Name: Sagar Patil
Roll No: 8061

Assignment 3

Title : In the following diagram let blue circles indicate positive examples and orange squares indicate negative examples. We want to use the k-NN algorithm for classifying the points. If k=3, find the class of the point (6,6).

Objectives : After completion of this assignment students are able Implement code for KNN Classification for Classify Positive and Negative Points in given example also Extend the same example for Distance-Weighted k-NN and locally weighted Averaging

Theory : KNN can be used for both classification and regression predictive problems. However, it is more widely used in classification problems in the industry. To evaluate any technique we generally look at 3 important aspects:

1. Ease to interpret output

2. Calculation time

3. Predictive Power

The KNN Algorithm

1. Load the data
2. Initialize K to your chosen number of neighbors
3. For each example in the data

   3.1 Calculate the distance between the query example and the current example from the data.

   3.2 Add the distance and the index of the example to an ordered collection.
4. Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances
5. Pick the first K entries from the sorted collection
6. Get the labels of the selected K entries
7. If regression, return the mean of the K labels
8. If classification, return the mode of the K labels

Code and Output

```
import pandas as pd
import numpy as np

#Read dataset
dataset = pd.read_csv("kdata.csv")
X = dataset.iloc[:,:-1].values
y = dataset.iloc[:,2].values

#import KNeighborshood Classifier and create object of it
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors=3)
classifier.fit(X,y)

#predict the class for the point(6,6)
X_test = np.array([6,6])
y_pred = classifier.predict([X_test])
print('General KNN',y_pred)

classifier = KNeighborsClassifier(n_neighbors=3,weights='distance')
classifier.fit(X,y)

#predict the class for the point(6,6)
X_test = np.array([6,2])
y_pred = classifier.predict([X_test])
print('Distance Weighted KNN',y_pred)
'''
OUTPUT
General KNN ['negative']
Distance Weighted KNN ['positive']

'''
```

Conclusion : In this way we learn KNN Classification to predict the General and Distance Weighted KNN for Given data point in terms of Positive or Negative.