SUPPORT VECTOR MACHINE ASSIGNMENT:

1. Prepare a classification model using SVM for salary data

CODE:

# -\*- coding: utf-8 -\*-

"""

Created on Tue Nov 17 13:39:47 2020

@author: sunil

"""

import pandas as pd

import numpy as np

import seaborn as sns

from sklearn import preprocessing

salary = pd.read\_csv("SalaryData\_Test.csv")

salary.drop(["capitalloss","capitalgain"],axis=1,inplace=True)

salary.columns

string\_columns=['workclass','education','maritalstatus','occupation','relationship','race','sex','native','Salary']

number = preprocessing.LabelEncoder()

for i in string\_columns:

salary[i] = number.fit\_transform(salary[i])

salary.head()

salary.describe()

salary.columns

sns.boxplot(x="age",y="Salary",data=salary,palette = "hls")

sns.boxplot(x="native",y="age",data=salary,palette = "hls")

from sklearn.svm import SVC

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

train,test = train\_test\_split(salary,test\_size = 0.3,random\_state=0)

test.head()

train\_X = train.iloc[:,:11]

train\_y = train.iloc[:,11]

test\_X = test.iloc[:,:11]

test\_y = test.iloc[:,11]

model\_linear = SVC(kernel = "linear")

model\_linear.fit(train\_X,train\_y)

pred\_test\_linear = model\_linear.predict(test\_X)

np.mean(pred\_test\_linear==test\_y) # Accuracy = 80.23

# Kernel = poly

model\_poly = SVC(kernel = "poly")

model\_poly.fit(train\_X,train\_y)

pred\_test\_poly = model\_poly.predict(test\_X)

np.mean(pred\_test\_poly==test\_y) # Accuracy = 79.9

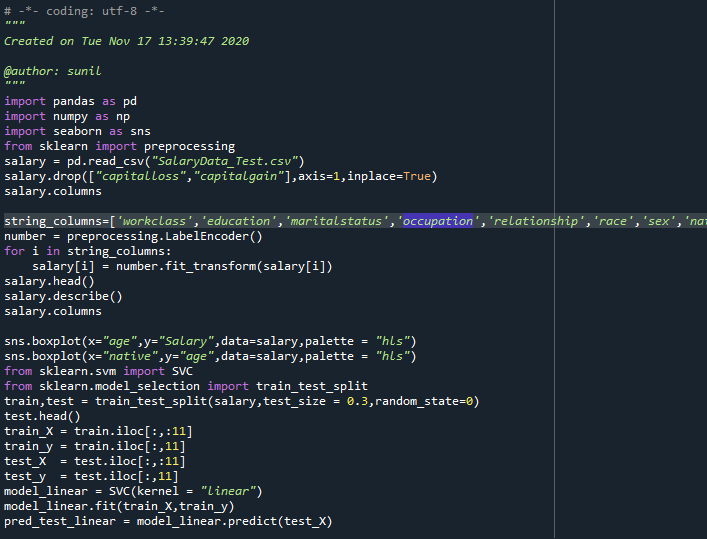
# kernel = rbf

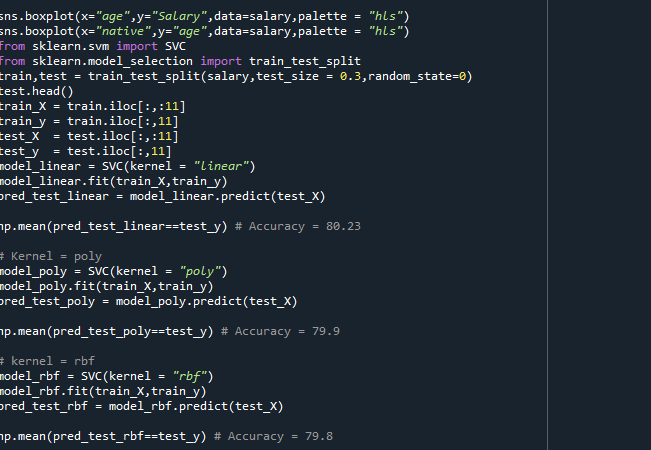
model\_rbf = SVC(kernel = "rbf")

model\_rbf.fit(train\_X,train\_y)

