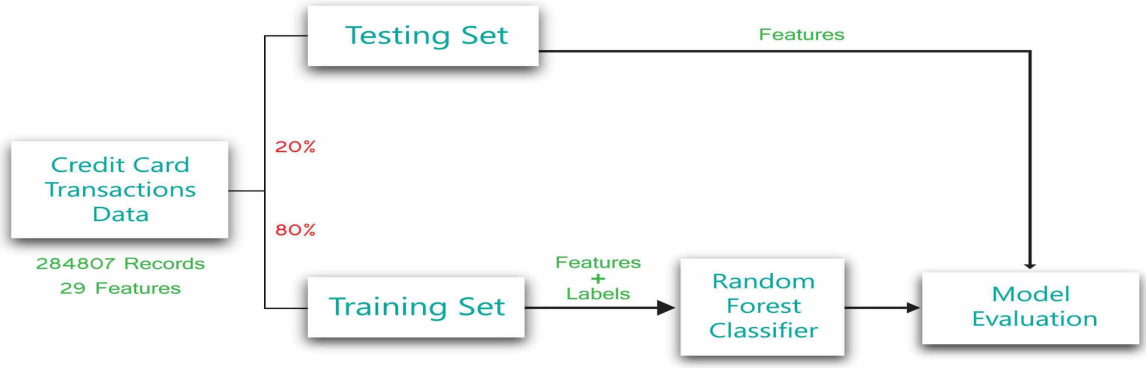
PDS Project Report

Date of Submission: 28/04/2022

Credit Card Fraud Detection



TEAM MEMBERS

NAME	ROLL NUMBER	EMAIL
SIRIKONDA KRUSHANG	S20190010164	krushang.s19@iiits.in
SRI LAKSHMI PRASANNA KONERU	S20190010168	srilakshmiprasanna.k19@iiits.in
KRISHNA NIVEDA	S20190020224	niveda.k19@iiits.in
ANUHYA SARABU	S20190020251	anuhya.s19@iiits.in

INTRODUCTION

Credit card fraud occurs when a scammer uses your credit card number and PIN, or a stolen credit card, to conduct financial transactions from your account without your permission. When an individual uses someone else's credit card information illegally or for uninformed personal spending, it is classified as credit card fraud. Credit card scams fall under identity theft and have become increasingly common nowadays. Scammers steal your card information and use it to perform unethical transactions on your account without your knowledge. Credit card security has become necessary as a result of this. With the advancement of modern technology and the creation of global communication superhighways, credit card fraud is already on the upswing.

Credit card fraud costs consumers and financial institutions billions of dollars each year, and criminals are always looking for new ways to commit cybercrimes.

Therefore , It is important that credit card companies are able to recognize fraudulent credit card transactions so that customers are not charged for items that they did not purchase, and fraud detection systems help with this as they help banks and financial institutions to reduce the damage caused by these cybercrimes.

DATA(Just a Glance)

This dataset contains the real bank transactions made by European cardholders in the year 2013. As a security concern, the actual variables are not being shared but — they have been transformed versions of PCA. As a result, we can find 29 feature columns and 1 final class column. Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. We have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

	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	0.128539
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
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
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	0.647376
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	-0.206010

5 rows × 31 columns

V22	V23	V24	V25	V26	V27	V28	Amount	Class
0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.62	0
-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.69	0
0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0
0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458	123.50	0
0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.99	0

IMPORTING LIBRARIES

For this credit card data, the features that we have in the dataset are the transformed version of PCA, so we will not need to perform the feature selection again.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.svm import SVC

from sklearn.svm import LinearSVC
from sklearn.metrics import plot_confusion_matrix
from sklearn.metrics import precision_recall_fscore_support as score
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report,roc_auc_score,roc_curve
import warnings
import datetime as dt
from datetime import datetime
warnings.filterwarnings("ignore")
```

IMPLEMENTATION OF PROJECT (Results)

Data Pre-processing and understanding :

The dataset is imbalanced towards a feature. Which seems pretty valid for such kind of data. Because today many banks have adopted different security mechanisms, so it is harder for hackers to make such moves. Still, sometimes when there is some vulnerability in the system the chance of such activities can increase.That's why we can see the majority of transactions belonging to our datasets are normal and only a few percentages of transactions are fraudulent.

```
df[df['Class'] == 1]['Class'].count()
```

```
492
```

The total number of transactions are 284807, out of which 284315 are normal transactions and 492 are fraudulent transactions.

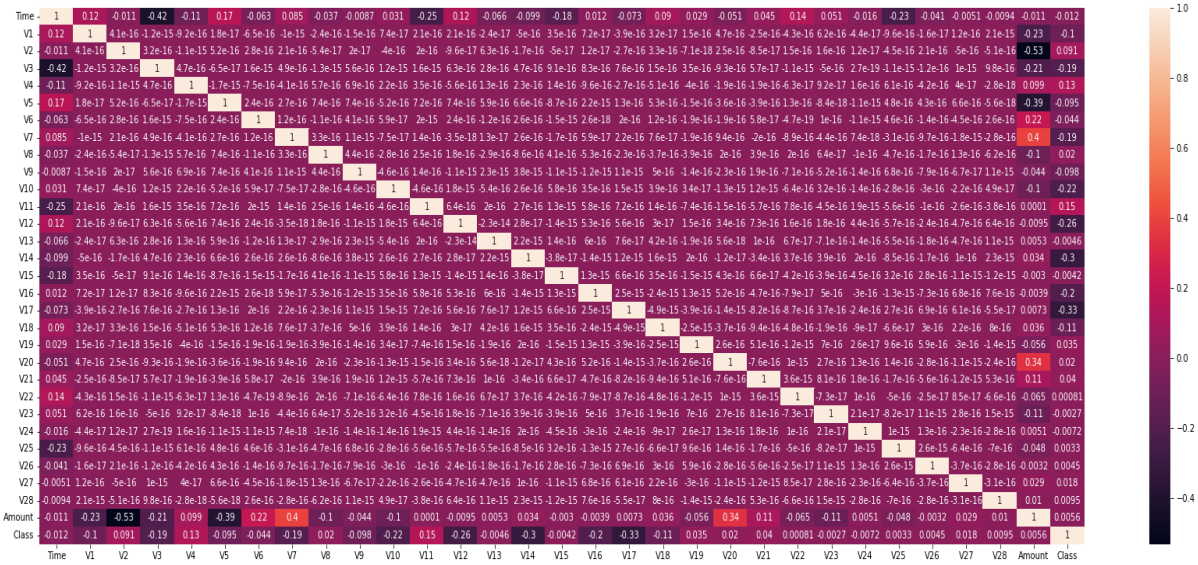
Checking for the null values:

```
 #   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
```

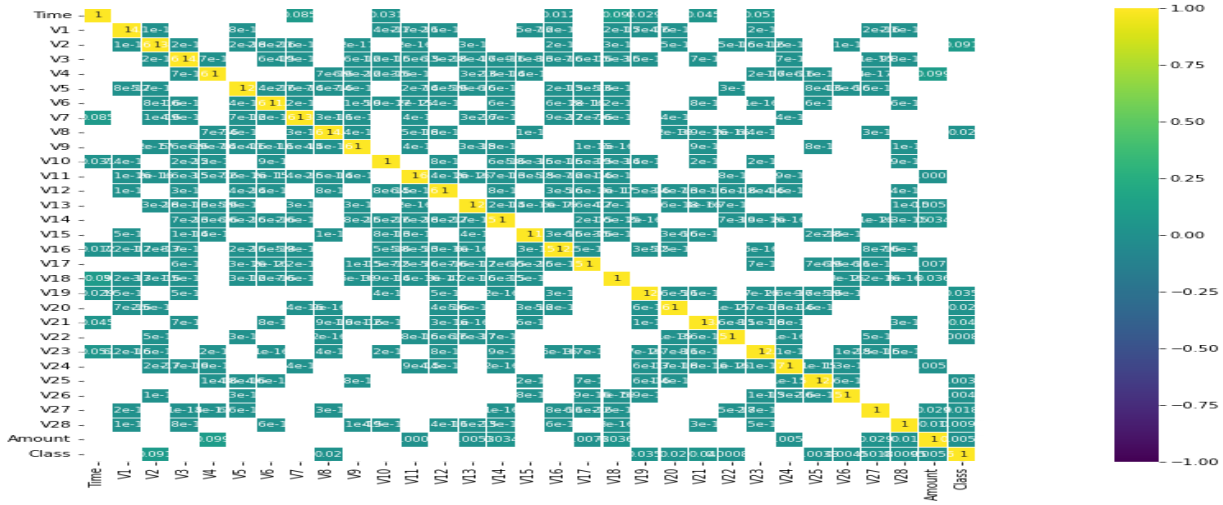
As per the count per column, we have no null values. Also, feature selection is not the case for this use case. Anyway, you can try applying feature selection mechanisms to check if the results are optimized.

we have observed in our data 28 features are transformed versions of PCA but the Amount is the original one. And, while checking the minimum and maximum is in the amount, we found the difference is huge and can deviate our result.

SNS Heatmap :



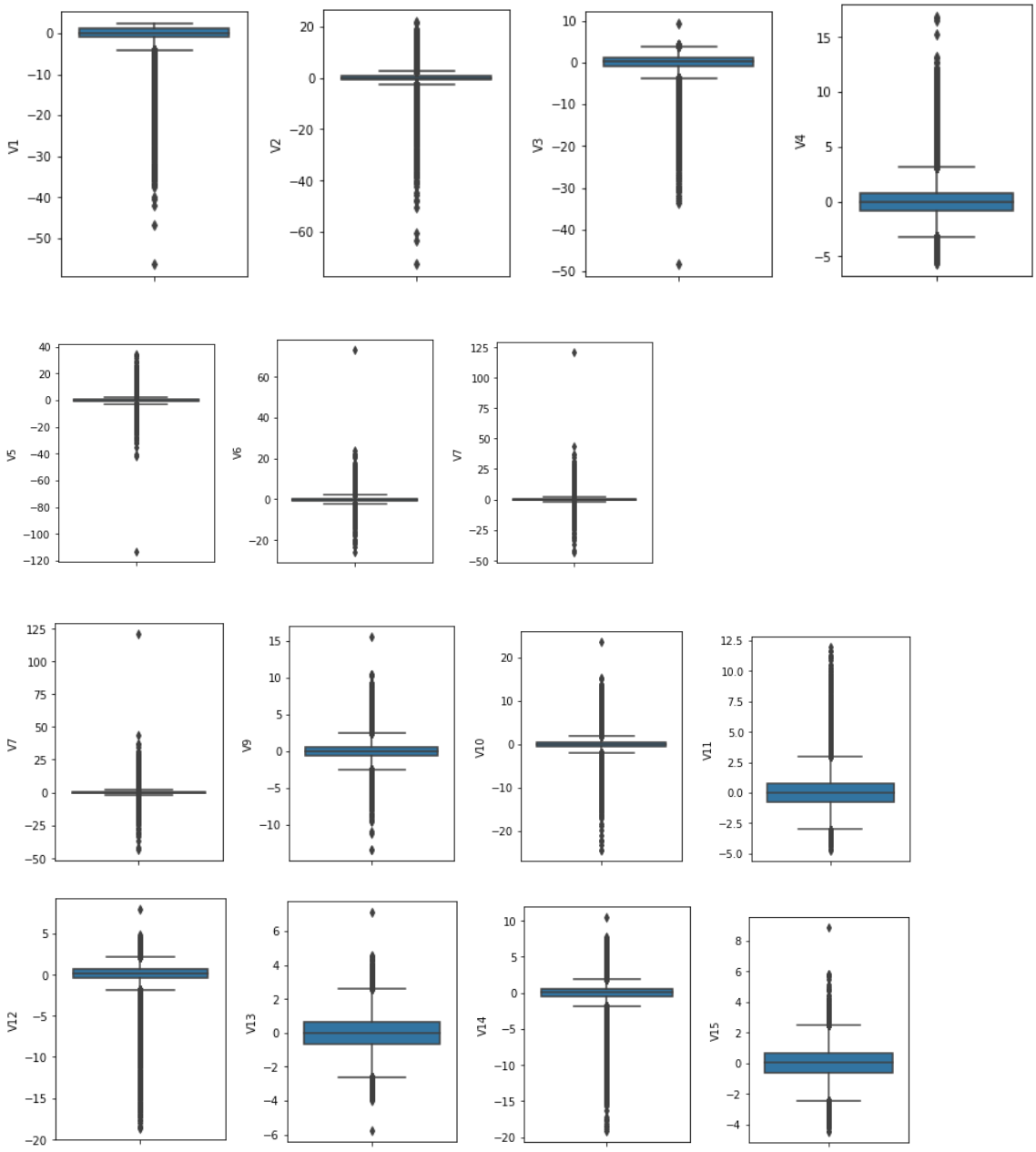
In the HeatMap we can clearly see that most of the features do not correlate to other features but there are some features that either have a positive or a negative correlation with each other. For example, V2 and V5 are highly negatively correlated with the feature Amount. We also see some correlation with V20 and Amount. This gives us a deeper understanding of the Data available to us.

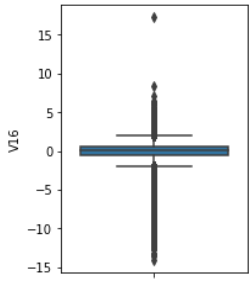


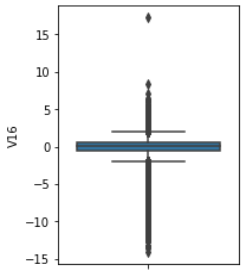
EDA:

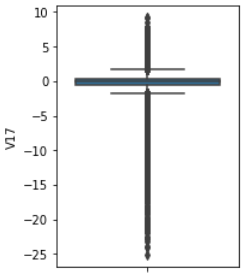
SNS BOX PLOTS:

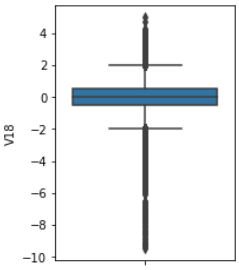
A box plot is a method for graphically depicting groups of numerical data through their quartiles.

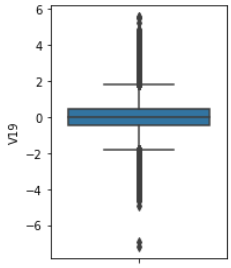


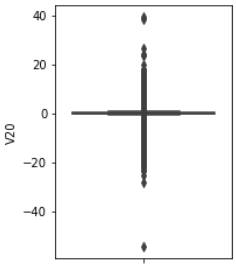

V16

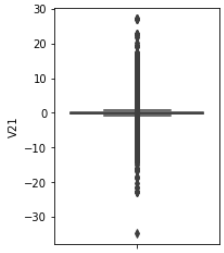

V16

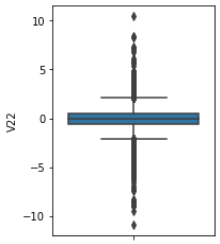

V17

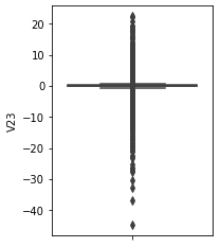

V18

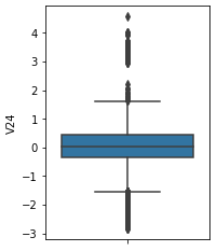

V19

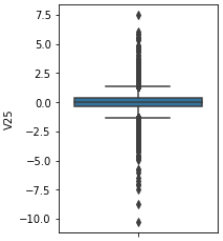

V20


V21

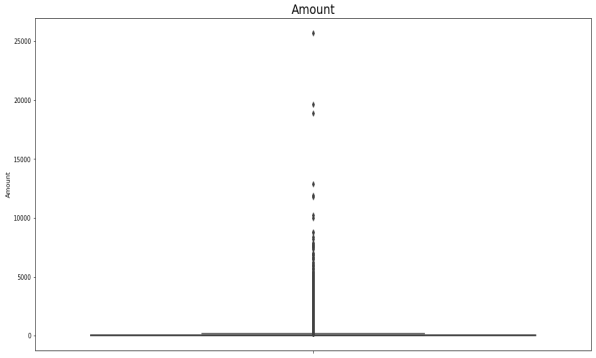

V22


V23

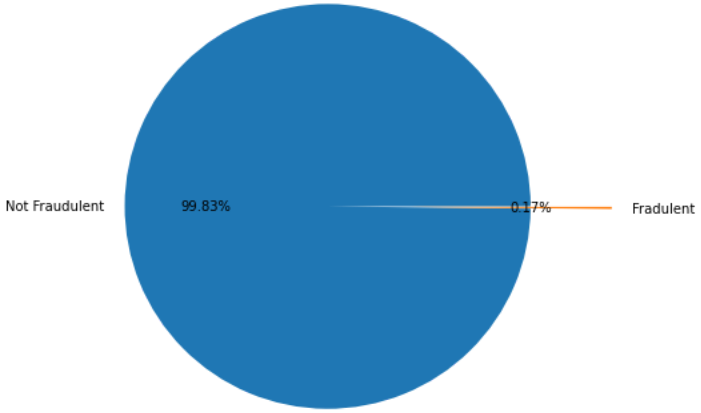

V24


V25

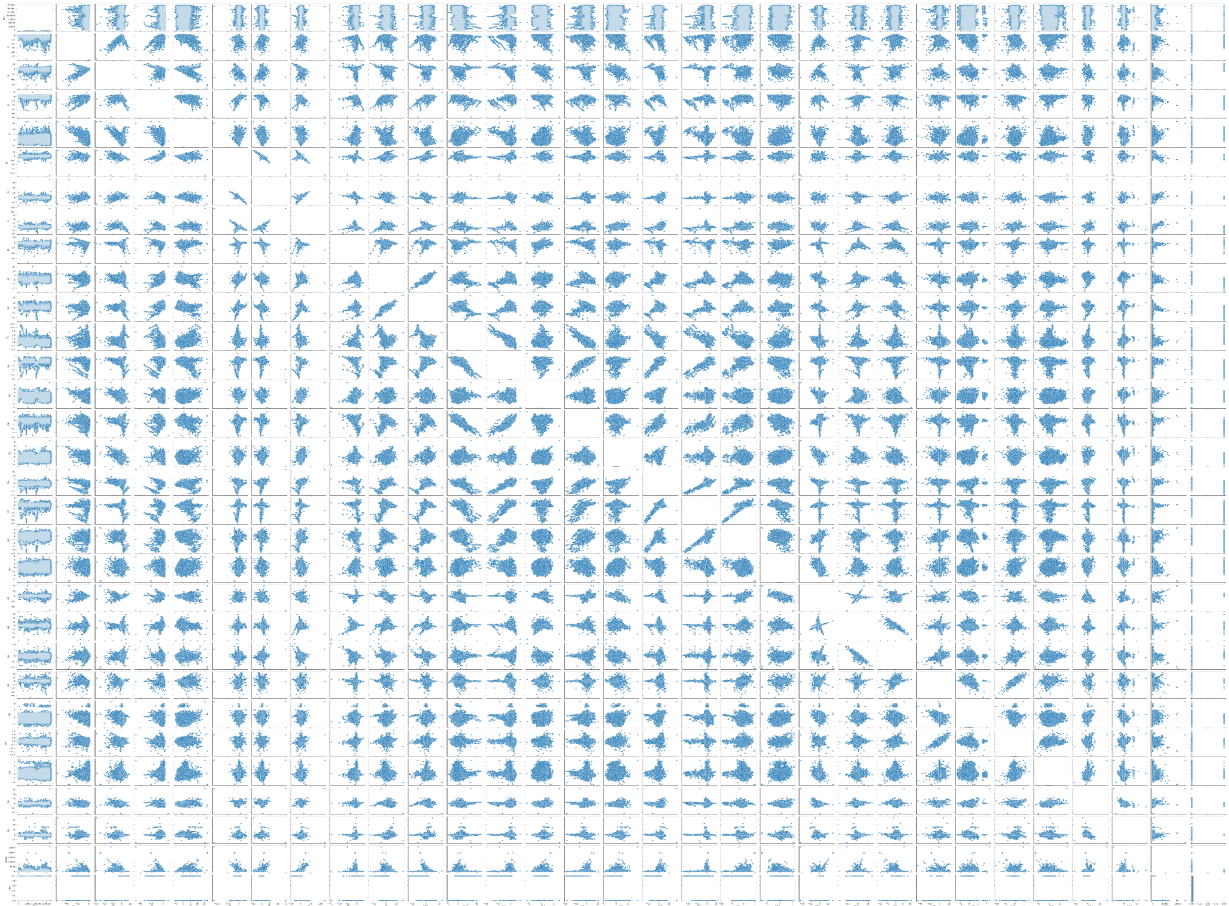
:


Amount
Amount

NON FRAUDULENT VS FRAUDULENT PIE CHART :

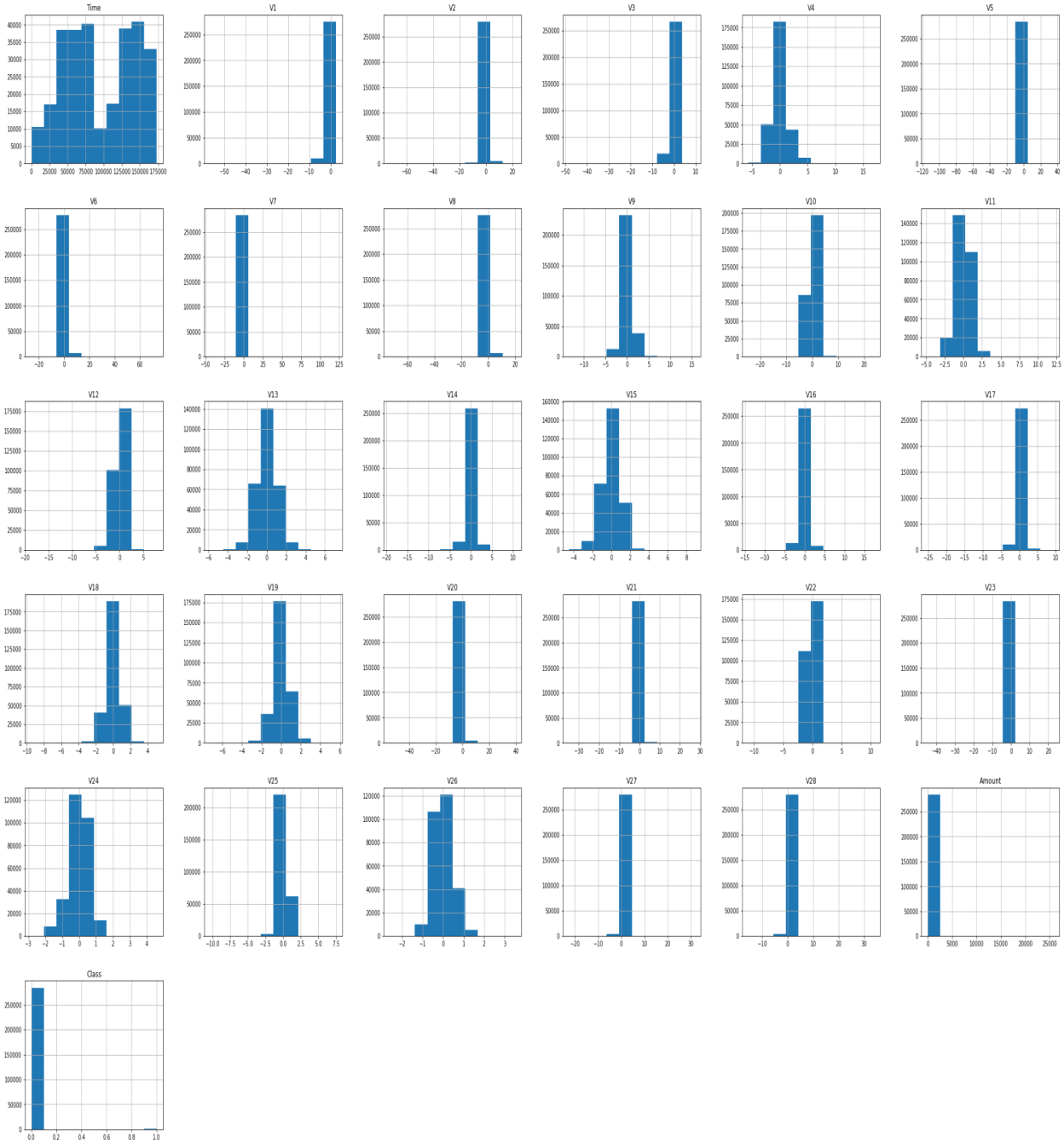


PAIRPLOTS:



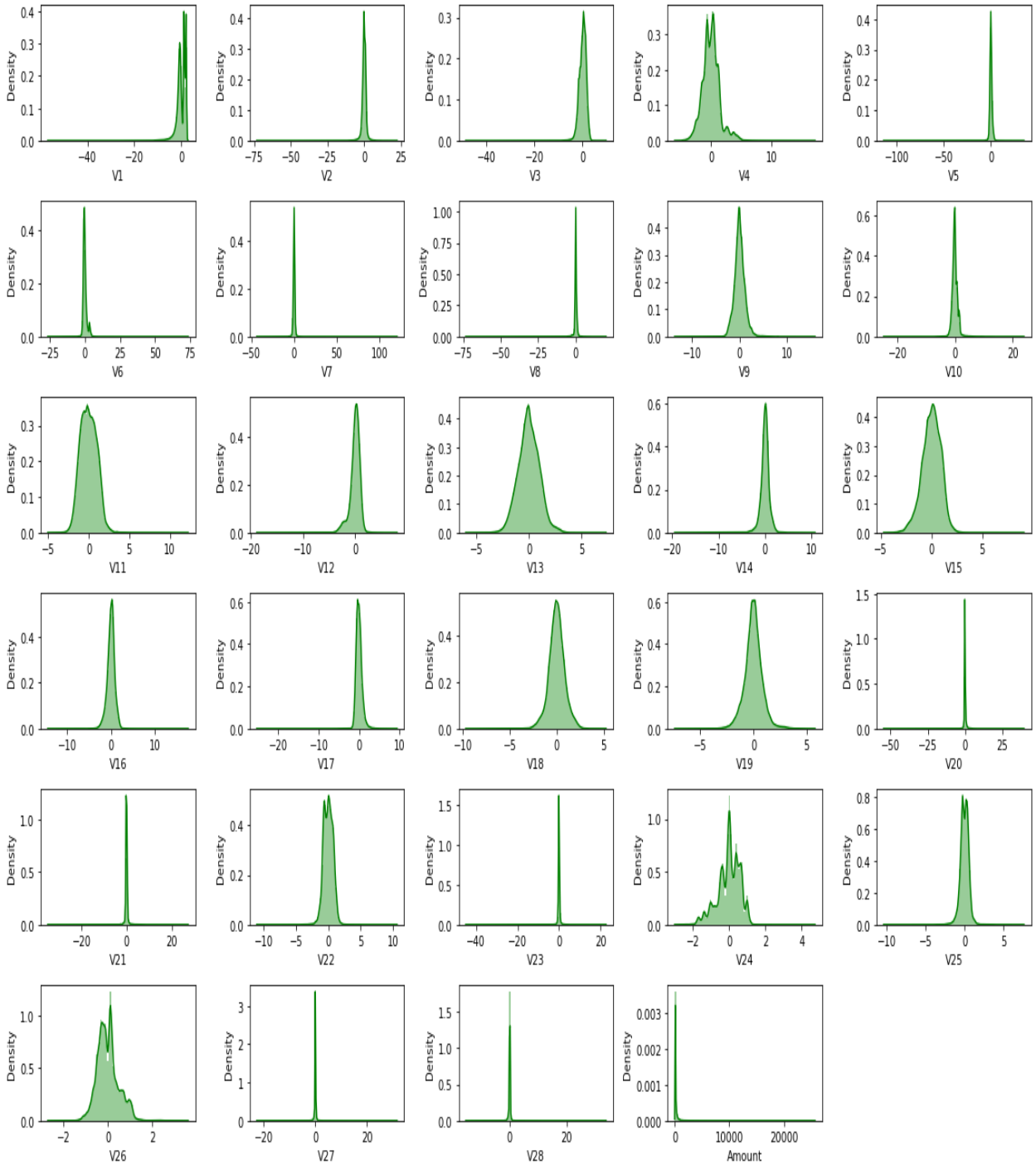
HISTOGRAM :

A histogram is an accurate representation of the distribution of numerical data. It is an estimate of the probability distribution of a continuous variable.



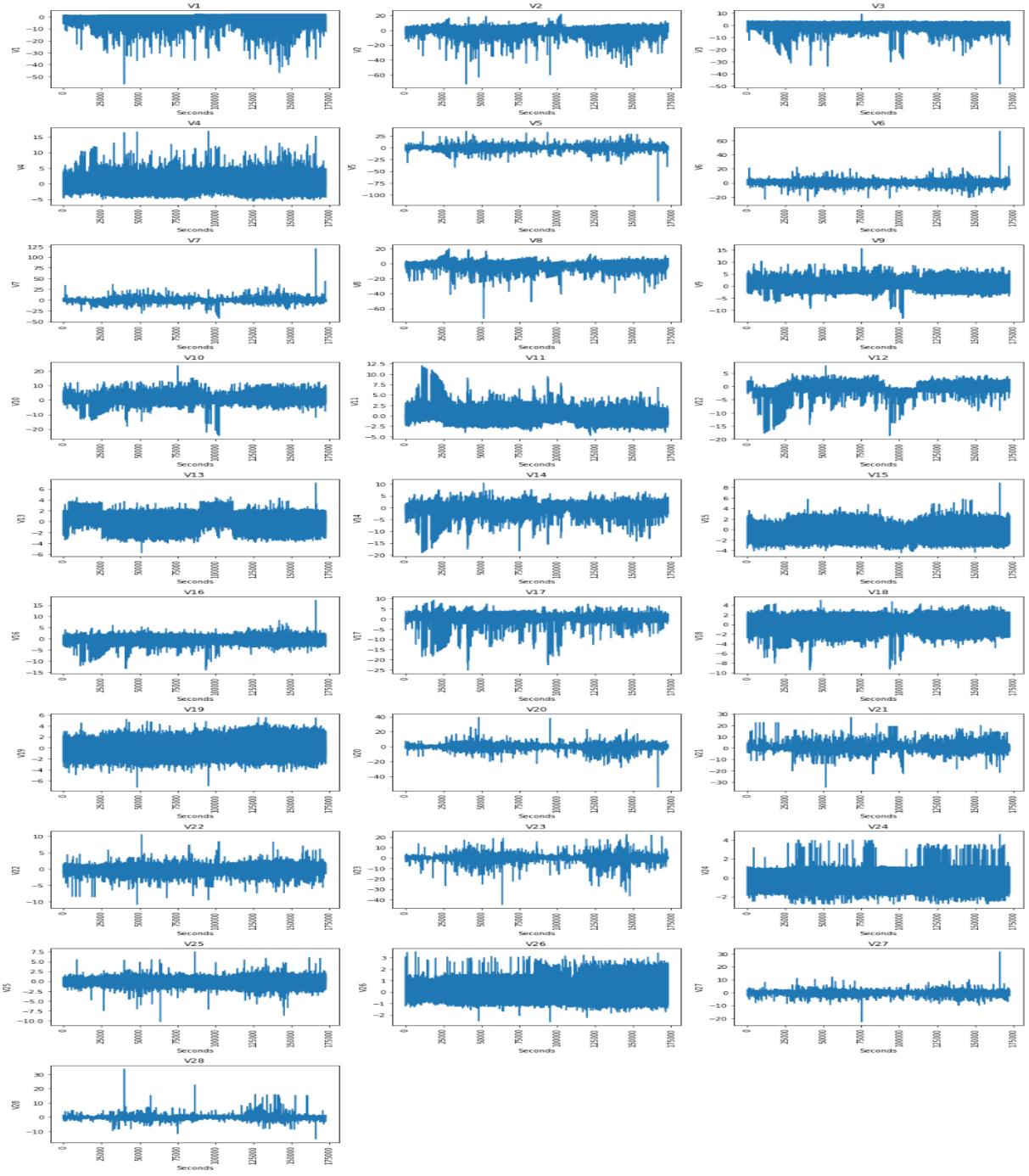
SNS DISTRIBUTION PLOT :

The distplot represents the univariate distribution of data i.e. data distribution of a variable against the density distribution.



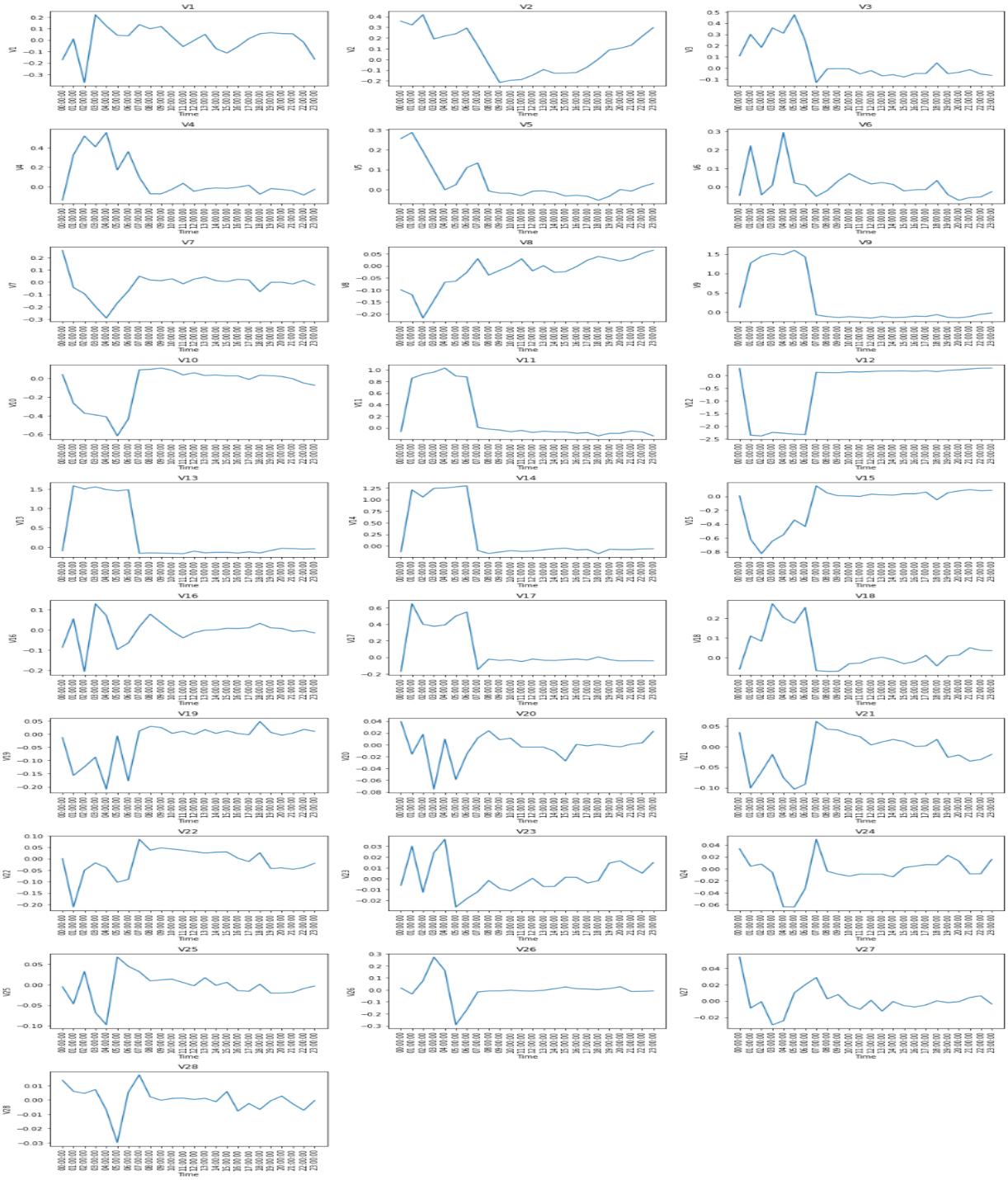
GRAPH :

A time series plot is a graph where some measure of time makes up the units on the x-axis. This is now called the time-axis, and the y-axis contains the data regarding what is being measured.



Averaged Hourly Time Series data:

We resampled the time series data into hourly time series data and averaged it over all the days in the given dataset. The resultant time series plots are as follows:



ML Algorithms and Visualization:

SCALING :

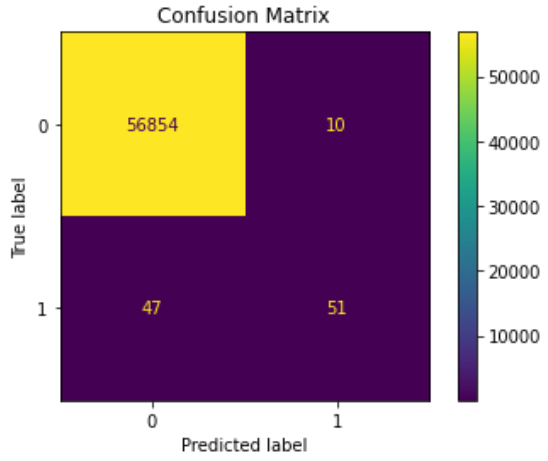
We used a standard scaler for scaling. This is an important technique that we performed as a preprocessing step before machine learning models, in order to standardize the range of functionality of the input dataset.

SVM CLASSIFIER :

- Training Accuracy: 0.9991090434286467
- Test Accuracy: 0.9989993328885924

	Precision	Recall	f1-score	support
0	1.00	1.00	1.00	56864
1	0.84	0.52	0.64	98
Accuracy			1.00	56962
Macro Average	0.92	0.76	0.82	56962
Weighted Average	1.00	1.00	1.00	56962

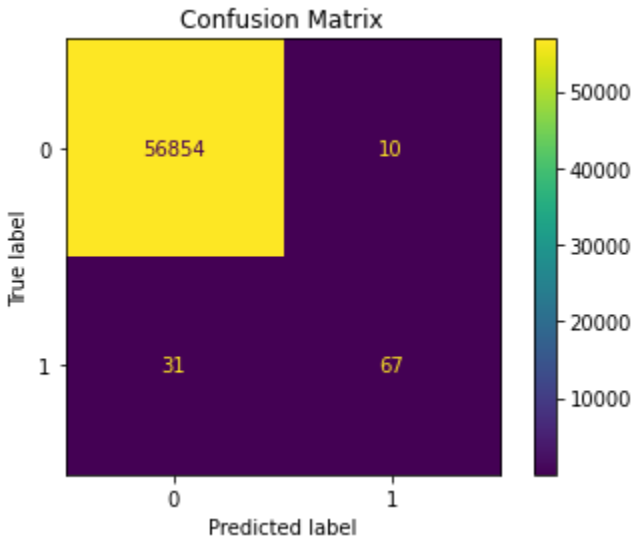
Confusion Matrix for SVM vs Xtest vs Ytest



RANDOM FOREST CLASSIFIER:

- Training Accuracy: 0.9999034431301982
- Test Accuracy: 0.9994733330992591

	Precision	Recall	f1-score	support
0	1.00	1.00	1.00	56864
1	0.96	0.72	0.83	98
Accuracy			1.00	56962
Macro Average	0.98	0.86	0.91	56962
