REPORT
Manish Jaisinghani

1. **Objective:** Demonstrate decision tree and KNN algorithms with 5 fold cross validation and evaluate classification results with accuracy and F-Measure.

2. **Data Collection:** The data collection process is completed in two steps as described below:

   a. Website List: I collected the website list by using online website scrapper http://www.xsitemap.com. The scrapper parses all the sub URLs from a given URL and provides a list of the parsed URLs. The URL list has been divided into 4 categories and 25 URLs have been taken from each category:
      - Politics – 25 articles
      - Health – 25 articles
      - World – 25 articles
      - Technology – 25 articles

   b. Website Scrapper: I have written HTML website scrapper in python using beautiful soup and NLTK as packages to parse the HTML packages and remove words from the collected list that do not add to meaning of the article, for example “a, the, he, she, these etc.” A complete list of the words which have been remove or not taken into consideration has been attached by the file name “stopwords”.

3. **Data Processing:** The data processing is done in two parts as described below:

   a. Website parsing: A list of URLs is provided in the file by name “**website_list**”. The python script parses through each website one by one. The script makes use of beautiful soup, parses the website and pulls out text data eliminating all the symbols, punctuations, styles and scripts. Once the data is collected the data is recorded in a list data structure and then stored making sure that there are no duplicate words in the list. The same procedure is repeated for every URL. While parsing the text of current article frequency of words occurring in the article is also recorded. The final step is just writing this data to a CSV file(“**data.csv”**). In the csv file each article is represented as a row and each word is recorded in a column and the final matrix consists of the frequency of words.

   b. Data manipulation: For decision tree algorithm the data was required to be cleaned and manipulated. The reason behind this task was that many of the words were discovered only once in a single article and were not adding any features to the decision making algorithm however affecting the functionality of prediction tests. I removed all the attributes which were not adding any value to the algorithm and ran the algorithm over less number and more meaningful attributes.

4.Module specifications:

The algorithms have been implemented using python 3 and below listed libraries have been used:

- **Pandas** – Deals with csv data and helps in manipulation.
- **Numpy** – Deals with data in the form of arrays and assists in manipulation
- **Matplotlib** – Helps to visualize the data.
- **Seaborn** – Another library which helps in data visualization, the only difference here is seaborn is vast than matplotlib.
- **Sklearn** – Helps to implement algorithms such as the decision classifier and KNN classifier and generate data models based on preprocessed data.
- **StratifiedKFold** – Helps to implement cross validation and generate data training and testing data with respect to predictor and target labels.
- **KNeighborsClassifier** – Specific module for implementing KNN.
- **DecisionTreeClassifier** – Specific module for implementing Decision tree

5. Implementation details:

Below are the details on how the algorithms have been written and pre-requisites for successfully running the same:-

- **DecisionKNN.py** - The file contains the implementation of both KNN as well as decision tree algorithm.
- **Data.csv** – This file contains the data that has been considered as data set for evaluation of decision tree and KNN classifiers.
- **Constraints** – The system should be installed with all python modules listed in "Module Specifications" section.
- The python script should be executed using python 3.0 or higher.
- The decision tree classifier also generates graph for all the executions.
- Once the module is executed we get a comparison of accuracy and precision for decision tree classifier with KNN classifier.

