

# anurag-dsbdal-pr5-1

February 21, 2024

```
[39]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, classification_report, \
    accuracy_score, precision_score, recall_score, f1_score
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

```
[8]: df = pd.read_csv("Social_Network_Ads.csv")
```

```
[9]: df.head()
```

```
[9]:
```

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0

```
[10]: df.shape
```

```
[10]: (400, 5)
```

```
[11]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  -
0   User ID         400 non-null   int64
1   Gender          400 non-null   object
2   Age             400 non-null   int64
```

```
3 EstimatedSalary 400 non-null int64
4 Purchased      400 non-null int64
dtypes: int64(4), object(1)
memory usage: 15.8+ KB
```

```
[12]: df.describe()
```

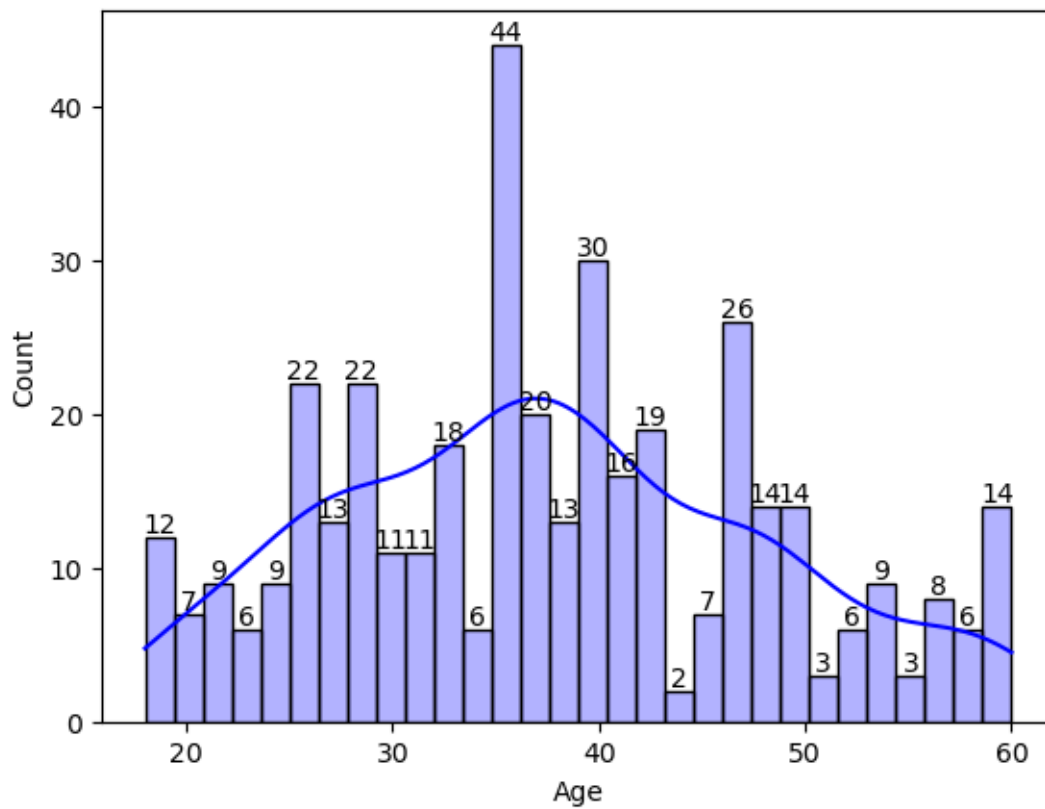
```
[12]:
```

	User ID	Age	EstimatedSalary	Purchased
count	4.000000e+02	400.000000	400.000000	400.000000
mean	1.569154e+07	37.655000	69742.500000	0.357500
std	7.165832e+04	10.482877	34096.960282	0.479864
min	1.556669e+07	18.000000	15000.000000	0.000000
25%	1.562676e+07	29.750000	43000.000000	0.000000
50%	1.569434e+07	37.000000	70000.000000	0.000000
75%	1.575036e+07	46.000000	88000.000000	1.000000
max	1.581524e+07	60.000000	150000.000000	1.000000

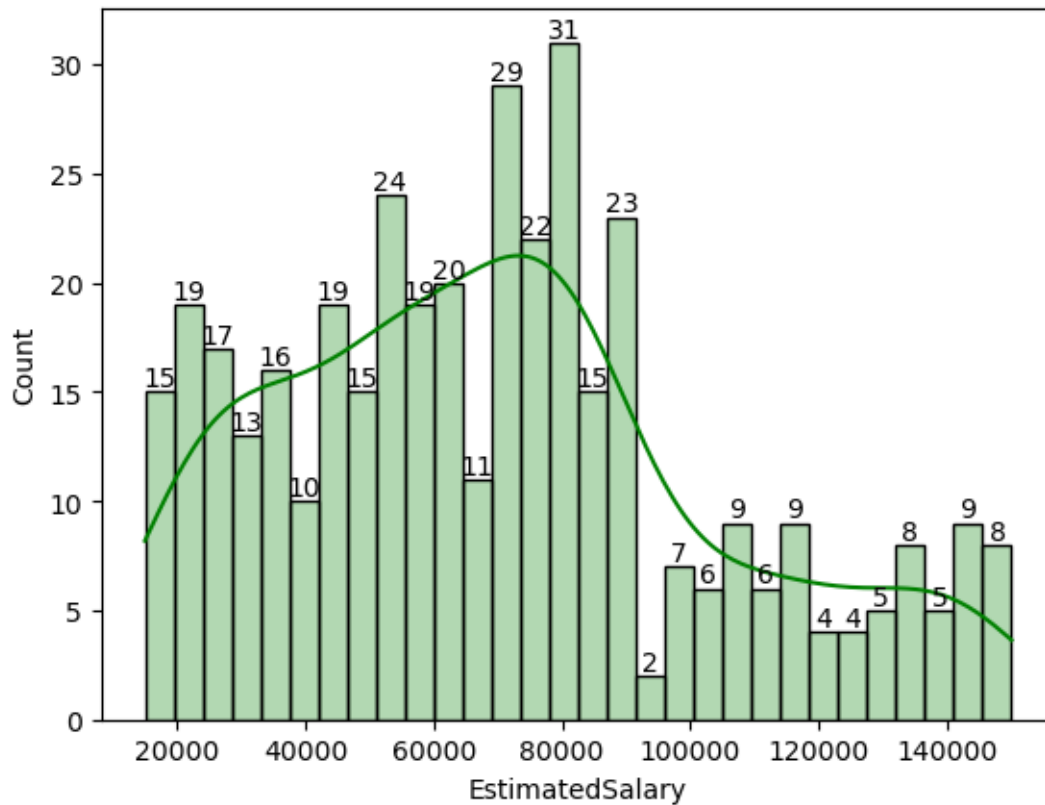
```
[13]: df.isna().sum()
```

```
[13]: User ID      0
      Gender      0
      Age         0
      EstimatedSalary  0
      Purchased    0
      dtype: int64
```

```
[17]: histplot = sns.histplot(df['Age'], kde=True, bins=30, color='blue', alpha=0.3)
      for i in histplot.containers:
          histplot.bar_label(i,)
      plt.show()
```



```
[16]: histplot = sns.histplot(df['EstimatedSalary'], kde=True, bins=30,
    color='green', alpha=0.3)
for i in histplot.containers:
    histplot.bar_label(i,)
plt.show()
```



```
[18]: df["Gender"].value_counts()
```

```
[18]: Gender
      Female    204
      Male     196
      Name: count, dtype: int64
```

```
[19]: def gender_encoder(value):
      if (value == "Male"):
          return 1
      elif (value == "Female"):
          return 0
      else:
          return -1
```

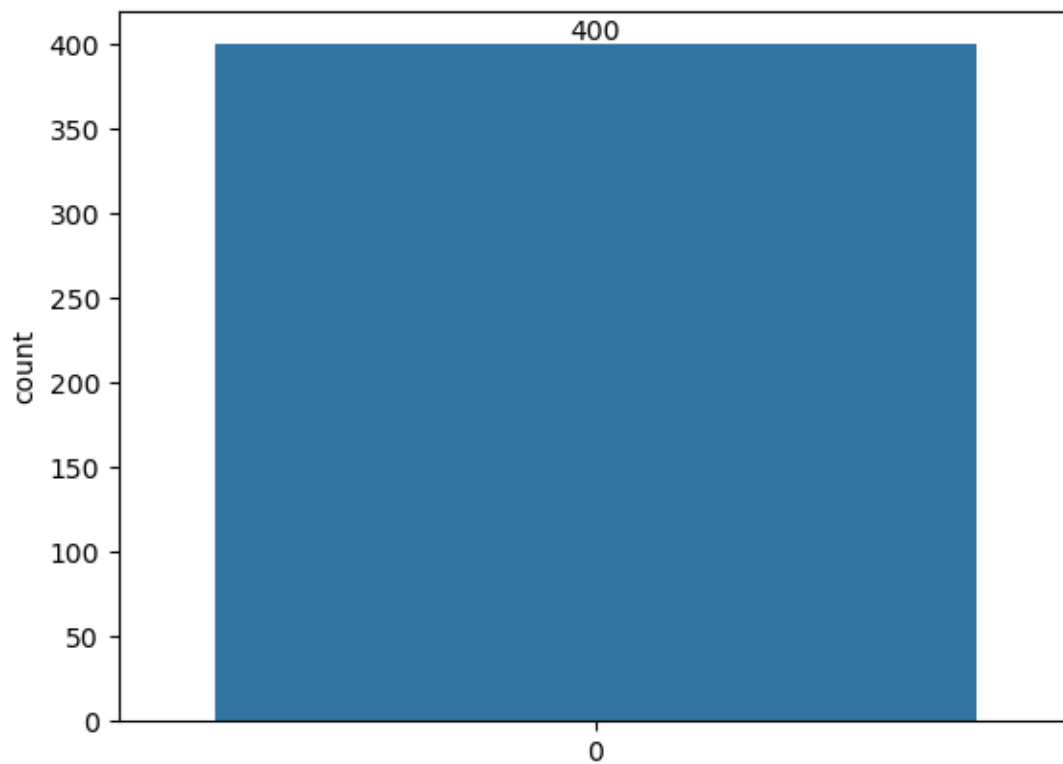
```
[20]: df["Gender"] = df["Gender"].apply(gender_encoder)
```

```
[21]: df["Purchased"].value_counts()
```

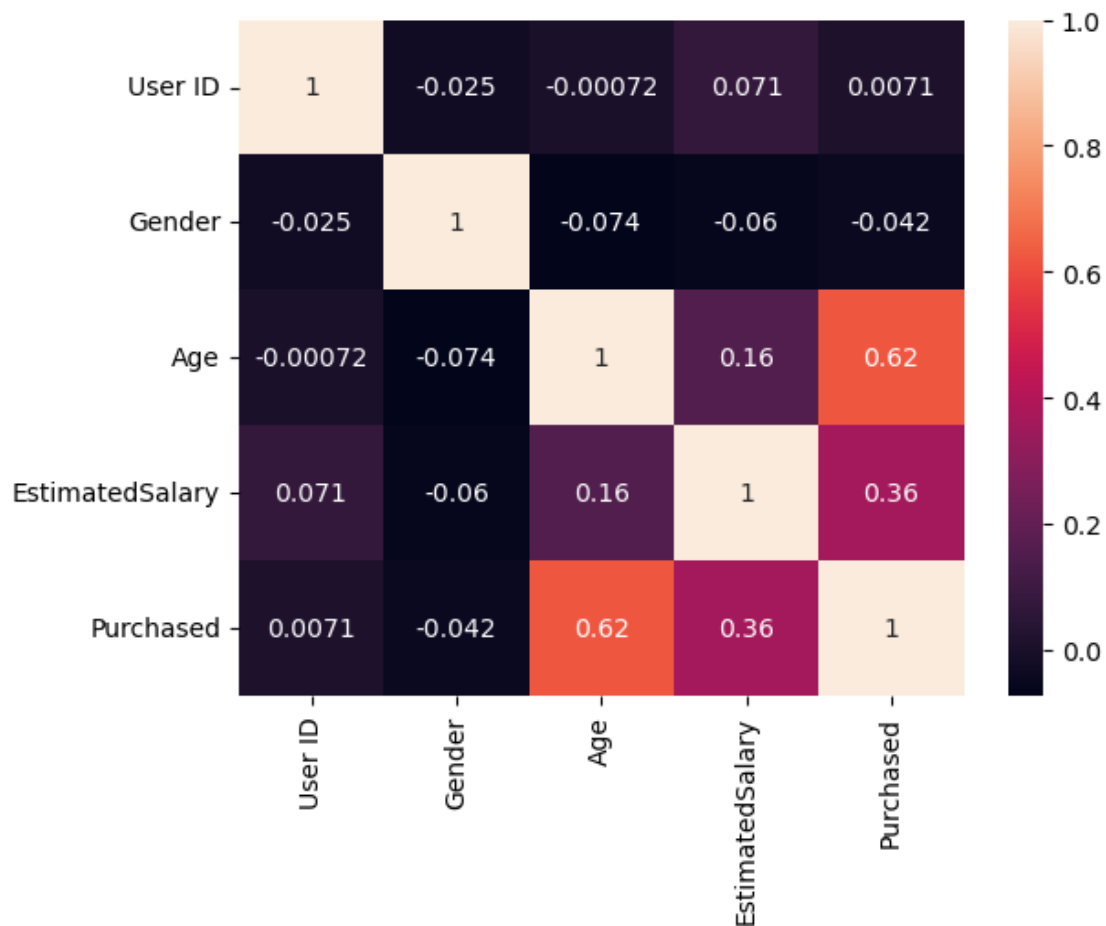
```
[21]: Purchased
      0    257
```

