Credit Card Fraud Detection-Capstone Project

Created By:
Namia Modamed Ali

Problem Statement

- Finex is a leading financial service provider based out of Florida, US. It offers a wide range of products and business services to customers through different channels, ranging from in-person banking and ATMs to online banking. Over the last few years, Finex has observed that a significantly large number of unauthorized transactions are being made, due to which the bank has been facing a huge revenue and profitability crisis.

- Customers have been complaining about unauthorized transactions being made through their credit/debit cards. It has been reported that fraudsters use stolen/lost cards and hack private systems to access the personal and sensitive data of many cardholders.

- ATM skimming at various POS terminals such as gas stations, shopping malls, and ATMs that do not send alerts or do not have OTP systems through banks has also been reported

- Since these fraudulent activities occurs at non peak and odd hours, this has led to late complaint registration with Finex and by the time the case is flagged fraudulent, the bank incurs heavy losses and ends up paying the lost amount to the cardholders.

Business Objective

- In recent times, the number of fraud transactions has increased drastically due to which credit card companies are facing a lot of challenges. For many banks, retaining high profitable customers is the most important business goal. Banking fraud, however, poses a significant threat to this goal.

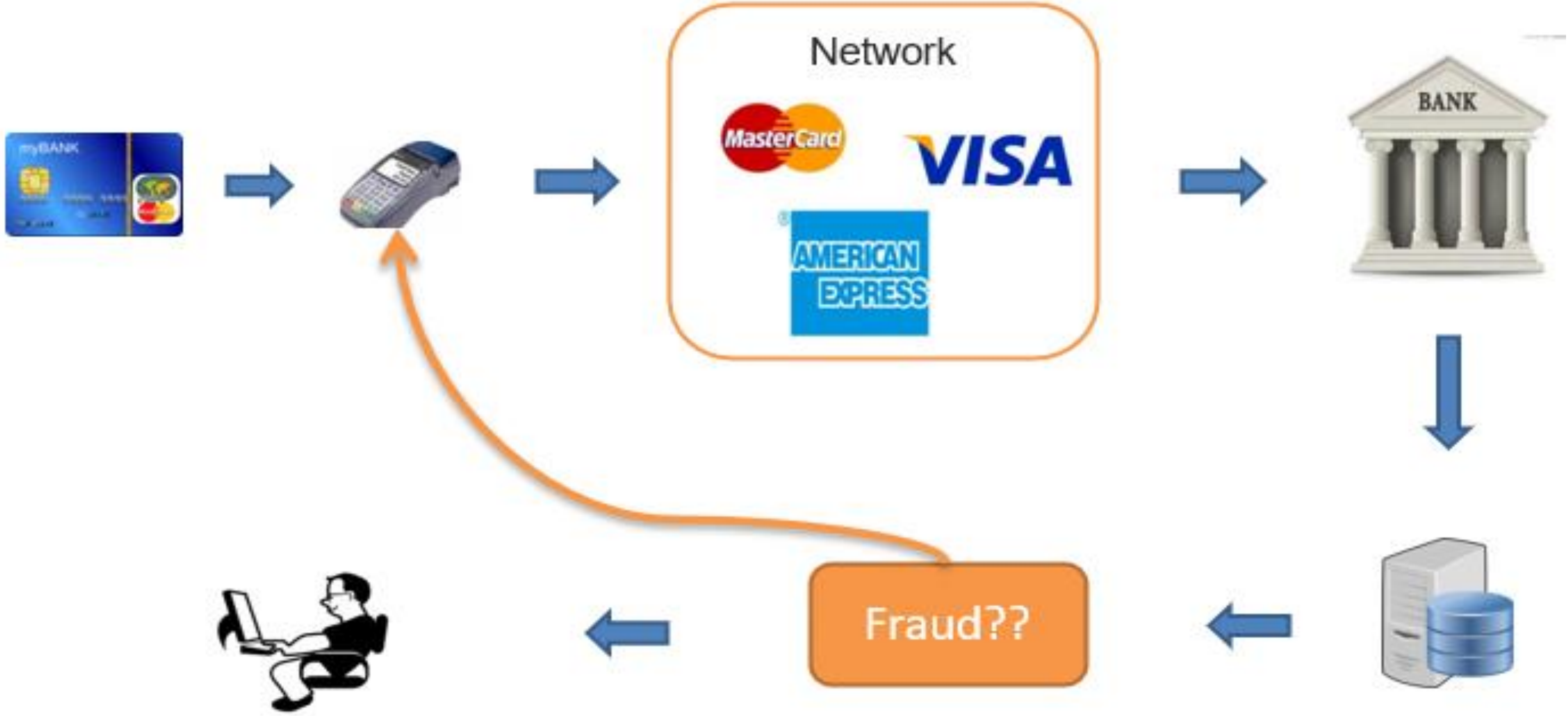
- The Branch Manager is worried about the ongoing situation and wants to identify the possible root causes and action areas to come up with a long-term solution that would help the bank generate high revenue with minimal losses.

- The aim of this capstone project is to identify and predict fraudulent credit card transactions using machine learning models.

APPROACH

- Reading and understanding the data
- Data Cleaning
- EDA
- Dealing with imbalance - Sampling
- Creating Dummy Variables.
- Feature Scaling
- Splitting the data into train-test dataset
- Model Building
- Model Evaluation- Accuracy, Sensitivity, Specificity, Precision, recall
- Making predictions on test set.

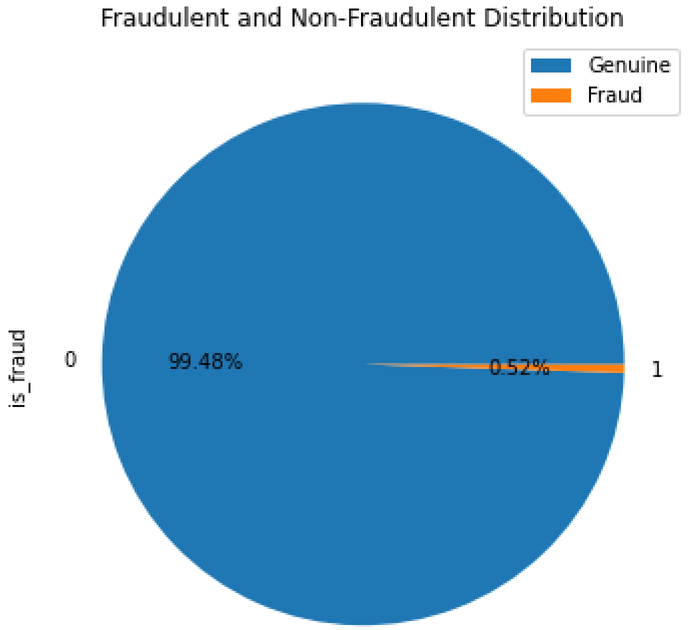
Simplify transaction flow

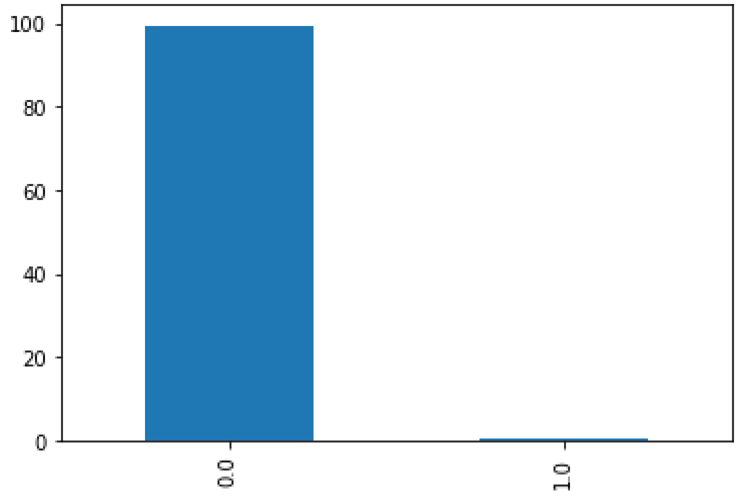


Data Understanding Cleaning

- Firstly, we analyzed the data to know more about its Attributes.
- The datasets contains transactions made by credit cards, where we have **9651** frauds out of **1852394** transactions. The dataset is highly unbalanced, the positive class (frauds) account for **0.52%** of all transactions.
- Un named column and those with null values are dropped.
- Data type conversion is done for trans_date_trans_time column from object to date time
- Transaction time , month , year , week , week number , minute the transaction occurred were extracted.

Imbalance in Data



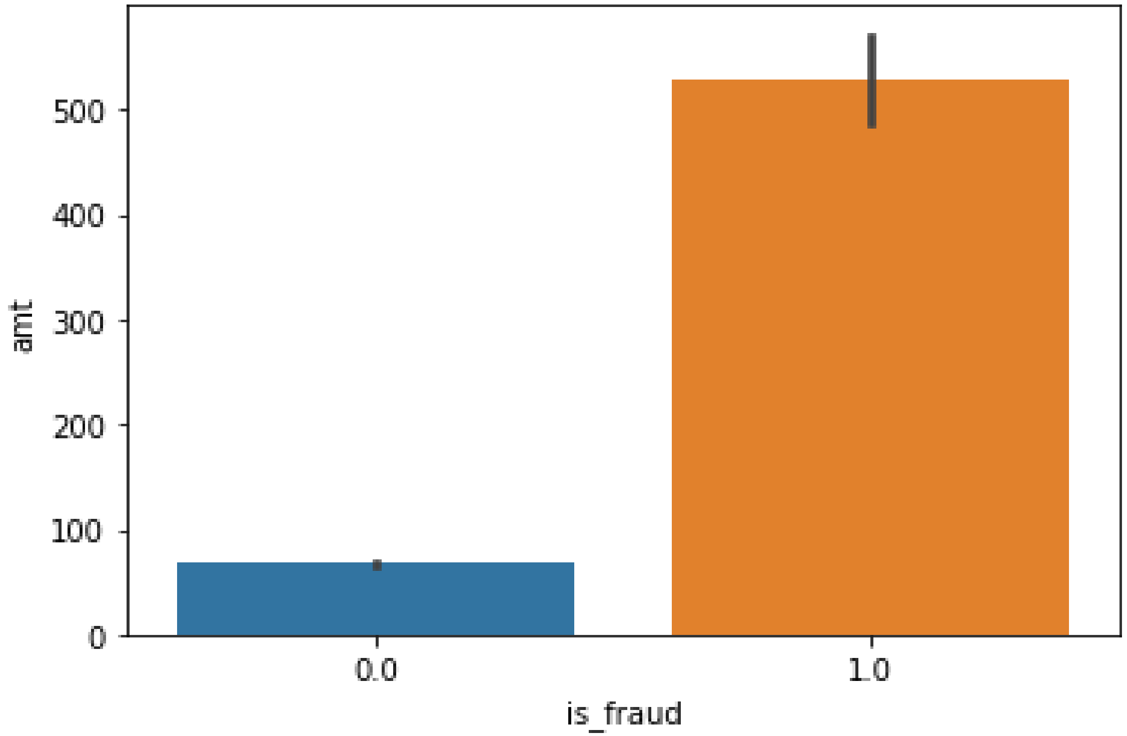


High Imbalance in the dataset, displayed by:

- Genuine as valid transactions and fraud as fraudulent transaction.

- **1** is denoted for fraudulent cases whereas **0** represent genuine transactions.