6. Data Analysis:

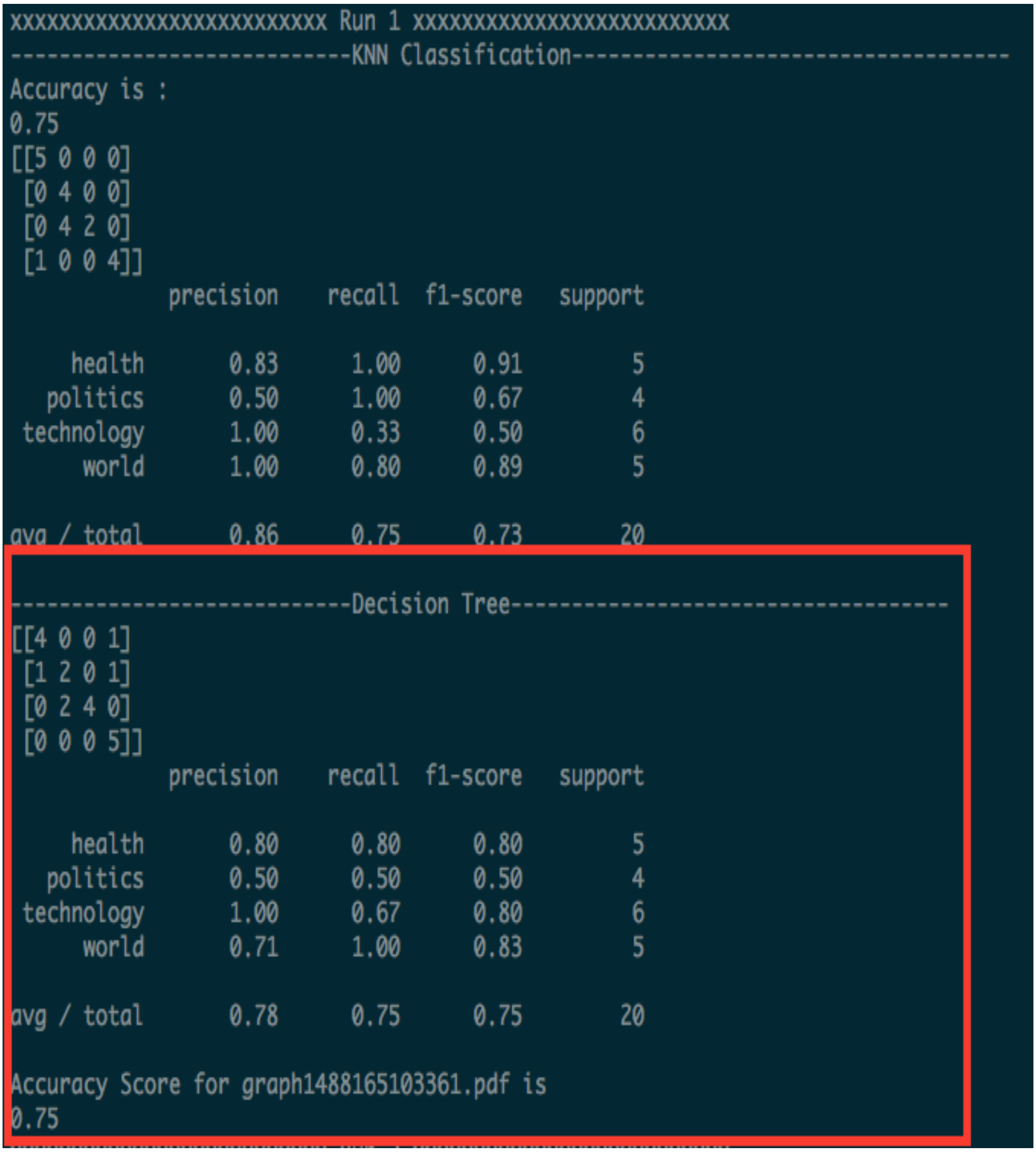
Execution of Decision tree classifier:

- **Cross Validation:** In this section we divide the data in 5 sets of training and testing data.
- For each set of training and testing data set we execute the decision tree classifier and report all the parameters
- **Accuracy:** We report accuracy of the run using sklearn module.
- **Confusion matrix:** We report confusion matrix for each run and demonstrate correct and incorrect number of predictions in the testing data.
- **Precision:** We report precision on each of the category labels and then calculate cumulative precision of each run.
- **Recall:** We report recall on each category levels and then calculate cumulative recall for each run of dataset.
- **F1-Score:** We calculate f measure for each category and a cumulative f measure as well. We also display the comparison results of f-measure for both KNN and Decision tree algorithms.
- **Accuracy:** We calculate accuracy for each category and a cumulative accuracy as well. We also display the comparison results of f-measure for both KNN and Decision tree algorithms.

