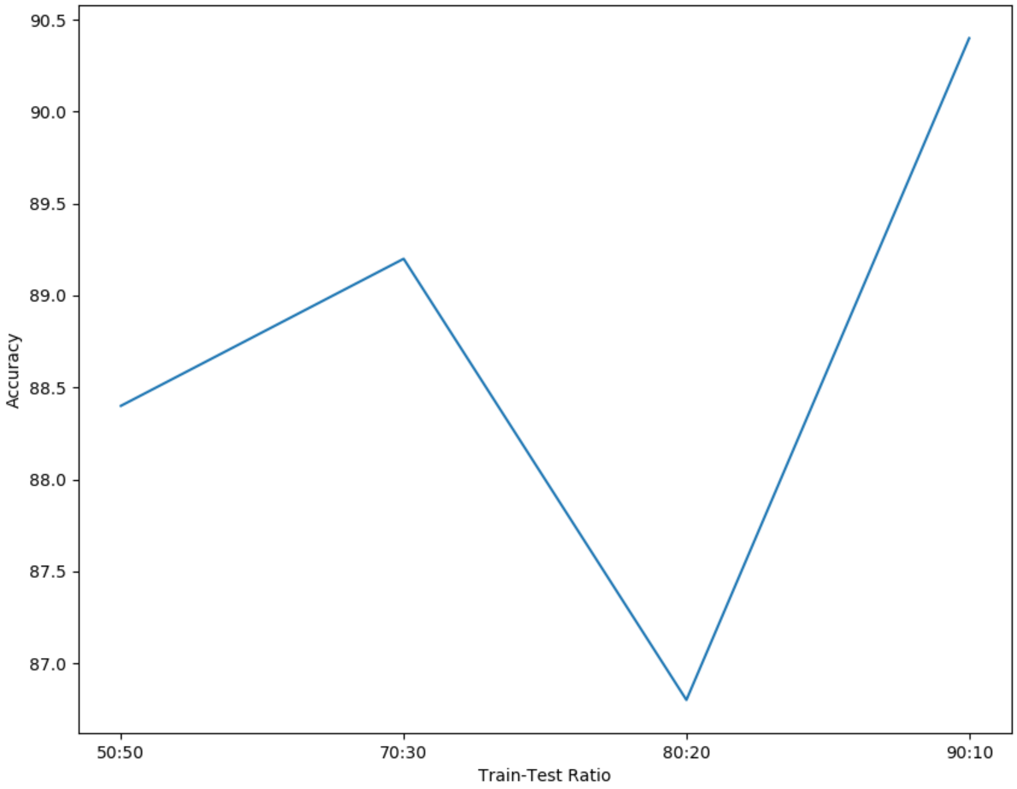
true label
predicted label

Accuracy= 90.4%



K vs Accuracy

From the graph, we can infer that the accuracy of the KNN increases with the increase in the K Value.



Accuracy vs Train-Test Ratio (at best value of K)

We can infer that the best accuracy occurs at 90:10 train test ratio with a value of **90.4%**

Comparison

KNN Classification	**90.4 %**
Rocchio Classification	**95.4 %**
Naïve Bayes Classification	**96.8 %**

Naïve Bayes Classification achieves an accuracy of 96.8% which is higher than the Rocchio and KNN classification. KNN achieves only 90.4% because the train dataset available is small. Rocchio classification achieves 95.4% which is quite close to Naïve Bayes classification. We can see that Naïve Bayes classification outperforms the KNN and Rocchio classification.