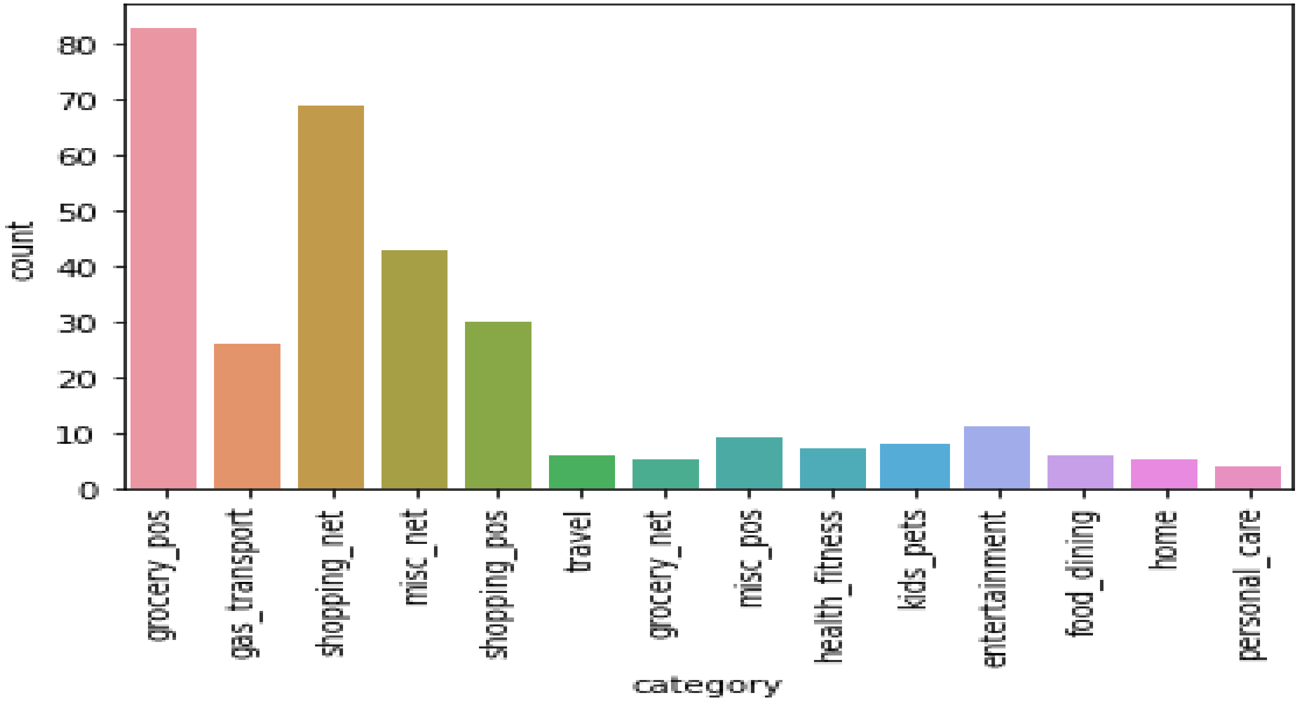
Exploratory Data Analysis



Amount for fraud and valid transactions

- Even though the fraud transaction is very less in number the average amount for fraud transaction is higher compared to the valid transaction.

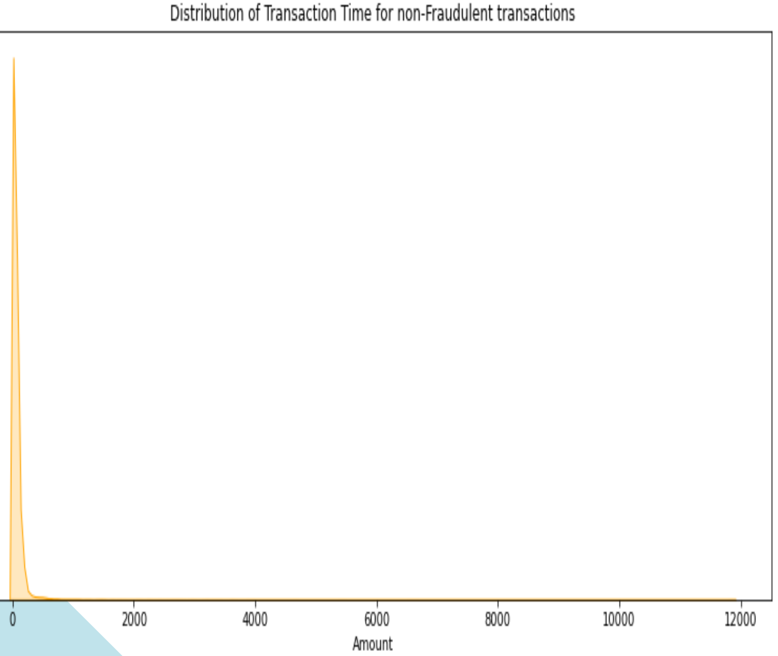
Exploratory Data Analysis

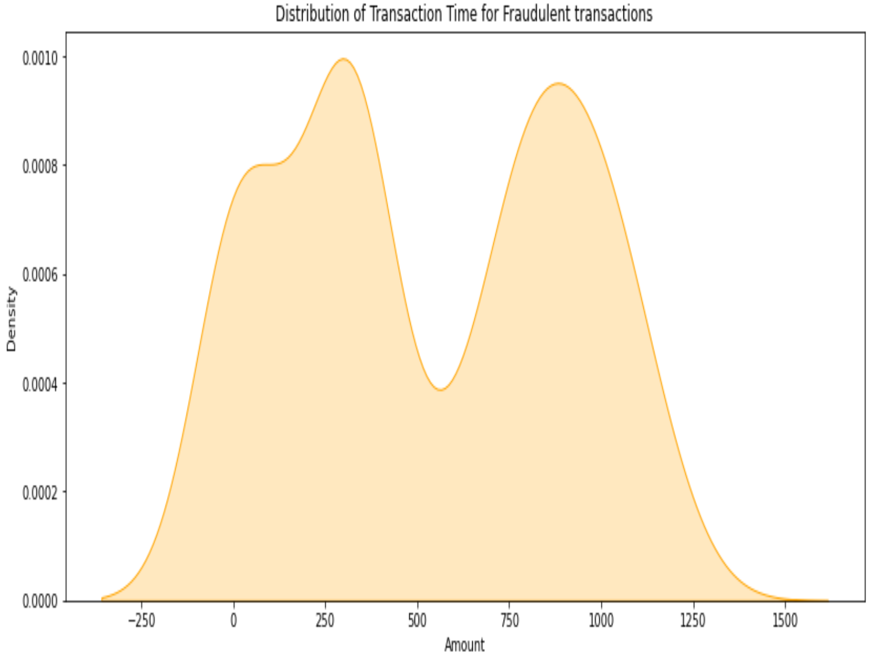


- From the plot it is evident that Grocery POS and shopping online has higher fraudulent transactions.