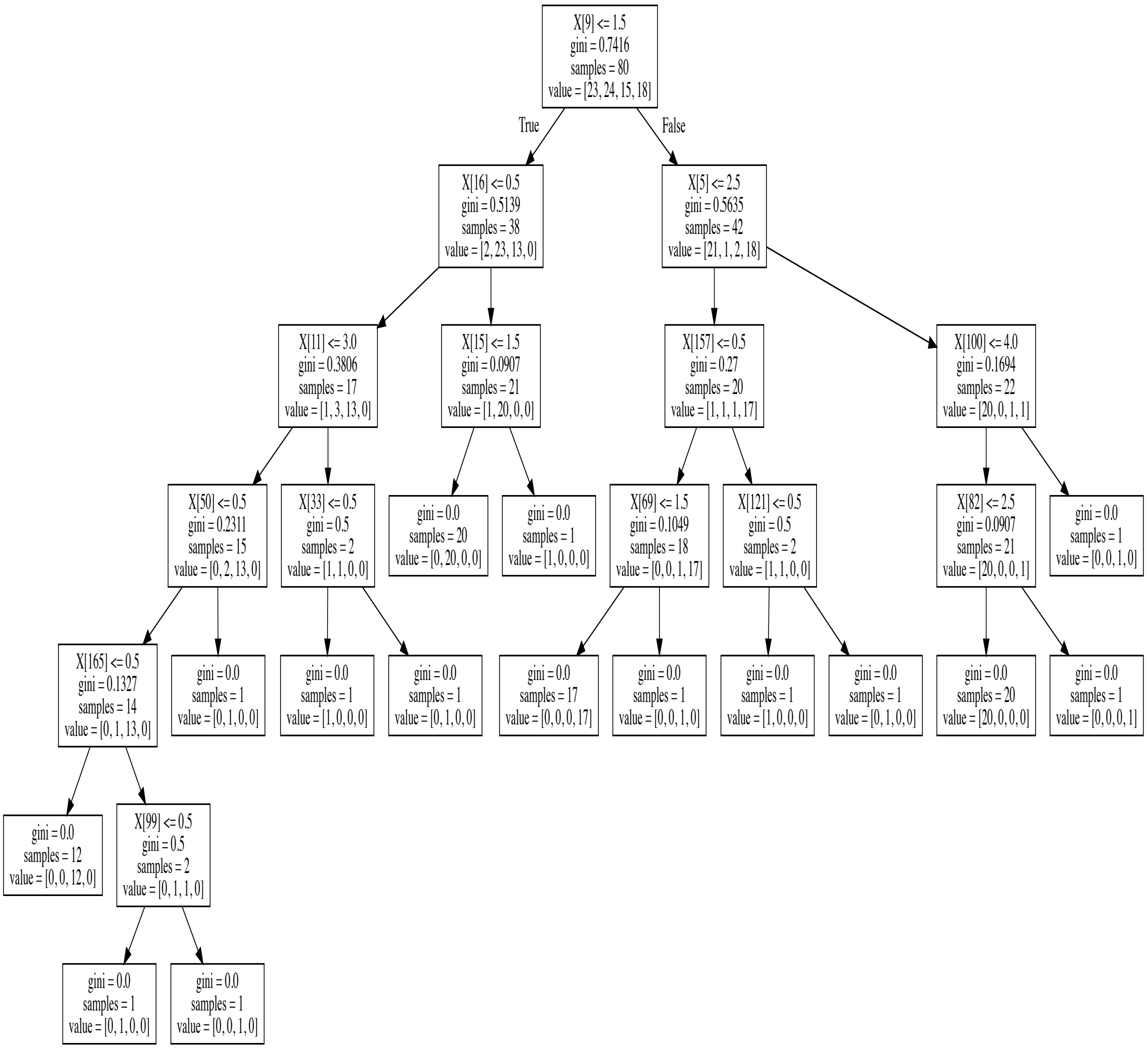
a. **Decision Tree analysis:** We will take a sample execution run and demonstrate data analysis for decision tree.

- The highlighted part shows all the parameters for decision tree execution.
- First is the confusion matrix
- Second comes the precision, recall F-Measure and support for each category as well as average for one run.
- Then comes the graph for the execution

- Finally it provides the accuracy for the execution.
- Below is the graph generated for execution run

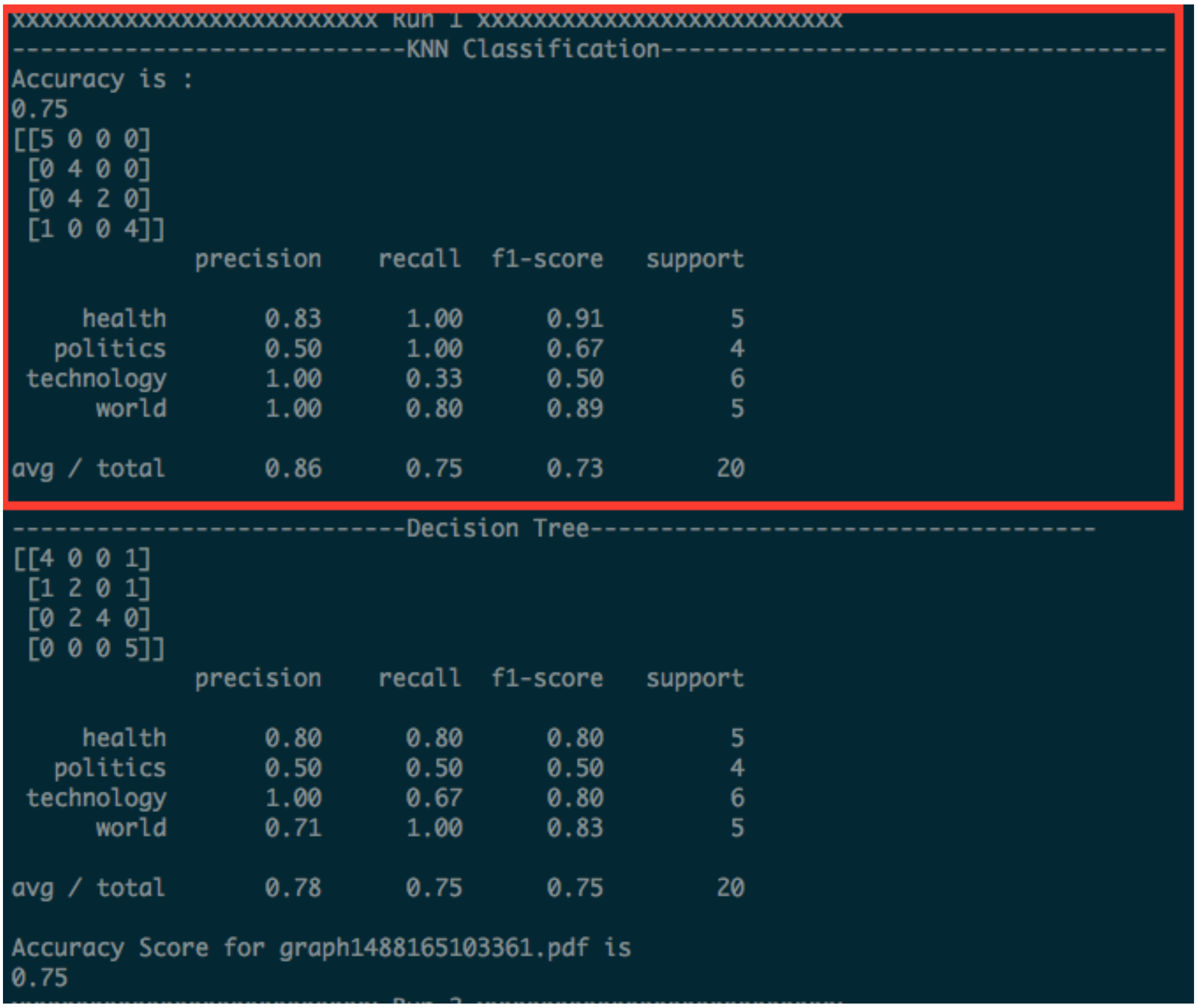


```
xxxxxxxxxxxxxxxxxxxxxxxxxx Run 1 xxxxxxxxxxxxxxxxxxxxxxxxxx
----------------------------KNN Classification-----------------------------------
Accuracy is :
0.75
[[5 0 0 0]
 [0 4 0 0]
 [0 4 2 0]
 [1 0 0 4]]
             precision    recall  f1-score   support

     health       0.83      1.00      0.91         5
   politics       0.50      1.00      0.67         4
 technology       1.00      0.33      0.50         6
      world       1.00      0.80      0.89         5

avg / total       0.86      0.75      0.73        20

----------------------------Decision Tree----------------------------------
[[4 0 0 1]
 [1 2 0 1]
 [0 2 4 0]
 [0 0 0 5]]
             precision    recall  f1-score   support

     health       0.80      0.80      0.80         5
   politics       0.50      0.50      0.50         4
 technology       1.00      0.67      0.80         6
      world       0.71      1.00      0.83         5

avg / total       0.78      0.75      0.75        20

Accuracy Score for graph1488165103361.pdf is
0.75
```



