In [1]:
```
!pip install matplotlib
!pip install seaborn
!pip install plotly-express
```

```
Requirement already satisfied: matplotlib in c:\users\comic\anaconda3\lib\site-packages (3.7.1)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\comic\anaconda3\lib\site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\comic\anaconda3\lib\site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\comic\anaconda3\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\comic\anaconda3\lib\site-packages (from matplotlib) (4.39.3)
Requirement already satisfied: cycler>=0.10 in c:\users\comic\anaconda3\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: packaging>=20.0 in c:\users\comic\anaconda3\lib\site-packages (from matplotlib) (23.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\comic\anaconda3\lib\site-packages (from matplotlib) (1.0.7)
Requirement already satisfied: pillow>=6.2.0 in c:\users\comic\anaconda3\lib\site-packages (from matplotlib) (9.4.0)
Requirement already satisfied: numpy>=1.20 in c:\users\comic\anaconda3\lib\site-packages (from matplotlib) (1.24.2)
Requirement already satisfied: six>=1.5 in c:\users\comic\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Requirement already satisfied: seaborn in c:\users\comic\anaconda3\lib\site-packages (0.12.2)
Requirement already satisfied: numpy!=1.24.0,>=1.17 in c:\users\comic\anaconda3\lib\site-packages (from seaborn) (1.24.2)
Requirement already satisfied: pandas>=0.25 in c:\users\comic\anaconda3\lib\site-packages (from seaborn) (2.0.0)
Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in c:\users\comic\anaconda3\lib\site-packages (from seaborn) (3.7.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\comic\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (4.3
9.3)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\comic\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn)
(2.8.2)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\comic\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (3.0.
9)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\comic\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (1.0.
7)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\comic\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (1.4.
4)
Requirement already satisfied: cycler>=0.10 in c:\users\comic\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (0.11.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\comic\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (9.4.0)
Requirement already satisfied: packaging>=20.0 in c:\users\comic\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (23.0)
Requirement already satisfied: tzdata>=2022.1 in c:\users\comic\anaconda3\lib\site-packages (from pandas>=0.25->seaborn) (2023.3)
Requirement already satisfied: pytz>=2020.1 in c:\users\comic\anaconda3\lib\site-packages (from pandas>=0.25->seaborn) (2022.7)
Requirement already satisfied: six>=1.5 in c:\users\comic\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.1->s
eaborn) (1.16.0)
Requirement already satisfied: plotly-express in c:\users\comic\anaconda3\lib\site-packages (0.4.1)
Requirement already satisfied: plotly>=4.1.0 in c:\users\comic\anaconda3\lib\site-packages (from plotly-express) (5.14.1)
Requirement already satisfied: statsmodels>=0.9.0 in c:\users\comic\anaconda3\lib\site-packages (from plotly-express) (0.13.5)
Requirement already satisfied: numpy>=1.11 in c:\users\comic\anaconda3\lib\site-packages (from plotly-express) (1.24.2)
Requirement already satisfied: patsy>=0.5 in c:\users\comic\anaconda3\lib\site-packages (from plotly-express) (0.5.3)
Requirement already satisfied: pandas>=0.20.0 in c:\users\comic\anaconda3\lib\site-packages (from plotly-express) (2.0.0)
Requirement already satisfied: scipy>=0.18 in c:\users\comic\anaconda3\lib\site-packages (from plotly-express) (1.10.1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\comic\anaconda3\lib\site-packages (from pandas>=0.20.0->plotly-express) (2023.3)
Requirement already satisfied: pytz>=2020.1 in c:\users\comic\anaconda3\lib\site-packages (from pandas>=0.20.0->plotly-express) (2022.7)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\comic\anaconda3\lib\site-packages (from pandas>=0.20.0->plotly-express)
(2.8.2)
Requirement already satisfied: six in c:\users\comic\anaconda3\lib\site-packages (from patsy>=0.5->plotly-express) (1.16.0)
Requirement already satisfied: packaging in c:\users\comic\anaconda3\lib\site-packages (from plotly>=4.1.0->plotly-express) (23.0)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\comic\anaconda3\lib\site-packages (from plotly>=4.1.0->plotly-express) (8.2.2)
```