- This is because fraudulent transaction occurs at pos during non suspicious place and timing , as well as digital transactions cannot be flagged fraudulent sooner.

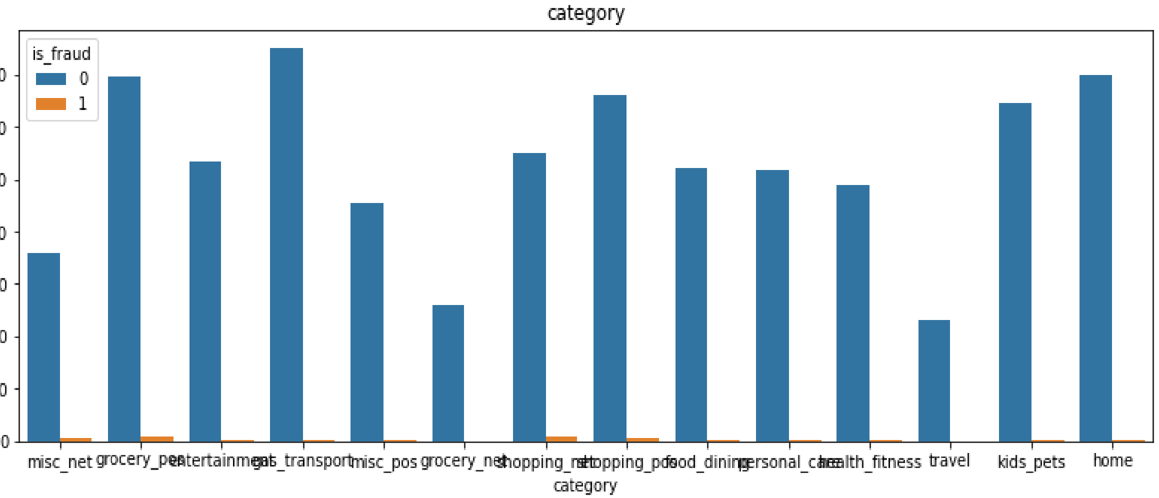
Exploratory Data Analysis

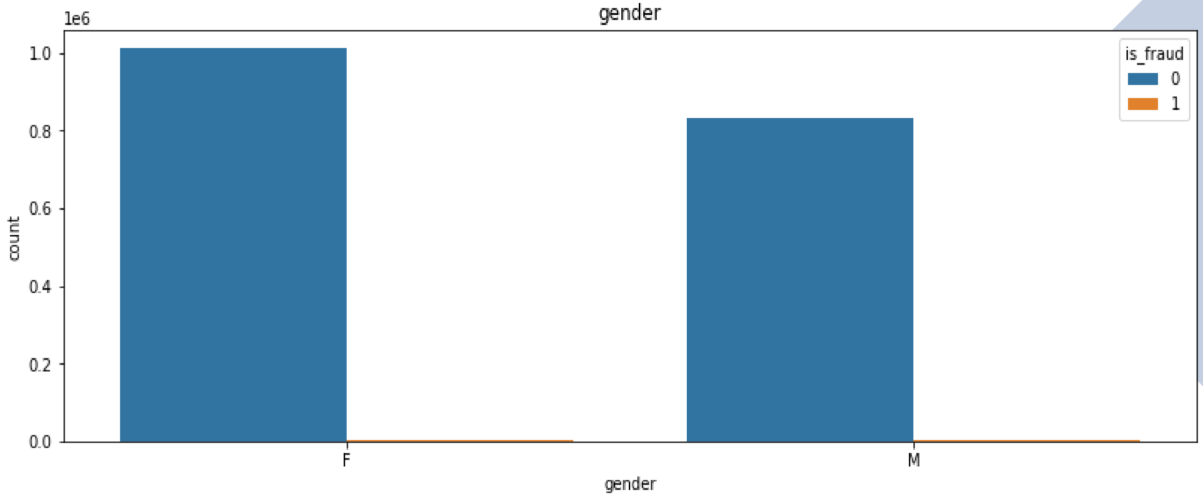


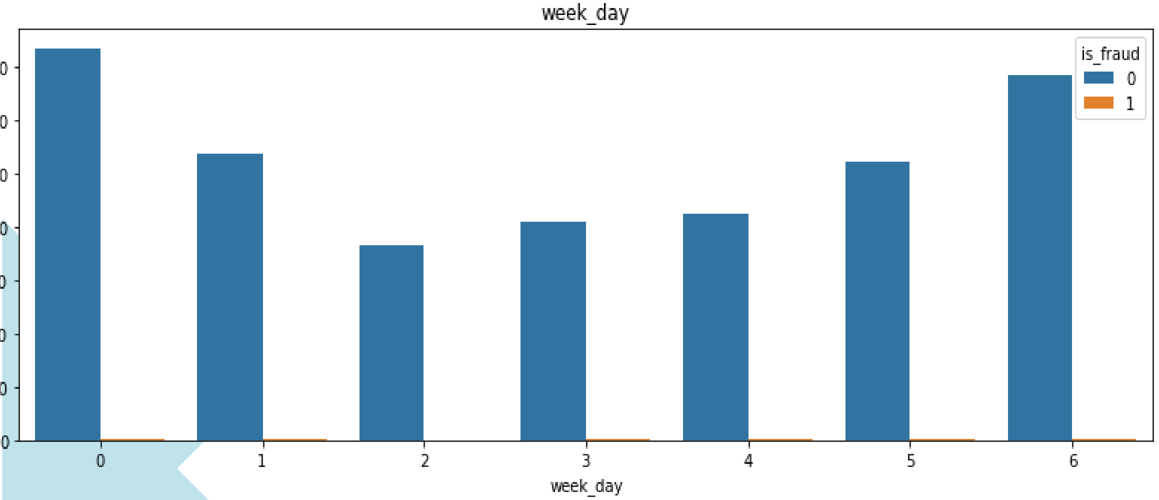


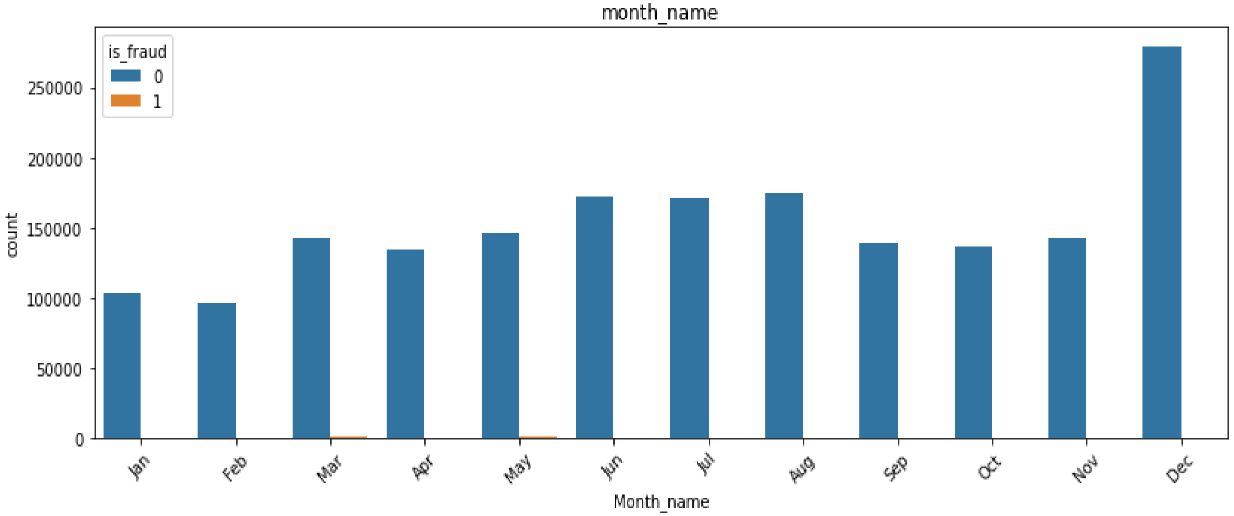
- **Distribution of transaction time** for each transaction is displayed here.
- We can see that Distribution of transaction time for Fraudulent transaction is more spread.

Exploratory Data Analysis

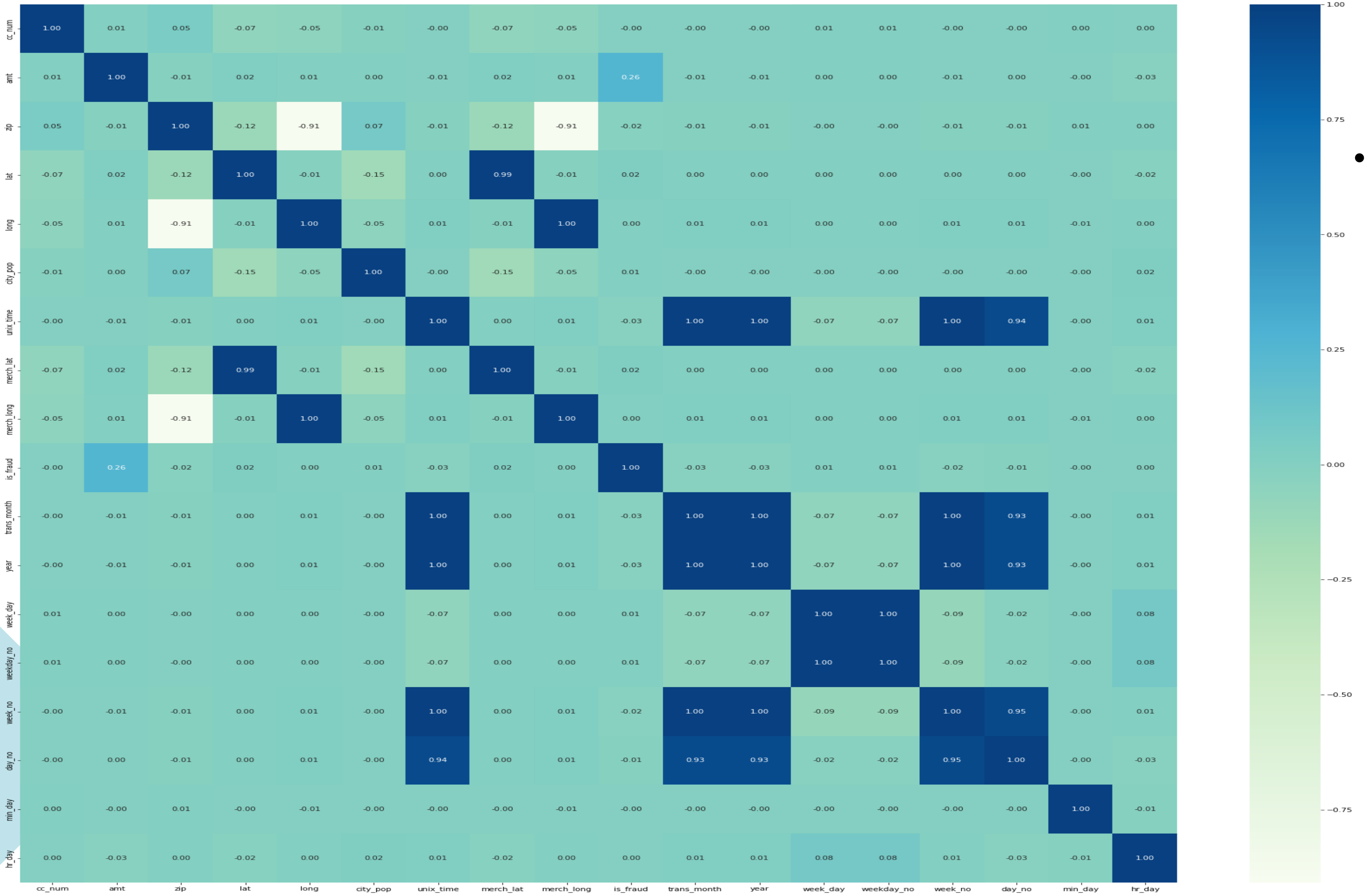








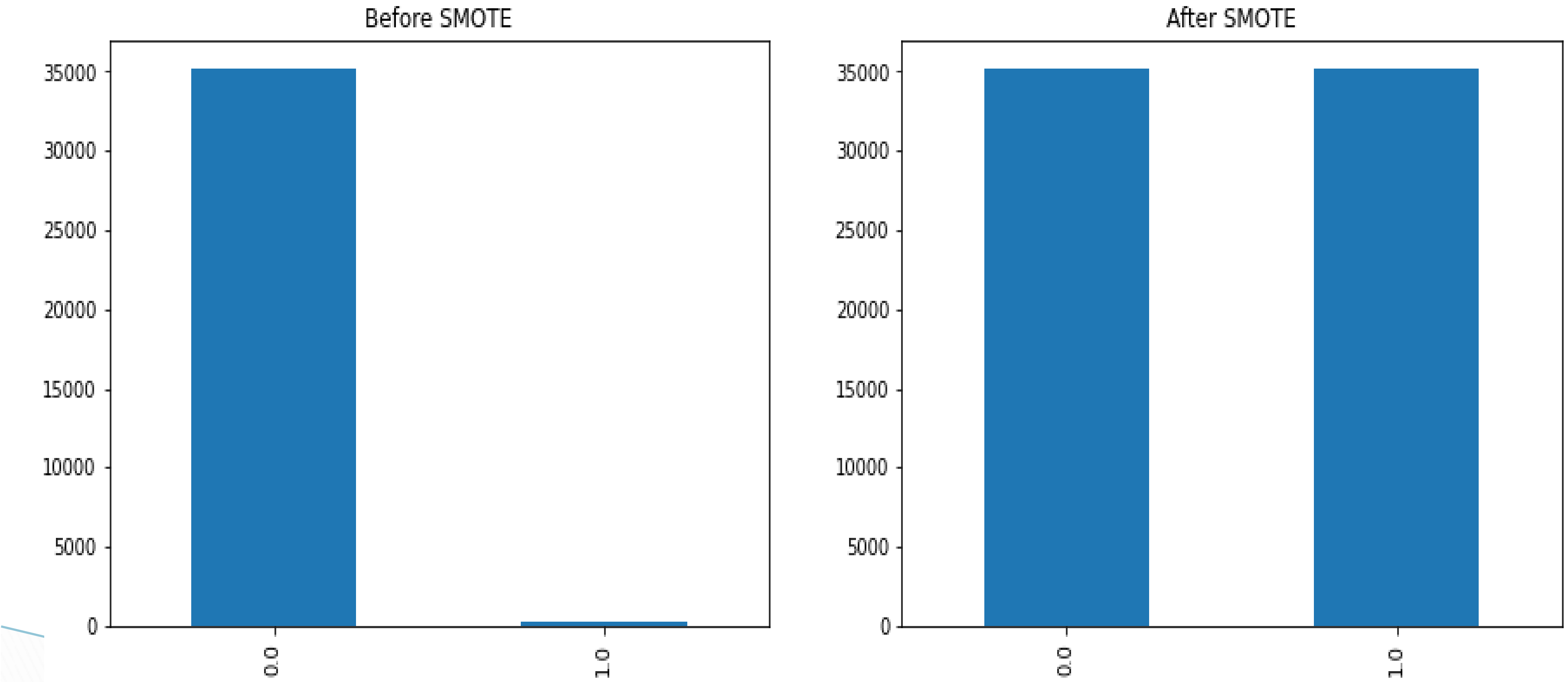
Correlation



	cc_num	amt	zip	lat	long	city_pop	unix_time	merch_lat	merch_long	is_fraud	trans_month	year	week_day	weekday_no	week_no	day_no	min_day	hr_day
cc_num	1.00	0.01	0.05	-0.07	-0.05	-0.01	-0.00	-0.07	-0.05	-0.00	-0.00	-0.00	0.01	0.01	-0.00	-0.00	0.00	0.00
amt	0.01	1.00	-0.01	0.02	0.01	0.00	-0.01	0.02	0.01	0.26	-0.01	-0.01	0.00	0.00	-0.01	0.00	-0.00	-0.03
zip	0.05	-0.01	1.00	-0.12	-0.91	0.07	-0.01	-0.12	-0.91	-0.02	-0.01	-0.01	-0.00	-0.00	-0.01	-0.01	0.01	0.00
lat	-0.07	0.02	-0.12	1.00	-0.01	-0.15	0.00	0.99	-0.01	0.02	0.00	0.00	0.00	0.00	0.00	0.00	-0.00	-0.02
long	-0.05	0.01	-0.91	-0.01	1.00	-0.05	0.01	-0.01	1.00	0.00	0.01	0.01	0.00	0.00	0.01	0.01	-0.01	0.00
city_pop	-0.01	0.00	0.07	-0.15	-0.05	1.00	-0.00	-0.15	-0.05	0.01	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	0.02
unix_time	-0.00	-0.01	-0.01	0.00	0.01	-0.00	1.00	0.00	0.01	-0.03	1.00	1.00	-0.07	-0.07	1.00	0.94	-0.00	0.01
merch_lat	-0.07	0.02	-0.12	0.99	-0.01	-0.15	0.00	1.00	-0.01	0.02	0.00	0.00	0.00	0.00	0.00	0.00	-0.00	-0.02