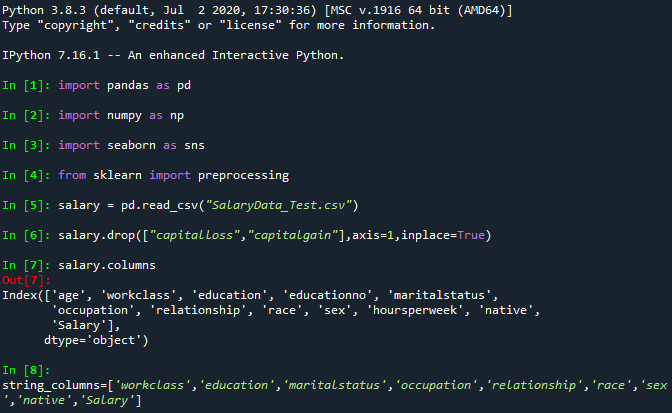
pred\_test\_rbf = model\_rbf.predict(test\_X)

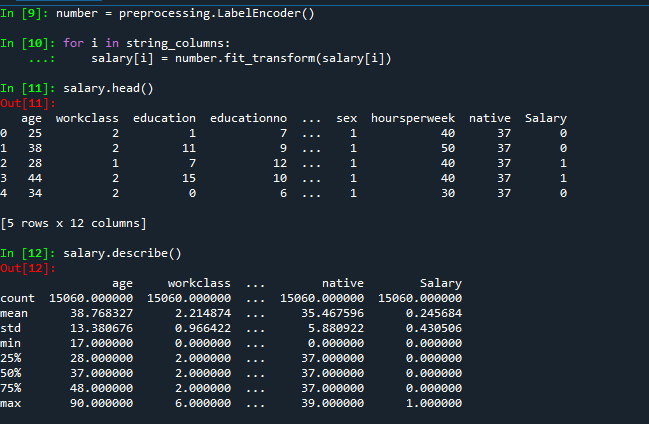
np.mean(pred\_test\_rbf==test\_y) # Accuracy = 79.8

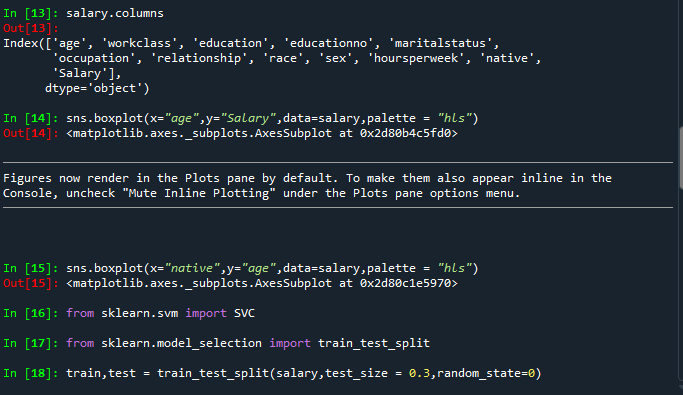


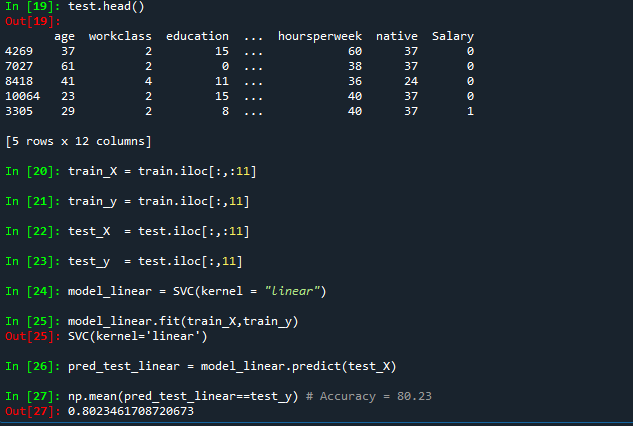


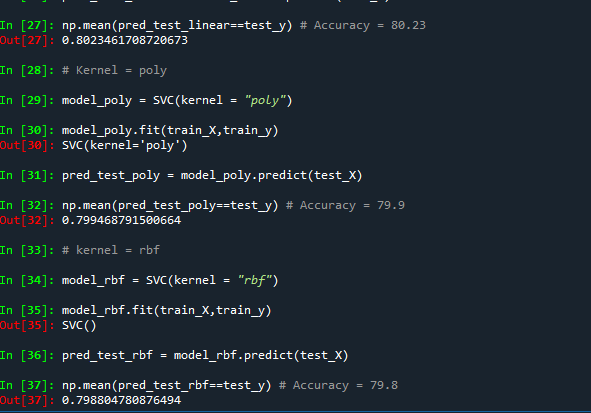
**Output:**



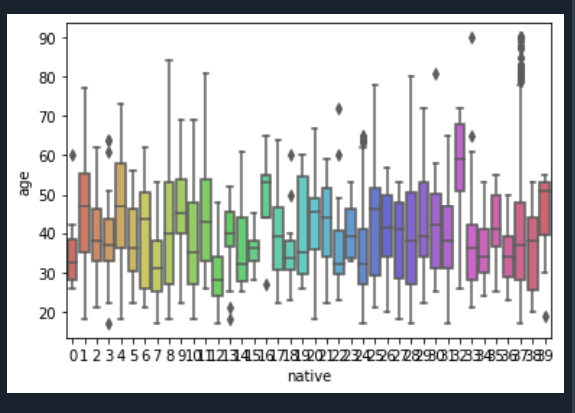


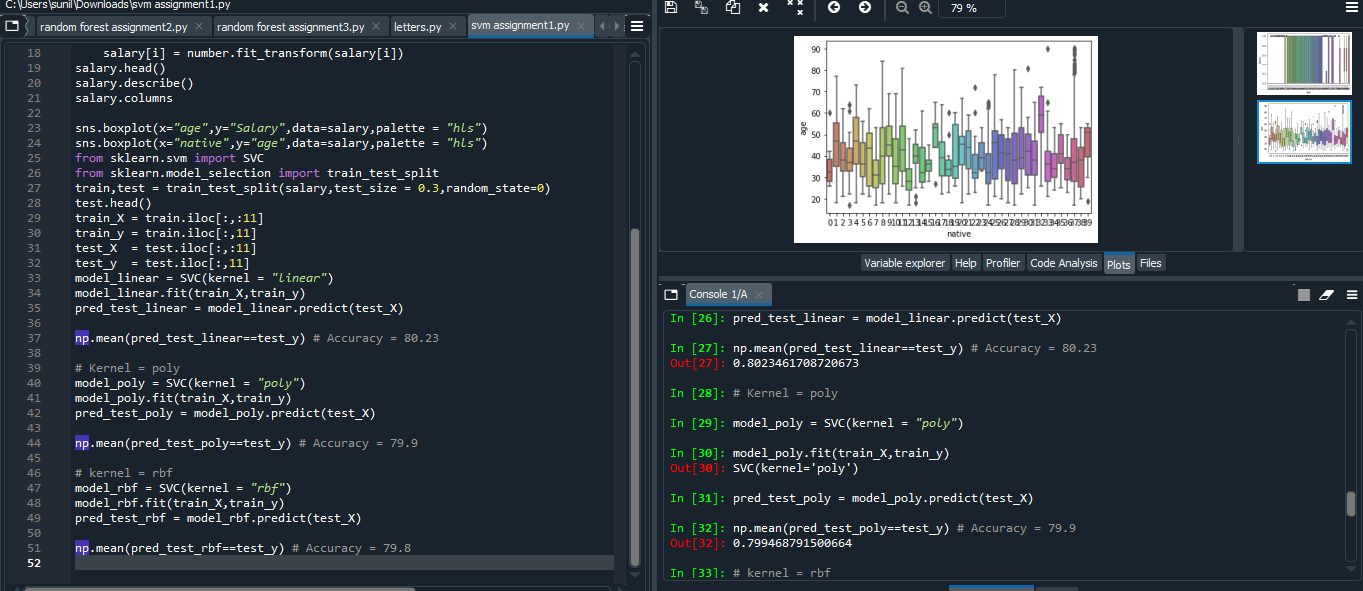






**PLOTS:**





**2)WITH “FORESTFRIES.CSV” DATASETS:**

**CODE:**

# -\*- coding: utf-8 -\*-

"""

Created on Tue Nov 17 12:27:37 2020

@author: sunil

"""

import pandas as pd

import numpy as np

import seaborn as sns

from sklearn import preprocessing

data = pd.read\_csv("forestfires.csv")

string\_columns=['day','size\_category']

number = preprocessing.LabelEncoder()

for i in string\_columns:

data[i] = number.fit\_transform(data[i])

data.head()

data.describe()

data.columns

sns.boxplot(x="month",y="FFMC",data=data,palette = "hls")

sns.boxplot(x="FFMC",y="month",data=data,palette = "hls")

#sns.pairplot(data=letters)

from sklearn.svm import SVC

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

train,test = train\_test\_split(data,test\_size = 0.3,random\_state=0)

test.head()

train\_X = train.iloc[:,1:]

train\_y = train.iloc[:,0]

test\_X = test.iloc[:,1:]

test\_y = test.iloc[:,0]

model\_linear = SVC(kernel = "linear")

model\_linear.fit(train\_X,train\_y)

pred\_test\_linear = model\_linear.predict(test\_X)

np.mean(pred\_test\_linear==test\_y) # Accuracy = 94.23

# Kernel = poly

model\_poly = SVC(kernel = "poly")

model\_poly.fit(train\_X,train\_y)

pred\_test\_poly = model\_poly.predict(test\_X)

np.mean(pred\_test\_poly==test\_y) # Accuracy = 75.0

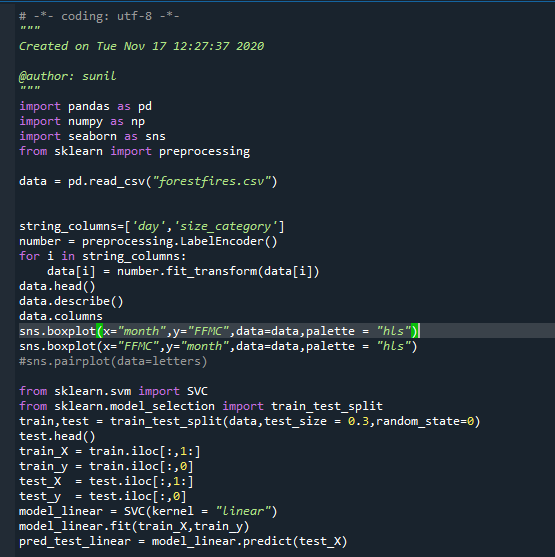
# kernel = rbf

model\_rbf = SVC(kernel = "rbf")

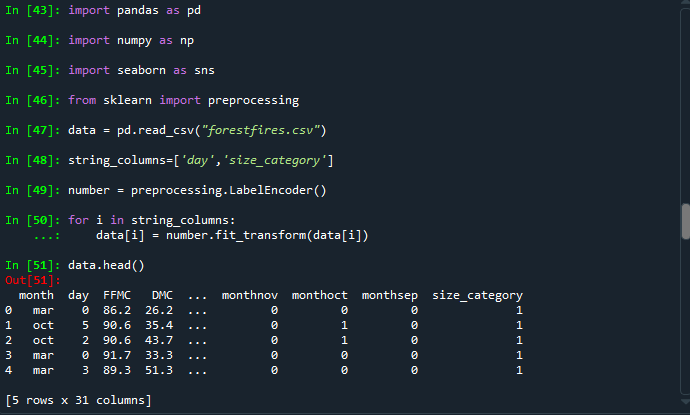
model\_rbf.fit(train\_X,train\_y)

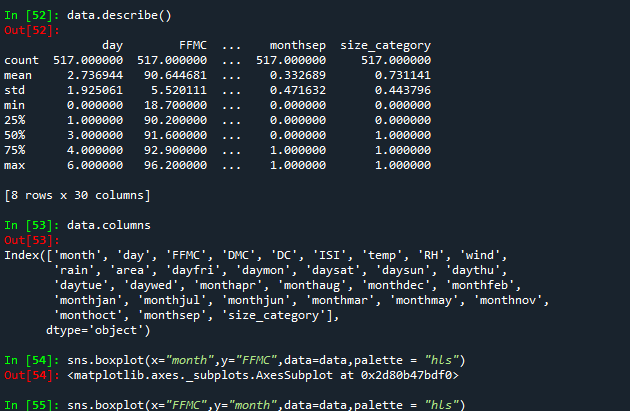
pred\_test\_rbf = model\_rbf.predict(test\_X)

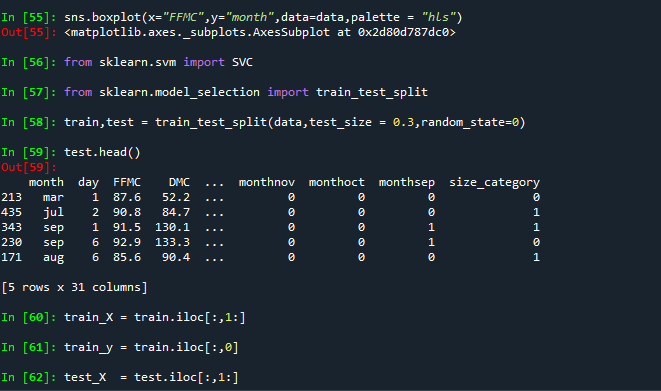
np.mean(pred\_test\_rbf==test\_y) # Accuracy = 76.9

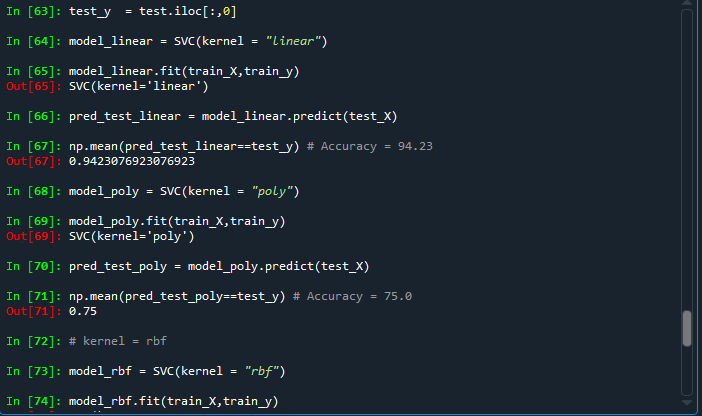


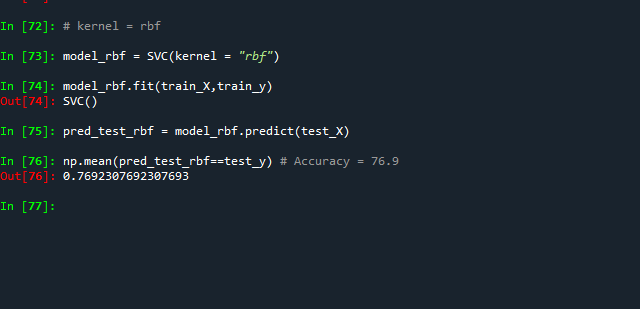
OUTPUT:











PLOTS:

