

# Credit Card Fraud Anomaly Detection

## Introduction:

Credit card fraud is a major worldwide problem that costs people, companies, and financial institutions a lot of money. The complexity and diversity of data patterns have made it harder to detect fraudulent conduct as the number of digital transactions has increased. By examining departures from typical transaction behavior, anomaly detection techniques are essential in spotting questionable activity. These techniques use statistical models and sophisticated machine learning algorithms to identify minor, odd trends that might point to fraud. In the digital economy, higher security, lower financial risk, and increased customer and service provider trust are all guaranteed by efficient fraud detection systems.

## Data Information:

Dataset link: <https://www.kaggle.com/code/rimshavirmani/anomaly-detection/notebook>

Used the Creditcard.csv file to train the models. The dataset information is mentioned below:

```
<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
...
29  Amount      284807 non-null float64
30  Class       284807 non-null int64  
dtypes: float64(30), int64(1)
memory usage: 67.4 MB
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount
0	0.0	-1.339807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	..	-0.018307	0.277838	-0.110474	0.066608	0.128539	-0.189115	0.133558	-0.021053	149.62
1	0.0	1.191837	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	-0.008983	0.014724	2.69
2	1.0	-1.338354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	..	0.247998	0.771679	0.909412	-0.689381	-0.327642	-0.138097	-0.055353	-0.059752	378.66
3	1.0	-0.966272	-0.185326	1.792993	-0.063291	-0.010309	1.247203	0.237809	0.377436	-1.387024	..	-0.108300	0.095274	-0.190321	-1.175575	0.647376	-0.221939	0.062723	0.061458	123.50
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.502941	-0.270533	0.817739	..	-0.009431	0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.99

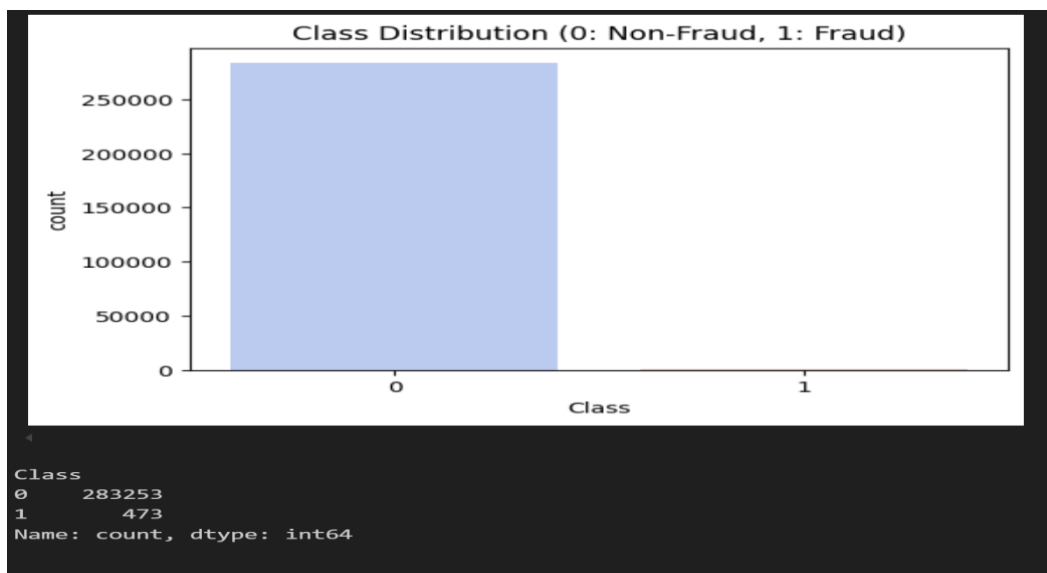
