




# TEXT CLASSIFICATION

Presented By : NITIN YADAV  
(16212304)



# Introduction: Motivation

- The process of identifying the class to which a text document belongs.
- The process get succeeded by comparing the content of the document with some predefined categories.
- Organize data automatically, where manual organisation of the data is impossible.
- A method to solve the real world problems.

# Problem Definition?

- This project involves writing an implementation from scratch of a basic multi-class text classification system based on a k-Nearest Neighbor (KNN) classifier. The project has the following tasks:
  1. Implement an unweighted k-NN classifier which uses an appropriate similarity measure for working with text data represented as a document-term matrix. The implementation is suitable for multi-class classification (i.e. a dataset with more than 2 class labels). The number of neighbors is user-specified parameter.
  2. There is an additional option for the classifier to allow weighted majority voting (i.e. a weighted KNN classifier), using an appropriate weighting scheme.
  3. It contains an evaluation measure to find overall accuracy of the classifiers on labelled data. Accuracy of both the unweighted and weighted k-NN classifiers for different values of k in the range [1,10] on the sample dataset is evaluated.

# Existing Methods?

- Decision Tree Classifier
- Probabilistic and Naïve Bayes Classifier
- SVM classifier
- KNN Classification

# Proposed Method?

- KNN Classification : Weighted vs Unweighted
  - Method 1 : Unweighted KNN Classification: A KNN classifier in its simplest form works under the assumption that all features are of equal value when, the classification problem at hand is concerned.
  - Problem of Unweighted: Irrelevant and noisy features influence the neighborhood search to the same degree as highly relevant features, the accuracy of the model is likely to deteriorate.
  - Method 2 : Weighted KNN Classification: Weighted method is used for the approximation of the optimal degree of influence of individual features using a training set. When it is successfully applied, relevant features are attributed a high weight value, whereas irrelevant features are given a weight value close to zero. It increases the accuracy of the classifier.

# About Dataset

- Dataset contains two files. (<https://github.com/NitinYadav20/Machine-Learning/tree/master/data> )

## 1. File: news\_articles.mtx

- Description : This is the sparse document-term matrix stored in the Matrix Market format (<http://math.nist.gov/MatrixMarket/>), with rows representing 1839 unique documents and the columns representing 4882 unique terms. An entry  $(i,j)$  with value  $f$  in the matrix indicates that the term  $j$  appears in document  $i$  with frequency  $f$ . Zero values are not stored in the file.

## 2. File: news\_articles.labels

- Description : This is a text file which lists the class label for each of the 1839 documents. Each line in the text file contains a document number and the name of its class, separated by a comma. So the line “6,business” indicates that document 6 has the class label 'business', while the line “1835,technology” indicates that document 1835 has the class label 'technology'.

# Evaluation Method

```
def getAccuracy(test_set, predictions, labels):  
    correct = 0  
    for x in range(len(test_set)):  
        if labels[test_set[x][-1]] == predictions[x]:  
            correct += 1  
  
    return (correct/float(len(test_set))) * 100.0
```

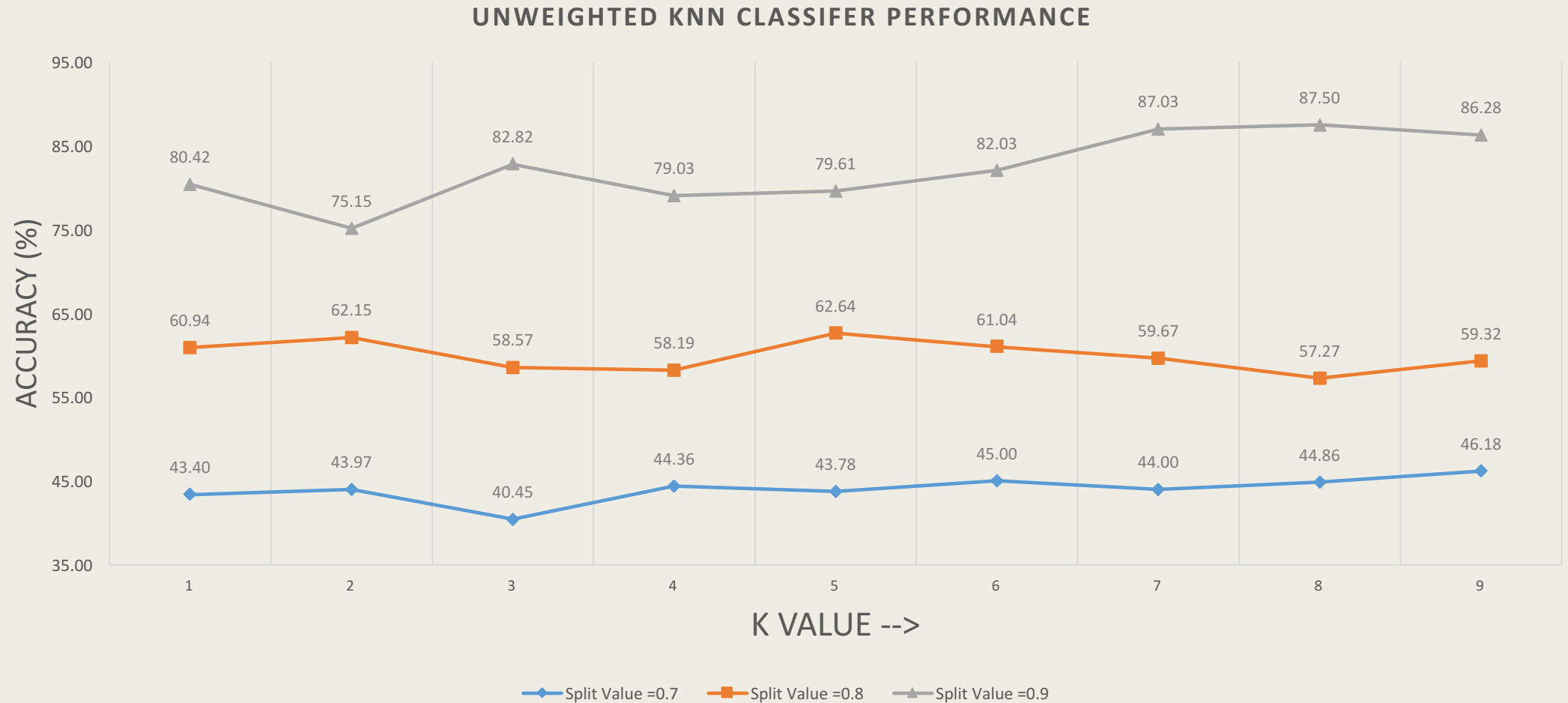
- Above the Accuracy() function calculates the ratio of the total correct predictions out of all predictions and sums the total correct predictions and returns the accuracy as a percentage of correct classifications.

# Question : Finding Optimal Value of K in KNN classifier.

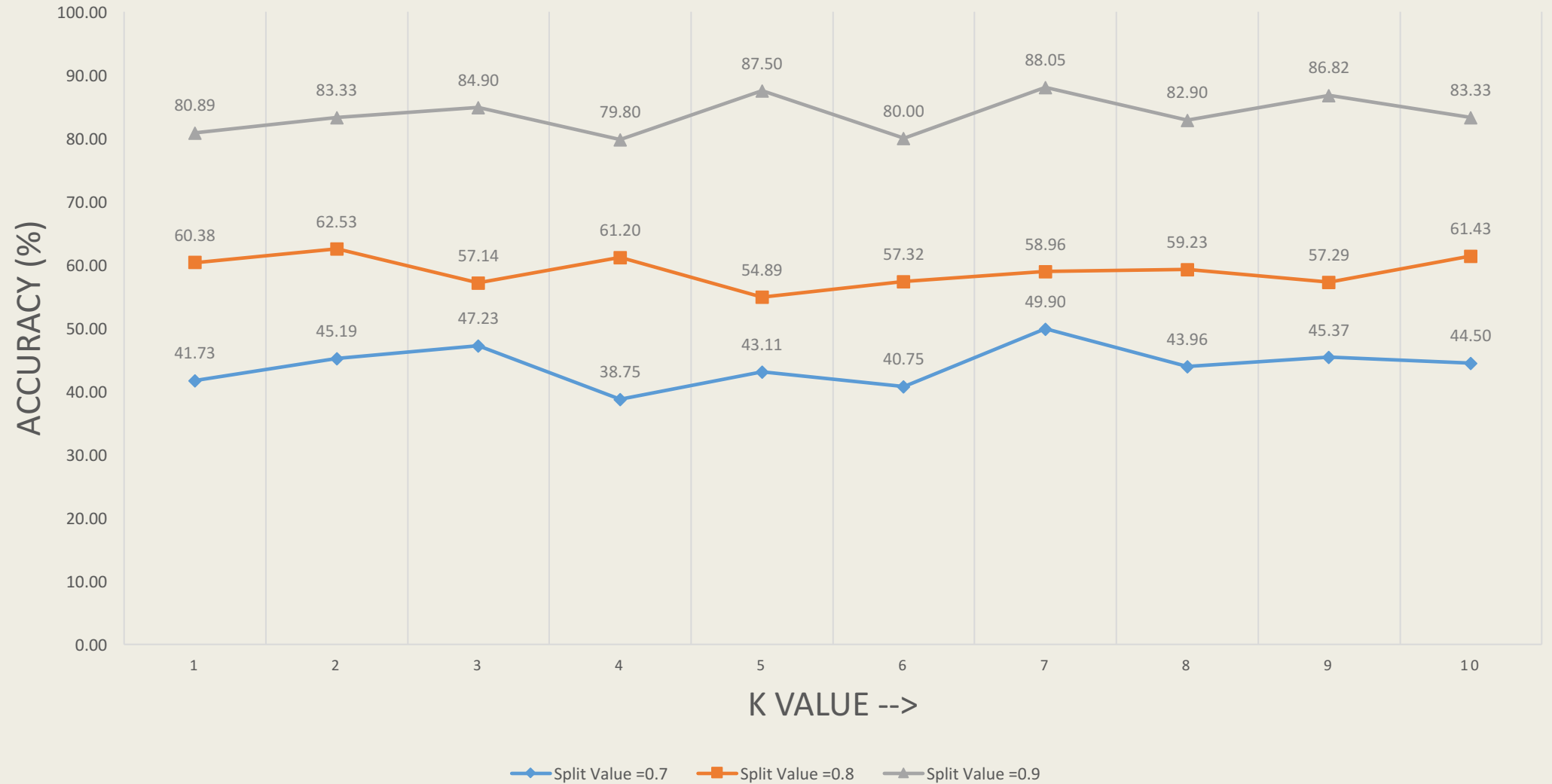
- Choice of k is very critical – A small value of k means that noise will have a higher influence on the result. A large value make it computationally expensive and kind of defeats the basic philosophy behind KNN (that points that are near might have similar densities or classes ). There is no rule of thumb to find the optimal value of K. Choice of K is driven by the end application as well as the dataset.



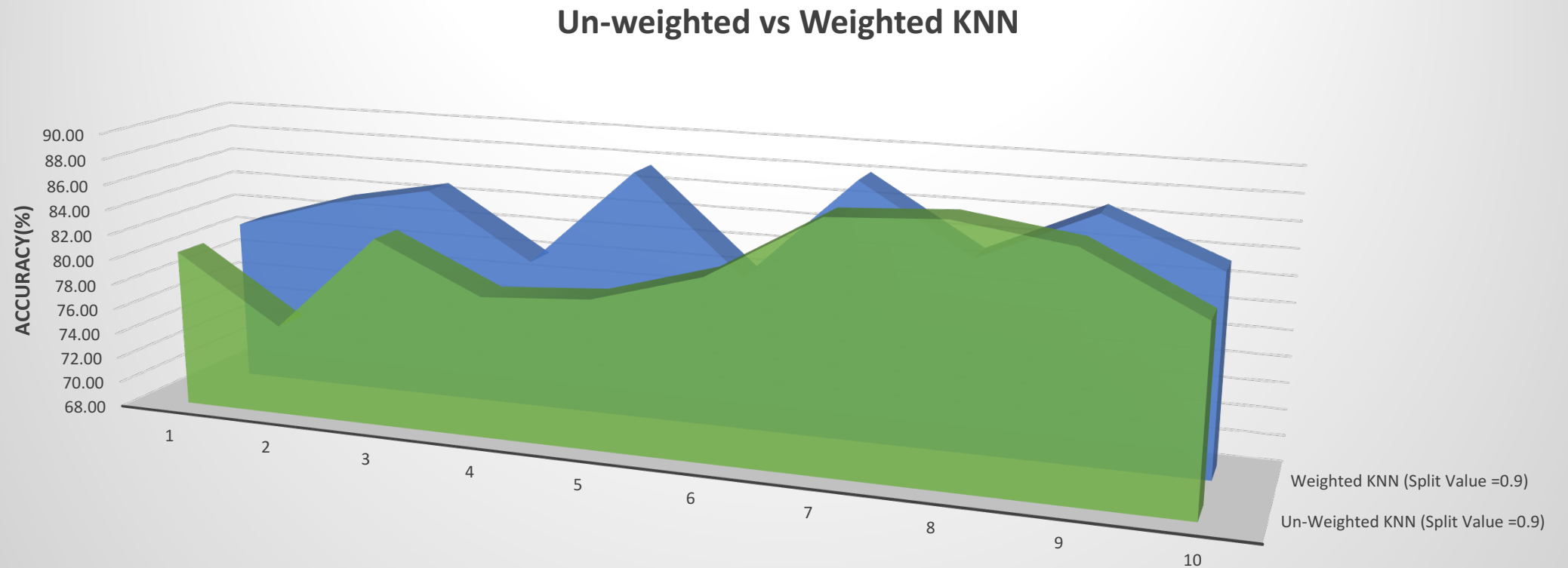
# Analysis



## PERFORMANCE OF WEIGHTED KNN CLASSIFIER



# Conclusion



	1	2	3	4	5	6	7	8	9	10
Un-Weighted KNN (Split Value =0.9)	80.42	75.15	82.82	79.03	79.61	82.03	87.03	87.50	86.28	82.00
Weighted KNN (Split Value =0.9)	80.89	83.33	84.90	79.80	87.50	80.00	88.05	82.90	86.82	83.33

Un-Weighted KNN (Split Value =0.9)    Weighted KNN (Split Value =0.9)

Any Queries?

[Nitin.yadav2@mail.dcu.ie](mailto:Nitin.yadav2@mail.dcu.ie)

Presentation Video: <https://youtu.be/VbqTw9uYJcM>

# Thank you