

# FRAUD DETECTION USING MACHINE LEARNING

BY : INPREET KAUR  
ASPIRING DATA SCIENTIST

**DOMAIN : BANKING**

**STAKEHOLDER : HEAD OF CREDIT  
CARD DEPARTMENT , CBA**

**BUSINESS PROBLEM : LOSS DUE  
TO CREDIT CARD FRAUD**

# AGENDA

A brief look at what we will discuss going further

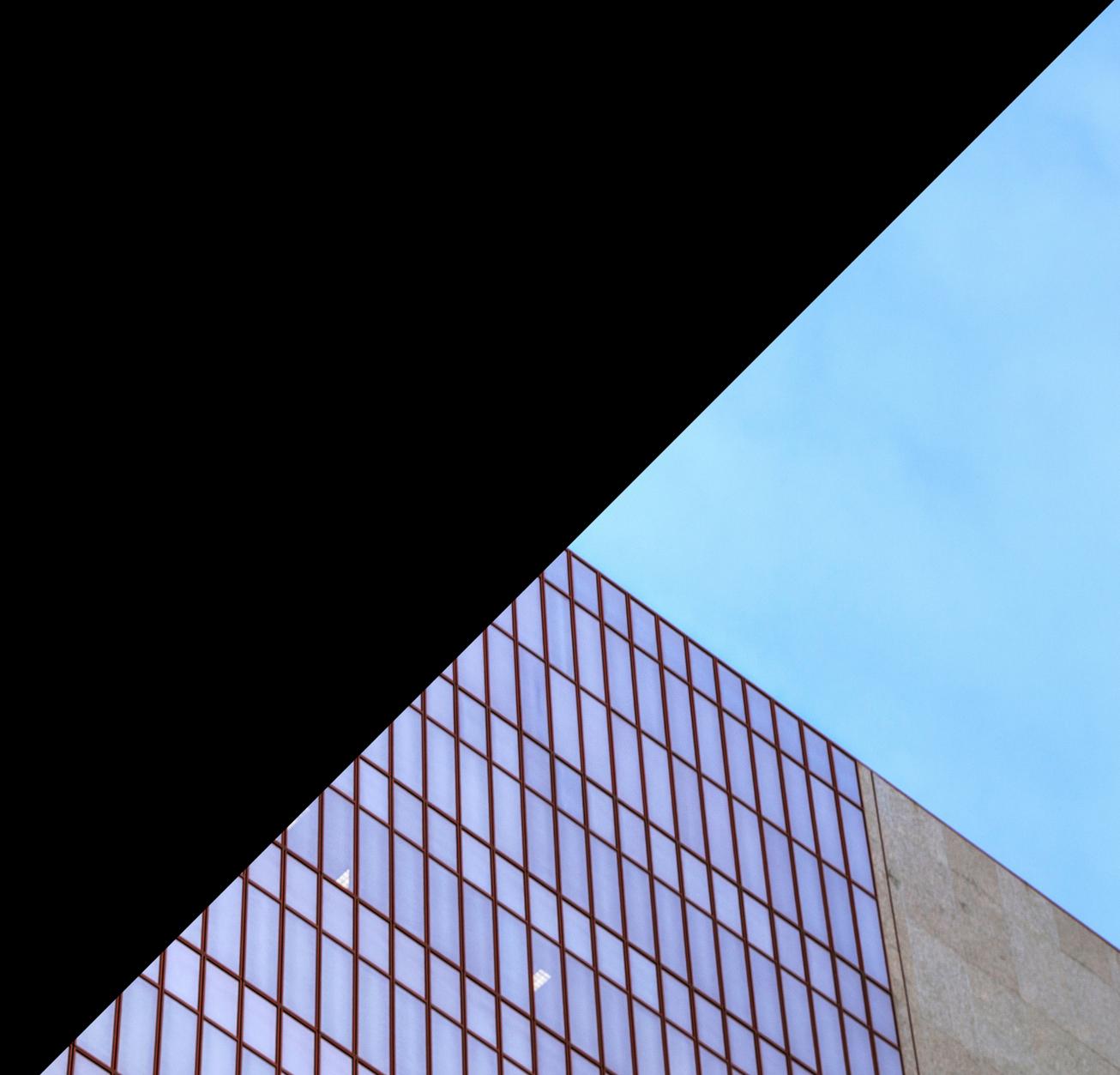
- Background
- Business Question Vs Data Question
- The Pipeline
- EDA and Feature Engineering
- Modelling
- Summary and Insights
- Cost Benefit Analysis

