In [1]:
```
!pip install -U imbalanced-learn
print("Imbalanced-learn installation/update completed.")
```

```
Collecting imbalanced-learnImbalanced-learn installation/update completed.
  Downloading imbalanced_learn-0.12.3-py3-none-any.whl (258 kB)

Requirement already satisfied, skipping upgrade: scikit-learn>=1.0.2 in c:\users\sanka\anaconda3\lib\site-packages (from imbala
nced-learn) (1.3.2)
Requirement already satisfied, skipping upgrade: numpy>=1.17.3 in c:\users\sanka\anaconda3\lib\site-packages (from imbalanced-l
earn) (1.19.2)
Requirement already satisfied, skipping upgrade: scipy>=1.5.0 in c:\users\sanka\anaconda3\lib\site-packages (from imbalanced-le
arn) (1.5.2)
Requirement already satisfied, skipping upgrade: joblib>=1.1.1 in c:\users\sanka\anaconda3\lib\site-packages (from imbalanced-l
earn) (1.4.0)
Requirement already satisfied, skipping upgrade: threadpoolctl>=2.0.0 in c:\users\sanka\anaconda3\lib\site-packages (from imbal
anced-learn) (2.1.0)
Installing collected packages: imbalanced-learn
Successfully installed imbalanced-learn-0.12.3
```

In [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 classification_report, confusion_matrix, precision_score, recall_score, f1_score
from imblearn.over_sampling import SMOTE
```

In [4]:
```
df = pd.read_csv('creditcard.csv')
```

In [5]:
```
df.info()
```

```
<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
```

In [7]:
```
df.shape
```

Out[7]:
```
(284807, 31)
```

In [8]:
```
df.head()
```

Out[8]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.12853
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.16717
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.32764
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.64737
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.20601

5 rows × 31 columns

In [9]:
```
df.tail()
```

Out[9]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24
284802	172786.0	-11.881118	10.071785	-9.834783	-2.066656	-5.364473	-2.606837	-4.918215	7.305334	1.914428	...	0.213454	0.111864	1.014480	-0.509348
284803	172787.0	-0.732789	-0.055080	2.035030	-0.738589	0.868229	1.058415	0.024330	0.294869	0.584800	...	0.214205	0.924384	0.012463	-1.016226
284804	172788.0	1.919565	-0.301254	-3.249640	-0.557828	2.630515	3.031260	-0.296827	0.708417	0.432454	...	0.232045	0.578229	-0.037501	0.640134
284805	172788.0	-0.240440	0.530483	0.702510	0.689799	-0.377961	0.623708	-0.686180	0.679145	0.392087	...	0.265245	0.800049	-0.163298	0.123205
284806	172792.0	-0.533413	-0.189733	0.703337	-0.506271	-0.012546	-0.649617	1.577006	-0.414650	0.486180	...	0.261057	0.643078	0.376777	0.008797

5 rows × 31 columns

In [11]:
```
df.columns
```

Out[11]:
```
Index(['Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10',
       'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19', 'V20',
       'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28', 'Amount',
       'Class'],
      dtype='object')
```

In [12]:
```
print("\nClass distribution:")
print(df['Class'].value_counts())
```

