

```
In [5]: import pandas as pd  
import numpy as np
```

```
In [15]: import pandas as pd  
  
data = pd.read_csv(r"C:\Users\kiruthika\Downloads\creditcard.csv.zip")  
print("Dataset loaded successfully")  
data.head()
```

Dataset loaded successfully

```
Out[15]:
```

	Time	V1	V2	V3	V4	V5	V6	V7	
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.0
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.0
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.2
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.3
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.2

5 rows × 31 columns



```
In [14]: import os  
  
os.path.exists(r"C:\Users\kiruthika\Downloads\creditcard.csv.zip")
```

```
Out[14]: True
```

```
In [16]: data.head()  
data.info()  
data['Class'].value_counts()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 284807 entries, 0 to 284806
Data columns (total 31 columns):
 #   Column   Non-Null Count   Dtype  
 ---  -- 
 0   Time     284807 non-null    float64
 1   V1       284807 non-null    float64
 2   V2       284807 non-null    float64
 3   V3       284807 non-null    float64
 4   V4       284807 non-null    float64
 5   V5       284807 non-null    float64
 6   V6       284807 non-null    float64
 7   V7       284807 non-null    float64
 8   V8       284807 non-null    float64
 9   V9       284807 non-null    float64
 10  V10      284807 non-null    float64
 11  V11      284807 non-null    float64
 12  V12      284807 non-null    float64
 13  V13      284807 non-null    float64
 14  V14      284807 non-null    float64
 15  V15      284807 non-null    float64
 16  V16      284807 non-null    float64
 17  V17      284807 non-null    float64
 18  V18      284807 non-null    float64
 19  V19      284807 non-null    float64
 20  V20      284807 non-null    float64
 21  V21      284807 non-null    float64
 22  V22      284807 non-null    float64
 23  V23      284807 non-null    float64
 24  V24      284807 non-null    float64
 25  V25      284807 non-null    float64
 26  V26      284807 non-null    float64
 27  V27      284807 non-null    float64
 28  V28      284807 non-null    float64
 29  Amount    284807 non-null    float64
 30  Class     284807 non-null    int64
dtypes: float64(30), int64(1)
memory usage: 67.4 MB
```

Out[16]: Class

0	284315
1	492

Name: count, dtype: int64

In [17]: X = data.drop('Class', axis=1)
y = data['Class']

In [18]: from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X['Amount'] = scaler.fit_transform(X[['Amount']])

In [19]: 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=42, stratify=y
)

In [22]: model = LogisticRegression(max_iter=5000) # increase iterations

```
model.fit(X_train, y_train)
```

Out[22]:

- ▼ LogisticRegression ⓘ ⓘ

- Parameters

In [23]:

```
y_pred = model.predict(X_test)
```

In [24]:

```
from sklearn.metrics import accuracy_score
```

```
print("Accuracy:", accuracy_score(y_test, y_pred))
```

Accuracy: 0.9991854161399961

In [25]:

```
from sklearn.metrics import confusion_matrix
```

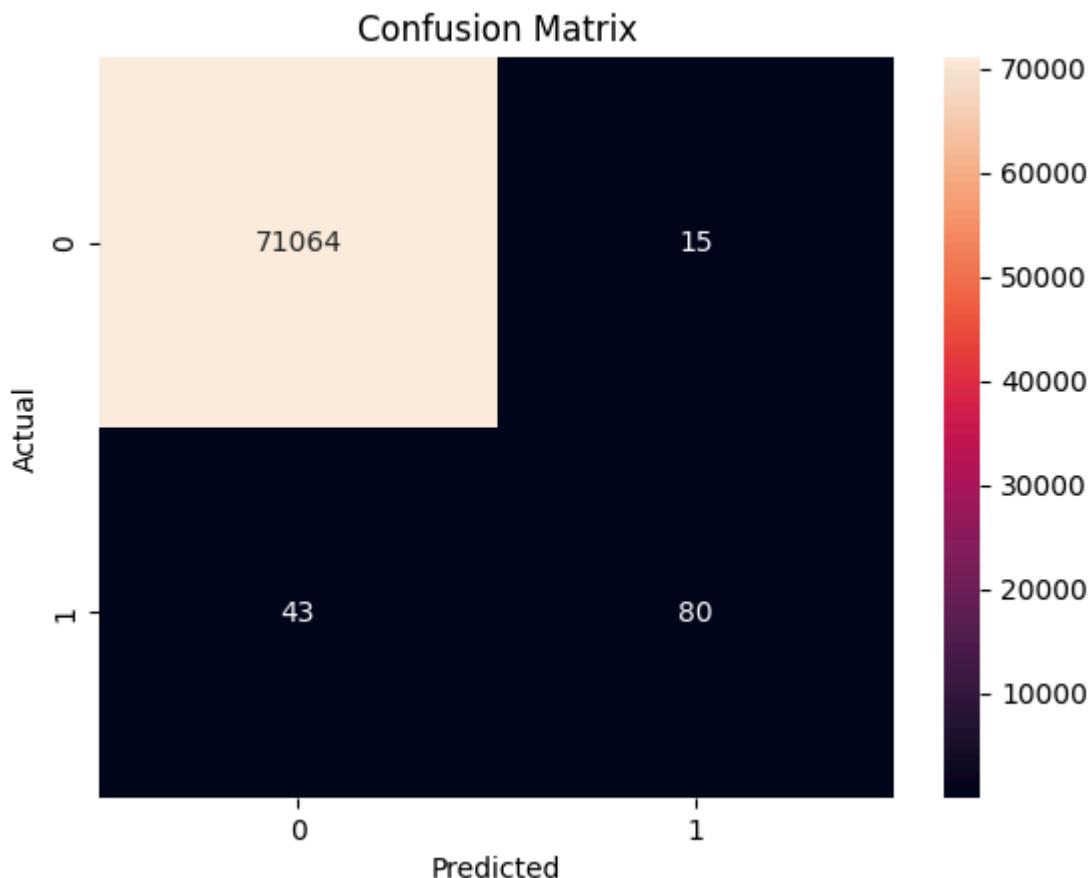
```
cm = confusion_matrix(y_test, y_pred)
print(cm)
```

```
[[71064    15]
 [   43    80]]
```

In [26]:

```
import matplotlib.pyplot as plt
import seaborn as sns
```

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



```
In [27]: from sklearn.metrics import classification_report  
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	71079
1	0.84	0.65	0.73	123
accuracy			1.00	71202
macro avg	0.92	0.83	0.87	71202
weighted avg	1.00	1.00	1.00	71202

```
In [ ]:
```