# BACKGROUND

**Credit card fraud is the fraudulent use of a credit card done so through the theft of the cardholder's personal details.**

**Thanks to the invention of the internet and the endless supply of eCommerce sites that came with it, credit card scammers now have an easier time than ever pinching your details**

**Data from the Australian Payments Network showed that card fraud cost \$447.2 million in the 2019/20 financial year. Most of this happened online through card-not-present (CNP) fraud, which made up the bulk of card fraud (87.7%)**



# **TYPES OF CREDIT CARD FRAUD**

**Card-not-present (CNP) fraud**

**Counterfeit and skimming fraud**

**Lost and stolen card fraud**

**Card-never arrived-fraud**

**False application fraud**



# **Business Question**

**How to detect the fraudulent transactions out of the non-fraudulent ones?**

# **Data Question**

**Which model can predict fraud transactions more accurately?**

# The Pipeline

01

## Identify the Business Question

How to detect and minimise fraud?

02

## Transform to Data Question

- How to build a model which can predict fraud?
- What data we require?
- What are the important features?
- Which model can predict fraud accurately?

03

## EDA and Feature Engineering

Explore the Data.  
Find and create the right features in order to predict fraud.

04

## Design the Model

Try various algorithms and compare.

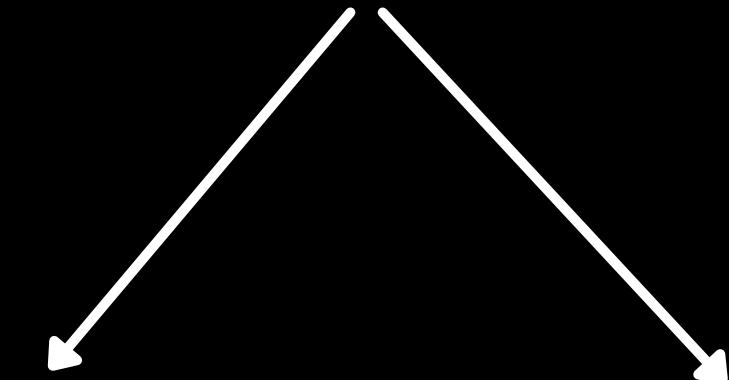
05

## Validation Testing

Select the best model that can predict fraud accurately

If classified as fraud,  
investigate further

If classified non fraud,  
good to go.



# How Frauds are detected?

- **Regional Statistics** : When purchases are made from different locations.
- **Transaction Statistics** : If a customer deviates from the regular buying pattern or time.
- **Time Based Number Of Transactions Statistics** : When large number of transactions are made from a card in short span of time.
- **Time Based Amount Statistics** : When suddenly costly items are purchased.

# THE DATASET

This is a simulated credit card transaction dataset from Kaggle containing legitimate and fraud transactions from the duration 1st Jan 2019 - 30th June 2020. It covers credit cards of 1000 customers doing transactions with a pool of 800 merchants.

