EXPERIMENT 8

KNN

Aim:

To understand the importance of KNN.

Algorithm:

1. Import the required libraries — pandas, train\_test\_split, KNeighborsClassifier, and evaluation metrics.

2. Load the Iris dataset using pd.read\_csv().

3. Separate the dataset into features (X) and labels (Y).

4. Split the data into training and testing sets using train\_test\_split().

5. Initialize the KNN classifier with n\_neighbors = 5.

6. Train the model using the training data with fit().

7. Evaluate model performance by printing training and testing accuracy.

8. Generate and display the confusion matrix and classification report for detailed performance analysis.

Program:

import pandas as pd

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

from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import confusion\_matrix, classification\_report

df = pd.read\_csv("C:\iris.csv")

features = df.iloc[:, :-1].values

labels = df.iloc[:, 4].values

xtrain, xtest, ytrain, ytest = train\_test\_split(features, labels, test\_size=0.2, random\_state=42)

model\_KNN = KNeighborsClassifier(n\_neighbors=5)

model\_KNN.fit(xtrain, ytrain)

print("Training Accuracy:", model\_KNN.score(xtrain, ytrain))

print("Testing Accuracy:", model\_KNN.score(xtest, ytest))

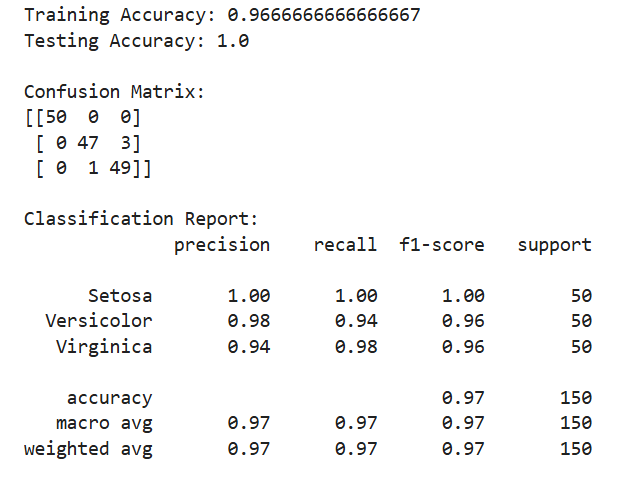
print("\nConfusion Matrix:")

print(confusion\_matrix(labels, model\_KNN.predict(features)))

print("\nClassification Report:")

print(classification\_report(labels, model\_KNN.predict(features)))

Output:



Result:

Hence a python program for KNN is written and executed successfully.