```
1    143
Name: count, dtype: int64
```

```
[22]: countplot = sns.countplot(df["Purchased"])
      for i in countplot.containers:
          countplot.bar_label(i,)
      plt.show()
```



```
[25]: sns.heatmap(df.corr(), annot=True)
      plt.show()
```



```
[26]: x = df[["Age", "EstimatedSalary"]]
      y = df["Purchased"]
```

```
[27]: scaler = StandardScaler()
      x = scaler.fit_transform(x)
```

```
[28]: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2,
      ↪ random_state=42)
```

```
[29]: x_train.shape, x_test.shape, y_train.shape, y_test.shape
```

```
[29]: ((320, 2), (80, 2), (320,), (80,))
```

```
[30]: model = LogisticRegression(n_jobs=-1)
```

```
[31]: model.fit(x_train, y_train)
```

```
[31]: LogisticRegression(n_jobs=-1)
```

```
[32]: y_pred = model.predict(x_test)
```

```
[33]: cm = confusion_matrix(y_test, y_pred)
      print(cm)
```

```
[[50  2]
 [ 9 19]]
```

```
[40]: cm = confusion_matrix(y_test, y_pred)
      print(cm)
```

```
[[50  2]
 [ 9 19]]
```

```
[44]: !pip install mlxtend
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting mlxtend
```

```
Obtaining dependency information for mlxtend from https://files.pythonhosted.org/packages/1c/07/512f6a780239ad6ce06ce2aa7b4067583f5ddcfc7703a964a082c706a070/mlxtend-0.23.1-py3-none-any.whl.metadata
```

```
Downloading mlxtend-0.23.1-py3-none-any.whl.metadata (7.3 kB)
```

```
Requirement already satisfied: scipy>=1.2.1 in
```

```
c:\programdata\anaconda3\lib\site-packages (from mlxtend) (1.11.1)
```

```
Requirement already satisfied: numpy>=1.16.2 in
```

```
c:\programdata\anaconda3\lib\site-packages (from mlxtend) (1.24.3)
```

```
Requirement already satisfied: pandas>=0.24.2 in
```

```
c:\programdata\anaconda3\lib\site-packages (from mlxtend) (2.0.3)
```

```
Requirement already satisfied: scikit-learn>=1.0.2 in
```

```
c:\users\admin\appdata\roaming\python\python311\site-packages (from mlxtend) (1.4.1.post1)
```

```
Requirement already satisfied: matplotlib>=3.0.0 in
```

```
c:\programdata\anaconda3\lib\site-packages (from mlxtend) (3.7.2)
```

```
Requirement already satisfied: joblib>=0.13.2 in
```

```
c:\programdata\anaconda3\lib\site-packages (from mlxtend) (1.2.0)
```

```
Requirement already satisfied: contourpy>=1.0.1 in
```

```
c:\programdata\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (1.0.5)
```

```
Requirement already satisfied: cycycler>=0.10 in
```

```
c:\programdata\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (0.11.0)
```

```
Requirement already satisfied: fonttools>=4.22.0 in
```

```
c:\programdata\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (4.25.0)
```

```
Requirement already satisfied: kiwisolver>=1.0.1 in
```

```
c:\programdata\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (1.4.4)
```

```
Requirement already satisfied: packaging>=20.0 in
```

```

c:\programdata\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend)
(23.1)
Requirement already satisfied: pillow>=6.2.0 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend)
(9.4.0)
Requirement already satisfied: pyparsing<3.1,>=2.3.1 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend)
(3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend)
(2.8.2)
Requirement already satisfied: pytz>=2020.1 in
c:\programdata\anaconda3\lib\site-packages (from pandas>=0.24.2->mlxtend)
(2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in
c:\programdata\anaconda3\lib\site-packages (from pandas>=0.24.2->mlxtend)
(2023.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in
c:\programdata\anaconda3\lib\site-packages (from scikit-learn>=1.0.2->mlxtend)
(2.2.0)
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-
packages (from python-dateutil>=2.7->matplotlib>=3.0.0->mlxtend) (1.16.0)
Downloading mlxtend-0.23.1-py3-none-any.whl (1.4 MB)
----- 0.0/1.4 MB ? eta -:-:--
----- 0.0/1.4 MB ? eta -:-:--
----- 0.0/1.4 MB 435.7 kB/s eta 0:00:04
- ----- 0.1/1.4 MB 653.6 kB/s eta 0:00:03
----- 0.3/1.4 MB 1.8 MB/s eta 0:00:01
----- 0.5/1.4 MB 2.5 MB/s eta 0:00:01
----- 1.0/1.4 MB 4.4 MB/s eta 0:00:01
----- 1.4/1.4 MB 5.1 MB/s eta 0:00:00
Installing collected packages: mlxtend
Successfully installed mlxtend-0.23.1

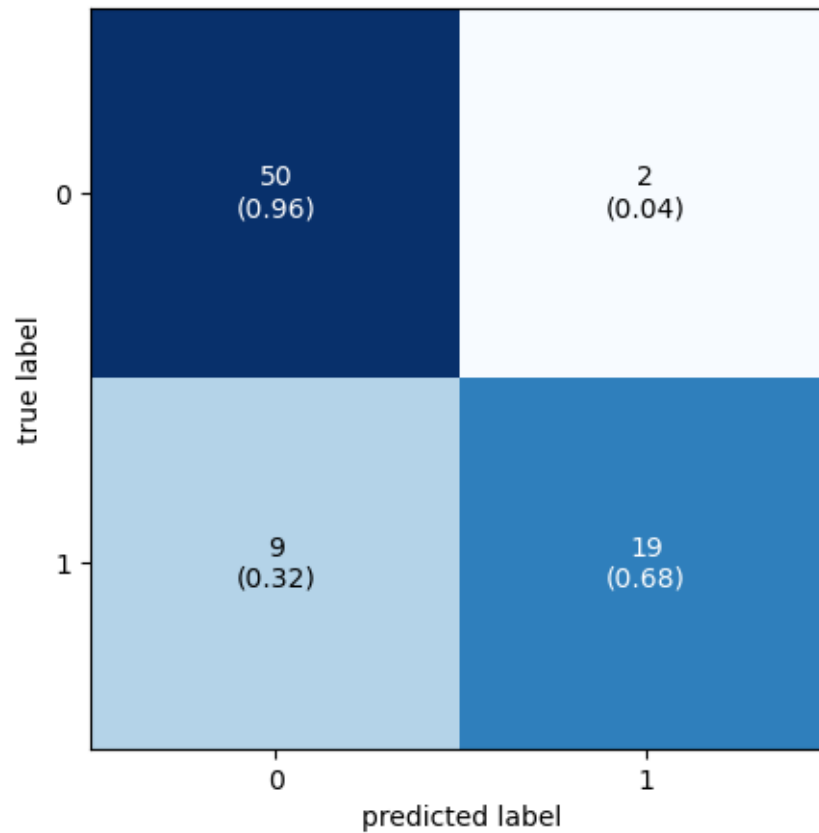
```

```
[46]: from mlxtend.plotting import plot_confusion_matrix
```

```
[47]: from sklearn.metrics import confusion_matrix, classification_report,
      accuracy_score, precision_score, recall_score, f1_score
```

```
[48]: plot_confusion_matrix(conf_mat=cm, figsize=(5,5), show_normed=True)
      plt.show()
```





```
[49]: print(f"TN value is {cm[0][0]}")
      print(f"FP value is {cm[0][1]}")
      print(f"FN value is {cm[1][0]}")
      print(f"TP value is {cm[1][1]}")
```

```
TN value is 50
FP value is 2
FN value is 9
TP value is 19
```

```
[50]: print(f"Accuracy score is {accuracy_score(y_test, y_pred)}")
```

```
Accuracy score is 0.8625
```

```
[51]: print(f"Error rate is {1-accuracy_score(y_test, y_pred)}")
```

```
Error rate is 0.13749999999999996
```

```
[52]: print(f"Precision score is {precision_score(y_test, y_pred)}")
```

```
Precision score is 0.9047619047619048
```

```
[53]: print(f"Recall score is {recall_score(y_test, y_pred)}")
```

Recall score is 0.6785714285714286

```
[54]: print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.85	0.96	0.90	52
1	0.90	0.68	0.78	28
accuracy			0.86	80
macro avg	0.88	0.82	0.84	80
weighted avg	0.87	0.86	0.86	80

```
[ ]: Name-Anurag Jadhav  
Roll No-13171  
Div-A
```