In [97]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [99]:
```
dataset = pd.read_csv("Social_Network_Ads.csv")
dataset.head()
```

Out[99]:

	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

In [101…
```
dataset.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 [103…
```
X = dataset.iloc[:, [2, 3]].values   # Age and EstimatedSalary
y = dataset.iloc[:, 4].values        # Purchased
```

In [105…
```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.25, random_state=0
)
```

In [107…
```
from sklearn.preprocessing import StandardScaler

sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

In [109…
```
from sklearn.linear_model import LogisticRegression

classifier = LogisticRegression(random_state=0)
classifier.fit(X_train, y_train)
```

Out[109…

▾ LogisticRegression ⓘ ?

▶ Parameters

In [111…
```
y_pred = classifier.predict(X_test)
y_pred
```

Out[111…
```
array([0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
       0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1])
```

In [113…
```
from sklearn.metrics import confusion_matrix

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

Out[113…
```
array([[65,  3],
       [ 8, 24]])
```

In [115…
```
TN = cm[0][0]
FP = cm[0][1]
FN = cm[1][0]
TP = cm[1][1]

print("True Positive (TP):", TP)
print("False Positive (FP):", FP)
print("True Negative (TN):", TN)
print("False Negative (FN):", FN)
```

```
True Positive (TP): 24
False Positive (FP): 3
True Negative (TN): 65
False Negative (FN): 8
```

In [117…
```
accuracy = (TP + TN) / (TP + TN + FP + FN)
accuracy
```

Out[117…
```
np.float64(0.89)
```

In [119…
```
error_rate = 1 - accuracy
error_rate
```

Out[119…
```
np.float64(0.10999999999999999)
```

In [121…
```
precision = TP / (TP + FP)
precision
```

Out[121…
```
np.float64(0.8888888888888888)
```

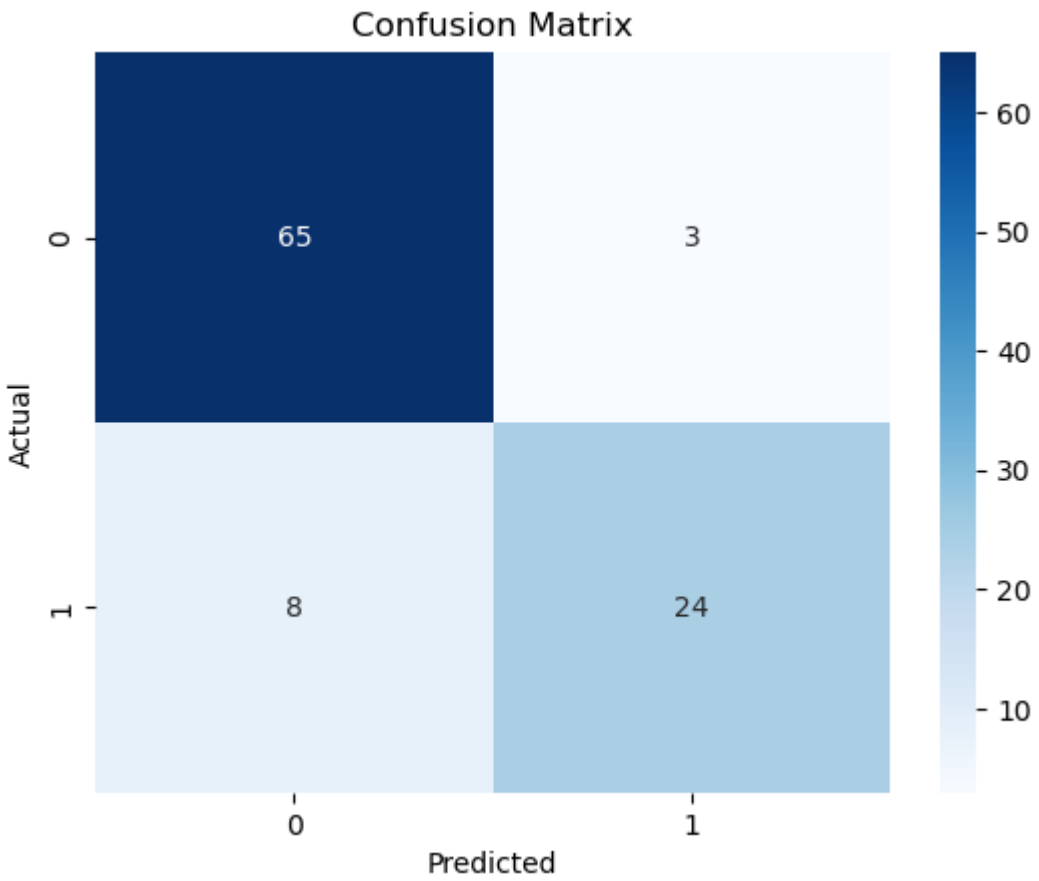
In [122…
```
recall = TP / (TP + FN)
recall
```

Out[122…
```
np.float64(0.75)
```

In [125…
```
print("Accuracy:", accuracy)
print("Error Rate:", error_rate)
print("Precision:", precision)
print("Recall:", recall)
```

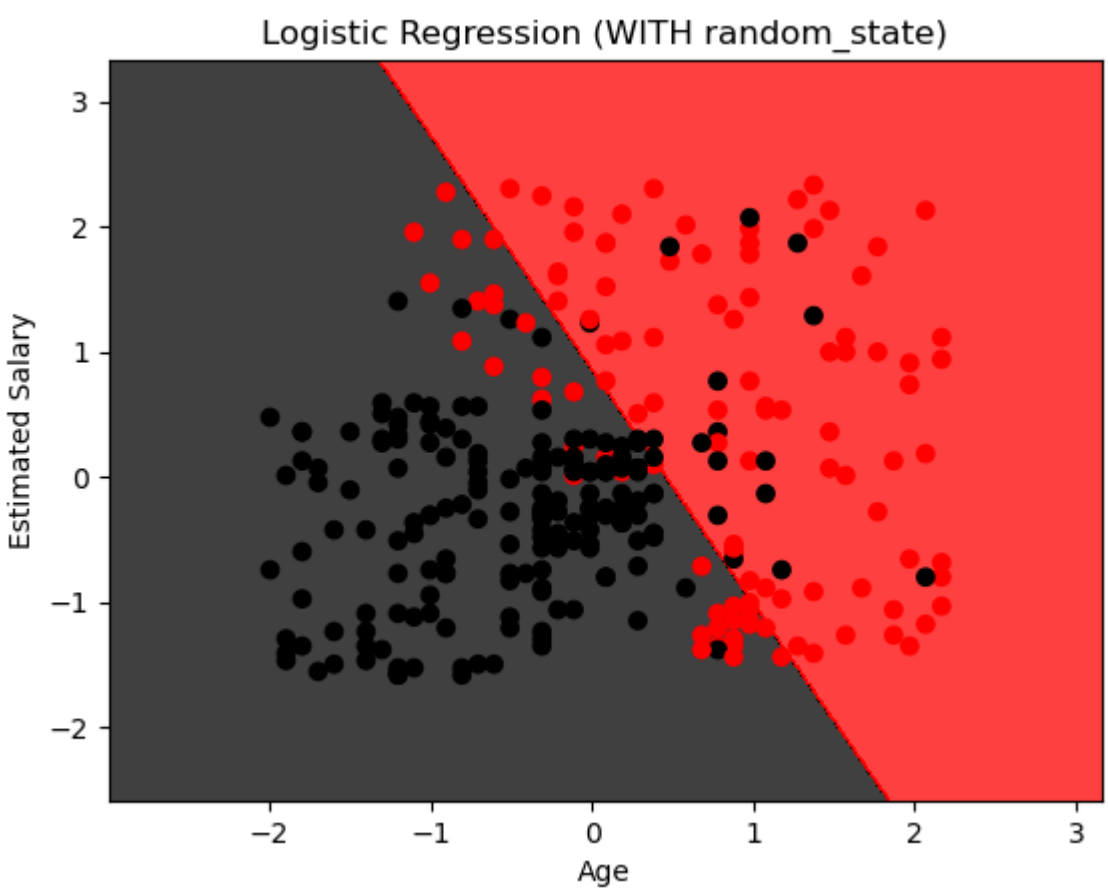
```
Accuracy: 0.89
Error Rate: 0.10999999999999999
Precision: 0.8888888888888888
Recall: 0.75
```

In [126…
```
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()
```



In [128…

```
from matplotlib.colors import ListedColormap

X_set, y_set = X_train, y_train
X1, X2 = np.meshgrid(
    np.arange(X_set[:, 0].min() - 1, X_set[:, 0].max() + 1, 0.01),
    np.arange(X_set[:, 1].min() - 1, X_set[:, 1].max() + 1, 0.01)
)

plt.contourf(
    X1, X2,
    classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
    alpha=0.75,
    cmap=ListedColormap(('black', 'red'))
)

plt.scatter(
    X_set[:, 0], X_set[:, 1],
    c=y_set,
    cmap=ListedColormap(('black', 'red'))
)

plt.title("Logistic Regression (WITH random_state)")
plt.xlabel("Age")
plt.ylabel("Estimated Salary")
plt.show()
```



In [131…
```
X_train2, X_test2, y_train2, y_test2 = train_test_split(
    X, y, test_size=0.25
)
```

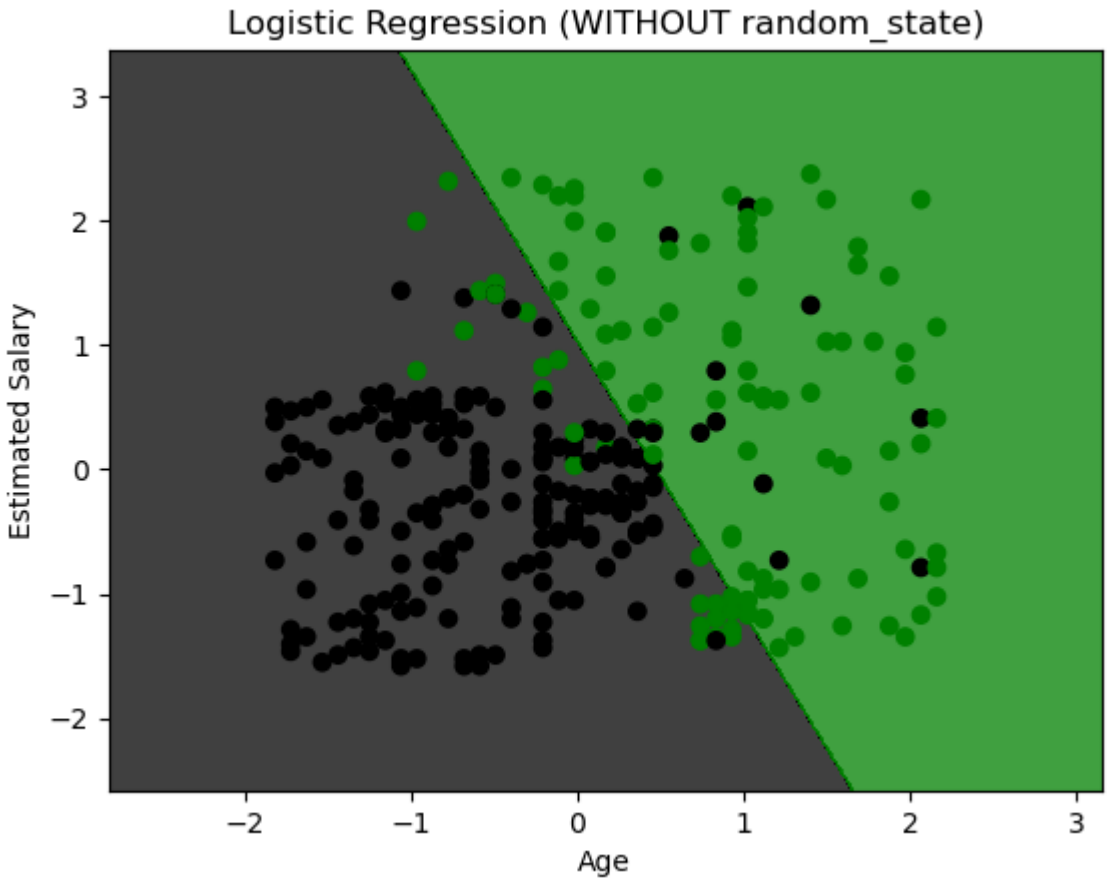
In [133…
```
sc2 = StandardScaler()
X_train2 = sc2.fit_transform(X_train2)
X_test2 = sc2.transform(X_test2)

classifier2 = LogisticRegression()
classifier2.fit(X_train2, y_train2)
```

Out[133…
▾ LogisticRegression ⓘ ⓘ

▶ Parameters

In [139…
```
X_set, y_set = X_train2, y_train2

X1, X2 = np.meshgrid(
    np.arange(X_set[:, 0].min() - 1, X_set[:, 0].max() + 1, 0.01),
    np.arange(X_set[:, 1].min() - 1, X_set[:, 1].max() + 1, 0.01)
)

plt.contourf(
    X1, X2,
    classifier2.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
    alpha=0.75,
    cmap=ListedColormap(('black', 'green'))
)

plt.scatter(
    X_set[:, 0], X_set[:, 1],
    c=y_set,
    cmap=ListedColormap(('black', 'green'))
)

plt.title("Logistic Regression (WITHOUT random_state)")
```

```
plt.xlabel("Age")
plt.ylabel("Estimated Salary")
plt.show()
```



In [ ]: