**Ex.no:7 DHAMODARAN.B**

**Date:13.09.2020** **1832016**

**K-NEAREST NEIGHBORS CLASSIFIER**

**Problem Statement:**

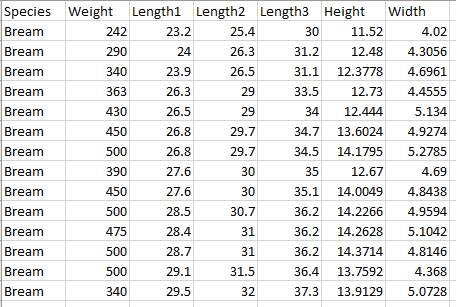
Fishes are one of the greatest sources of protein and the total turnover of the fish industry is about 45,000 crore Rupees. Thus, ensuring the quality of fish is necessary. Since the fishes are priced on their weight and how rare the species is, detecting the species of the fish is necessary to determine the price. Our motto is to predict the species of the fish with the physical measurements of the fish. By predicting the species of the fish, we can determine the price of the fish. This can be achieved by using K-nearest neighbor Classifier.

**Problem Description:**

The dataset had data on the body measurements of the fish such as width, height and breadth, weight etc... It is pre-known that the different species vary in their physical characteristics and body measurements too. The dependent variable or the target variable that we want to predict is ‘species’ and the independent variables are the ‘weight’, ‘length1’, ’length2’, ’lenght3’, ’height’ and ‘width’. All the independent features were continuous in nature. The dataset had information on 7 species but we have taken only two species i.e. ‘Bream’ and ‘Perch’. The information on two species ‘Bream’ and ‘Perch’ had about 91 data instances.

Initially, the data was checked for null values. The data are scaled using the standard scaling technique and later it was split into train and test in the ratio 85:15. This ratio of splitting had been done since the size of the dataset is quite less for training. Scaled data helps in making the computation faster. Thus, successfully preprocessed data can now be used for fitting of model. The K-nearest neighbor classifier model is implemented using the scikit’s learn’ s KNeighborsClassifier function. The model is validated under different metrics with the help of the test/validation dataset..

**Sample Dataset:**



**Code:**

**K-Nearest Neighbor Classifier:**

import pandas as pd

import numpy as np

dataset=pd.read\_csv('Fish.csv')

datasetprint("The Different Species are :",list(dataset['Species'].unique()))

print("The data for the species Bream and Perch are :")

dataframe=pd.DataFrame(dataset[dataset['Species'].isin(['Bream','Perch'])])

dataframe.index=range(len(dataframe))

dataframe.isnull()

from sklearn.preprocessing import StandardScaler

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

from sklearn.metrics import classification\_report

from sklearn.metrics import confusion\_matrix

scaler=StandardScaler()

dataframe.iloc[:,1:]=scaler.fit\_transform(dataframe.iloc[:,1:])

xtrain,xtest,ytrain,ytest=train\_test\_split(dataframe.iloc[:,1:].values,dataframe['Species'].values,test\_size=0.15)

from sklearn.neighbors import KNeighborsClassifier

knn\_model=KNeighborsClassifier()

knn\_model.fit(xtrain,ytrain)

ypred=knn\_model.predict(xtest)

print("Predictions \n",ypred)

print("True Values \n",ytest)

conf\_mat=confusion\_matrix(ytest,ypred)

print("Confusion Matrix \n",conf\_mat)

print(classification\_report(ytest,ypred))

print("Accuracy : " ,(conf\_mat[0][0]+conf\_mat[1][1])/len(ytest) )

probs=knn\_model.predict\_proba(xtest)

probs=probs[:,1]

from sklearn.metrics import roc\_curve

from sklearn.metrics import roc\_auc\_score

import matplotlib.pyplot as plt

from sklearn.preprocessing import LabelBinarizer

binarizer=LabelBinarizer()

ytest\_b=binarizer.fit\_transform(ytest)

fpr,tpr,\_=roc\_curve(ytest\_b,probs)

random\_probs = [0 for \_ in range(len(ytest))]

p\_fpr,p\_tpr,\_ = roc\_curve(ytest\_b,random\_probs)

auc\_score=roc\_auc\_score(ytest,probs)

print("AUC SCORE : " ,auc\_score)

plt.plot(p\_fpr, p\_tpr, linestyle='--')

plt.plot(fpr, tpr, marker='.', label='KNN Classifier (area=%0.2f)'% auc\_score)

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title("ROC-AUC CURVE for KNN Classifier")

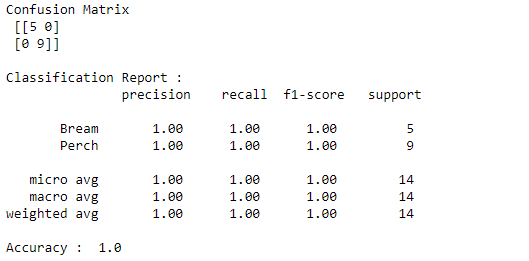
plt.legend()

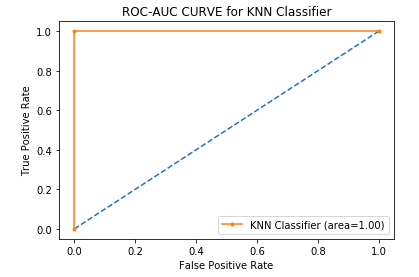
plt.show()

**Output:**









**Inference**:

The K-nearest neighbor classifier has produced an accuracy of 100% which implements that the model correctly classifies all the test instances. Precision is the ratio of true positive to the total no. of instances classified as positive. Precision of 100% for Bream class indicates that the all the instances classified as Bream are actually Bream in nature. Also, the precision for Perch species is 100% which indicates that all the instances classified as Perch are actually Perch in reality and there were no misclassifications. The AUC score of the model is 1.00 which implies that the model is very well capable of the distinguishing the species.

**Conclusion:**

Looking at the performance of the model, with the help of AUC score, we can conclude that the model is very well capable of distinguishing the species and also a precision of 100% for both the species makes the model’s predictions more reliable. Thus, our classifier model implemented with K-Nearest neighbors can be used for classifying the fish species to determine the prices.