In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.model_selection import train_test_split
```

#Data Preprocessing

In [3]:

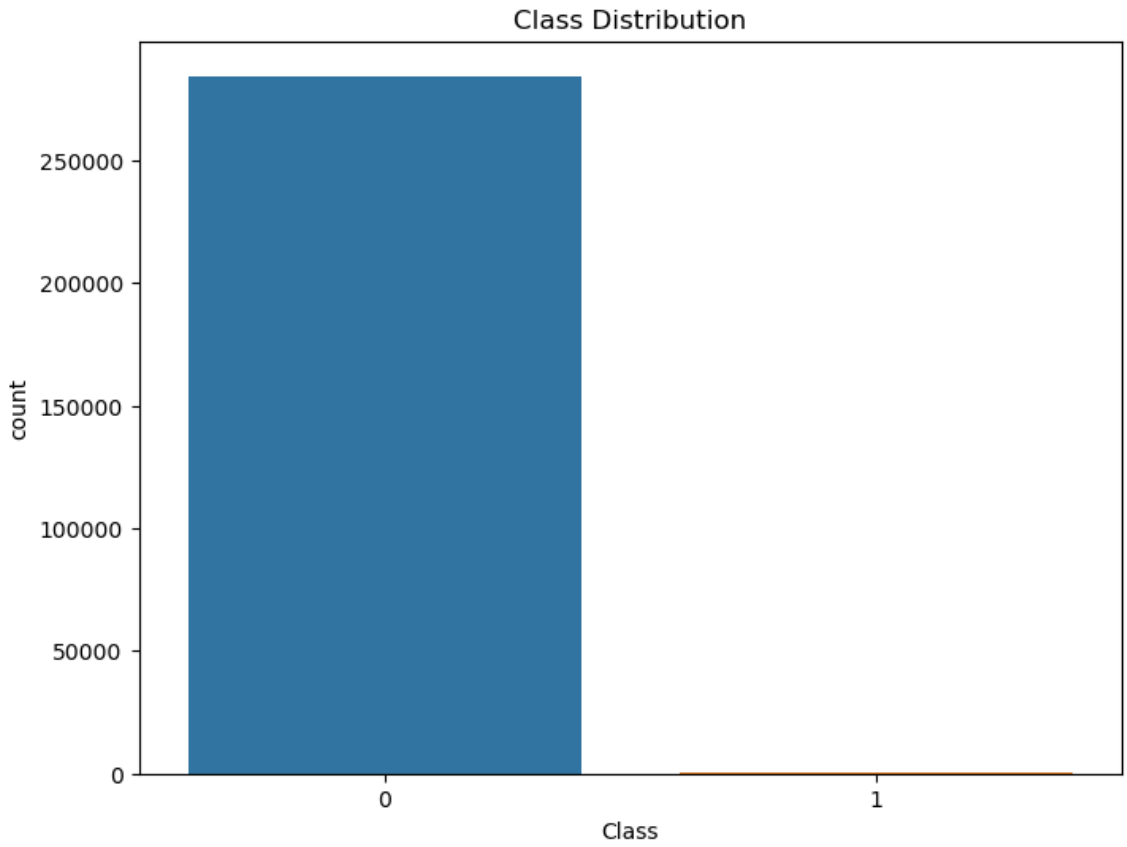
```
df = pd.read_csv('creditcard.csv')

# Check for missing values
df.isnull().sum()

# Check class distribution
df['Class'].value_counts()

# Scale the features
from sklearn.preprocessing import StandardScaler
df['Amount'] = StandardScaler().fit_transform(df['Amount'].values.reshape(-1, 1))
df.drop('Time', axis=1, inplace=True)
```

Data Visualization

In [4]:

```
plt.figure(figsize=(8, 6))
sns.countplot(df['Class'])
plt.title('Class Distribution')
plt.show()
```

```
C:\Users\shaha\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futu
reWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterp
retation.
  warnings.warn(
```



#Splitting Data into Train and Test sets

In [5]:

```
X = df.drop('Class', axis=1)
y = df['Class']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42
```

#Building Machine Learning Models

Decision Tree

In [6]:

```
from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier()
dtc.fit(X_train, y_train)

y_pred = dtc.predict(X_test)
print('Decision Tree Classifier Results:')
print('Confusion Matrix:')
print(confusion_matrix(y_test, y_pred))
print('Classification Report:')
print(classification_report(y_test, y_pred))
```

```
Decision Tree Classifier Results:
Confusion Matrix:
[[85269    38]
 [   25   111]]
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85307
           1       0.74      0.82      0.78       136

    accuracy                           1.00     85443
   macro avg       0.87      0.91      0.89     85443
weighted avg       1.00      1.00      1.00     85443
```

#Model Evaluation

In [8]:

```
# Decision Tree Results
dtc_pred = dtc.predict(X_test)
print('Decision Tree Results:')
print('Confusion Matrix:')
print(confusion_matrix(y_test, dtc_pred))
print('Classification Report:')
print(classification_report(y_test, dtc_pred))
```

```
Decision Tree Results:
Confusion Matrix:
[[85269    38]
 [   25   111]]
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85307
           1       0.74      0.82      0.78       136

    accuracy                           1.00     85443
   macro avg       0.87      0.91      0.89     85443
weighted avg       1.00      1.00      1.00     85443
```

In [ ]:

```

```