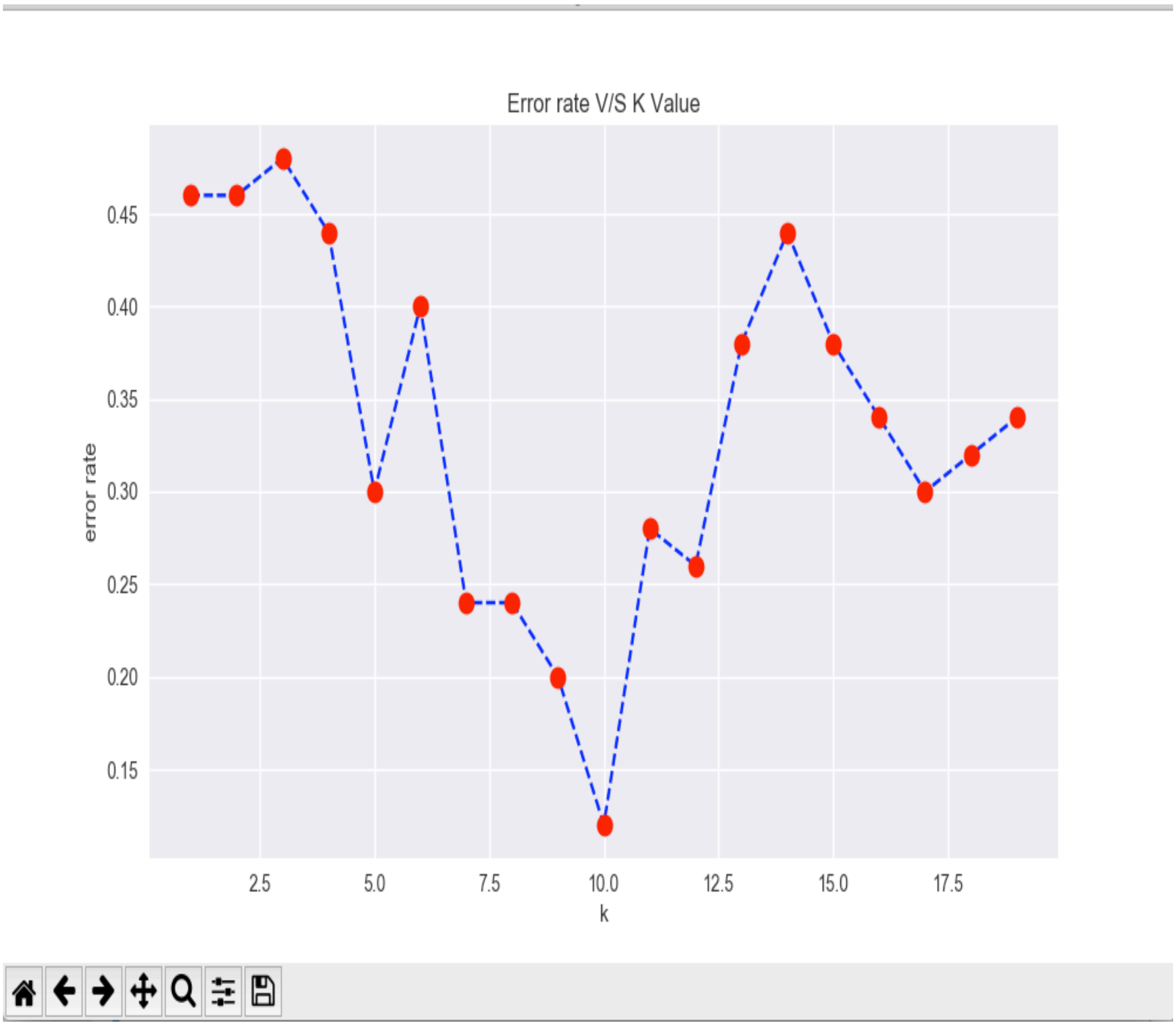
- All the graph files have been provided with the data.

b. **KNN Algorithm:** We will take a sample execution run and demonstrate data analysis for KNN algorithm



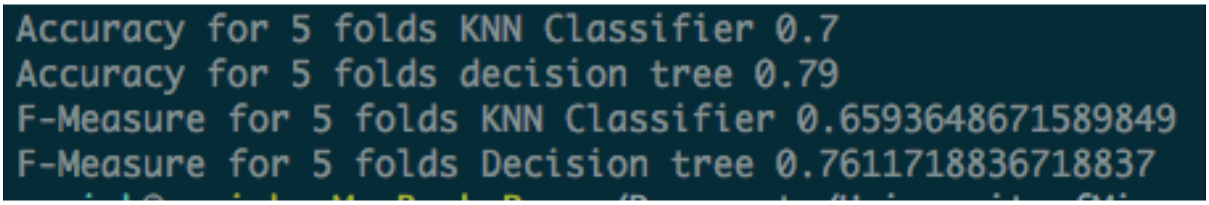
```
xxxxxxxxxxxxxxxxxxxxxxxxxx Run 1 xxxxxxxxxxxxxxxxxxxxxxxxxx
---------------------------KNN Classification-----------------------------------
Accuracy is :
0.75
[[5 0 0 0]
 [0 4 0 0]
 [0 4 2 0]
 [1 0 0 4]]
             precision    recall  f1-score   support

     health       0.83      1.00      0.91         5
   politics       0.50      1.00      0.67         4
 technology       1.00      0.33      0.50         6
      world       1.00      0.80      0.89         5

avg / total       0.86      0.75      0.73        20
---------------------------Decision Tree----------------------------------
[[4 0 0 1]
 [1 2 0 1]
 [0 2 4 0]
 [0 0 0 5]]
             precision    recall  f1-score   support

     health       0.80      0.80      0.80         5
   politics       0.50      0.50      0.50         4
 technology       1.00      0.67      0.80         6
      world       0.71      1.00      0.83         5

avg / total       0.78      0.75      0.75        20

Accuracy Score for graph1488165103361.pdf is
0.75
xxxxxxxxxxxxxxxxxxxxxxxxxx Run 2 xxxxxxxxxxxxxxxxxxxxxxxxxx
```

- The highlighted part shows all the parameters for KNN algorithm run.
- First is the accuracy score.
- Second comes the confusion matrix
- Third comes the precision, recall F-Measure and support for each category as well as average for one run.
- The initial execution of the algorithm is for k Value 4.
- I ran KNN for values 1 to 20 and plotted the graph to record the error rate for each execution.
- Found that for value of K = 10 the error rate was least and this only gives us the best classification results.


Error rate V/S K Value
error rate
0.45
0.40
0.35
0.30
0.25
0.20
0.15
2.5
5.0
7.5
10.0
12.5
15.0
17.5
k

Comparison and analysis:

The comparative analysis of the f measure and accuracy for Decision tree and KNN algorithm are as below:-



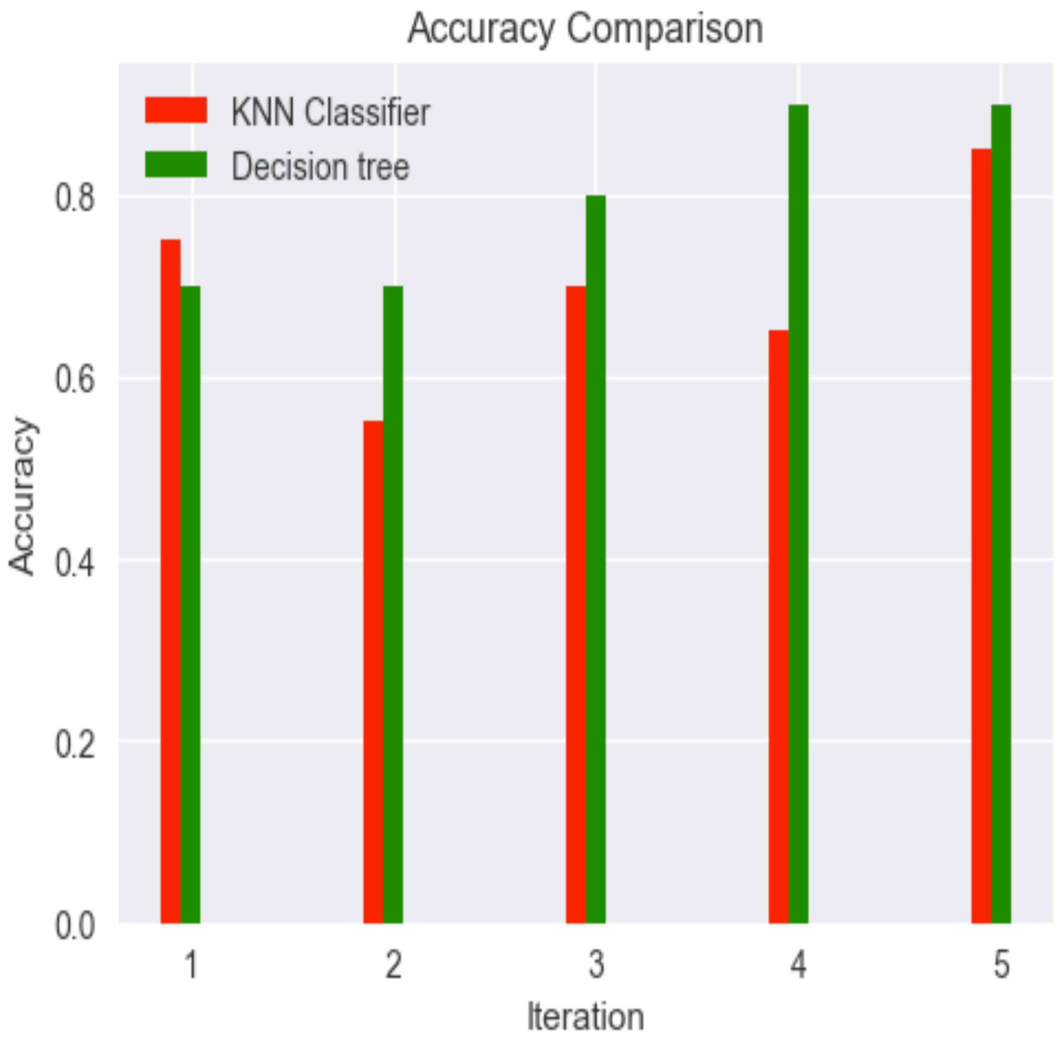
```
Accuracy for 5 folds KNN Classifier 0.7
Accuracy for 5 folds decision tree 0.79
F-Measure for 5 folds KNN Classifier 0.6593648671589849
F-Measure for 5 folds Decision tree 0.7611718836718837
```

Clearly accuracy for decision tree algorithm is 79% and for KNN classifier algorithm is 70%. I chose the value of K neighbours as 9 as this gave maximum accuracy and minimum error rate for the algorithm.