merch_long	-0.05	0.01	-0.91	-0.01	1.00	-0.05	0.01	-0.01	1.00	0.00	0.01	0.01	0.00	0.00	0.01	0.01	-0.01	0.00
is_fraud	-0.00	0.26	-0.02	0.02	0.00	0.01	-0.03	0.02	0.00	1.00	-0.03	-0.03	0.01	0.01	-0.02	-0.01	-0.00	0.00
trans_month	-0.00	-0.01	-0.01	0.00	0.01	-0.00	1.00	0.00	0.01	-0.03	1.00	1.00	-0.07	-0.07	1.00	0.93	-0.00	0.01
year	-0.00	-0.01	-0.01	0.00	0.01	-0.00	1.00	0.00	0.01	-0.03	1.00	1.00	-0.07	-0.07	1.00	0.93	-0.00	0.01
week_day	0.01	0.00	-0.00	0.00	0.00	-0.00	-0.07	0.00	0.00	0.01	-0.07	-0.07	1.00	1.00	-0.09	-0.02	-0.00	0.08
weekday_no	0.01	0.00	-0.00	0.00	0.00	-0.00	-0.07	0.00	0.00	0.01	-0.07	-0.07	1.00	1.00	-0.09	-0.02	-0.00	0.08
week_no	-0.00	-0.01	-0.01	0.00	0.01	-0.00	1.00	0.00	0.01	-0.02	1.00	1.00	-0.09	-0.09	1.00	0.95	-0.00	0.01
day_no	-0.00	0.00	-0.01	0.00	0.01	-0.00	0.94	0.00	0.01	-0.01	0.93	0.93	-0.02	-0.02	0.95	1.00	-0.00	-0.03
min_day	0.00	-0.00	0.01	-0.00	-0.01	-0.00	-0.00	-0.00	-0.01	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	1.00	-0.01
hr_day	0.00	-0.03	0.00	-0.02	0.00	0.02	0.01	-0.02	0.00	0.00	0.01	0.01	0.08	0.08	0.01	-0.03	-0.01	1.00

- We can see that there is some correlation between transaction day , month , year , week number ,which can be further analyzed using VIF.

Resampling The Data - Handling Imbalance



- **Under sampling** is basically downsizing (sampling) the majority class (normal class).
- **Oversampling** is basically making duplicates of the rare class (anomalous class) examples.
- **SMOTE** is basically used to generate artificial anomalies (not duplicates).

Model Building and evaluation

- We will start building the model with the train-test split. (At least 100 class 1 rows should be there in the test split), use the stratified split here. (80-20 ratio can be used) We need to find which ML model works good with the imbalance data and have better results on the test data.

- **Logistic regression** works best when the data is linearly separable and needs to be interpretable.

- **KNN** is also highly interpretable, but not preferred when we have a huge amount of data as it will consume a lot of computation.

- **The decision tree** model is the first choice when we want the output to be intuitive, but they tend to overfit if left unchecked.

- **Naive Bayes** algorithms are mostly used in sentiment analysis, spam filtering, recommendation systems etc. They are fast and easy to implement but their biggest disadvantage is that the requirement of predictors to be independent. In most of the real life cases, the predictors are dependent, this hinders the performance of the classifier.

Contd..

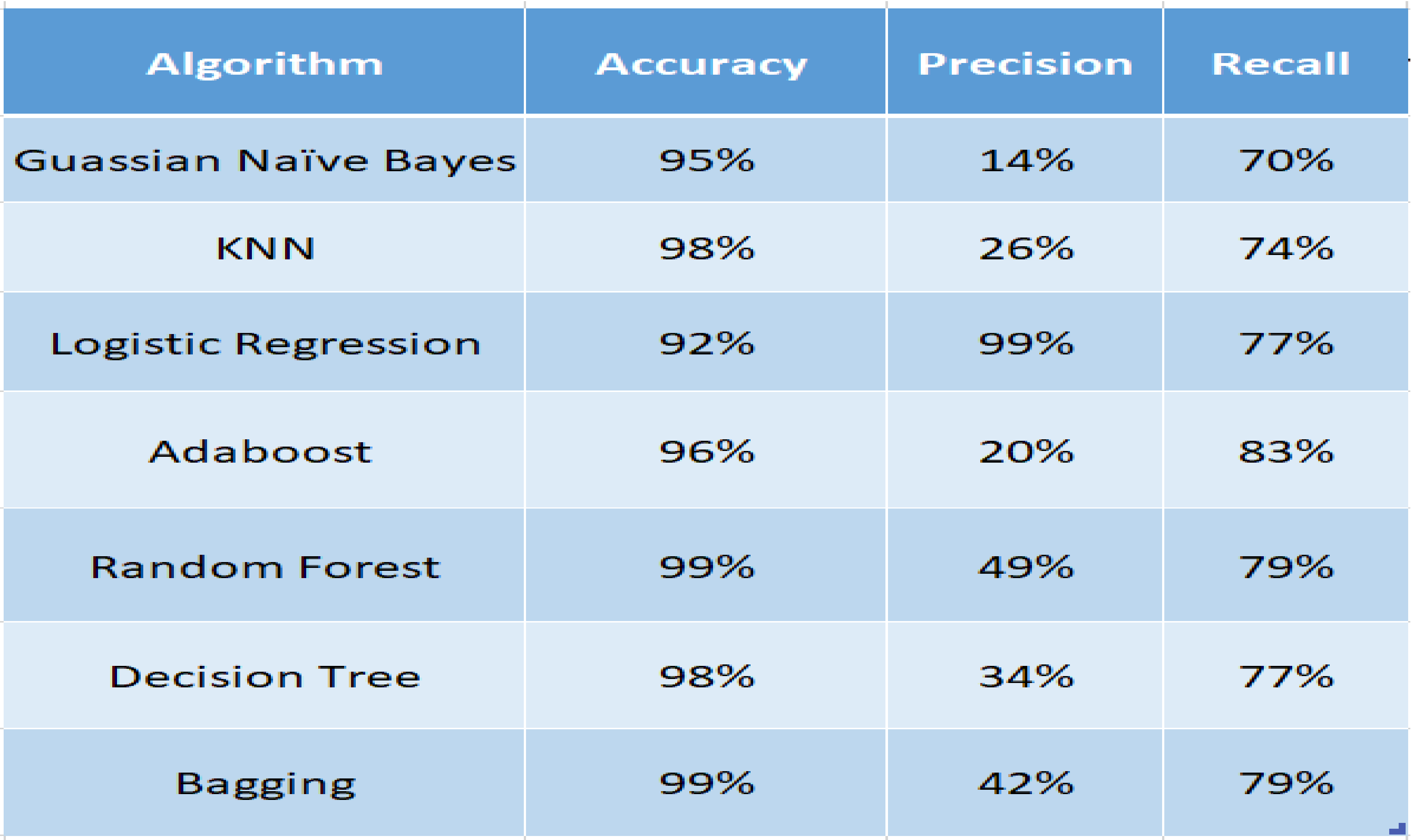
- **AdaBoost** can be used to **boost the performance of any machine learning algorithm**. It is best used with weak learners. These are models that achieve accuracy just above random chance on a classification problem. The most suited and therefore most common algorithm used with AdaBoost are decision trees with one level.

