

```

# @title Import Required Libraries
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 classification_report, confusion_matrix,
accuracy_score

# @title Load the Dataset
df = pd.read_csv("creditcard.csv")    # use your file name
df.head()

{"type": "dataframe", "variable_name": "df"}

# @title Understand the Dataset
df.info()

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5848 entries, 0 to 5847
Data columns (total 31 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   Time        5848 non-null   int64
 1   V1          5848 non-null   float64
 2   V2          5848 non-null   float64
 3   V3          5848 non-null   float64
 4   V4          5848 non-null   float64
 5   V5          5848 non-null   float64
 6   V6          5848 non-null   float64
 7   V7          5848 non-null   float64
 8   V8          5848 non-null   float64
 9   V9          5848 non-null   float64
10  V10         5848 non-null   float64
11  V11         5848 non-null   float64
12  V12         5848 non-null   float64
13  V13         5848 non-null   float64
14  V14         5848 non-null   float64
15  V15         5848 non-null   float64
16  V16         5848 non-null   float64
17  V17         5848 non-null   float64
18  V18         5848 non-null   float64
19  V19         5848 non-null   float64
20  V20         5848 non-null   float64
21  V21         5848 non-null   float64
22  V22         5848 non-null   float64

```

```
23 V23      5848 non-null float64
24 V24      5848 non-null float64
25 V25      5848 non-null float64
26 V26      5847 non-null float64
27 V27      5847 non-null float64
28 V28      5847 non-null float64
29 Amount    5847 non-null float64
30 Class     5847 non-null float64
dtypes: float64(30), int64(1)
memory usage: 1.4 MB
```

```
# @title
df.isnull().sum()
```

```
Time      0
V1        0
V2        0
V3        0
V4        0
V5        0
V6        0
V7        0
V8        0
V9        0
V10       0
V11       0
V12       0
V13       0
V14       0
V15       0
V16       0
V17       0
V18       0
V19       0
V20       0
V21       0
V22       0
V23       0
V24       0
V25       0
V26       1
V27       1
V28       1
Amount    1
Class     1
dtype: int64
```

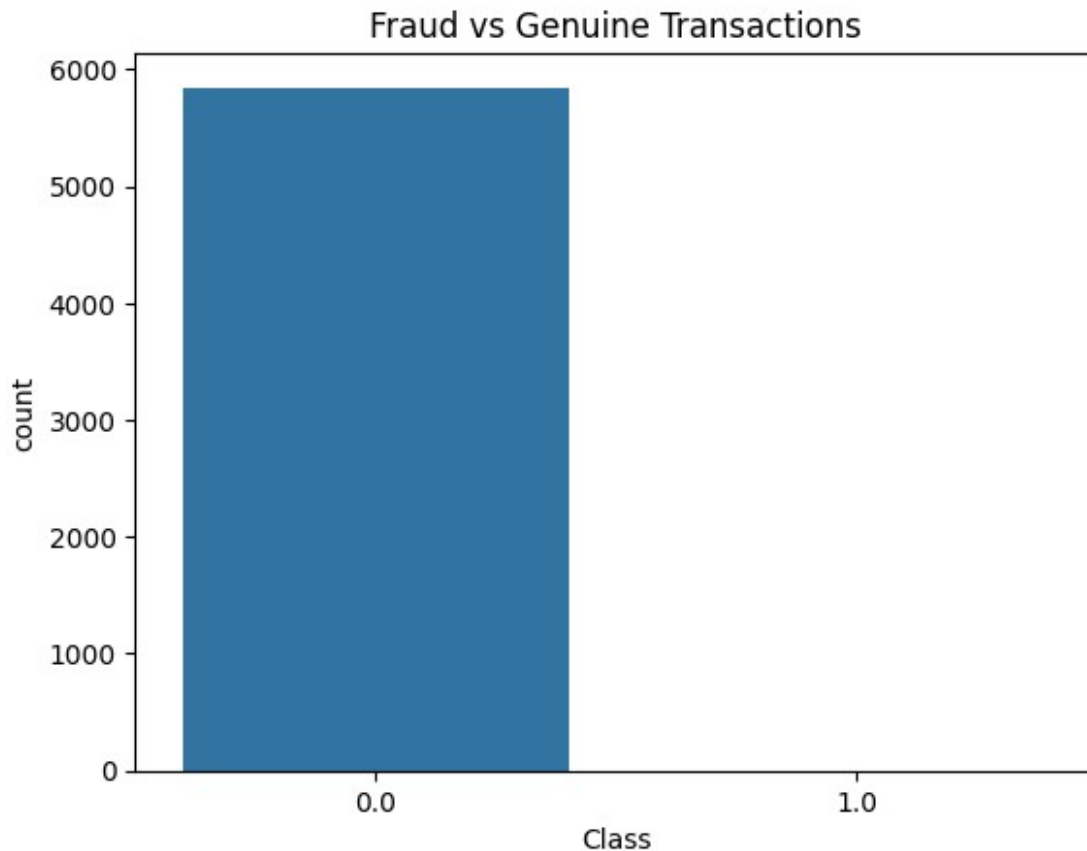
```
df['Class'].value_counts()
```

```

Class
0.0    5844
1.0      3
Name: count, dtype: int64

# @title Visualize Class Imbalance
sns.countplot(x='Class', data=df)
plt.title("Fraud vs Genuine Transactions")
plt.show()

