

```

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 confusion_matrix, accuracy_score, classification_report

data = pd.read_csv("/content/suv_data.csv")

print(data.head())
print(data.info())

```

```

      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
<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
None

```

```

X = data[["Age", "EstimatedSalary"]]
y = data["Purchased"]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)

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

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

y_pred = model.predict(X_test_scaled)

cm = confusion_matrix(y_test, y_pred)
print("\nConfusion Matrix:")
print(cm)

plt.figure(figsize=(5,4))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix - Logistic Regression (SUV Dataset)")
plt.show()

test_accuracy = accuracy_score(y_test, y_pred)
train_accuracy = model.score(X_train_scaled, y_train)

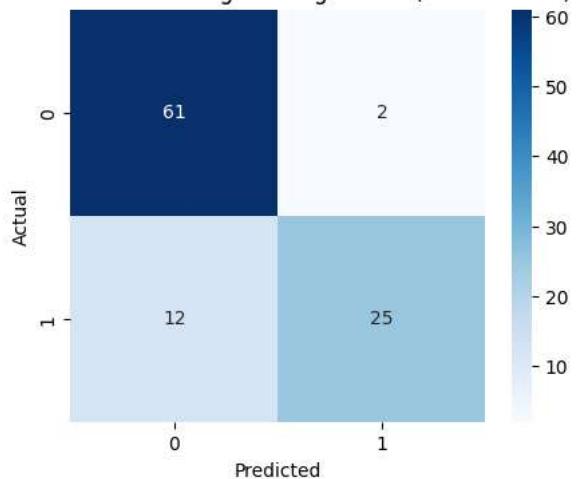
print("\nTraining Accuracy:", train_accuracy)
print("Test Accuracy:", test_accuracy)
print("\nClassification Report:\n", classification_report(y_test, y_pred))

```

Confusion Matrix:

```
[[61  2]
 [12 25]]
```

Confusion Matrix - Logistic Regression (SUV Dataset)



Training Accuracy: 0.8366666666666667

Test Accuracy: 0.86

Classification Report:

	precision	recall	f1-score	support
0	0.84	0.97	0.90	63
1	0.93	0.68	0.78	37
accuracy			0.86	100
macro avg	0.88	0.82	0.84	100
weighted avg	0.87	0.86	0.85	100