

```
In [ ]: NAME : SHINDE SHUBHAM DNYANDEV, ROLL NO. : EN23107121, BATCH : C
```

```
In [85]: import pandas as pd  
import seaborn as sns  
import matplotlib.pyplot as plt
```

```
In [87]: df = pd.read_csv("/home/admin1/networkadds.csv")  
df
```

```
Out[87]:
```

|     | 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         |
| ... | ...      | ...    | ... | ...             | ...       |
| 395 | 15691863 | Female | 46  | 41000           | 1         |
| 396 | 15706071 | Male   | 51  | 23000           | 1         |
| 397 | 15654296 | Female | 50  | 20000           | 1         |
| 398 | 15755018 | Male   | 36  | 33000           | 0         |
| 399 | 15594041 | Female | 49  | 36000           | 1         |

400 rows × 5 columns

```
In [89]: df.describe()
```

```
Out[89]:
```

|       | 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   |

```
In [91]: df.isnull().sum()
```

```
Out[91]:
```

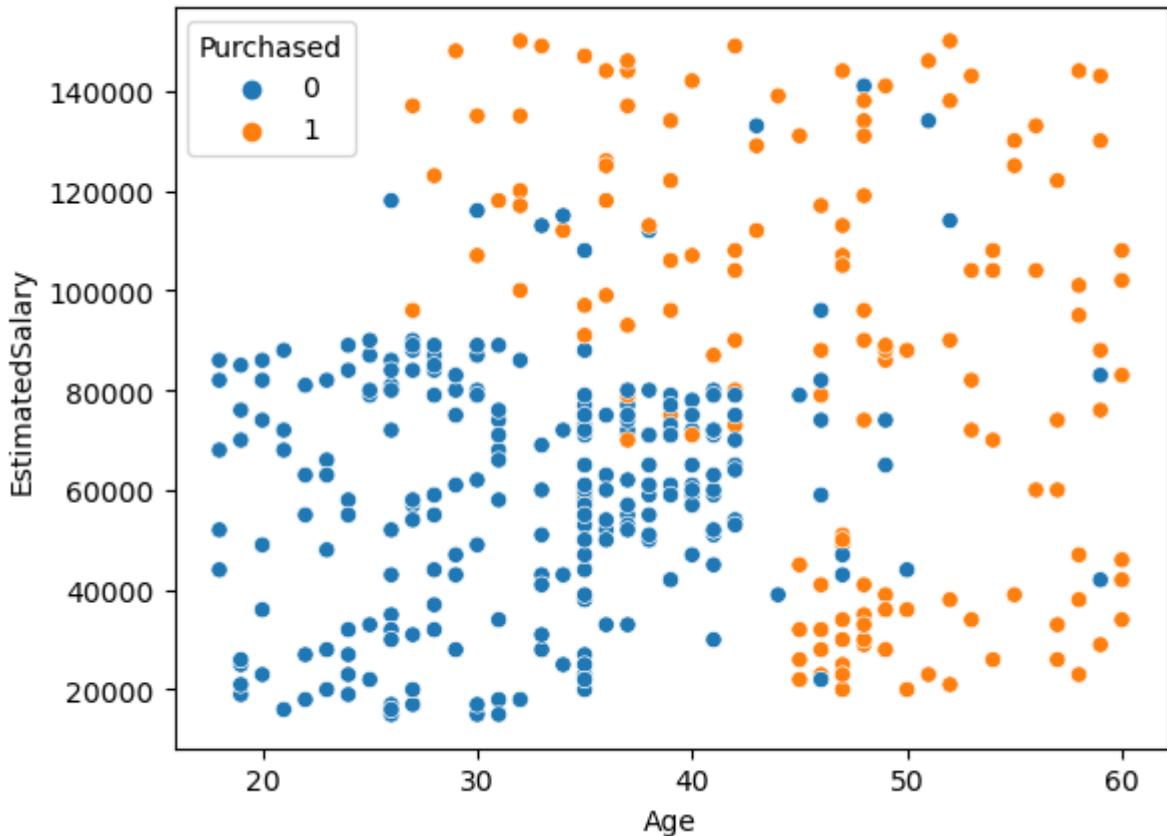
|                 |   |
|-----------------|---|
| User ID         | 0 |
| Gender          | 0 |
| Age             | 0 |
| EstimatedSalary | 0 |
| Purchased       | 0 |
| dtype: int64    |   |

```
In [93]: 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
```

```
In [95]: sns.scatterplot(data = df, x = 'Age', y = 'EstimatedSalary', hue = 'Purchased')
```

```
Out[95]: <Axes: xlabel='Age', ylabel='EstimatedSalary'>
```



```
In [97]: x = df.drop(columns = ['Purchased', 'Gender'])
y = df['Purchased']
```

```
In [99]: x
```