In [1]:
```
# Data exploration dependencies
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

In [2]:
```
# Importing credit card dataset
df = pd.read_csv(r"C:\Users\comic\OneDrive\Desktop\creditcard.csv")
```

In [3]:
```
# How is the data organized?
df.head(3)
```

Out[3]:

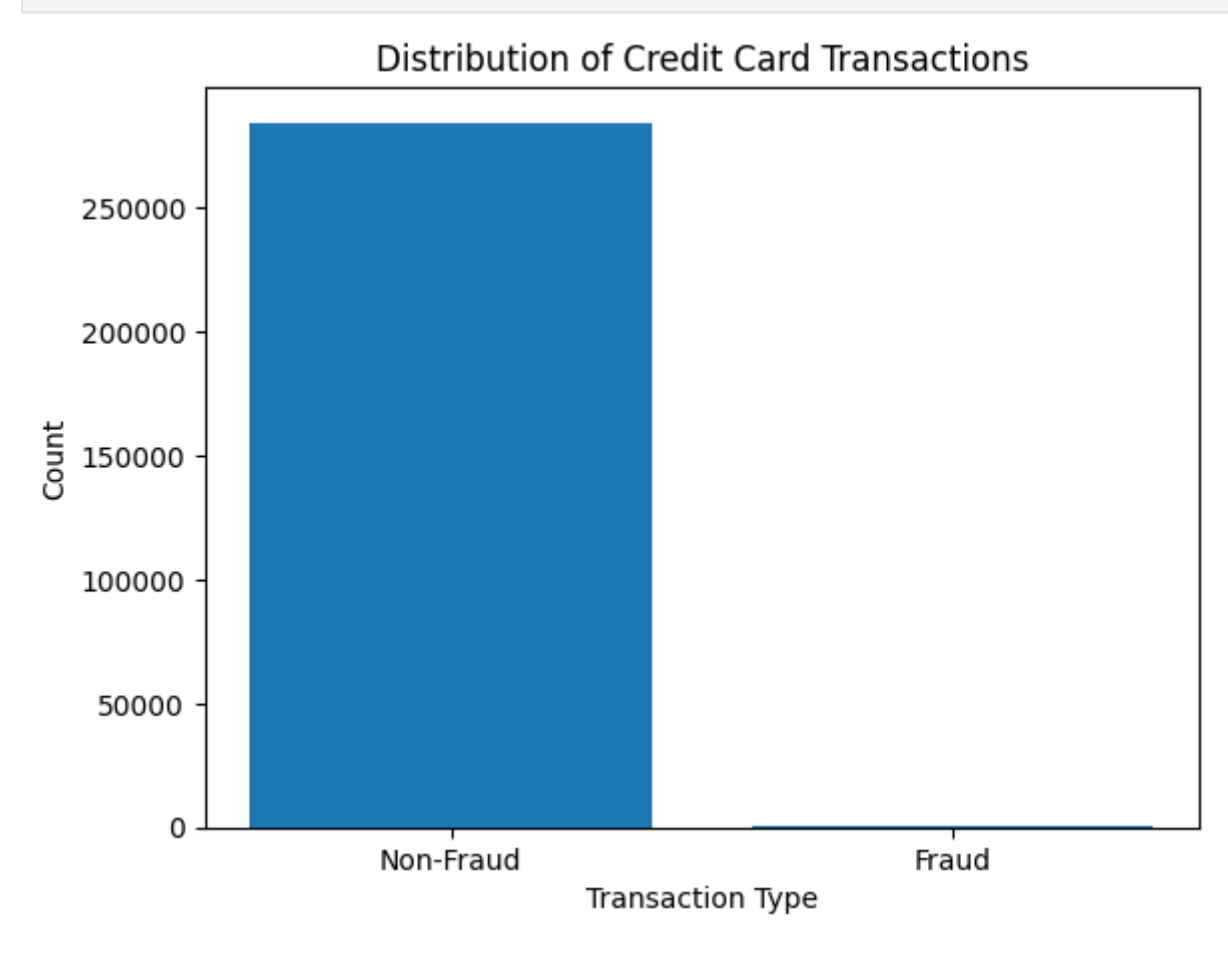
	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.128539	-0.189115
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.167170	0.125895
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	-0.327642	-0.139097

3 rows × 31 columns

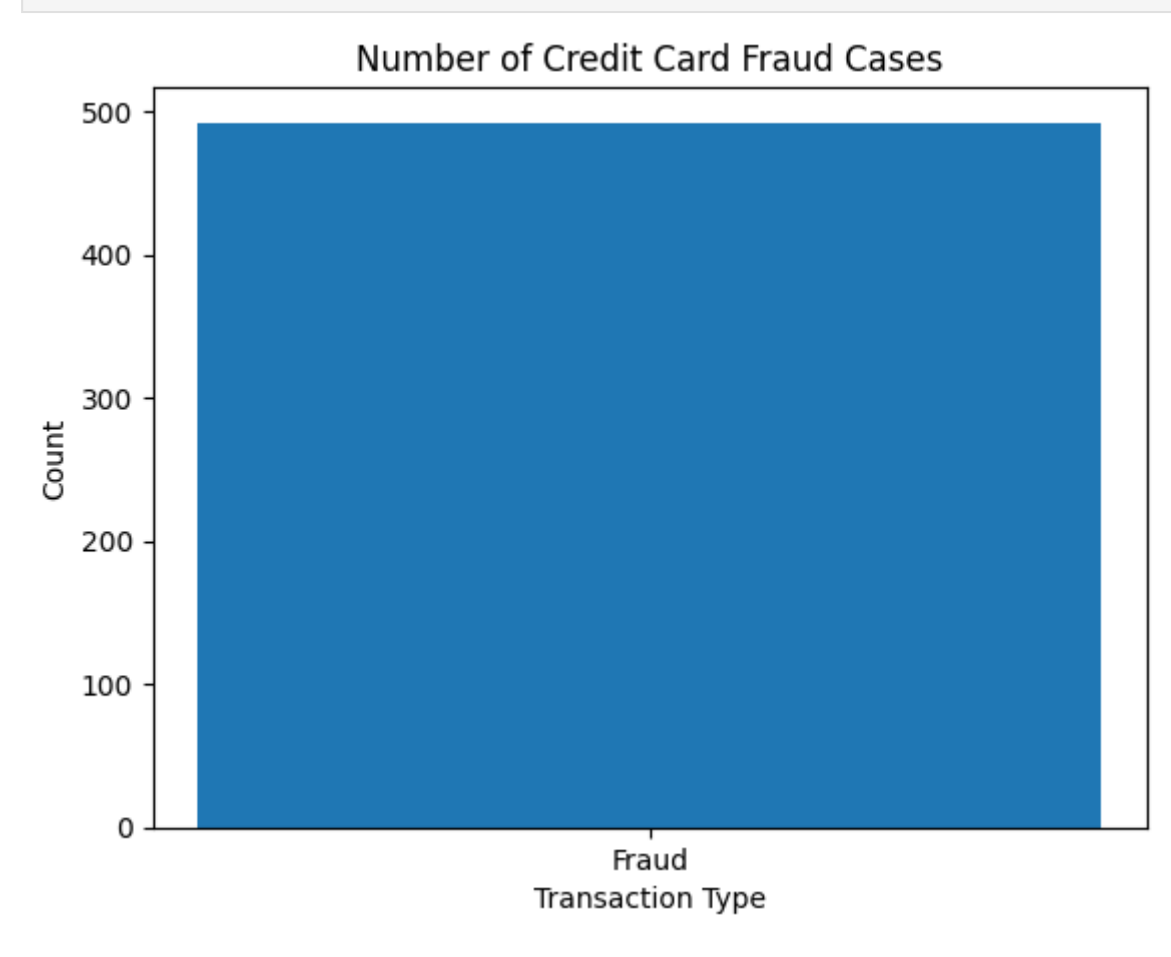
In [4]:
```
# What types of data do we have?
df.info()
```