```
Class distribution:
0    284315
1       492
Name: Class, dtype: int64
```

In [15]:
```
df.describe()
```

Out[15]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...
count	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	...
mean	-1.050379e-14	3.919560e-15	5.688174e-16	-8.769071e-15	2.782312e-15	-1.552563e-15	2.010663e-15	-1.694249e-15	-1.927028e-16	-3.137024e-15	...
std	1.000002e+00	1.958696e+00	1.651309e+00	1.516255e+00	1.415869e+00	1.380247e+00	1.332271e+00	1.237094e+00	1.194353e+00	1.098632e+00	...
min	-1.996583e+00	-5.640751e+01	-7.271573e+01	-4.832559e+01	-5.683171e+00	-1.137433e+02	-2.616051e+01	-4.355724e+01	-7.321672e+01	-1.343407e+01	...
25%	-8.552120e-01	-9.203734e-01	-5.985499e-01	-8.903648e-01	-8.486401e-01	-6.915971e-01	-7.682956e-01	-5.540759e-01	-2.086297e-01	-6.430976e-01	...
50%	-2.131453e-01	1.810880e-02	6.548556e-02	1.798463e-01	-1.984653e-02	-5.433583e-02	-2.741871e-01	4.010308e-02	2.235804e-02	-5.142873e-02	...
75%	9.372174e-01	1.315642e+00	8.037239e-01	1.027196e+00	7.433413e-01	6.119264e-01	3.985649e-01	5.704361e-01	3.273459e-01	5.971390e-01	...
max	1.642058e+00	2.454930e+00	2.205773e+01	9.382558e+00	1.687534e+01	3.480167e+01	7.330163e+01	1.205895e+02	2.000721e+01	1.559499e+01	...

8 rows × 31 columns

In [17]:
```
df.isnull().sum()
```

Out[17]:
```
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       0
V27       0
V28       0
Amount    0
Class     0
dtype: int64
```

In [20]:
```
df.corr()
```

Out[20]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	
Time	1.000000	1.173963e-01	-1.059333e-02	-4.196182e-01	-1.052602e-01	1.730721e-01	-6.301647e-02	8.471437e-02	-3.694943e-02	-8.660434e-03	...	4.4735
V1	0.117396	1.000000e+00	4.697350e-17	-1.424390e-15	1.755316e-17	6.391162e-17	2.398071e-16	1.991550e-15	-9.490675e-17	2.169581e-16	...	-1.75
V2	-0.010593	4.697350e-17	1.000000e+00	2.512175e-16	-1.126388e-16	-2.039868e-16	5.024680e-16	3.966486e-16	-4.413984e-17	-5.728718e-17	...	8.4444
V3	-0.419618	-1.424390e-15	2.512175e-16	1.000000e+00	-3.416910e-16	-1.436514e-15	1.431581e-15	2.168574e-15	3.433113e-16	-4.233770e-16	...	-2.97
V4	-0.105260	1.755316e-17	-1.126388e-16	-3.416910e-16	1.000000e+00	-1.940929e-15	-2.712659e-16	1.556330e-16	5.195643e-16	3.859585e-16	...	-9.97
V5	0.173072	6.391162e-17	-2.039868e-16	-1.436514e-15	-1.940929e-15	1.000000e+00	7.926364e-16	-4.209851e-16	7.589187e-16	4.205206e-16	...	-1.36
V6	-0.063016	2.398071e-16	5.024680e-16	1.431581e-15	-2.712659e-16	7.926364e-16	1.000000e+00	1.429426e-16	-1.707421e-16	1.114447e-16	...	-1.57
V7	0.084714	1.991550e-15	3.966486e-16	2.168574e-15	1.556330e-16	-4.209851e-16	1.429426e-16	1.000000e+00	-8.691834e-17	7.933251e-16	...	1.9386
V8	-0.036949	-9.490675e-17	-4.413984e-17	3.433113e-16	5.195643e-16	7.589187e-16	-1.707421e-16	-8.691834e-17	1.000000e+00	2.900829e-16	...	-2.41
V9	-0.008660	2.169581e-16	-5.728718e-17	-4.233770e-16	3.859585e-16	4.205206e-16	1.114447e-16	7.933251e-16	2.900829e-16	1.000000e+00	...	4.5783
V10	0.030617	7.433820e-17	-4.782388e-16	6.289267e-16	6.055490e-16	-6.601716e-16	2.850776e-16	3.043333e-17	9.051847e-17	-2.771761e-16	...	8.0895
V11	-0.247689	2.438580e-16	9.468995e-16	-5.501758e-17	-2.083600e-16	7.342759e-16	4.865799e-16	-1.084105e-15	1.954747e-16	4.682341e-16	...	-3.91
V12	0.124348	2.422086e-16	-6.588252e-16	2.206522e-16	-5.657963e-16	3.761033e-16	2.140589e-16	1.510045e-15	-6.266057e-17	-2.445230e-15	...	3.2295
V13	-0.065902	-2.115458e-16	3.854521e-16	-6.883375e-16	-1.506129e-16	-9.578659e-16	-2.268061e-16	-9.892325e-17	-2.382948e-16	-2.650351e-16	...	9.4991
V14	-0.098757	9.352582e-16	-2.541036e-16	4.271336e-16	-8.522435e-17	-3.634803e-16	3.452801e-16	-1.729462e-16	-1.131098e-16	2.343317e-16	...	1.6341
V15	-0.183453	-3.252451e-16	2.831060e-16	1.122756e-16	-1.507718e-16	-5.132620e-16	-6.368111e-18	1.936832e-17	2.021491e-16	-1.588105e-15	...	1.9474
V16	0.011903	6.308789e-16	4.934097e-17	1.183364e-15	-6.939204e-16	-3.517076e-16	-2.477917e-16	2.893672e-16	5.027192e-16	-3.251906e-16	...	-3.92
V17	-0.073297	-5.011524e-16	-9.883008e-16	4.576619e-17	-4.397925e-16	1.425729e-16	3.567582e-16	1.149692e-15	-3.508777e-16	6.535992e-16	...	-7.75
V18	0.090438	2.870125e-16	2.636654e-16	5.427965e-16	1.493667e-16	1.109525e-15	2.811474e-16	-1.116789e-16	-4.093852e-16	1.203843e-16	...	-1.14
V19	0.028975	1.818128e-16	9.528280e-17	2.576773e-16	-2.656938e-16	-3.138234e-16	2.717167e-16	-2.874017e-16	-5.339821e-16	1.120752e-16	...	4.0325
V20	-0.050866	1.036959e-16	-9.309954e-16	-9.429297e-16	-3.223123e-16	2.076048e-16	1.898638e-16	1.744242e-16	-1.095534e-16	-4.340941e-16	...	-1.12
V21	0.044736	-1.755072e-16	8.444409e-17	-2.971969e-17	-9.976950e-17	-1.368701e-16	-1.575903e-16	1.938604e-16	-2.412439e-16	4.578389e-17	...	1.0000
V22	0.144059	7.477367e-17	2.500830e-16	4.648259e-16	2.099922e-16	5.060029e-16	-3.362902e-16	-1.058131e-15	5.475559e-16	2.871855e-17	...	3.9059
V23	0.051142	9.808705e-16	1.059562e-16	2.115206e-17	6.002528e-17	1.637596e-16	-7.232186e-17	2.327911e-16	3.897104e-16	5.929286e-16	...	6.1273
V24	-0.016182	7.354269e-17	-8.142354e-18	-9.351637e-17	2.229738e-16	-9.286095e-16	-1.261867e-15	-2.589727e-17	-1.802967e-16	-2.346385e-16	...	1.2982
V25	-0.233083	-9.805358e-16	-4.261894e-17	4.771164e-16	5.394585e-16	5.625102e-16	1.081933e-15	1.174169e-15	-1.390791e-16	1.099645e-15	...	-2.82
V26	-0.041407	-8.621897e-17	2.601622e-16	6.521501e-16	-6.179751e-16	9.144690e-16	-2.378414e-16	-7.334507e-16	-1.209975e-16	-1.388725e-15	...	-4.90
V27	-0.005135	3.208233e-17	-4.478472e-16	6.239832e-16	-6.403423e-17	4.465960e-16	-2.623818e-16	-5.886825e-16	1.733633e-16	-2.287414e-16	...	-1.03
V28	-0.009413	9.820892e-16	-3.676415e-16	7.726948e-16	-5.863664e-17	-3.299167e-16	4.813155e-16	-6.836764e-17	-4.484325e-16	9.146779e-16	...	5.1322
Amount	-0.010596	-2.277087e-01	-5.314089e-01	-2.108805e-01	9.873167e-02	-3.863563e-01	2.159812e-01	3.973113e-01	-1.030791e-01	-4.424560e-02	...	1.0599
Class	-0.012323	-1.013473e-01	9.128865e-02	-1.929608e-01	1.334475e-01	-9.497430e-02	-4.364316e-02	-1.872566e-01	1.987512e-02	-9.773269e-02	...	4.0413

31 rows × 31 columns

In [21]:
```
scaler = StandardScaler()
df['Amount'] = scaler.fit_transform(df[['Amount']])
df['Time'] = scaler.fit_transform(df[['Time']])
```

In [22]:
```
# Define features and target variable
X = df.drop('Class', axis=1)
y = df['Class']
```

In [23]:
```
# Handle class imbalance using SMOTE (Synthetic Minority Over-sampling Technique)
smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X, y)
```

In [24]:
```
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_resampled, y_resampled, test_size=0.2, random_state=42, stratify=y_resample
```

In [25]:
```
# Train a Logistic Regression model
model = LogisticRegression(max_iter=1000, random_state=42)
model.fit(X_train, y_train)
```

Out[25]:
```
LogisticRegression(max_iter=1000, random_state=42)
```

In [26]:
```
# Make predictions
y_pred = model.predict(X_test)
```

In [29]:
```
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```

```
Confusion Matrix:
[[55447  1416]
 [ 4447 52416]]
```

In [30]:
```
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

```
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.98      0.95     56863
           1       0.97      0.92      0.95     56863

    accuracy                           0.95    113726
   macro avg       0.95      0.95      0.95    113726
weighted avg       0.95      0.95      0.95    113726
```

In [31]:
```
precision = precision_score(y_test, y_pred)
print(f"\nPrecision: {precision:.4f}")
recall = recall_score(y_test, y_pred)
print(f"Recall: {recall:.4f}")
f1 = f1_score(y_test, y_pred)
print(f"F1-Score: {f1:.4f}")
```

```
Precision: 0.9737
Recall: 0.9218
F1-Score: 0.9470
```

In [ ]: