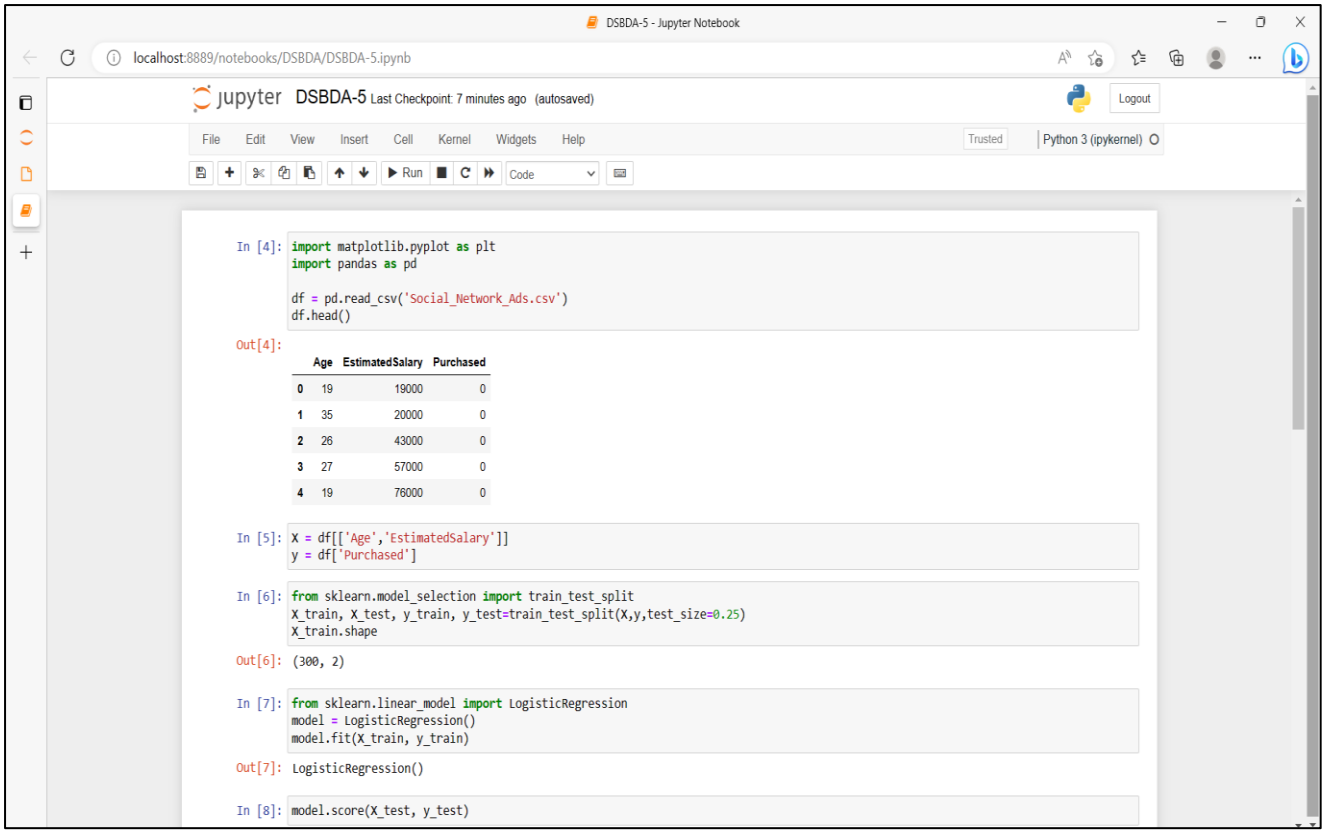
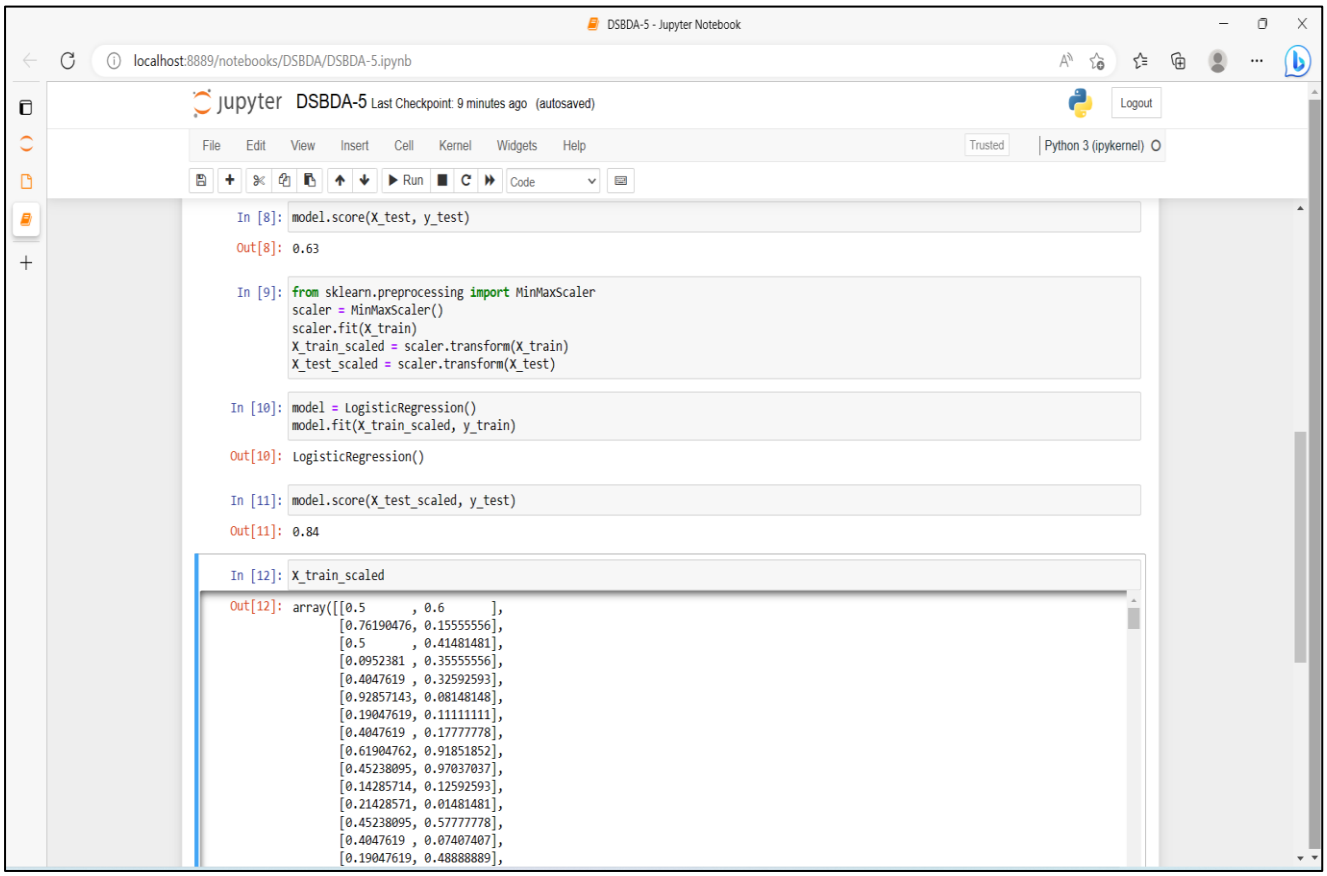
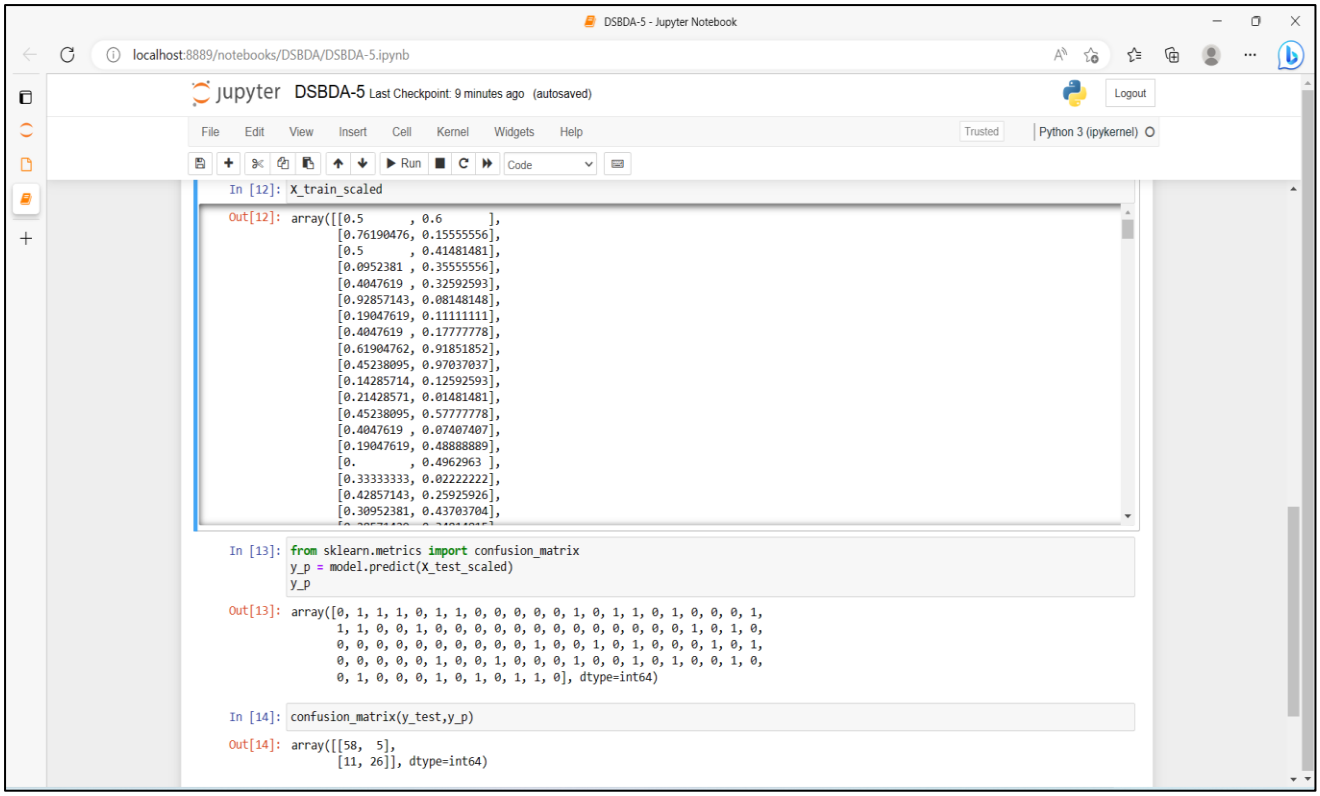

DSBDA-5 - Jupyter Notebook
localhost:8889/notebooks/DSBDA/DSBDA-5.ipynb
jupyter DSBDA-5 Last Checkpoint: 7 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [4]:
import matplotlib.pyplot as plt
import pandas as pd
df = pd.read_csv('Social_Network_Ads.csv')
df.head()
Out[4]:
Age EstimatedSalary Purchased
0 19 19000 0
1 35 20000 0
2 26 43000 0
3 27 57000 0
4 19 76000 0
In [5]:
X = df[['Age','EstimatedSalary']]
y = df['Purchased']
In [6]:
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test=train_test_split(X,y,test_size=0.25)
X_train.shape
Out[6]:
(300, 2)
In [7]:
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(X_train, y_train)
Out[7]:
LogisticRegression()
In [8]:
model.score(X_test, y_test)


DSBDA-5 - Jupyter Notebook
localhost:8889/notebooks/DSBDA/DSBDA-5.ipynb
jupyter DSBDA-5 Last Checkpoint: 9 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [8]:
model.score(X_test, y_test)
Out[8]:
0.63
In [9]:
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
scaler.fit(X_train)
X_train_scaled = scaler.transform(X_train)
X_test_scaled = scaler.transform(X_test)
In [10]:
model = LogisticRegression()
model.fit(X_train_scaled, y_train)
Out[10]:
LogisticRegression()
In [11]:
model.score(X_test_scaled, y_test)
Out[11]:
0.84
In [12]:
X_train_scaled
Out[12]:
array([[0.5 , 0.6 ],
[0.76190476, 0.15555556],
[0.5 , 0.41481481],
[0.0952381 , 0.35555556],
[0.4047619 , 0.32592593],
[0.92857143, 0.08148148],
[0.19047619, 0.11111111],
[0.4047619 , 0.17777778],
[0.61904762, 0.91851852],
[0.45238095, 0.97037037],
[0.14285714, 0.12592593],
[0.21428571, 0.01481481],
[0.45238095, 0.57777778],
[0.4047619 , 0.07407407],
[0.19047619, 0.48888889],


DSBDA-5 - Jupyter Notebook
localhost:8889/notebooks/DSBDA/DSBDA-5.ipynb
jupyter DSBDA-5 Last Checkpoint: 9 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [12]: X_train_scaled
Out[12]: array([[0.5 , 0.6 ],
[0.76190476, 0.15555556],
[0.5 , 0.41481481],
[0.0952381 , 0.35555556],
[0.4047619 , 0.32592593],
[0.92857143, 0.08148148],
[0.19047619, 0.11111111],
[0.4047619 , 0.17777778],
[0.61904762, 0.91851852],
[0.45238095, 0.97037037],
[0.14285714, 0.12592593],
[0.21428571, 0.01481481],
[0.45238095, 0.57777778],
[0.4047619 , 0.07407407],
[0.19047619, 0.48888889],
[0. , 0.4962963 ],
[0.33333333, 0.02222222],
[0.42857143, 0.25925926],
[0.30952381, 0.43703704],
In [13]: from sklearn.metrics import confusion_matrix
y_p = model.predict(X_test_scaled)
y_p
Out[13]: array([0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1,
1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1,
0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0,
0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0], dtype=int64)
In [14]: confusion_matrix(y_test,y_p)
Out[14]: array([[58, 5],
[11, 26]], dtype=int64)