Lab-9: Classification with K-Nearest Neighbors

9.1 Objectives:

1. To learn and Implement K-Nearest Neighbor machine learning technique

9.2 K-Nearest Neighbors

KNN is a simple model for regression and classification tasks. It is so simple that its name describes most of its learning algorithm. The titular neighbors are representations of training instances in a metric space. A metric space is a feature space in which the distances between all members of a set are defined. In the previous chapter's pizza problem, our training instances were represented in a metric space because the distances between all the pizza diameters was defined. These neighbors are used to estimate the value of the response variable for a test instance. The hyperparameter k specifies how many neighbors can be used in the estimation. A hyperparameter is a parameter that controls how the algorithm learns; hyperparameters are not estimated from the training data and are sometimes set manually. Finally, the k neighbors that are selected are those that are nearest to the test instance, as measured by some distance function.

For classification tasks, a set of tuples of feature vectors and class labels comprise the training set. KNN is a capable of binary, multi-class, and multi-label classification; we will define these tasks later, and we will focus on binary classification in this chapter. The simplest KNN classifiers use the mode of the KNN labels to classify test instances, but other strategies can be used. The k is often set to an odd number to prevent ties.

In regression tasks, the feature vectors are each associated with a response variable that takes a real-valued scalar instead of a label. The prediction is the mean or weighted mean of the KNN response variables.

Example:

Suppose there are some shaded red and blue circles that are labeled as '+' and '-' respectively, as shown in Figure 4. In order to determine the class of a test sample shown as '?' with k=5, the votes of five nearest neighbors are considered. In this case, as nearest neighbors to both test samples are mostly red shaded circles, so both of them are identified as '+'.

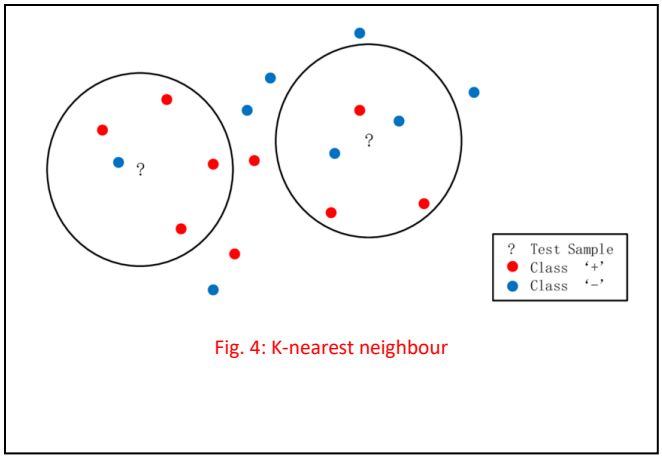

Fig. 4: K-nearest neighbour

Figure 1: K-NN example

9.1.1 Implementation of K-NN Classifier

Dataset

The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant.

Attribute Information:

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm
5. class:
-- Iris Setosa
-- Iris Versicolour
-- Iris Virginica

	A	B	C	D	E
1	sepal_length	sepal_width	petal_length	petal_width	species
2	5.1	3.5	1.4	0.2	setosa
3	5.7	2.8	4.1	1.3	versicolor
4	6.3	3.3	6	2.5	virginica

Figure 2: sample of dataset

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
data=pd.read_csv(r'F:\path\iris_data_2.csv')
print(data.head())

x=data.drop('species','columns')
y=data['species']
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20)
#print(x_train)
scaler=StandardScaler()
scaler.fit(x_train)
#print(x_train)
x_train=scaler.transform(x_train)
x_test=scaler.transform(x_test)
print(x_train)
print(x_test)
from sklearn.neighbors import KNeighborsClassifier
classifier=KNeighborsClassifier(n_neighbors=3)
classifier.fit(x_train,y_train)
result=classifier.predict(x_test)
print(result)
from sklearn.metrics import confusion_matrix,classification_report
print(classification_report(y_test,result))
print(confusion_matrix(y_test,result))
```

TASK: Implement support vector machine for the same classification problem

1. Attach the screenshot of the code and output