```
<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
```

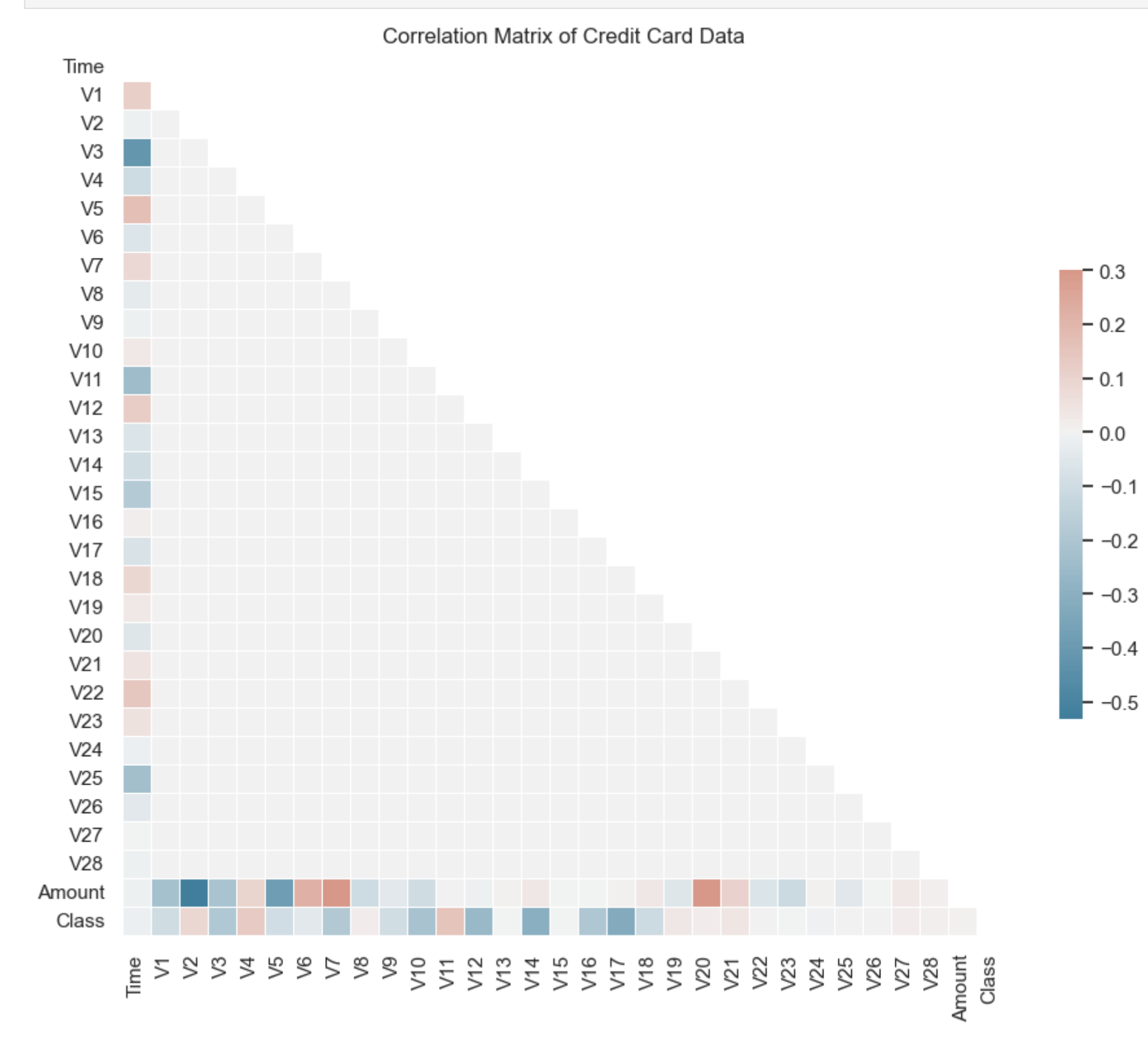
In [5]:
```
#As the data is Clean we can directly get into the Visualization
# Count the number of non-fraud and fraud transactions
counts = df['Class'].value_counts()

# Create a bar plot to visualize the distribution
plt.bar(['Non-Fraud', 'Fraud'], counts)
plt.title('Distribution of Credit Card Transactions')
plt.xlabel('Transaction Type')
plt.ylabel('Count')
plt.show()
```



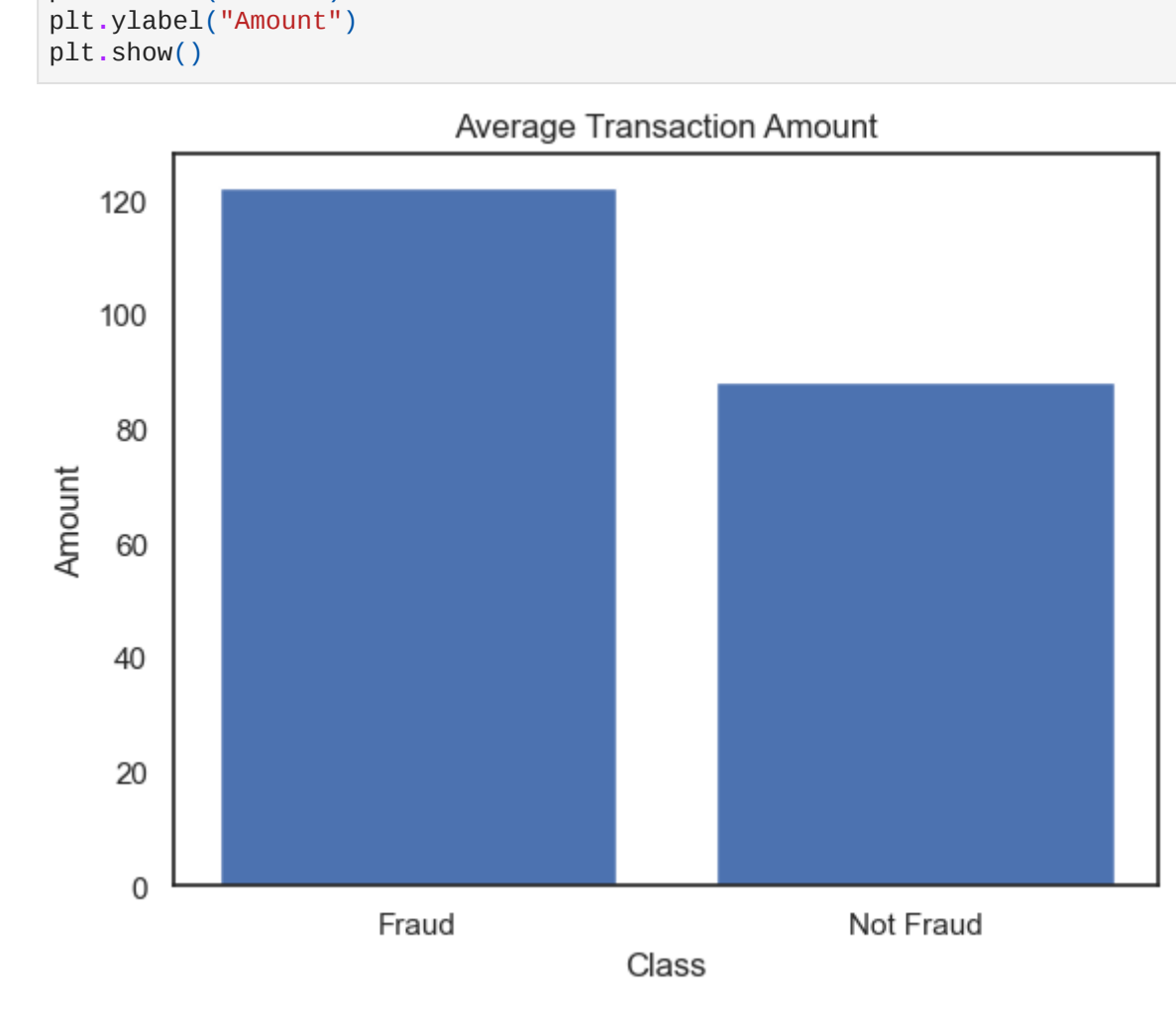
In [6]:
```
#As we cannot clearly see how many number of fraud cases are there lets Visualize it seperately
# Filter the DataFrame to include only fraud cases
fraud_cases = df[df['Class'] == 1]

# Count the number of fraud cases
fraud_count = len(fraud_cases)

# Create a bar plot to visualize the count of fraud cases
plt.bar(['Fraud'], fraud_count)
plt.title('Number of Credit Card Fraud Cases')
plt.xlabel('Transaction Type')
plt.ylabel('Count')
plt.show()
```



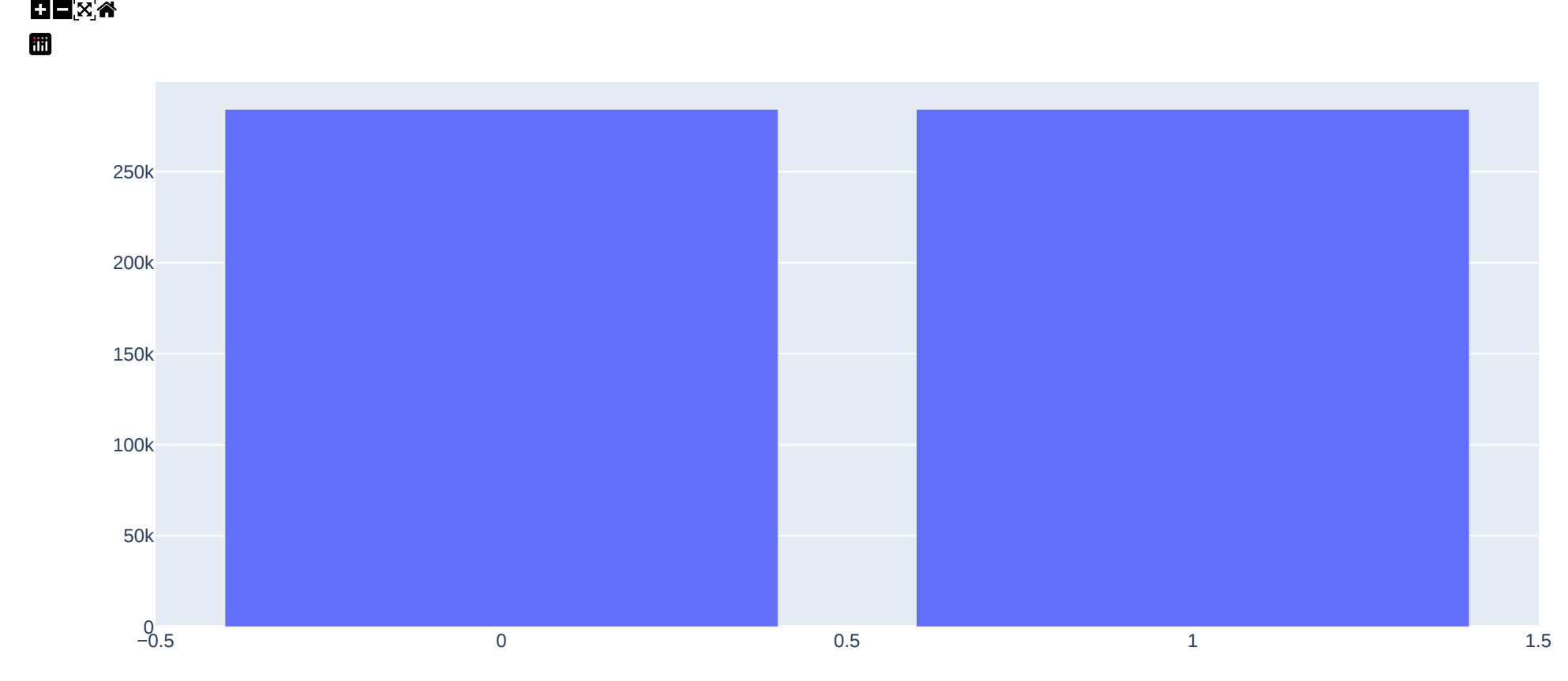
In [7]:
```
#Fraud cases make up approximately 0.2% of the total cases which implies the dataset is Highly Imbalanced
#We tried to find correlations between the different features and the fraud class using a correlation matrix, but we did not find any signif
# Calculate the correlation matrix
corr = df.corr()

# Calculate the correlation matrix
corr = df.corr()

# Create a heatmap plot of the correlation matrix using seaborn
sns.set(style="white")
mask = np.triu(np.ones_like(corr, dtype=bool))
fig, ax = plt.subplots(figsize=(11, 9))
cmap = sns.diverging_palette(230, 20, as_cmap=True)
sns.heatmap(corr, mask=mask, cmap=cmap, vmax=.3, center=0,
            square=True, linewidths=.5, cbar_kws={"shrink": .5})

plt.title('Correlation Matrix of Credit Card Data')
plt.show()
```



In [25]:
```
#Fraud transactions have an average amount which is 50% higher than non fraud transactions!
import pandas as pd
import matplotlib.pyplot as plt


# Group transactions by class and calculate the mean amount
fraud_mean = df[df["Class"] == 1]["Amount"].mean()
not_fraud_mean = df[df["Class"] == 0]["Amount"].mean()

# Plot the results
plt.bar(["Fraud", "Not Fraud"], [fraud_mean, not_fraud_mean])
plt.title("Average Transaction Amount")
plt.xlabel("Class")
plt.ylabel("Amount")
plt.show()
```



In [9]:
```
#SMOTE
#We have to do SMOTE only on the trainig set, never on the test set.
# Library that will handle SMOTE
from imblearn.over_sampling import SMOTE
# Selecting features and target
X_orig = df.drop(['Class', 'Time'], axis=1)
y_orig = df['Class']
# Transforming dataset
oversample = SMOTE()
X, y  = oversample.fit_resample(X_orig, y_orig)
# Verifying transformation
px.bar(y.value_counts(), title='Fraudulent Charge Distrobution post-SMOTE')
```



In [10]:
```
#We will use the LazyPredict library to compare multiple ML models from sklearn and find the best one for our task.
# Installing the library
!pip install lazypredict
```

```
Requirement already satisfied: lazypredict in c:\users\comic\anaconda3\lib\site-packages (0.2.12)
Requirement already satisfied: tqdm in c:\users\comic\anaconda3\lib\site-packages (from lazypredict) (4.65.0)
Requirement already satisfied: pandas in c:\users\comic\anaconda3\lib\site-packages (from lazypredict) (2.0.0)
Requirement already satisfied: lightgbm in c:\users\comic\anaconda3\lib\site-packages (from lazypredict) (3.3.5)
Requirement already satisfied: xgboost in c:\users\comic\anaconda3\lib\site-packages (from lazypredict) (1.7.5)
Requirement already satisfied: scikit-learn in c:\users\comic\anaconda3\lib\site-packages (from lazypredict) (1.2.2)
Requirement already satisfied: click in c:\users\comic\anaconda3\lib\site-packages (from lazypredict) (8.0.4)
Requirement already satisfied: joblib in c:\users\comic\anaconda3\lib\site-packages (from lazypredict) (1.2.0)
Requirement already satisfied: colorama in c:\users\comic\anaconda3\lib\site-packages (from click->lazypredict) (0.4.6)
Requirement already satisfied: scipy in c:\users\comic\anaconda3\lib\site-packages (from lightgbm->lazypredict) (1.10.1)
Requirement already satisfied: wheel in c:\users\comic\anaconda3\lib\site-packages (from lightgbm->lazypredict) (0.38.4)
Requirement already satisfied: numpy in c:\users\comic\anaconda3\lib\site-packages (from lightgbm->lazypredict) (1.24.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\comic\anaconda3\lib\site-packages (from scikit-learn->lazypredict) (3.1.0)
Requirement already satisfied: pytz>=2020.1 in c:\users\comic\anaconda3\lib\site-packages (from pandas->lazypredict) (2022.7)
Requirement already satisfied: tzdata>=2022.1 in c:\users\comic\anaconda3\lib\site-packages (from pandas->lazypredict) (2023.3)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\comic\anaconda3\lib\site-packages (from pandas->lazypredict) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\comic\anaconda3\lib\site-packages (from python-dateutil>=2.8.2->pandas->lazypredict) (1.
16.0)
```

In [11]:
```
# Importing machine learning model dependencies
from lazypredict.Supervised import LazyClassifier
from sklearn.model_selection import train_test_split
```

In [12]:
```
      # Importing machine learning model dependencies
from lazypredict.Supervised import LazyClassifier
from sklearn.model_selection import train_test_split
```

In [13]:
```
# Splitting data into testing and training partitions
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size = 0.95)
```

In [14]:
```
# Creating LazyPredict-specific train-test splits
Lazy_X_train, Lazy_X_test, Lazy_y_train, Lazy_y_test = train_test_split(X_test, y_test)
```

In [15]:
```
# Creating model(s) and fitting them to partitioned training data
clf = LazyClassifier(verbose=0, ignore_warnings=True, custom_metric=None)
models, predictions = clf.fit(Lazy_X_train, Lazy_X_test, Lazy_y_train, Lazy_y_test)
```

```
100%|██████████████████████████████████████████████████████████████████████████████| 29/29 [02:51<00:00,  5.90s/it]
```

In [16]:
```
# Getting the models that performed the best in the partition set
models.sort_values(by='Accuracy', ascending=False)
```

Out[16]:

Model	Accuracy	Balanced Accuracy	ROC AUC	F1 Score	Time Taken
LGBMClassifier	1.00	1.00	1.00	1.00	0.36
XGBClassifier	1.00	1.00	1.00	1.00	2.16
ExtraTreesClassifier	1.00	1.00	1.00	1.00	2.22
LabelSpreading	0.99	0.99	0.99	0.99	46.20
LabelPropagation	0.99	0.99	0.99	0.99	32.83
RandomForestClassifier	0.99	0.99	0.99	0.99	14.53
BaggingClassifier	0.99	0.99	0.99	0.99	8.95
KNeighborsClassifier	0.99	0.99	0.99	0.99	0.60
ExtraTreeClassifier	0.98	0.98	0.98	0.98	0.09
DecisionTreeClassifier	0.98	0.98	0.98	0.98	1.28
SVC	0.97	0.97	0.97	0.97	5.52
AdaBoostClassifier	0.97	0.97	0.97	0.97	7.55
LogisticRegression	0.96	0.96	0.96	0.96	0.25
SGDClassifier	0.96	0.96	0.96	0.96	0.13
LinearSVC	0.95	0.95	0.95	0.95	1.35
CalibratedClassifierCV	0.95	0.95	0.95	0.95	5.81
PassiveAggressiveClassifier	0.95	0.95	0.95	0.95	0.14
QuadraticDiscriminantAnalysis	0.94	0.94	0.94	0.94	0.11
Perceptron	0.94	0.94	0.94	0.94	0.08
RidgeClassifierCV	0.92	0.92	0.92	0.92	0.13
LinearDiscriminantAnalysis	0.92	0.92	0.92	0.92	0.21
RidgeClassifier	0.92	0.92	0.92	0.92	0.17
GaussianNB	0.92	0.92	0.92	0.92	0.08
BernoulliNB	0.91	0.91	0.91	0.91	0.09
NuSVC	0.91	0.91	0.91	0.91	39.99
NearestCentroid	0.88	0.88	0.88	0.87	0.06
DummyClassifier	0.50	0.50	0.50	0.33	0.06

In [26]:
```
!pip install scikit-plot
!pip install xgboost
```

```
Collecting scikit-plot
  Downloading scikit_plot-0.3.7-py3-none-any.whl (33 kB)
Requirement already satisfied: scipy>=0.9 in c:\users\comic\anaconda3\lib\site-packages (from scikit-plot) (1.10.1)
Requirement already satisfied: joblib>=0.10 in c:\users\comic\anaconda3\lib\site-packages (from scikit-plot) (1.2.0)
Requirement already satisfied: scikit-learn>=0.18 in c:\users\comic\anaconda3\lib\site-packages (from scikit-plot) (1.2.2)
Requirement already satisfied: matplotlib>=1.4.0 in c:\users\comic\anaconda3\lib\site-packages (from scikit-plot) (3.7.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\comic\anaconda3\lib\site-packages (from matplotlib>=1.4.0->scikit-plot) (2.
8.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\comic\anaconda3\lib\site-packages (from matplotlib>=1.4.0->scikit-plot) (1.0.7)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\comic\anaconda3\lib\site-packages (from matplotlib>=1.4.0->scikit-plot) (3.0.9)
Requirement already satisfied: numpy>=1.20 in c:\users\comic\anaconda3\lib\site-packages (from matplotlib>=1.4.0->scikit-plot) (1.24.2)
Requirement already satisfied: packaging>=20.0 in c:\users\comic\anaconda3\lib\site-packages (from matplotlib>=1.4.0->scikit-plot) (23.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\comic\anaconda3\lib\site-packages (from matplotlib>=1.4.0->scikit-plot) (9.4.0)
Requirement already satisfied: cycler>=0.10 in c:\users\comic\anaconda3\lib\site-packages (from matplotlib>=1.4.0->scikit-plot) (0.11.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\comic\anaconda3\lib\site-packages (from matplotlib>=1.4.0->scikit-plot) (1.4.4)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\comic\anaconda3\lib\site-packages (from matplotlib>=1.4.0->scikit-plot) (4.39.
3)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\comic\anaconda3\lib\site-packages (from scikit-learn>=0.18->scikit-plot) (3.
1.0)
Requirement already satisfied: six>=1.5 in c:\users\comic\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib>=1.4.0->scikit-
plot) (1.16.0)
Installing collected packages: scikit-plot
Successfully installed scikit-plot-0.3.7
Requirement already satisfied: xgboost in c:\users\comic\anaconda3\lib\site-packages (1.7.5)
Requirement already satisfied: scipy in c:\users\comic\anaconda3\lib\site-packages (from xgboost) (1.10.1)
Requirement already satisfied: numpy in c:\users\comic\anaconda3\lib\site-packages (from xgboost) (1.24.2)
```

In [29]:
```
!pip install scikit-learn
```

