**Name:- Prajwal Santosh Mankar**

**Roll No: - 96**

**Practical No: 6**

**Practical Name: Write a program to implement the k-Nearest Neighbour algorithm to classify the iris dataset. Print both correct and wrong predictions. Java/Python ML library classes can be used for this problem**

**------------------------------------------------------------------------------------------------**

**Code: For Iris Data Set**

from pandas import DataFrame

from sklearn.datasets import load\_iris

#from sklearn.datasets import load\_breast\_cancer

from sklearn.neighbors import KNeighborsClassifier

from sklearn import metrics

from sklearn.metrics import confusion\_matrix

from sklearn.model\_selection import train\_test\_split

data\_b = load\_iris()

#data\_b = load\_breast\_cancer()

df = DataFrame(data\_b.data, columns=data\_b.feature\_names)

df['target'] = data\_b.target

# print(df)

# print(data\_b.DESCR)

print("Dataset Labels=", data\_b.target\_names)

X\_train, X\_test, Y\_train, y\_test = train\_test\_split(df[data\_b.feature\_names], df['target'], random\_state=1)

print(X\_train.head(6))

print(Y\_train.head(6))

print(X\_test.head())

clf = KNeighborsClassifier(n\_neighbors=6)

clf.fit(X\_train, Y\_train) # model is trained

y\_pred = clf.predict(X\_test)

# print(y\_test, y\_pred)

print("Accuracy:", metrics.accuracy\_score(y\_test, y\_pred))

cm = confusion\_matrix(y\_test, y\_pred)

print("Confusion Matrix:")

print(cm)

# corr = cm[0, 0] + cm[1, 1] + cm[2, 2] # ----for iris

# corr = cm[0, 0] + cm[1, 1] #----for breast cancer

corr = 0

for i in range(len(data\_b.target\_names)):

corr = corr + cm[i, i]

wrg = len(y\_test) - corr

print("Number of correct predictions=", corr)

print("Number of wrong predictions = ", wrg)

**OUTPUT:**

C:\Users\sejal\MCA-I\_ML\Scripts\python.exe C:/Users/sejal/PycharmProjects/MCA-I\_ML/KNN.py

Dataset Labels= ['setosa' 'versicolor' 'virginica']

sepal length (cm) sepal width (cm) petal length (cm) petal width (cm)

54 6.5 2.8 4.6 1.5

108 6.7 2.5 5.8 1.8

112 6.8 3.0 5.5 2.1

17 5.1 3.5 1.4 0.3

119 6.0 2.2 5.0 1.5

103 6.3 2.9 5.6 1.8

54 1

108 2

112 2

17 0

119 2

103 2

Name: target, dtype: int32

sepal length (cm) sepal width (cm) petal length (cm) petal width (cm)

14 5.8 4.0 1.2 0.2

98 5.1 2.5 3.0 1.1

75 6.6 3.0 4.4 1.4

16 5.4 3.9 1.3 0.4

131 7.9 3.8 6.4 2.0

Accuracy: 1.0

Confusion Matrix:

[[13 0 0]

[ 0 16 0]

[ 0 0 9]]

Number of correct predictions= 38

Number of wrong predictions = 0

Process finished with exit code 0

**Code: For Breast Cancer Data Set**

from pandas import DataFrame

#from sklearn.datasets import load\_iris

from sklearn.datasets import load\_breast\_cancer

from sklearn.neighbors import KNeighborsClassifier

from sklearn import metrics

from sklearn.metrics import confusion\_matrix

from sklearn.model\_selection import train\_test\_split

#data\_b = load\_iris()

data\_b = load\_breast\_cancer()

df = DataFrame(data\_b.data, columns=data\_b.feature\_names)

df['target'] = data\_b.target

# print(df)

# print(data\_b.DESCR)

print("Dataset Labels=", data\_b.target\_names)

X\_train, X\_test, Y\_train, y\_test = train\_test\_split(df[data\_b.feature\_names], df['target'], random\_state=1)

print(X\_train.head(6))

print(Y\_train.head(6))

print(X\_test.head())

clf = KNeighborsClassifier(n\_neighbors=6)

clf.fit(X\_train, Y\_train) # model is trained

y\_pred = clf.predict(X\_test)

# print(y\_test, y\_pred)

print("Accuracy:", metrics.accuracy\_score(y\_test, y\_pred))

cm = confusion\_matrix(y\_test, y\_pred)

print("Confusion Matrix:")

print(cm)

# corr = cm[0, 0] + cm[1, 1] + cm[2, 2] # ----for iris

# corr = cm[0, 0] + cm[1, 1] #----for breast cancer

corr = 0

for i in range(len(data\_b.target\_names)):

corr = corr + cm[i, i]

wrg = len(y\_test) - corr

print("Number of correct predictions=", corr)

print("Number of wrong predictions = ", wrg)

**OUTPUT:**

C:\Users\sejal\MCA-I\_ML\Scripts\python.exe C:/Users/sejal/PycharmProjects/MCA-I\_ML/KNN.py

Dataset Labels= ['malignant' 'benign']

mean radius mean texture ... worst symmetry worst fractal dimension

562 15.22 30.62 ... 0.4089 0.14090

291 14.96 19.10 ... 0.2962 0.08472

16 14.68 20.13 ... 0.3029 0.08216

546 10.32 16.35 ... 0.2681 0.07399

293 11.85 17.46 ... 0.3101 0.07007

350 11.66 17.07 ... 0.2731 0.06825

[6 rows x 30 columns]

562 0

291 1

16 0

546 1

293 1

350 1

Name: target, dtype: int32

mean radius mean texture ... worst symmetry worst fractal dimension

421 14.69 13.98 ... 0.2827 0.09208

47 13.17 18.66 ... 0.3900 0.11790

292 12.95 16.02 ... 0.3380 0.09584

186 18.31 18.58 ... 0.3206 0.06938

414 15.13 29.81 ... 0.3233 0.06165

[5 rows x 30 columns]

Accuracy: 0.9370629370629371

Confusion Matrix:

[[51 4]

[ 5 83]]

Number of correct predictions= 134

Number of wrong predictions = 9

Process finished with exit code 0