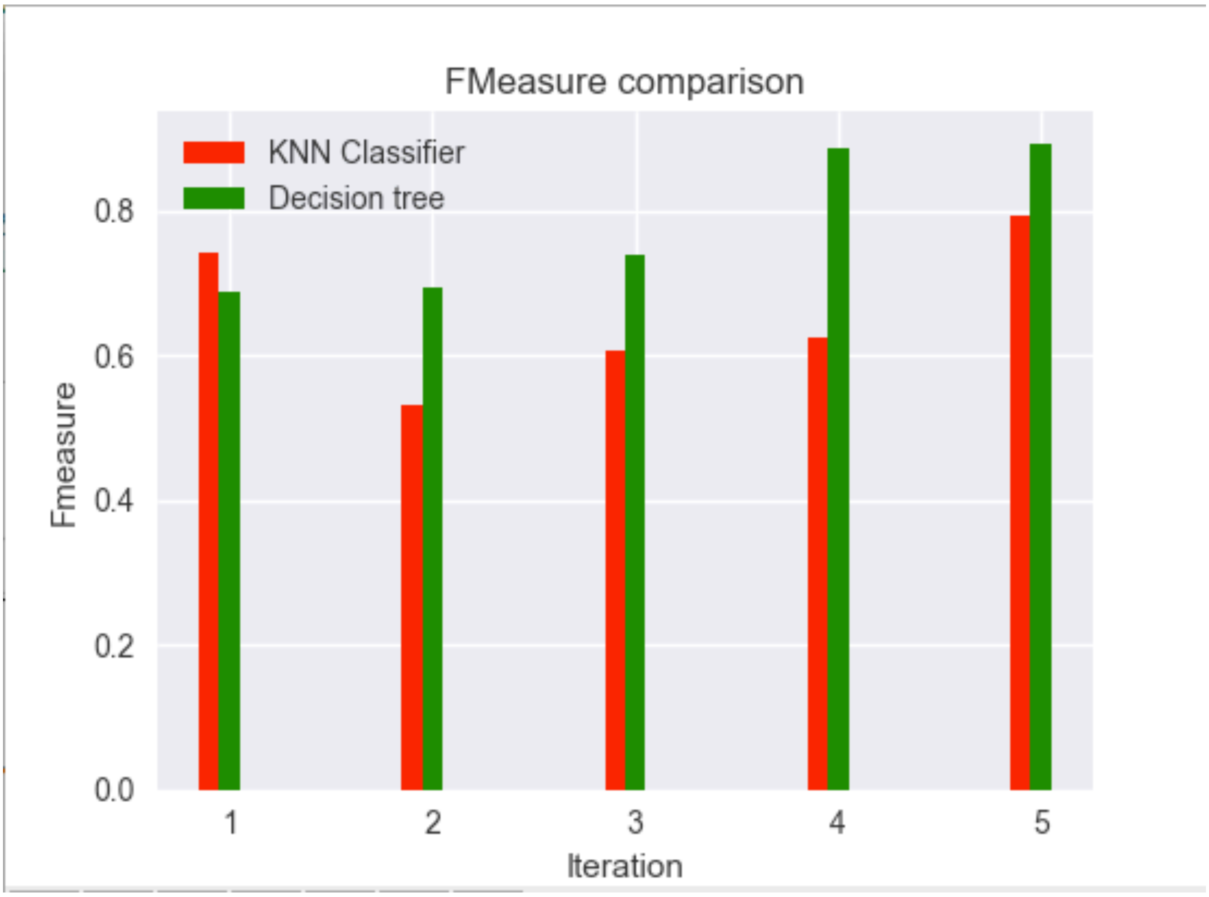
NOTE – Accuracy and F-Measure for KNN algorithm varies as per the number of neighbour chosen and is plotted against error rate as show in graph above.

Below are the Comparison graphs for accuracy and f-Measure for decision tree and KNN algorithms:

Accuracy comparison:



F Measure Comparison



7. Conclusion: Clearly the results are **better classified using decision tree method**. Hence decision tree method is the better method for data analysis out of the two methods. Here we also have to make a note about the correct K value which is required in KNN algorithm. The results of the algorithm vary with a different K value and hence the K value should be chose only after calculating error rate for a range of k values.