```
Requirement already satisfied: scikit-learn in c:\users\comic\anaconda3\lib\site-packages (1.2.2)
Requirement already satisfied: joblib>=1.1.1 in c:\users\comic\anaconda3\lib\site-packages (from scikit-learn) (1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\comic\anaconda3\lib\site-packages (from scikit-learn) (3.1.0)
Requirement already satisfied: numpy>=1.17.3 in c:\users\comic\anaconda3\lib\site-packages (from scikit-learn) (1.24.2)
Requirement already satisfied: scipy>=1.3.2 in c:\users\comic\anaconda3\lib\site-packages (from scikit-learn) (1.10.1)
```

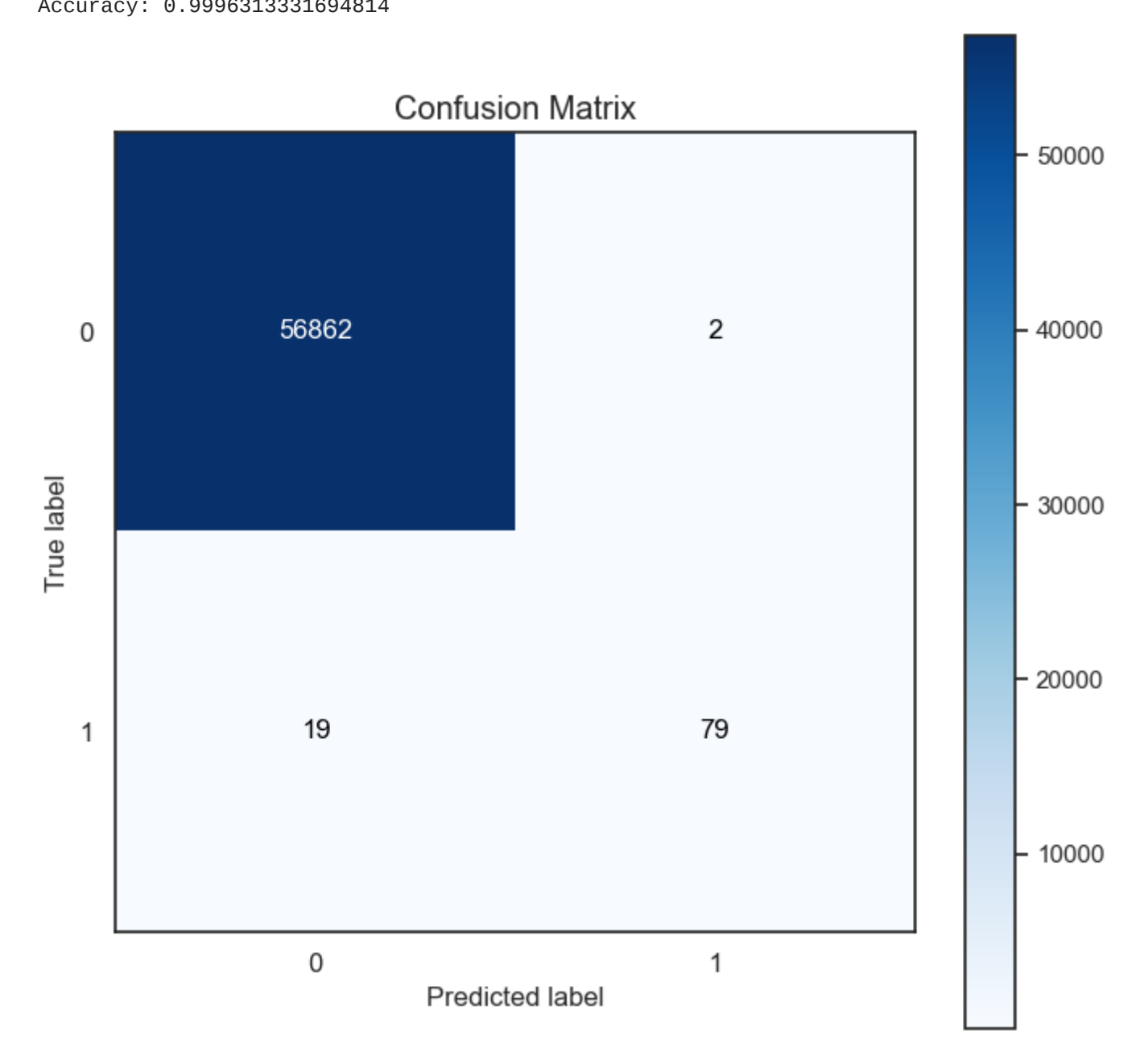
In [31]:
```
!pip install --upgrade scikit-learn
```

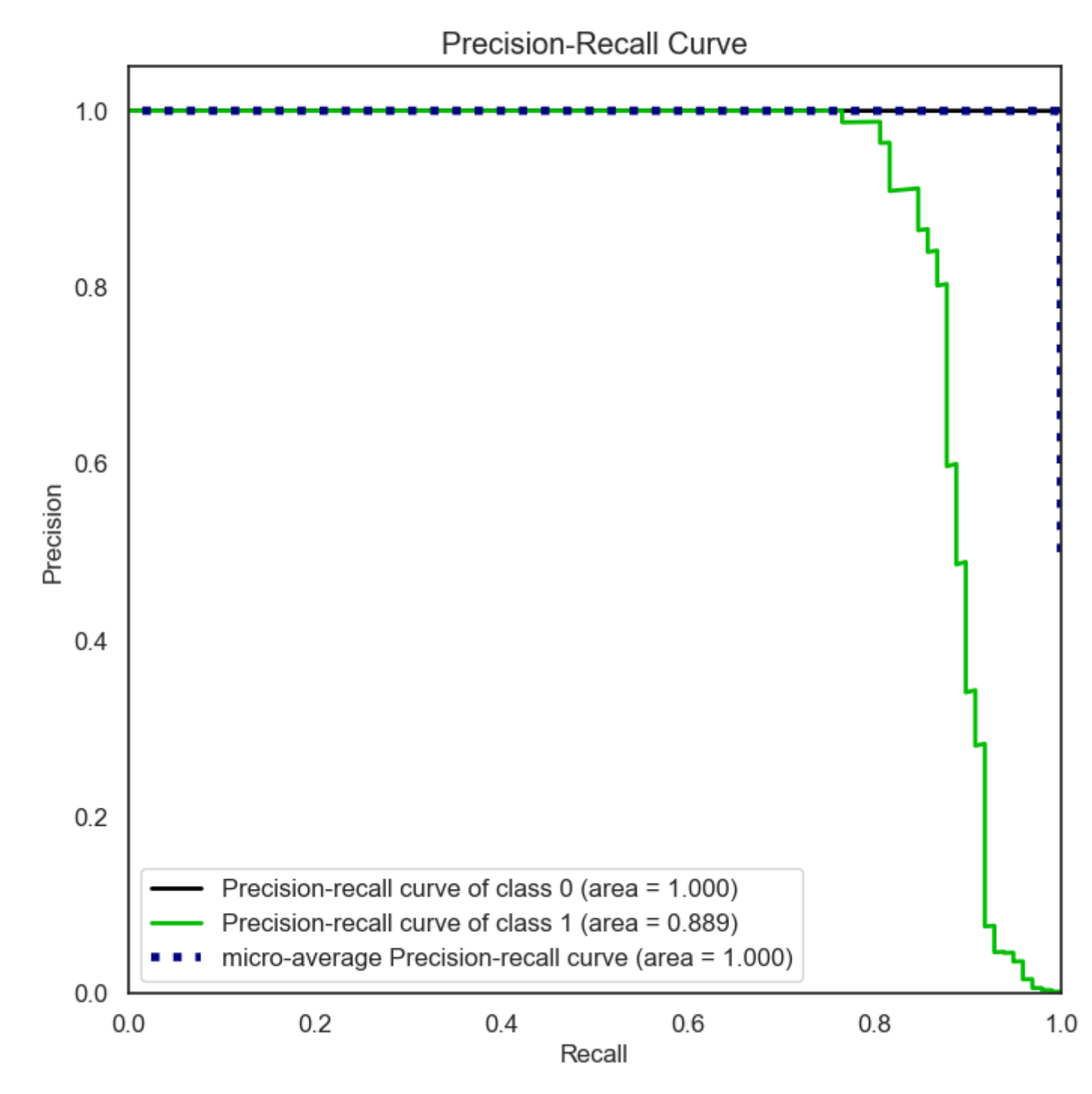
```
Requirement already satisfied: scikit-learn in c:\users\comic\anaconda3\lib\site-packages (1.2.2)
Requirement already satisfied: scipy>=1.3.2 in c:\users\comic\anaconda3\lib\site-packages (from scikit-learn) (1.10.1)
Requirement already satisfied: joblib>=1.1.1 in c:\users\comic\anaconda3\lib\site-packages (from scikit-learn) (1.2.0)
Requirement already satisfied: numpy>=1.17.3 in c:\users\comic\anaconda3\lib\site-packages (from scikit-learn) (1.24.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\comic\anaconda3\lib\site-packages (from scikit-learn) (3.1.0)
```

In [ ]:

In [22]:
```
#The machine learning model got extremely good results where in we got a Precision-Recall curve of 0.889 without any hyperparameter tuning!!
#The Model was able to predict 79 out of the 98 fraudulent cases accurately!
import pandas as pd
from sklearn.model_selection import train_test_split, cross_val_score
from xgboost import XGBClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
import scikitplot as skplt
import matplotlib.pyplot as plt

# Load data and define X and y
df = pd.read_csv(r"C:\Users\comic\OneDrive\Desktop\creditcard.csv")
X = df.drop(["Class", "Time"], axis=1)
y = df["Class"]

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create and train the XGBoost model
model = XGBClassifier()
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Calculate the accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

# Generate and plot confusion matrix
cm = confusion_matrix(y_test, y_pred)
skplt.metrics.plot_confusion_matrix(y_test, y_pred, figsize=(8, 8))
plt.show()

# Generate and plot precision-recall curve
skplt.metrics.plot_precision_recall(y_test, model.predict_proba(X_test), figsize=(8, 8))
plt.show()
```

```
Accuracy: 0.9996313331694814
```





In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: