Aim: - To implement and observe output of Random Forest on IDLE.

CODE: -

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

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

from sklearn.preprocessing import RobustScaler

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

from sklearn.ensemble import RandomForestClassifier

from sklearn import metrics

Bill\_Auth\_data = pd.read\_csv('bill\_authentication.csv')

print(Bill\_Auth\_data.head())

print(Bill\_Auth\_data.info())

Bill\_Auth\_data = Bill\_Auth\_data.sample(frac=1.0).reset\_index(drop=True)

print(Bill\_Auth\_data.head())

print(Bill\_Auth\_data.describe())

Bill\_Auth\_data.iloc[:,0:-1].plot(kind='hist',subplots=True,layout=(2,2),figsize=(15,6));

Bill\_Auth\_data.iloc[:,0:-1].plot(kind='box',subplots=True,layout=(2,2),figsize=(15,6));

sns.pairplot(data=Bill\_Auth\_data,hue='Class');

X = Bill\_Auth\_data.drop('Class',axis=1)

Y = Bill\_Auth\_data['Class']

RS = RobustScaler()

X = RS.fit\_transform(X)

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X,Y,test\_size=0.2,random\_state=10)

RFC = RandomForestClassifier()

RFC.fit(X\_train,Y\_train)

Y\_pred = RFC.predict(X\_test)

print(metrics.accuracy\_score(Y\_test,Y\_pred))

print(metrics.confusion\_matrix(Y\_test,Y\_pred))

print(metrics.classification\_report(Y\_test,Y\_pred))

accuracy\_list =[]

for n\_est in range(50,201,10):

RFC = RandomForestClassifier(n\_estimators=n\_est)

RFC.fit(X\_train,Y\_train)

accuracy\_list.append(round(metrics.accuracy\_score(Y\_test,RFC.predict(X\_test))\*100,2))

plt.figure(figsize=(12, 6))

plt.plot(range(50, 201, 10), accuracy\_list, color='red',linestyle='dashed', marker='o', markerfacecolor='blue',markersize=10)

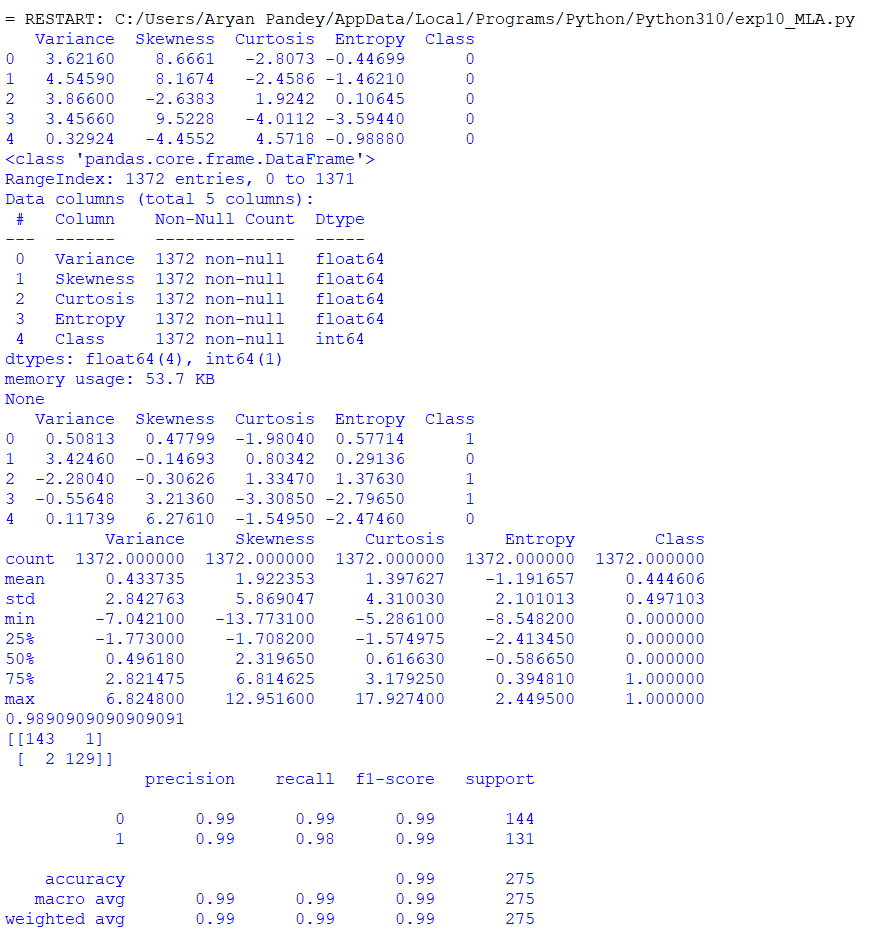
plt.xticks(range(50, 201, 10))

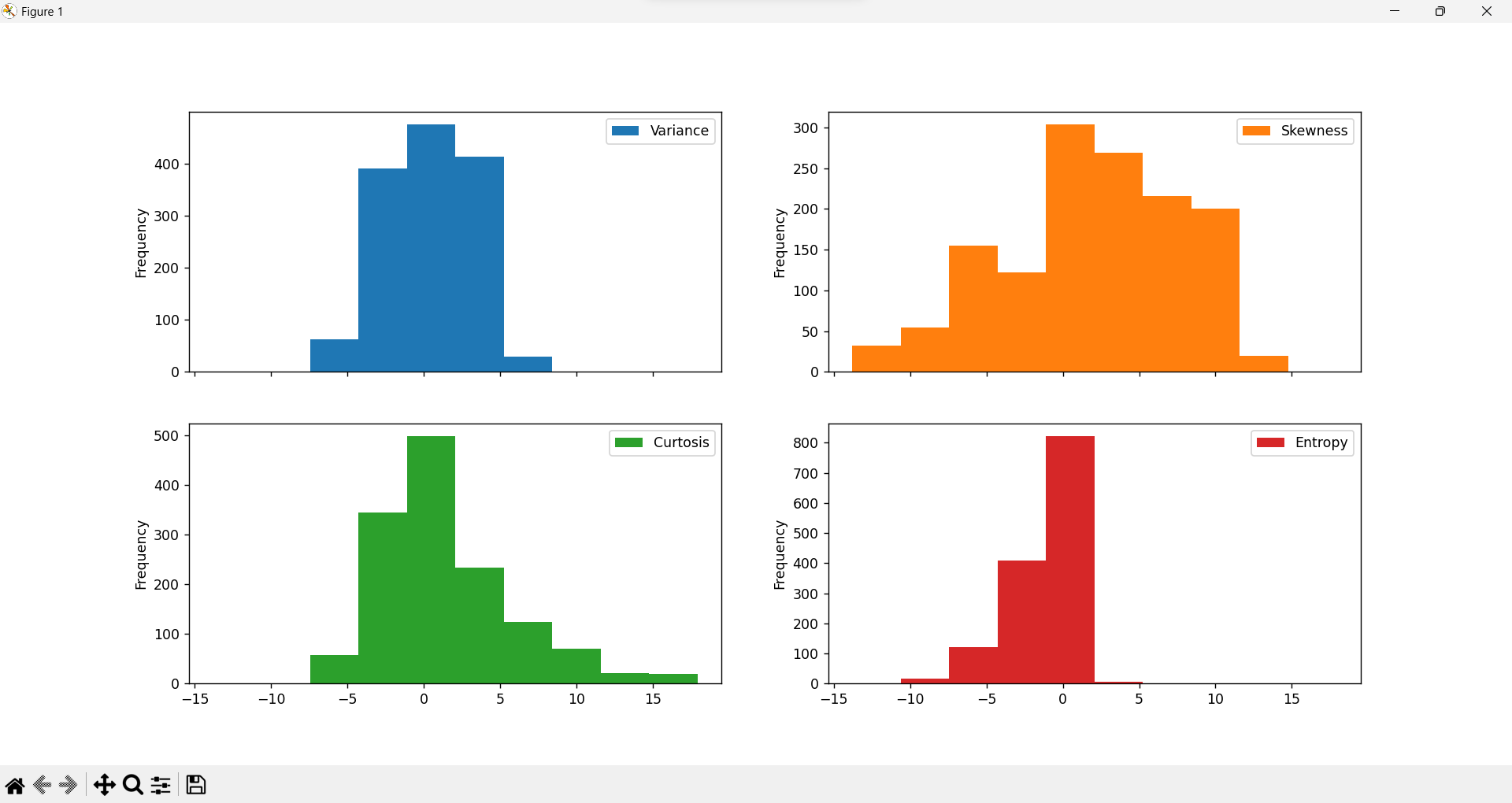
plt.xlabel('n\_estimator Value')

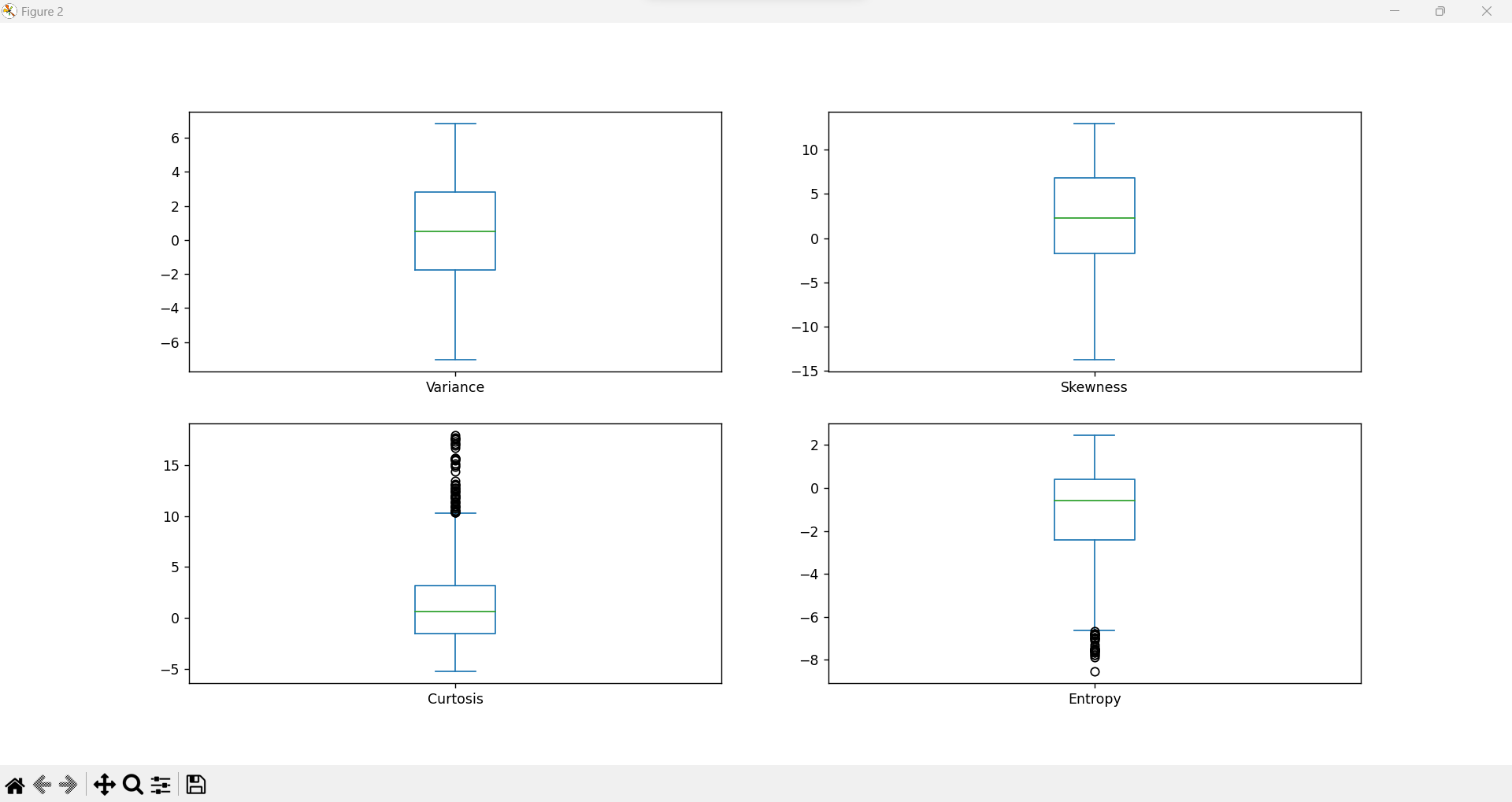
plt.ylabel('Accuracy');

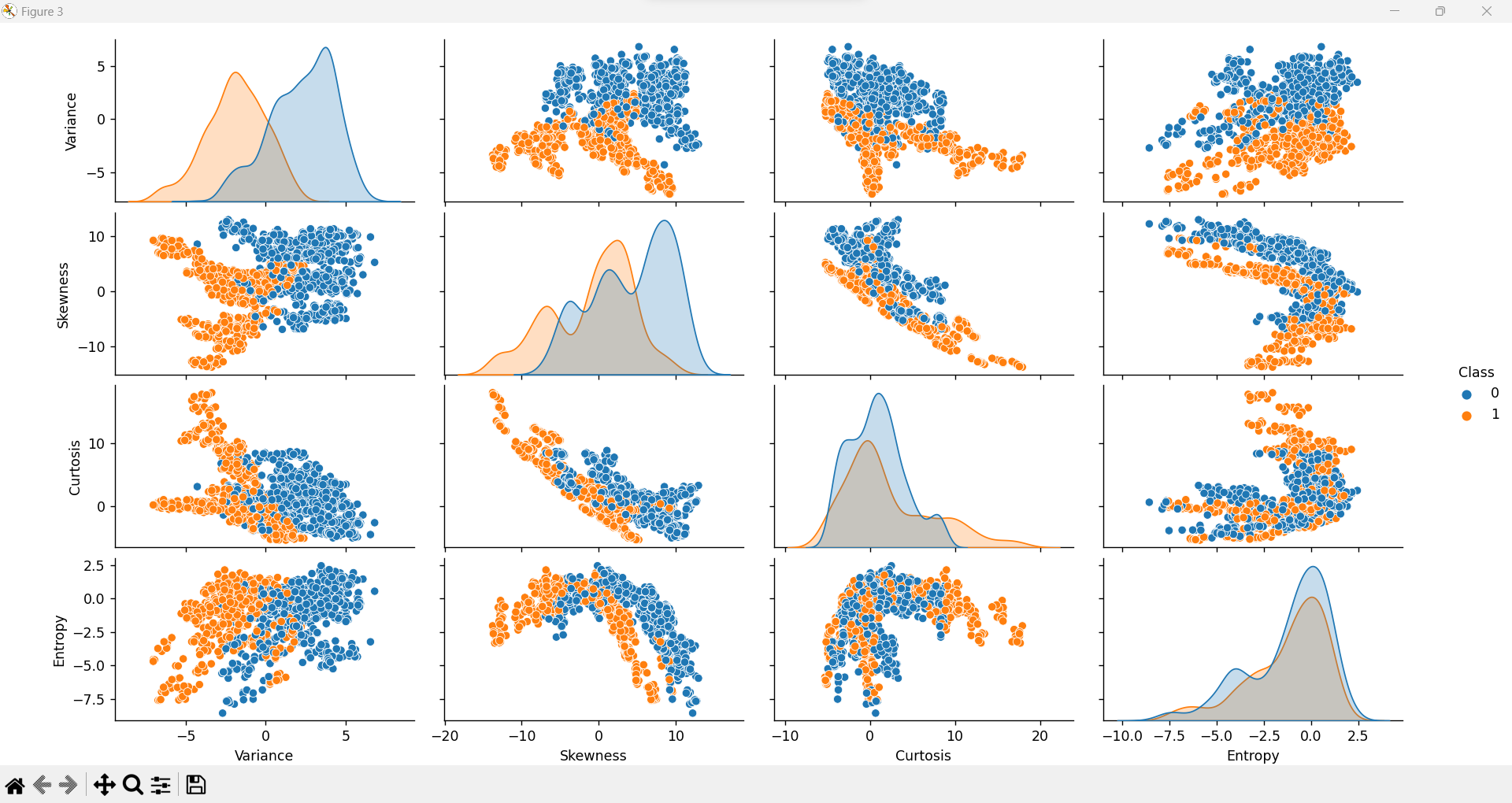
plt.show()

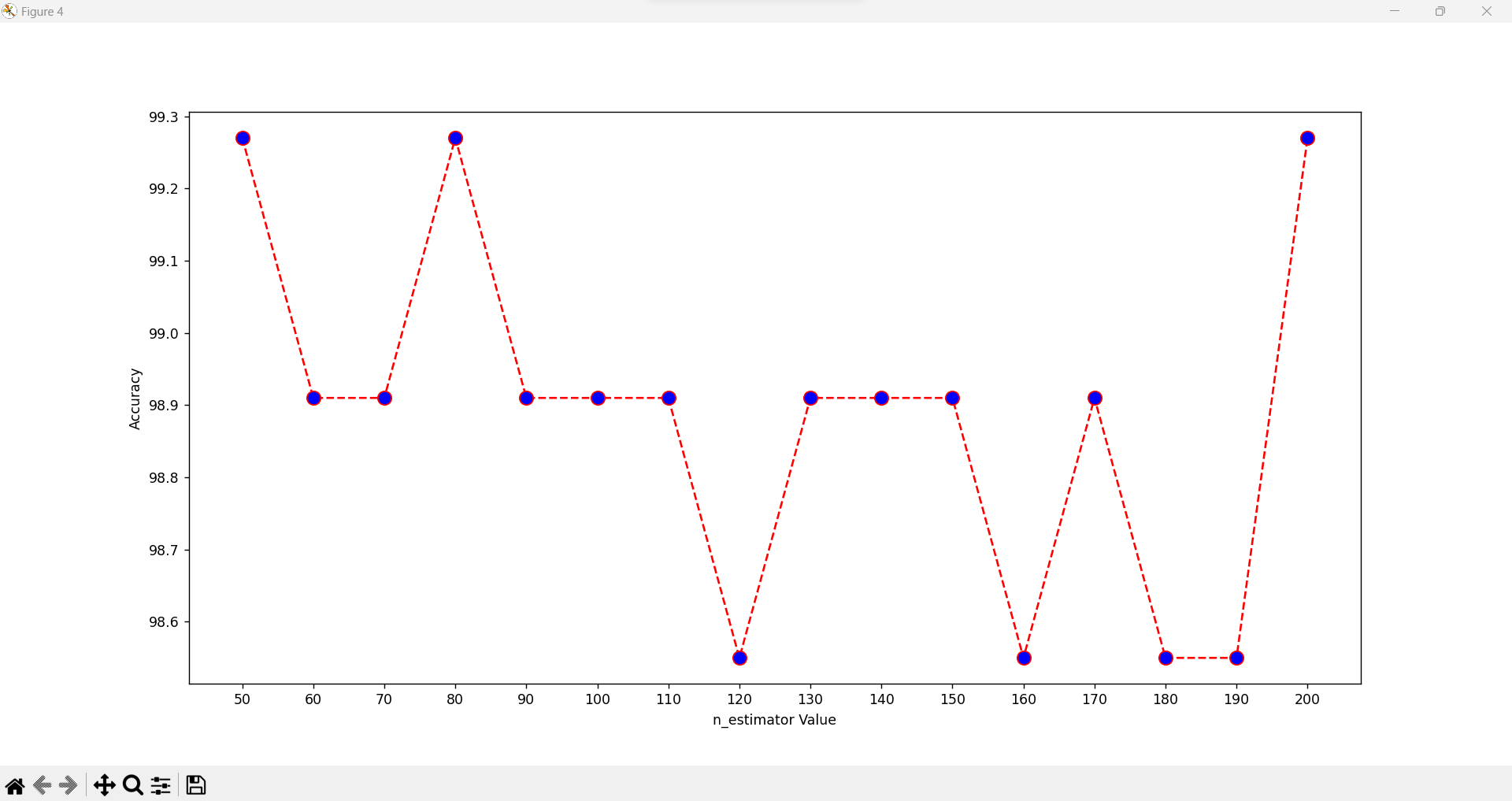
Output: -











Result: -

Hence verified and got the output of the tasks on IDLE Python.