```



```

# @title Feature Scaling
scaler = StandardScaler()

df['Amount'] = scaler.fit_transform(df[['Amount']])
df['Time'] = scaler.fit_transform(df[['Time']])

# @title Feature Selection
X = df.drop(columns=['Class'])
y = df['Class']

# @title Train-Test Split
df_cleaned = df.dropna(subset=['Class'])
X_cleaned = df_cleaned.drop(columns=['Class'])

```

```

y_cleaned = df_cleaned['Class']

X_train, X_test, y_train, y_test = train_test_split(
    X_cleaned, y_cleaned, test_size=0.2, random_state=42,
    stratify=y_cleaned
)

model = LogisticRegression(max_iter=1000, class_weight='balanced')
model.fit(X_train, y_train)

LogisticRegression(class_weight='balanced', max_iter=1000)

# @title Build Logistic Regression Model
y_pred = model.predict(X_test)

# @title Make Predictions
print("Accuracy:", accuracy_score(y_test, y_pred))

Accuracy: 1.0

# @title Model Evaluation
print(classification_report(y_test, y_pred))

```

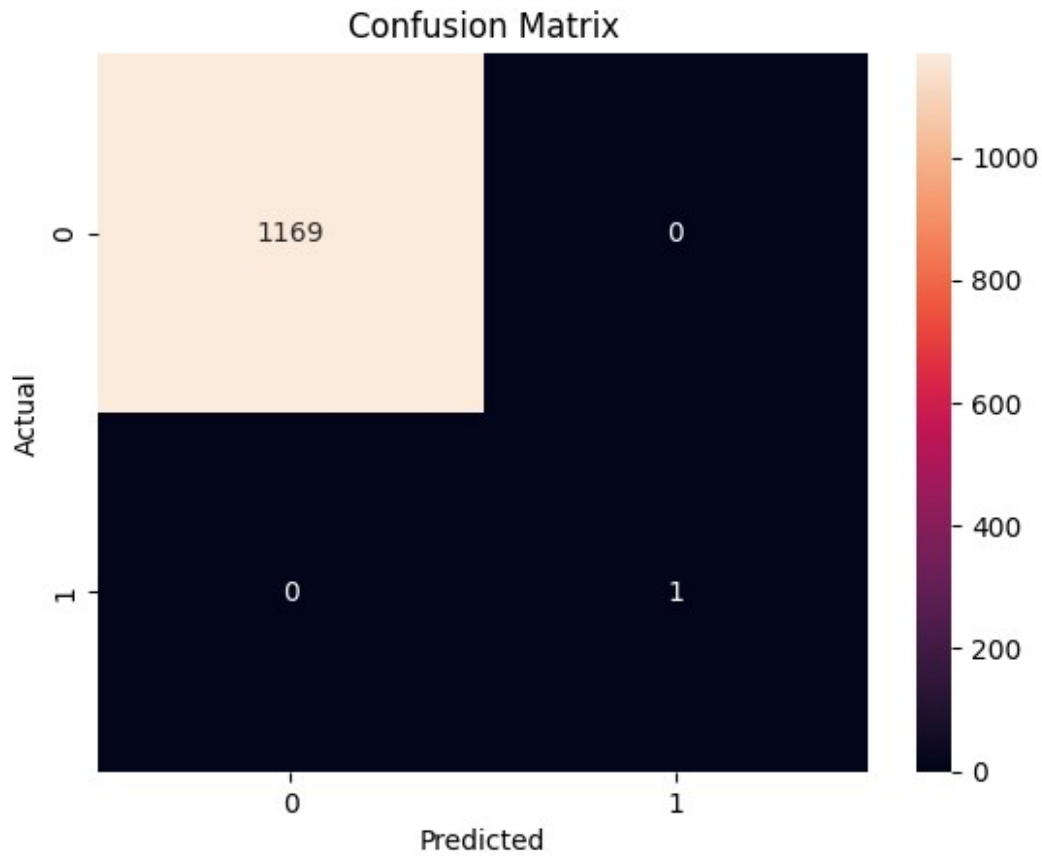
	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	1169
1.0	1.00	1.00	1.00	1
accuracy			1.00	1170
macro avg	1.00	1.00	1.00	1170
weighted avg	1.00	1.00	1.00	1170

```

# @title
cm = confusion_matrix(y_test, y_pred)

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

```



```
# @title Conclusion
```

```
print("Credit Card Fraud Detection model built successfully.")  
print("Logistic Regression effectively detects fraudulent  
transactions.")  
print("Handling class imbalance improves fraud detection  
performance.")
```

```
Credit Card Fraud Detection model built successfully.  
Logistic Regression effectively detects fraudulent transactions.  
Handling class imbalance improves fraud detection performance.
```