Weighted Average	1.00	1.00	1.00	56962

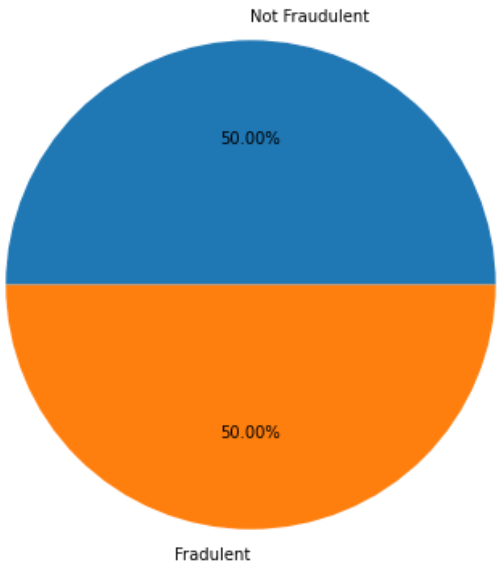


We can see that the recall of class 1 is 0.52 and 0.72 in SVM and Random Forest Classifiers. To solve this issue, we are using undersampling and oversampling.

UNDER SAMPLING :

Random undersampling involves randomly selecting examples from the majority class and deleting them from the training dataset.

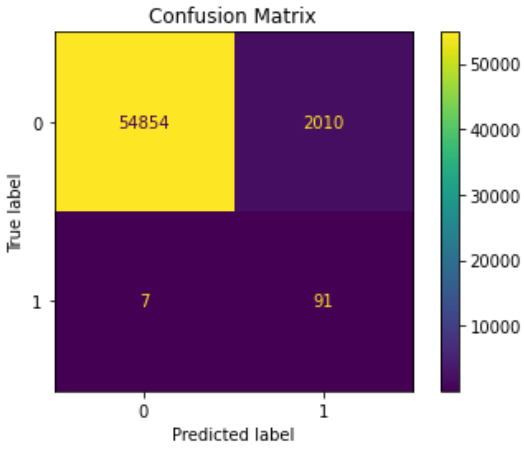
NON FRAUDULENT VS FRAUDULENT PIE CHART USING UNDER SAMPLING :



SVM CLASSIFIER USING UNDER SAMPLING :

- Training Accuracy: 0.9517766497461929
- Test Accuracy: 0.9645904287068572

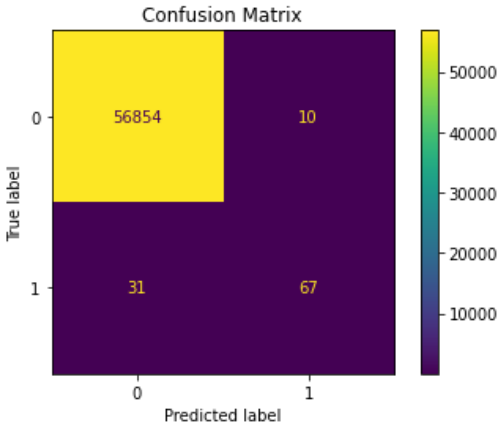
	Precision	Recall	f1-score	support
0	1.00	1.00	0.98	56864
1	0.04	0.52	0.08	98
Accuracy			0.96	56962
Macro Average	0.52	0.95	0.53	56962
Weighted Average	1.00	0.96	0.98	56962



RANDOM FOREST CLASSIFIER USING UNDER SAMPLING :

- Training Accuracy: 0.9928861788617886
- Test Accuracy: 0.9789508795337243

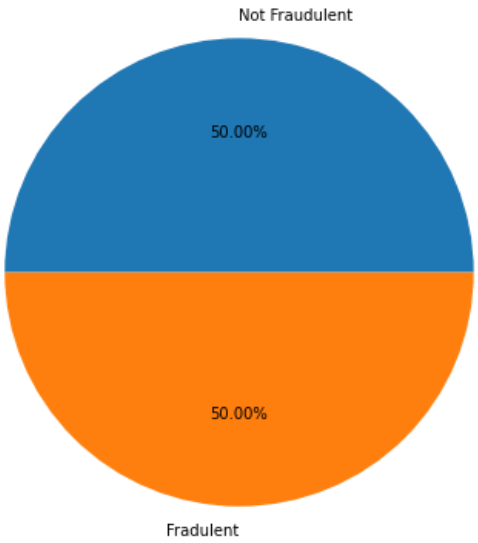
	Precision	Recall	f1-score	support
0	1.00	0.98	0.99	56864
1	0.07	0.99	0.14	98
Accuracy			0.98	56962
Macro Average	0.54	0.98	0.56	56962
Weighted Average	1.00	0.98	0.99	56962



OVER SAMPLING :

Random oversampling involves randomly selecting examples from the minority class, with replacement, and adding them to the training dataset.

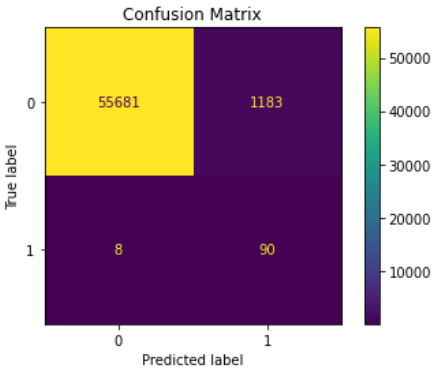
NON FRAUDULENT VS FRAUDULENT PIE CHART USING OVER SAMPLING :



SVM CLASSIFIER USING OVER SAMPLING :

- Training Accuracy: 0.9465071597838656
- Test Accuracy: 0.9790913240405885

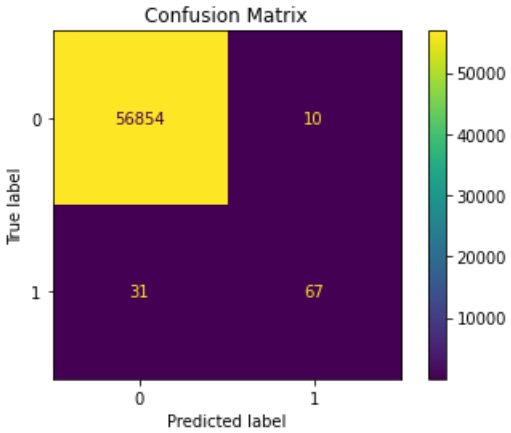
	Precision	Recall	f1-score	support
0	1.00	0.98	0.99	56864
1	0.07	0.92	0.13	98
Accuracy			0.98	56962
Macro Average	0.54	0.95	0.56	56962
Weighted Average	1.00	0.98	0.99	56962



RANDOM FOREST CLASSIFIER USING OVER SAMPLING :

- Training Accuracy: 0.9999978017243274
- Test Accuracy: 0.9994908886626171

	Precision	Recall	f1-score	support
0	1.00	1.00	1.00	56864
1	0.88	0.82	0.85	98
Accuracy			1.00	56962
Macro Average	0.94	0.91	0.92	56962
Weighted Average	1.00	1.00	1.00	56962



CONCLUSION

	Training Accuracy	Testing Accuracy
SVM Classifier	0.9991090434286467	0.9989993328885924
Random Forest Classifier	0.9999034431301982	0.9994733330992591
SVM Classifier using under sampling	0.9517766497461929	0.9645904287068572
Random Forest Classifier using under sampling	0.9928861788617886	0.9789508795337243
SVM Classifier using Over sampling	0.9465071597838656	0.9790913240405885
Random Forest Classifier using Over sampling	0.9999978017243274	0.9994908886626171

From the above table we can conclude that Random Forest Classifier is better than SVM for this particular dataset.

As we can observe from the above table, the results are more accurate if we use oversampling.