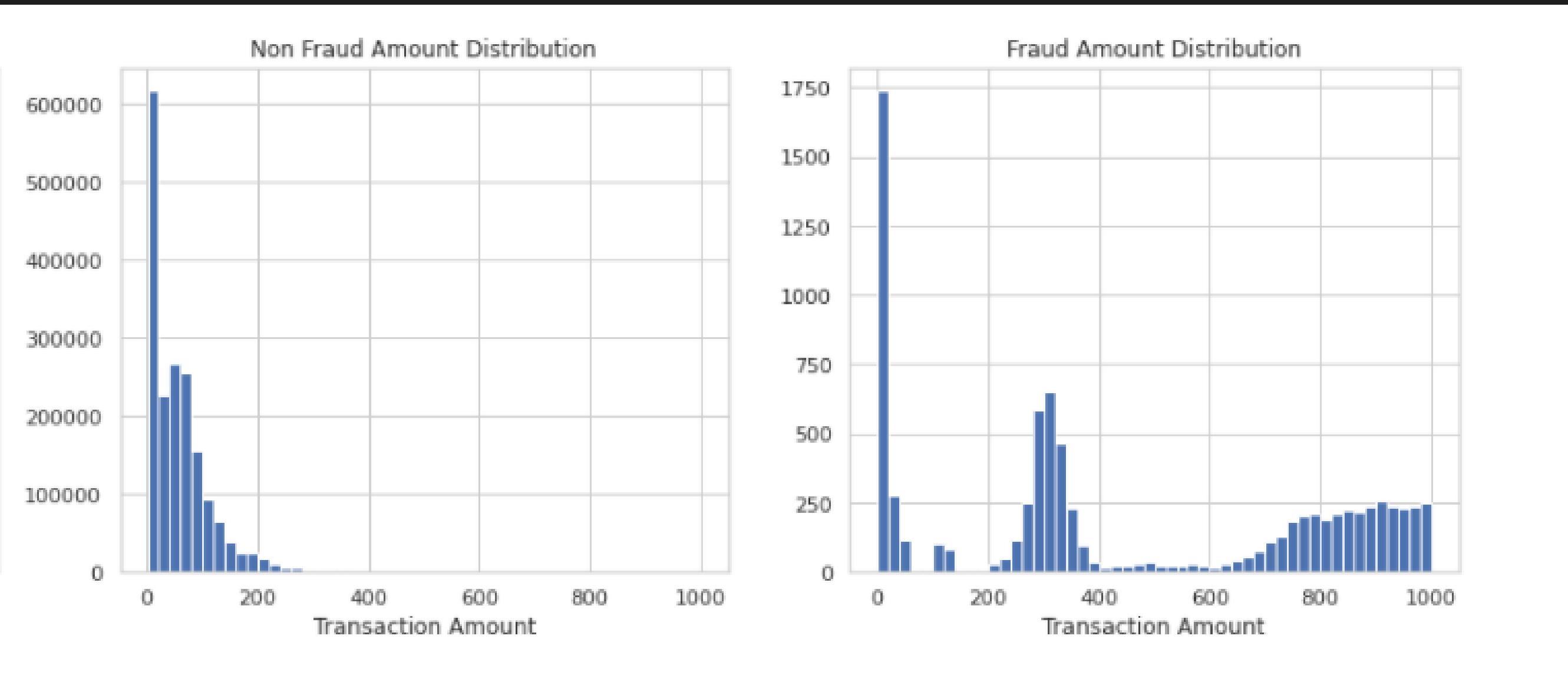
Number of rows = 1,852,394

Number of columns = 22

	is_fraud	count	percentage
0	0	1842743	99.478999
1	1	9651	0.521001

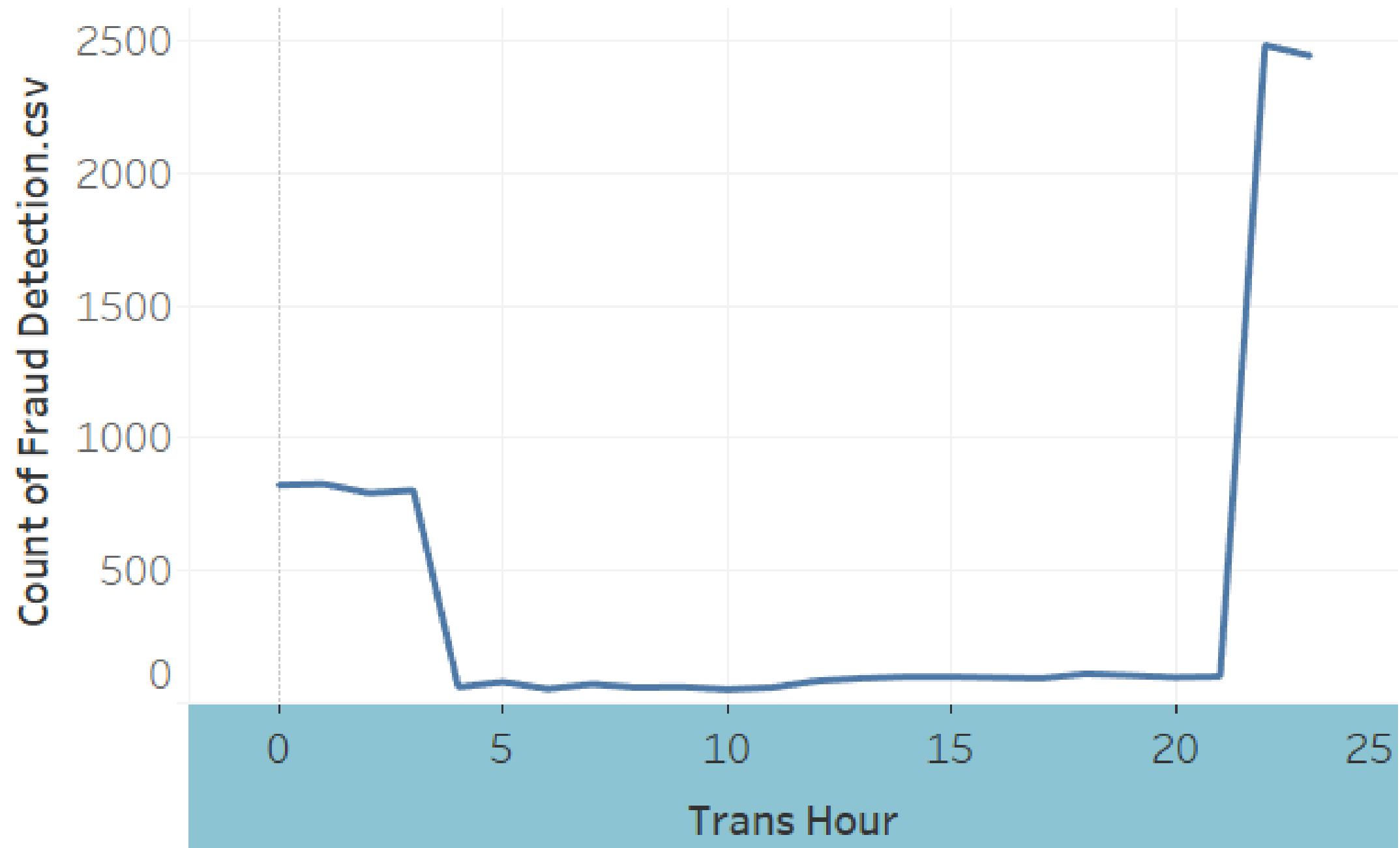
```
trans_date_trans_time      datetime64[ns]
cc_num                      int64
merchant                   object
category                   object
amt                         float64
first                       object
last                        object
gender                      object
street                      object
city                        object
state                       object
zip                          int64
lat                          float64
long                         float64
city_pop                    int64
job                          object
dob                          object
trans_num                   object
unix_time                   int64
merch_lat                   float64
merch_long                  float64
is_fraud                     int64
trans_hour                  int64
day_of_week                 object
year_month                  period[M]
hourEnc                      int64
dtype: object
```

# Exploratory Data Analysis



**Mean of Fraudulent Transactions = 530**  
**Mean of Non Fraud Transactions = 67**

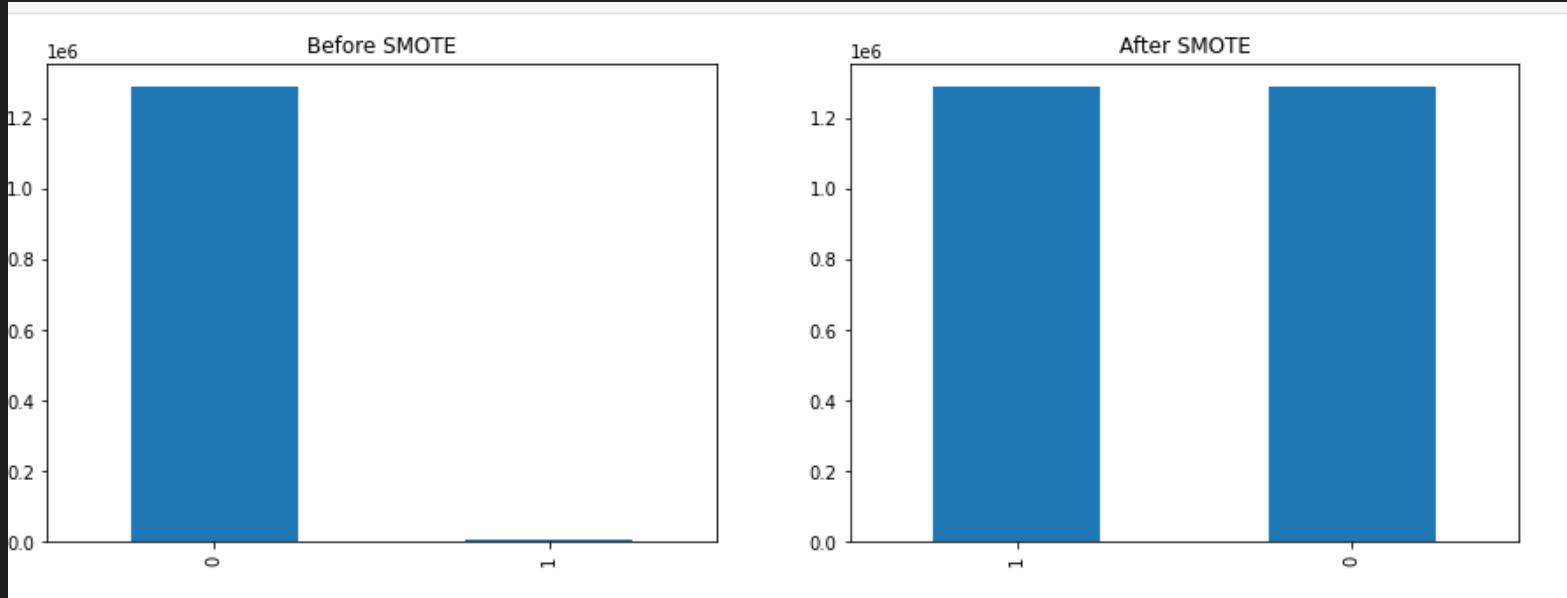
## Fraud Transactions by hour of the day



**Clearly we can see, most of the fraud transactions are done in the abnormal hours!**

# Feature Engineering

This Data was highly imbalanced so we have done oversampling with SMOTE to balance the data.



Extracted trans\_hour, transday\_of\_week and transmonth\_year from "trans\_date\_trans\_time" column

Created new features which are as follows:

- Hourenc {Encoded transactions done in normal hours 0500-2100 as normal (0) and transactions done in abnormal hours 2100-0500 as abnormal (1)}
- Derive age of the Customer (Transaction date - DOB)
- Calculated the distance between the Customer and the merchant from Latitude and longitude features.
- Extracted frequencies of transactions and Total Avg spend in last 60 days by customer
- Extracted frequencies of transactions in last 1 day by customer
- Extracted frequencies of fraud transactions in last 1 day by customer

Encoded the Categorical features

# Feature Selection

:	is_fraud	1.000000
	hist_fraud_trans_24h	0.772578
	amt	0.209307
	hourEnc	0.095764
	hist_trans_avg_amt_60d	0.084064
	hist_trans_60d	0.047788
	category_shopping_net	0.042452
	category_grocery_pos	0.033483
	category_misc_net	0.024667
	category_home	0.016623
	category_kids_pets	0.014307
	category_food_dining	0.013939
	category_health_fitness	0.013681
	unix_time	0.013330
	trans_hour	0.013196
	category_personal_care	0.011378
	age	0.010686
	category_misc_pos	0.008514
	week_Monday	0.008270
	category_grocery_net	0.006649
	category_travel	0.006286
	gender_M	0.005844
	week_Thursday	0.005712
	category_gas_transport	0.005155
	category_shopping_pos	0.004948
	week_Wednesday	0.004183
	week_Sunday	0.003870
	week_Tuesday	0.003026
	lat	0.002903
	merch_lat	0.002777
	week_Saturday	0.002612
	zip	0.002190
	cc_num	0.001125
	long	0.001021
	merch_long	0.000999
	hist_trans_24h	0.000486
	distance	0.000359
	city_pop	0.000325
	val_for_agg	NaN
	Name: is_fraud, dtype:	float64

# MODELLING

MODEL	PRECISION	RECALL	AUC
Logistic Regression	0.54	0.90	0.94
Gaussian Naive Bayes	0.49	0.86	0.92
Random Forest	0.90	0.90	0.95
Decision Tree with Bagging	0.87	0.89	0.94
Decision Tree with Adaptive Boosting	0.49	0.93	0.96
XG Boost	0.66	0.94	0.97

# SUMMARY AND INSIGHTS

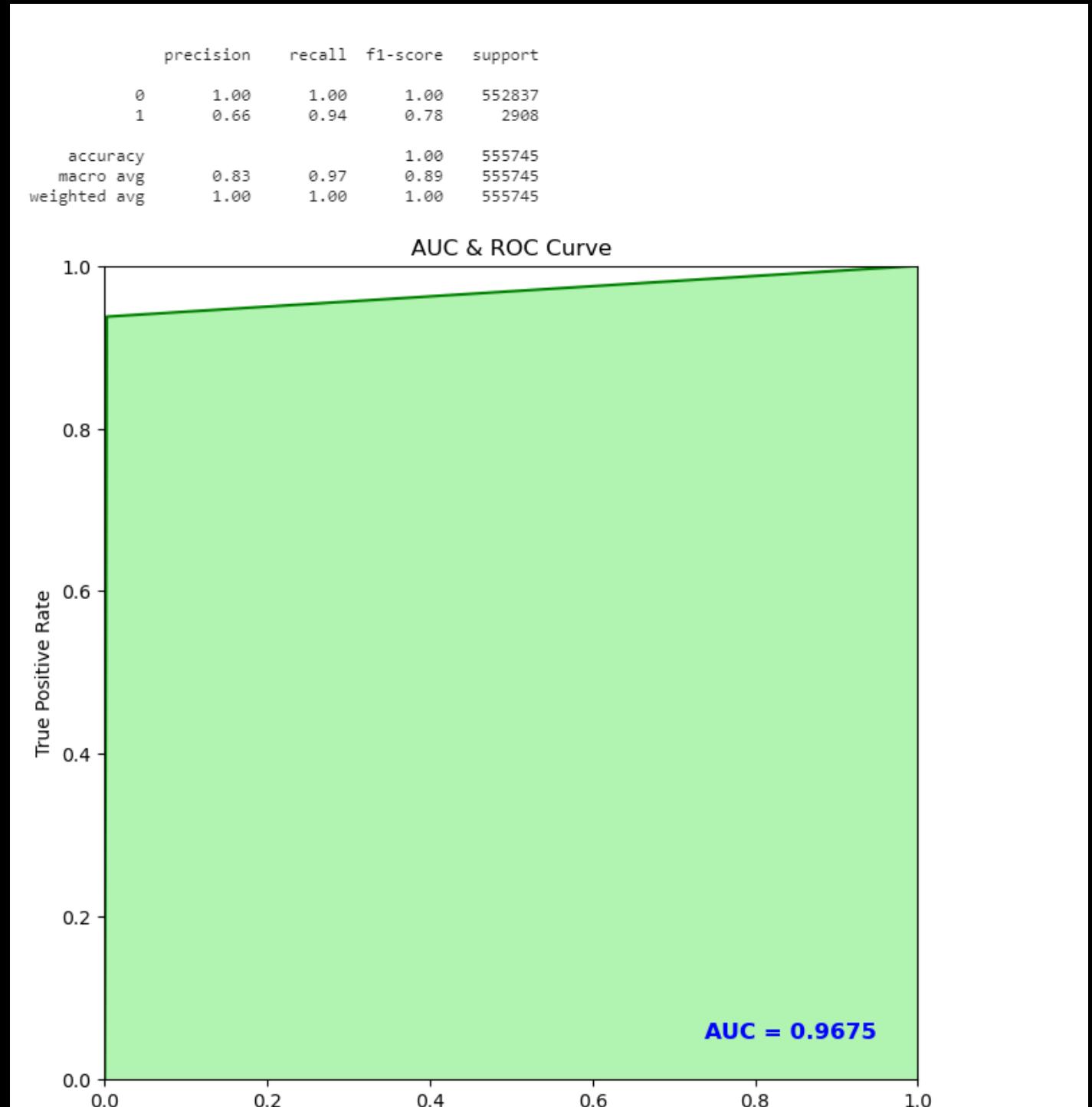
XGBoost Model can predict fraud with 94% accuracy.

```
[[551457 1380]
 [ 182 2726]]
```

## Limitations

- This is a simulated dataset and may not be comparable to the real life problem.

The cost of False Negatives is much higher than the cost of False Positives. By looking at the problem among all the metrics, we need to focus on getting high recall and at the same time, can evaluate the model performance by looking at precision, and AUC score.



# COST BENEFIT ANALYSIS

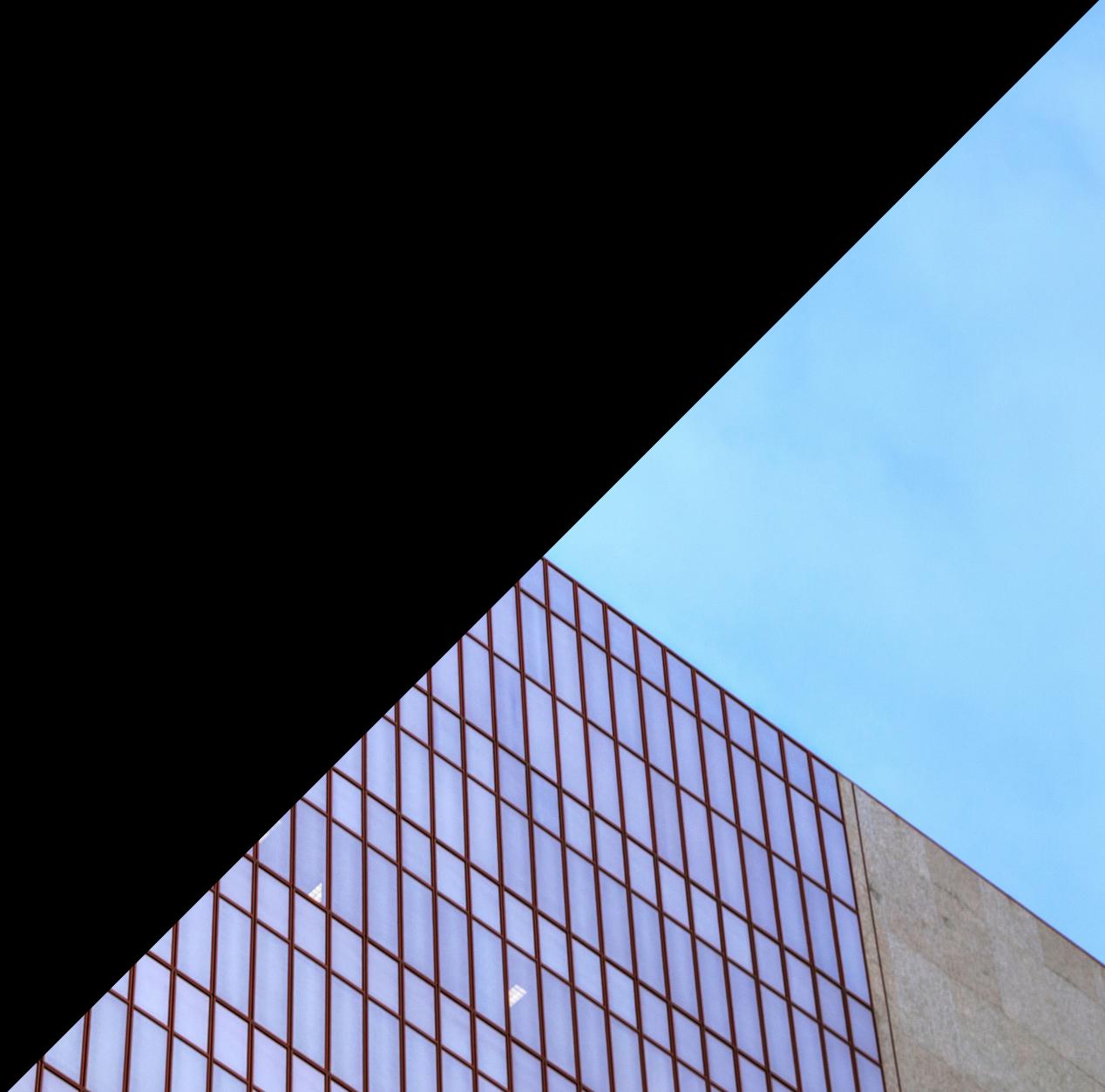
Average number of transactions per month : 77,183

Average number of fraudulent transaction per month : 402

Average amount per fraud transaction: 530

Annual Loss from Fraud : \$2,556,720

Even if we detect 94% of fraud, we can save \$2,403,316.



# Thank you for listening

Inpreetkaur.1988@gmail.com  
+61404402560

<https://github.com/InpreetKaur88/Capstone-Project---Credit-Card-Fraud-Detection>

# References

<https://www.savings.com.au/credit-cards/credit-card-fraud>

<https://www.kaggle.com/kaanboke/the-most-common-evaluation-metrics-a-gentle-intro>

<https://towardsdatascience.com/credit-card-fraud-detection-9bc8db79b956>

<https://medium.com/@ogunbajoadeyinka/credit-card-fraud-detection-with-machine-learning-in-python-5323ded33472>