k-Nearest Neighbour algorithm

```
In [1]:
         from sklearn.model_selection import train_test_split
         from sklearn.neighbors import KNeighborsClassifier
         from sklearn.metrics import classification_report,confusion_matrix
In [2]:
         from sklearn import datasets
         iris=datasets.load_iris()
         iris_data=iris.data
         iris_label=iris.target
         print(iris_data)
         print(iris_label)
         x_train,x_test,y_train,y_test=train_test_split(iris_data,iris_label)
        [[5.1 3.5 1.4 0.2]
         [4.9 3. 1.4 0.2]
         [4.7 3.2 1.3 0.2]
         [4.6 3.1 1.5 0.2]
         [5. 3.6 1.4 0.2]
         [5.4 3.9 1.7 0.4]
         [4.6 3.4 1.4 0.3]
         [5. 3.4 1.5 0.2]
         [4.4 2.9 1.4 0.2]
         [4.9 3.1 1.5 0.1]
         [5.4 3.7 1.5 0.2]
         [4.8 3.4 1.6 0.2]
         [4.8 3. 1.4 0.1]
         [4.3 3. 1.1 0.1]
         [5.8 4. 1.2 0.2]
         [5.7 4.4 1.5 0.4]
         [5.4 3.9 1.3 0.4]
         [5.1 3.5 1.4 0.3]
         [5.7 3.8 1.7 0.3]
         [5.1 3.8 1.5 0.3]
         [5.4 3.4 1.7 0.2]
         [5.1 3.7 1.5 0.4]
         [4.6 3.6 1. 0.2]
         [5.1 3.3 1.7 0.5]
         [4.8 3.4 1.9 0.2]
         [5. 3. 1.6 0.2]
         [5. 3.4 1.6 0.4]
         [5.2 3.5 1.5 0.2]
         [5.2 3.4 1.4 0.2]
         [4.7 3.2 1.6 0.2]
         [4.8 3.1 1.6 0.2]
         [5.4 3.4 1.5 0.4]
         [5.2 4.1 1.5 0.1]
         [5.5 4.2 1.4 0.2]
         [4.9 3.1 1.5 0.2]
         [5. 3.2 1.2 0.2]
         [5.5 3.5 1.3 0.2]
         [4.9 3.6 1.4 0.1]
         [4.4 3. 1.3 0.2]
         [5.1 3.4 1.5 0.2]
         [5. 3.5 1.3 0.3]
         [4.5 2.3 1.3 0.3]
         [4.4 3.2 1.3 0.2]
         [5. 3.5 1.6 0.6]
         [5.1 3.8 1.9 0.4]
         [4.8 3. 1.4 0.3]
```

[5.1 3.8 1.6 0.2] [4.6 3.2 1.4 0.2] [5.3 3.7 1.5 0.2]

```
[5. 3.3 1.4 0.2]
[7. 3.2 4.7 1.4]
[6.4 3.2 4.5 1.5]
[6.9 3.1 4.9 1.5]
[5.5 2.3 4. 1.3]
[6.5 2.8 4.6 1.5]
[5.7 2.8 4.5 1.3]
[6.3 3.3 4.7 1.6]
[4.9 2.4 3.3 1. ]
[6.6 2.9 4.6 1.3]
[5.2 2.7 3.9 1.4]
[5. 2. 3.5 1.]
[5.9 3. 4.2 1.5]
[6. 2.2 4. 1.]
[6.1 2.9 4.7 1.4]
[5.6 2.9 3.6 1.3]
[6.7 3.1 4.4 1.4]
[5.6 3. 4.5 1.5]
[5.8 2.7 4.1 1. ]
[6.2 2.2 4.5 1.5]
[5.6 2.5 3.9 1.1]
[5.9 3.2 4.8 1.8]
[6.1 2.8 4. 1.3]
[6.3 2.5 4.9 1.5]
[6.1 2.8 4.7 1.2]
[6.4 2.9 4.3 1.3]
[6.6 3. 4.4 1.4]
[6.8 2.8 4.8 1.4]
[6.7 3. 5. 1.7]
[6. 2.9 4.5 1.5]
[5.7 2.6 3.5 1.]