5 rows x 31 columns

## Data preprocessing:

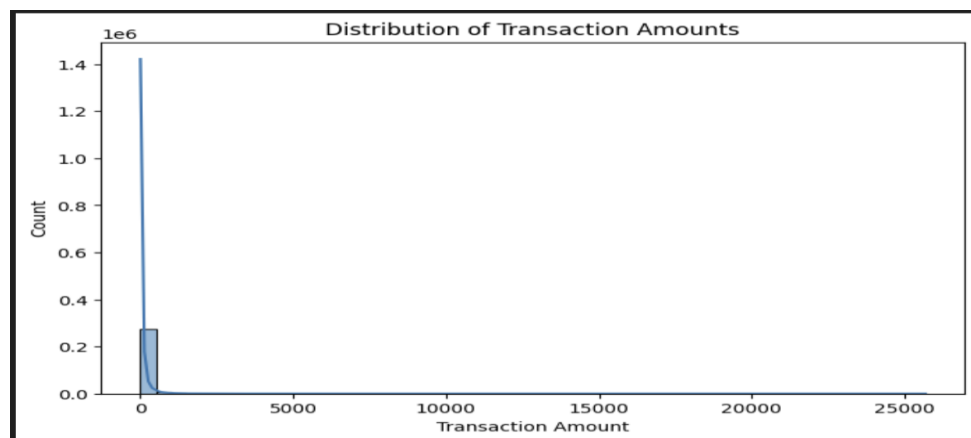
- Checking for datatypes and found that all columns are in numerical datatype.
- Checking for null values and found nothing.
- Removed duplicate rows.
- No removal of columns as they are required for model prediction.

## EDA and Visualization:

- Class Distribution and visualized it by bar graph. 473 fraud ones are found in dataset.

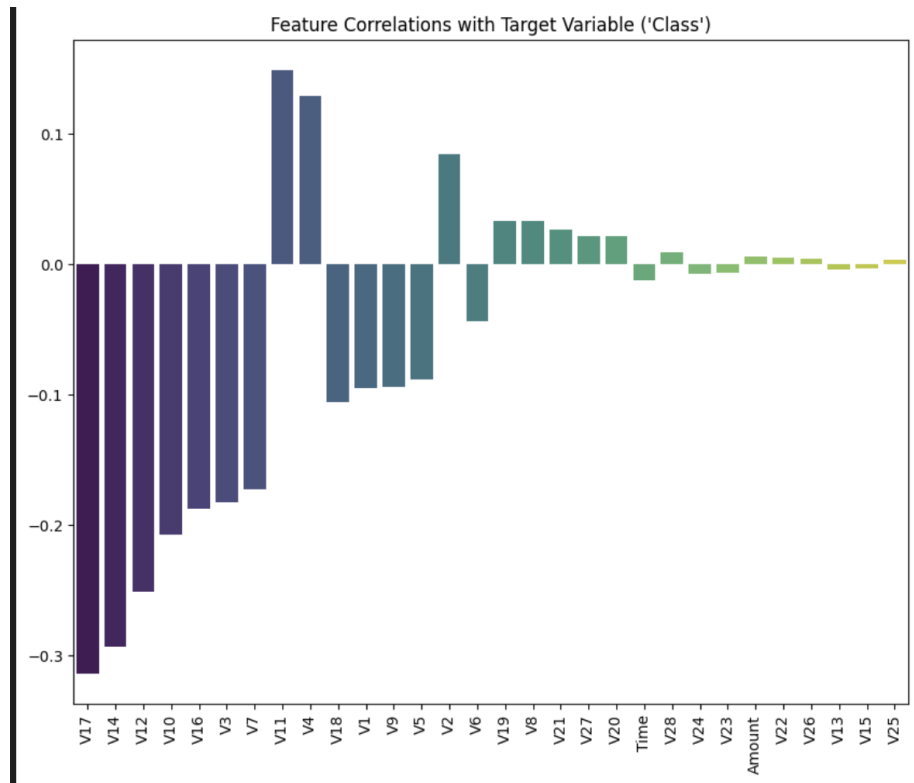


- Distribution of Transaction amounts.





- Normalization: Used standard scalar for it.
- Feature selection: Used all features as the data is highly differ for class 0 and 1.



Models:

- ISOLATION FOREST
- ONECLASS SVM
- LOCAL OUTLIER FACTOR

Evaluation metrics:

	Precision	Recall	F1-score	Accuracy	Anomalies
Isolation forest	Class 0: 1.00 Class 1: 0.20	Class 0: 1.00 Class 1: 0.21	1 and 0.21	0.99	94

Oneclass SVM	1 and 0.06	0.99 and 0.52	0.99 and 0.11	0.99	736
LOF	1 and 0.01	0.99 and 0.06	0.99 and 0.02	0.99	568

Out of all, the Isolation Forest works well and picked for model evaluation.

Model evaluation:

Used streamlit for app creation.

Link:

<https://huggingface.co/spaces/saikiranmansa/CreditFraudAnomalyDetection>

## Credit Card Fraud Anomaly Detection ML App

**Dataset Preview:**

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11
0	0	-1.3598	-0.0728	2.5363	1.3782	-0.3383	0.4624	0.2396	0.0987	0.3638	0.0908	-0
1	0	1.1919	0.2662	0.1665	0.4482	0.06	-0.0824	-0.0788	0.0851	-0.2554	-0.167	1
2	1	-1.3584	-1.3402	1.7732	0.3798	-0.5032	1.8005	0.7915	0.2477	-1.5147	0.2076	0
3	1	-0.9663	-0.1852	1.793	-0.8633	-0.0103	1.2472	0.2376	0.3774	-1.387	-0.055	-0
4	2	-1.1582	0.8777	1.5487	0.403	-0.4072	0.0959	0.5929	-0.2705	0.8177	0.7531	-0

**Select a Row for Model Input:**

Select a Row

1

**Detect Anomaly**

**Row selected for anomaly detection:**

	1
Time	0
V1	1.1919
V2	0.2662
V3	0.1665
V4	0.4482
V5	0.06
V6	-0.0824
V7	-0.0788
V8	0.0851
V9	-0.2554

**Anomaly Detection Result: Not Anomaly**

**Download Selected Row Result**