Out[99]:

|     | User ID  | Age | EstimatedSalary |
|-----|----------|-----|-----------------|
| 0   | 15624510 | 19  | 19000           |
| 1   | 15810944 | 35  | 20000           |
| 2   | 15668575 | 26  | 43000           |
| 3   | 15603246 | 27  | 57000           |
| 4   | 15804002 | 19  | 76000           |
| ... | ...      | ... | ...             |
| 395 | 15691863 | 46  | 41000           |
| 396 | 15706071 | 51  | 23000           |
| 397 | 15654296 | 50  | 20000           |
| 398 | 15755018 | 36  | 33000           |
| 399 | 15594041 | 49  | 36000           |

400 rows × 3 columns

In [101...]

y

Out[101...]

```
0      0
1      0
2      0
3      0
4      0
       .
395    1
396    1
397    1
398    0
399    1
Name: Purchased, Length: 400, dtype: int64
```

In [103...]

```
from sklearn.model_selection import train_test_split
```

In [128...]

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2)
```

In [130...]

```
from sklearn.preprocessing import StandardScaler
Scaler = StandardScaler()
Scaler
```

Out[130...]

▼ StandardScaler  
StandardScaler()

In [132...]

```
x_train_scale = Scaler.fit_transform(x_train)
x_test_scale = Scaler.transform(x_test)
```

In [134...]

```
from sklearn.linear_model import LogisticRegression
```

In [136...]

```
LR = LogisticRegression()
```

```
LR.fit(x_train_scale, y_train)
```

```
Out[136]: LogisticRegression()
          LogisticRegression()

In [138]: y_pred = LR.predict(x_test_scale)

In [140]: y_pred
Out[140]: array([0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0,
   0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0,
   1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0])

In [142]: LR.score(x_test, y_test)
/home/admin1/anaconda3/lib/python3.9/site-packages/sklearn/base.py:457: UserWarning: X
has feature names, but LogisticRegression was fitted without feature names
warnings.warn(
Out[142]: 0.675

In [144]: from sklearn.metrics import accuracy_score

In [146]: accuracy = accuracy_score(y_test, y_pred)
accuracy
Out[146]: 0.85

In [148]: CM = metrics.confusion_matrix(y_pred, y_test)
CM
Out[148]: array([[51,  9],
   [ 3, 17]])

In [176]: from sklearn.metrics import precision_score, recall_score, f1_score, classification_r
In [178]: precision = precision_score(y_test, y_pred)
precision
Out[178]: 0.85

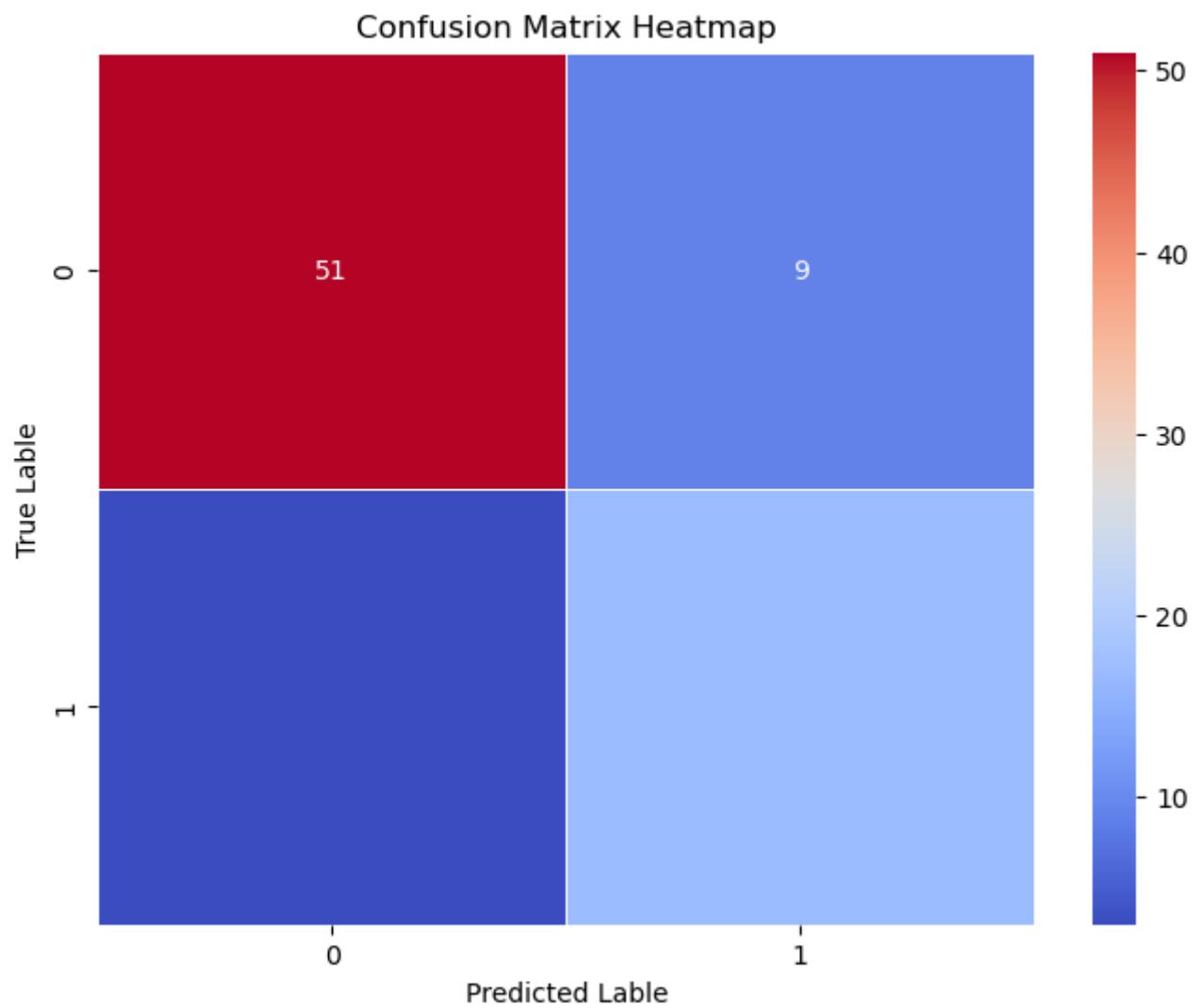
In [180]: recall = recall_score(y_test, y_pred)
recall
Out[180]: 0.6538461538461539

In [182]: f1 = f1_score(y_test, y_pred)
f1
Out[182]: 0.7391304347826088

In [184]: CR = classification_report(y_test, y_pred)
CR
Out[184]:          precision    recall  f1-score   support
          0       0.94      0.89      0.89      54\n
          1       0.82      0.85      0.83      80\n
  accuracy       0.82      0.85      0.83      134\n
   weighted avg       0.82      0.85      0.83      134

In [198]: plt.figure(figsize=(8, 6))
sns.heatmap(CM, annot=True, fmt='d', cmap='coolwarm', linewidths=0.5)
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
```

```
plt.title('Confusion Matrix Heatmap')
plt.show()
```



In [ ]: