```
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 mlxtend.plotting import plot_confusion_matrix
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score, precision_score, recall_score, f1_score
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

```
!pip install mlxtend
```

```
df = pd.read_csv("Customer_Behaviour.csv")
```

```
df.head()
```

```
df.shape
```

```
df.info()
```

```
df.describe()
```

```
df.isna().sum()
```

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

```
plt.show()


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


df["Gender"].value_counts()


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


df["Gender"] = df["Gender"].apply(gender_encoder)


df["Purchased"].value_counts()


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


countplot = sns.countplot(x="Purchased", hue="Gender", data=df, palette="twilight")
for i in countplot.containers:
    countplot.bar_label(i,)
plt.show()
```

```
sns.heatmap(df.corr(), annot=True)
plt.show()

x = df[["Age", "EstimatedSalary"]]
y = df["Purchased"]

scaler = StandardScaler()
x = scaler.fit_transform(x)

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)

x_train.shape, x_test.shape, y_train.shape, y_test.shape

model = LogisticRegression(n_jobs=-1)

model.fit(x_train, y_train)

y_pred = model.predict(x_test)

cm = confusion_matrix(y_test, y_pred)
print(cm)

plot_confusion_matrix(conf_mat=cm, figsize=(5,5), show_normed=True)
plt.show()

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]}")

print(f"Accuracy score is {accuracy_score(y_test, y_pred)}")
print(f"Error rate is {1-accuracy_score(y_test, y_pred)}")

print(f"Precision score is {precision_score(y_test, y_pred)}")

print(f"Recall score is {recall_score(y_test, y_pred)}")

print(classification_report(y_test, y_pred))
```