

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import precision_score, confusion_matrix,
accuracy_score, recall_score

df = pd.read_csv('Social_Network_Ads.csv')
df = pd.get_dummies(df, drop_first=True)
print(df.isnull().sum())

```

User ID	0
Age	0
EstimatedSalary	0
Purchased	0
Gender_Male	0

```

dtype: int64

X = df.drop('Purchased', axis=1)
y = df['Purchased']

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

logreg = LogisticRegression()
logreg.fit(X_train_scaled, y_train)
LogisticRegression()

y_train_pred = logreg.predict(X_train_scaled)
y_test_pred = logreg.predict(X_test_scaled)

train_accuracy = accuracy_score(y_train, y_train_pred)
train_precision = precision_score(y_train, y_train_pred)
train_recall = recall_score(y_train, y_train_pred)
train_cm = confusion_matrix(y_train, y_train_pred)

```

```

test_accuracy = accuracy_score(y_test, y_test_pred)
test_precision = precision_score(y_test, y_test_pred)
test_recall = recall_score(y_test, y_test_pred)
test_cm = confusion_matrix(y_test, y_test_pred)
print("Train Accuracy:", train_accuracy)
Train Accuracy: 0.821875
print("Train Precision:", train_precision)
Train Precision: 0.8295454545454546
print("Train Recall:", train_recall)
Train Recall: 0.6347826086956522
print("Train Confusion Matrix:")
Train Confusion Matrix:
print(train_cm)
[[190  15]
 [ 42  73]]
print("\nTest Accuracy:", test_accuracy)

Test Accuracy: 0.8875
print("Test Precision:", test_precision)
Test Precision: 0.9130434782608695
print("Test Recall:", test_recall)
Test Recall: 0.75
print("Test Confusion Matrix:")
Test Confusion Matrix:
print(test_cm)
[[50  2]
 [ 7 21]]

plt.figure(figsize=(8, 6))
sns.heatmap(test_cm, annot=True, fmt='d', cmap='Blues',
xticklabels=['Predicted 0', 'Predicted 1'], yticklabels=['Actual 0',
'Actual 1'])

```

```
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix (Test Set)')
plt.show()
```

