**# Implement Classification algorithm.**

import numpy as np

import pandas as pd

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

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score

from sklearn import datasets

from sklearn import svm

from sklearn.tree import DecisionTreeClassifier

from sklearn.naive\_bayes import GaussianNB

# import the iris dataset

iris = datasets.load\_iris()

X = iris.data

y = iris.target

# splitting X and y into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

X, y, test\_size=0.3, random\_state=1)

# GAUSSIAN NAIVE BAYES

gnb = GaussianNB()

# train the model

gnb.fit(X\_train, y\_train)

# make predictions

gnb\_pred = gnb.predict(X\_test)

# print the accuracy

print("Accuracy of Gaussian Naive Bayes: ",

accuracy\_score(y\_test, gnb\_pred))

# print other performance metrics

print("Precision of Gaussian Naive Bayes: ",

precision\_score(y\_test, gnb\_pred, average='weighted'))

print("Recall of Gaussian Naive Bayes: ",

recall\_score(y\_test, gnb\_pred, average='weighted'))

print("F1-Score of Gaussian Naive Bayes: ",

f1\_score(y\_test, gnb\_pred, average='weighted'))

# DECISION TREE CLASSIFIER

dt = DecisionTreeClassifier(random\_state=0)

# train the model

dt.fit(X\_train, y\_train)

# make predictions

dt\_pred = dt.predict(X\_test)

# print the accuracy

print("Accuracy of Decision Tree Classifier: ",

accuracy\_score(y\_test, dt\_pred))

# print other performance metrics

print("Precision of Decision Tree Classifier: ",

precision\_score(y\_test, dt\_pred, average='weighted'))

print("Recall of Decision Tree Classifier: ",

recall\_score(y\_test, dt\_pred, average='weighted'))

print("F1-Score of Decision Tree Classifier: ",

f1\_score(y\_test, dt\_pred, average='weighted'))

# SUPPORT VECTOR MACHINE

svm\_clf = svm.SVC(kernel='linear') # Linear Kernel

# train the model

svm\_clf.fit(X\_train, y\_train)

# make predictions

svm\_clf\_pred = svm\_clf.predict(X\_test)

# print the accuracy

print("Accuracy of Support Vector Machine: ",

accuracy\_score(y\_test, svm\_clf\_pred))

# print other performance metrics

print("Precision of Support Vector Machine: ",

precision\_score(y\_test, svm\_clf\_pred, average='weighted'))

print("Recall of Support Vector Machine: ",

recall\_score(y\_test, svm\_clf\_pred, average='weighted'))

print("F1-Score of Support Vector Machine: ",

f1\_score(y\_test, svm\_clf\_pred, average='weighted'))

**OUTPUT:**

Accuracy of Gaussian Naive Bayes: 0.9333333333333333

Precision of Gaussian Naive Bayes: 0.9352007469654529

Recall of Gaussian Naive Bayes: 0.9333333333333333

F1-Score of Gaussian Naive Bayes: 0.933615520282187

Accuracy of Decision Tree Classifier: 0.9555555555555556

Precision of Decision Tree Classifier: 0.9555555555555556

Recall of Decision Tree Classifier: 0.9555555555555556

F1-Score of Decision Tree Classifier: 0.9555555555555556

Accuracy of Support Vector Machine: 1.0

Precision of Support Vector Machine: 1.0

Recall of Support Vector Machine: 1.0

F1-Score of Support Vector Machine: 1.0

In [ ]:

1

​