Lab Assignment 6

Chaudhary Hamdan

1905387

Date: 09-03-2022

1. On ‘Income dataset’

# coding: utf-8

# In[1]:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

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

from sklearn import preprocessing

import seaborn as sns

from sklearn import svm

from sklearn.ensemble import GradientBoostingClassifier

from sklearn.ensemble import AdaBoostClassifier

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import accuracy\_score

from sklearn.metrics import recall\_score

from sklearn.metrics import precision\_score

from sklearn.metrics import f1\_score

# In[2]:

import warnings

warnings.filterwarnings('ignore')

# In[3]:

df = pd.read\_excel('income.xlsx')

# In[4]:

df.head()

# In[5]:

df.drop(columns=['capitalgain', 'capitalloss'], inplace=True)

# In[6]:

df.head()

# In[7]:

cols = ['JobType', 'EdType', 'maritalstatus', 'occupation', 'relationship', 'race',

'gender', 'nativecountry', 'SalStat']

for col in cols:

le = preprocessing.LabelEncoder()

df[col] = le.fit\_transform(df[col])

# In[8]:

df.head()

# In[9]:

df.SalStat.value\_counts()

# 1 : Less than or equal to 50k, 0 means less than 50k

# In[10]:

df.head()

# In[11]:

corr = df.corr()

corr.style.background\_gradient(cmap='coolwarm')

# In[12]:

x\_train, x\_test, y\_train, y\_test = train\_test\_split(df.drop(columns = ['SalStat']), df['SalStat'], test\_size = 0.2)

x\_train.shape, y\_train.shape, x\_test.shape, y\_test.shape

# In[13]:

algos = []

accuracy = []

recall = []

precision = []

f1Score = []

# In[14]:

algo = "SVM"

model = svm.SVC()

model.fit(x\_train, y\_train)

y\_pred = model.predict(x\_test)

print(algo)

print(confusion\_matrix(y\_test, y\_pred), '\n\n')

acc = accuracy\_score(y\_test, y\_pred) \* 100

print('Accuracy:', acc)

rec = recall\_score(y\_test, y\_pred) \* 100

print('Recall:', rec)

pre = precision\_score(y\_test, y\_pred) \* 100

print('Precision:', pre)

f1s = f1\_score(y\_test, y\_pred) \* 100

print('F score:', f1s)

algos.append(algo)

accuracy.append(acc)

recall.append(rec)

precision.append(pre)

f1Score.append(f1s)

# In[15]:

algo = "Gradient Boost"

model = GradientBoostingClassifier()

model.fit(x\_train, y\_train)

y\_pred = model.predict(x\_test)

print(algo)

print(confusion\_matrix(y\_test, y\_pred), '\n\n')

acc = accuracy\_score(y\_test, y\_pred) \* 100

print('Accuracy:', acc)

rec = recall\_score(y\_test, y\_pred) \* 100

print('Recall:', rec)

pre = precision\_score(y\_test, y\_pred) \* 100

print('Precision:', pre)

f1s = f1\_score(y\_test, y\_pred) \* 100

print('F score:', f1s)

algos.append(algo)

accuracy.append(acc)

recall.append(rec)

precision.append(pre)

f1Score.append(f1s)

# In[16]:

algo = "Ada Boost"

model = AdaBoostClassifier()

model.fit(x\_train, y\_train)

y\_pred = model.predict(x\_test)

print(algo)

print(confusion\_matrix(y\_test, y\_pred), '\n\n')

acc = accuracy\_score(y\_test, y\_pred) \* 100

print('Accuracy:', acc)

rec = recall\_score(y\_test, y\_pred) \* 100

print('Recall:', rec)

pre = precision\_score(y\_test, y\_pred) \* 100

print('Precision:', pre)

f1s = f1\_score(y\_test, y\_pred) \* 100

print('F score:', f1s)

algos.append(algo)

accuracy.append(acc)

recall.append(rec)

precision.append(pre)

f1Score.append(f1s)

# In[17]:

for i in range(3):

print(algos[i], ': ', accuracy[i],', ', recall[i],', ', precision[i],', ', f1Score[i])

# In[18]:

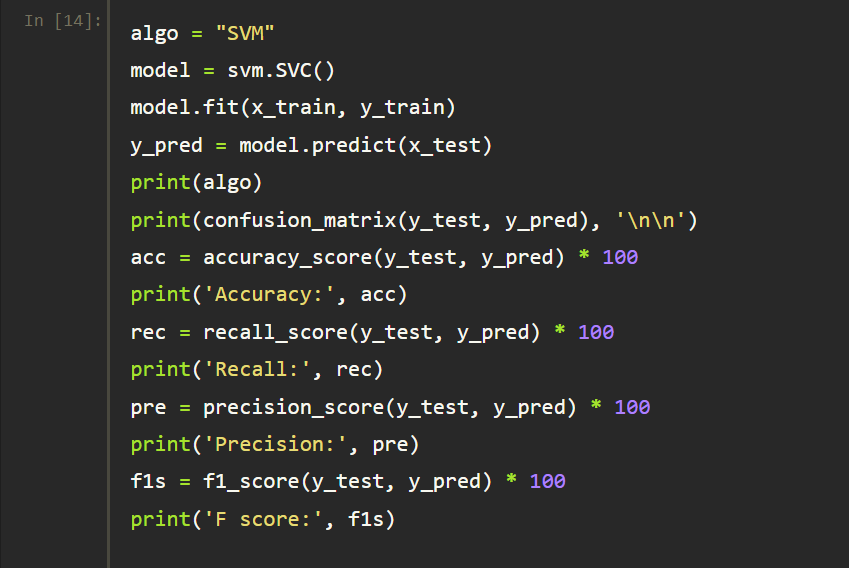
plt.bar(algos, accuracy)

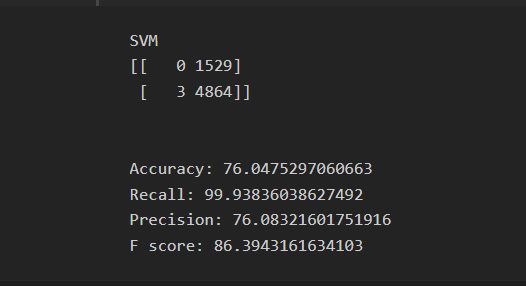
plt.show()

A)apply Support vector machine algorithm and find out the accuarcy in predicting whether the salary status is less than or equal to 5000 or it is greater tan 50000.

1. Confusion Matrix
2. Accuracy score

iii)recall, precision, f- score



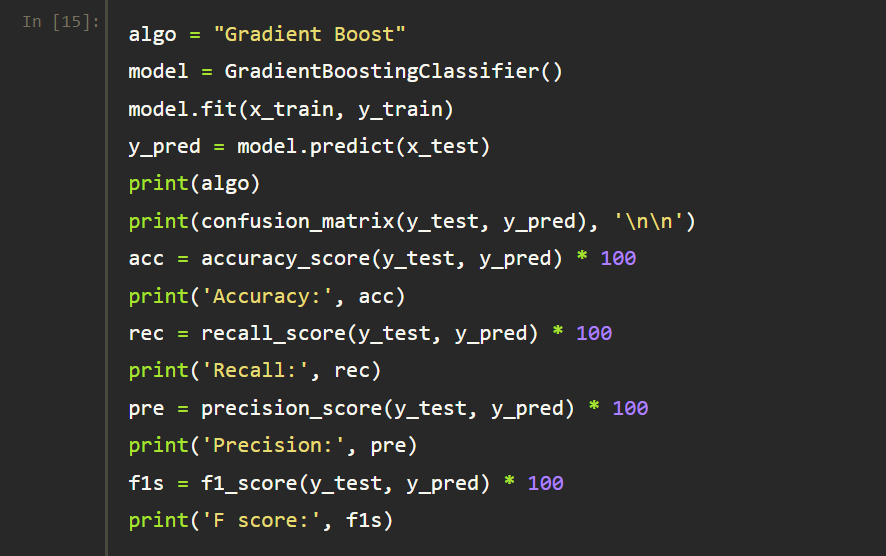


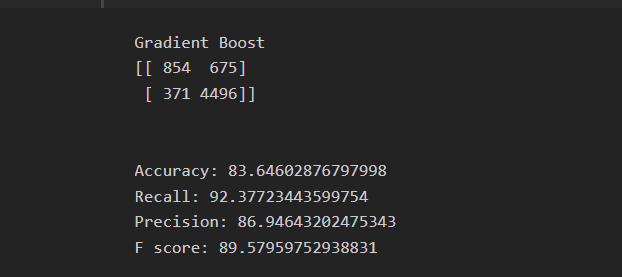
B)apply Gradient Boost algorithm and find out the accuarcy in predicting whether the salary status is less than or equal to 5000 or it is greater tan 50000.

I)Confusion Matrix

ii)Accuracy score

iii)recall, precision, f- score





C)apply Adaboost algorithm and find out the accuarcy in predicting whether the salary status is less than or equal to 5000 or it is greater tan 50000.

I)Confusion Matrix

ii)Accuracy score

iii)recall, precision, f- score

