import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

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

from sklearn.preprocessing import StandardScaler

from sklearn.naive\_bayes import GaussianNB

from sklearn.metrics import confusion\_matrix

…………

dataset=pd.read\_csv("F:\iris.csv")

………

Dataset

…….

#Spliting the dataset in independent and dependent variables

X = dataset.iloc[:,:4].values

y = dataset['Species'].values

# Splitting the dataset into the Training set and Test set

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.20, random\_state = 82)

# Feature Scaling to bring the variable in a single scale

sc = StandardScaler()

X\_train = sc.fit\_transform(X\_train)

X\_test = sc.transform(X\_test)

# Fitting Naive Bayes Classification to the Training set with linear kernel

nb = GaussianNB()

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

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

………

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

print(cm)

#finding accuracy from the confusion matrix.

a = cm.shape

corrPred = 0

falsePred = 0

for row in range(a[0]):

for c in range(a[1]):

if row == c:

corrPred +=cm[row,c]

else:

falsePred += cm[row,c]

print('Correct predictions: ', corrPred)

print('False predictions', falsePred)

print ('\n\nAccuracy of the Naive Bayes Clasification is: ', corrPred/(cm.sum()))