[2]:
```
import pandas as pd
import numpy as np
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

df=pd.read_csv(r"C:\Users\Admin\Downloads\archive\suv_data.csv")
print(df.head())

X = df[['Age', 'EstimatedSalary']]
y = df['Purchased']
```

```
    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
```

[3]:
```
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)
```

[4]:
```
model = LogisticRegression()
model.fit(X_train_scaled, y_train)
```

[4]:
```
LogisticRegression()
```

[7]:
```
y_pred = model.predict(X_test_scaled)
cm = confusion_matrix(y_test, y_pred)
tn, fp, fn, tp = cm.ravel()
print("\nConfusion Matrix:")
print(cm)
print(f"True Positives: {tp}")
print(f"True Negatives: {tn}")
print(f"False Positives: {fp}")
print(f"False Negatives: {fn}")
```

```
Confusion Matrix:
[[61  2]
 [12 25]]
True Positives: 25
True Negatives: 61
False Positives: 2
False Negatives: 12
```

[8]:
```
test_accuracy = model.score(X_test_scaled, y_test)
train_accuracy = model.score(X_train_scaled, y_train)
print(f"\nTest Accuracy: {test_accuracy:.4f}")
print(f"Training Accuracy: {train_accuracy:.4f}")
```

```
Test Accuracy: 0.8600
Training Accuracy: 0.8367
```