**PRACTICAL:6**

**AIM:**

Implement K-Nearest Neighbours, Support Vector Machine (SVM) and Naïve Bayes Classifier with python’s Scikit-Learn on different datasets. Compare the classifiers based on their evaluation measures.

**CODE:**

import pandas as pd

from sklearn.neighbors import KNeighborsClassifier

from sklearn.svm import LinearSVC

from sklearn.naive\_bayes import GaussianNB

from sklearn.metrics import classification\_report

df = pd.read\_csv("covtype.data")

df.describe()

df.isna().sum()

Y = df["1"].copy()

X = df[["51","510","221"]]

print(X.shape)

print(Y.shape)

neigh = KNeighborsClassifier(n\_neighbors=5)

neigh.fit(X,Y)

y\_knn = neigh.predict(X)

print(y\_knn.shape)

nb\_class = GaussianNB()

nb\_class.fit(X,Y)

y\_nb = nb\_class.predict(X)

print(y\_nb.shape)

svm\_class = LinearSVC()

svm\_class.fit(X,Y)

y\_svm = svm\_class.predict(X)

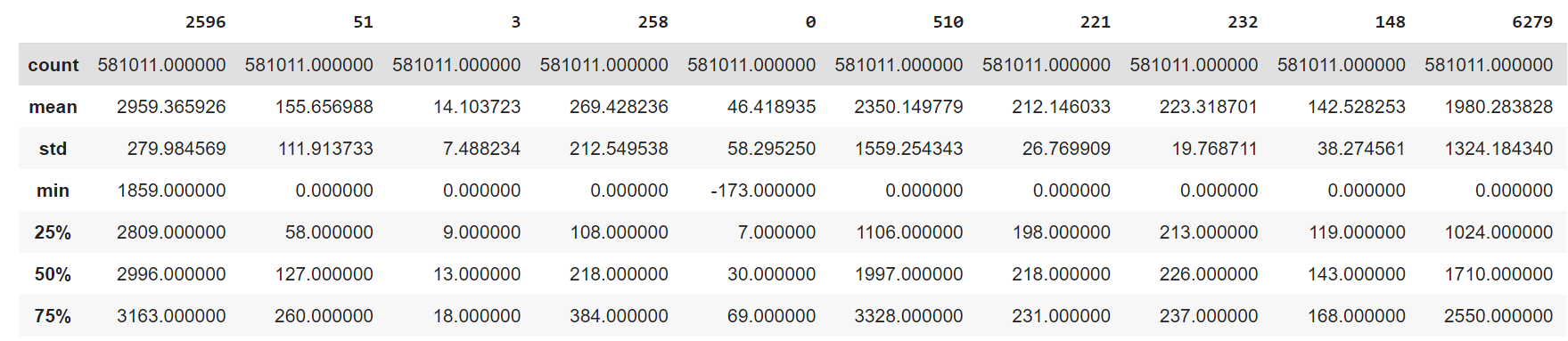
print(y\_svm.shape)

print(classification\_report(y\_knn, Y))

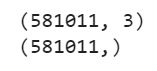
print(classification\_report(y\_nb, Y))

print(classification\_report(y\_svm, Y))

**OUTPUT:**



*Glimpses of describe()*



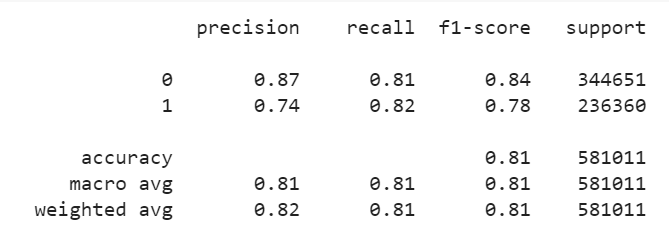
*Value of X,shape and Y.shape*



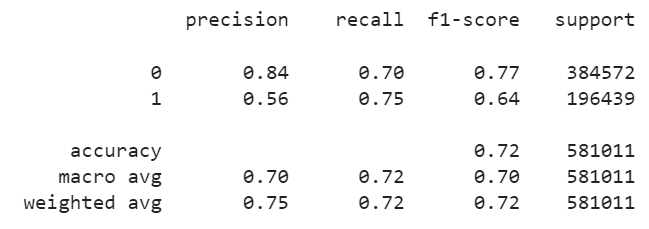
*Value of y\_knn\_shape*



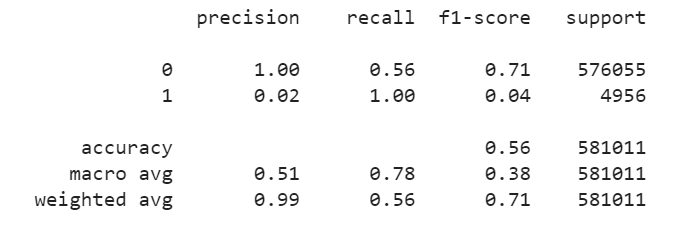
*Value of y\_nb\_shape*



*Classification report of (y\_knn,Y)*



*Classification report of (y\_nb,Y)*



*Classification report of (y\_svm,Ya)*