[5.5 2.4 3.8 1.1]
[5.5 2.4 3.7 1.]
[5.8 2.7 3.9 1.2]
[6. 2.7 5.1 1.6]
[5.4 3. 4.5 1.5]
[6. 3.4 4.5 1.6]
[6.7 3.1 4.7 1.5]
[6.3 2.3 4.4 1.3]
[5.6 3. 4.1 1.3]
[5.5 2.5 4. 1.3]
[5.5 2.6 4.4 1.2]
[6.1 3. 4.6 1.4]
[5.8 2.6 4. 1.2]
[5. 2.3 3.3 1.]
[5.6 2.7 4.2 1.3]
[5.7 3. 4.2 1.2]
[5.7 2.9 4.2 1.3]
[6.2 2.9 4.3 1.3]
[5.1 2.5 3. 1.1]
[5.7 2.8 4.1 1.3]
[6.3 3.3 6. 2.5]
[5.8 2.7 5.1 1.9]
[7.1 3. 5.9 2.1]
[6.3 2.9 5.6 1.8]
[6.5 3. 5.8 2.2]
[7.6 3. 6.6 2.1]
[4.9 2.5 4.5 1.7]
[7.3 2.9 6.3 1.8]
[6.7 2.5 5.8 1.8]
[7.2 3.6 6.1 2.5]
[6.5 3.2 5.1 2. ]
[6.4 2.7 5.3 1.9]
[6.8 3. 5.5 2.1]
[5.7 2.5 5. 2.]
[5.8 2.8 5.1 2.4]
[6.4 3.2 5.3 2.3]
```

[6.5 3. 5.5 1.8] [7.7 3.8 6.7 2.2]

```
[7.7 2.6 6.9 2.3]
       [6. 2.2 5. 1.5]
       [6.9 3.2 5.7 2.3]
       [5.6 2.8 4.9 2. ]
       [7.7 2.8 6.7 2. ]
       [6.3 2.7 4.9 1.8]
       [6.7 3.3 5.7 2.1]
       [7.2 3.2 6. 1.8]
       [6.2 2.8 4.8 1.8]
       [6.1 3. 4.9 1.8]
       [6.4 2.8 5.6 2.1]
       [7.2 3. 5.8 1.6]
       [7.4 2.8 6.1 1.9]
       [7.9 3.8 6.4 2. ]
       [6.4 2.8 5.6 2.2]
       [6.3 2.8 5.1 1.5]
       [6.1 2.6 5.6 1.4]
       [7.7 3. 6.1 2.3]
       [6.3 3.4 5.6 2.4]
       [6.4 3.1 5.5 1.8]
       [6. 3. 4.8 1.8]
       [6.9 3.1 5.4 2.1]
       [6.7 3.1 5.6 2.4]
       [6.9 3.1 5.1 2.3]
       [5.8 2.7 5.1 1.9]
       [6.8 3.2 5.9 2.3]
       [6.7 3.3 5.7 2.5]
       [6.7 3. 5.2 2.3]
       [6.3 2.5 5. 1.9]
       [6.5 3. 5.2 2.]
       [6.2 3.4 5.4 2.3]
       [5.9 3. 5.1 1.8]]
       2 2]
In [3]:
       classifier=KNeighborsClassifier(n_neighbors=5)
       classifier.fit(x_train,y_train)
       y_pred=classifier.predict(x_test)
       print('Confusion matrix is as follows')
       print(confusion_matrix(y_test,y_pred))
       print('Accuracy Metrics')
       print(classification_report(y_test,y_pred))
       Confusion matrix is as follows
       [[13 0 0]
       [ 0 13 0]
       [ 0 0 12]]
      Accuracy Metrics
                  precision
                            recall f1-score
                                            support
               0
                      1.00
                              1.00
                                      1.00
                                                13
               1
                      1.00
                              1.00
                                      1.00
                                                13
               2
                      1.00
                              1.00
                                      1.00
                                                12
          accuracy
                                      1.00
                                                38
         macro avg
                      1.00
                              1.00
                                      1.00
                                                38
      weighted avg
                      1.00
                              1.00
                                      1.00
                                                38
In [ ]:
```