- Adaboost is **less prone to overfitting as the input parameters are not jointly optimized**. The accuracy of weak classifiers can be improved by using Adaboost.

- **Random forest** is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the **predictive accuracy and control over-fitting.**

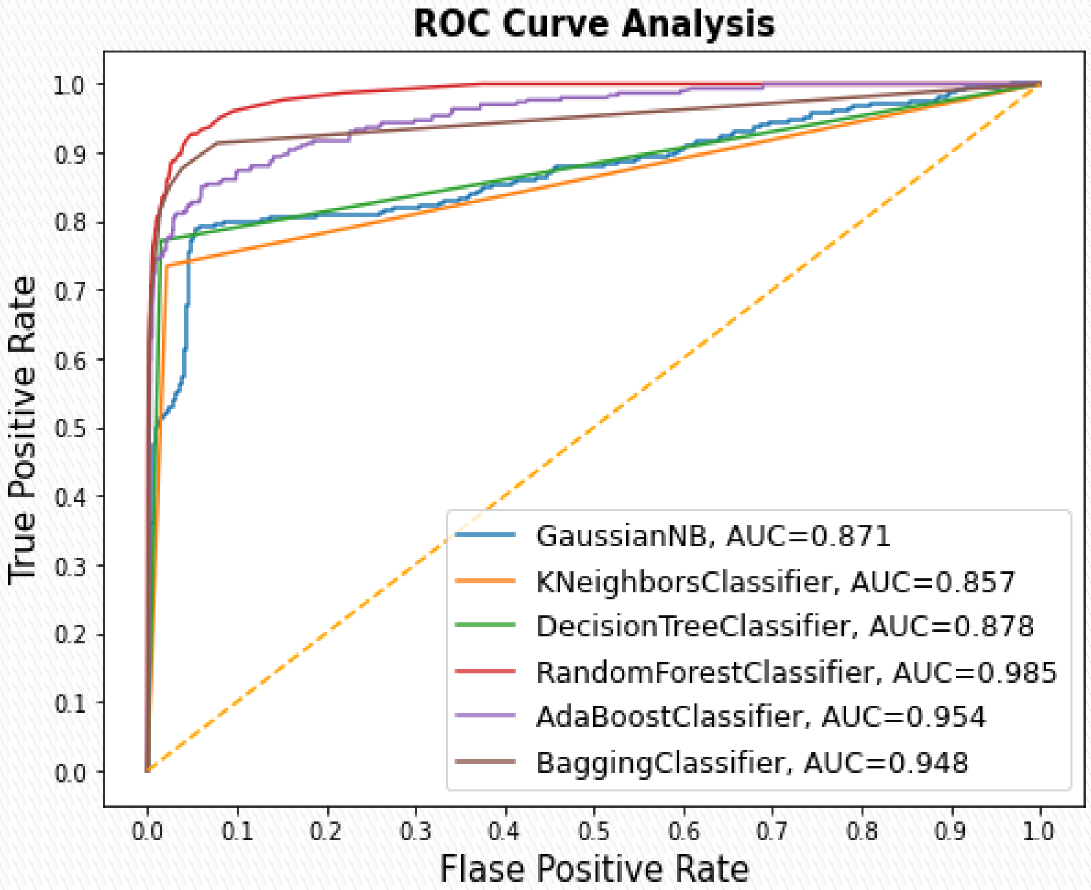
- The sub sample size is controlled with max_samples parameter if bootstrap=True , otherwise the whole dataset is used to build the tree

Evaluation



Algorithm	Accuracy	Precision	Recall
Guassian Naïve Bayes	95%	14%	70%
KNN	98%	26%	74%
Logistic Regression	92%	99%	77%
Adaboost	96%	20%	83%
Random Forest	99%	49%	79%
Decision Tree	98%	34%	77%
Bagging	99%	42%	79%

ROC Curve



- ROC curve is a trade off between True Positive Rate and False Positive Rate. A good ROC curve has a value close to 1. And our model’s value of ROC is 0.985 which is a very good value.

- After plotting the ROC curve we found that our Random forest model is performing better with respect to recall , precision and accuracy.

Important variables and their Impact

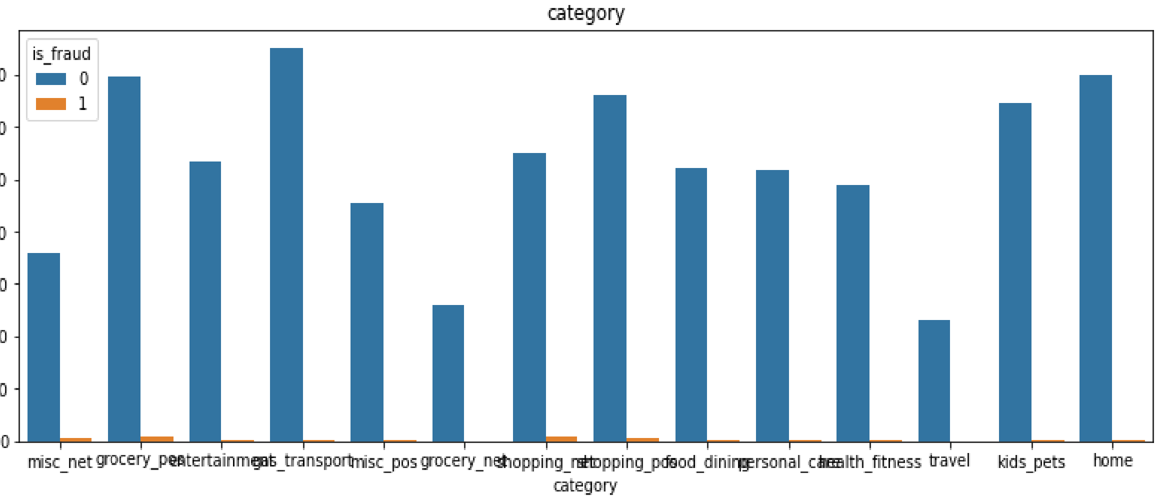
- Amount plays the most important role, since we can see that the mean amount in fraudulent transaction is higher than normal transaction.

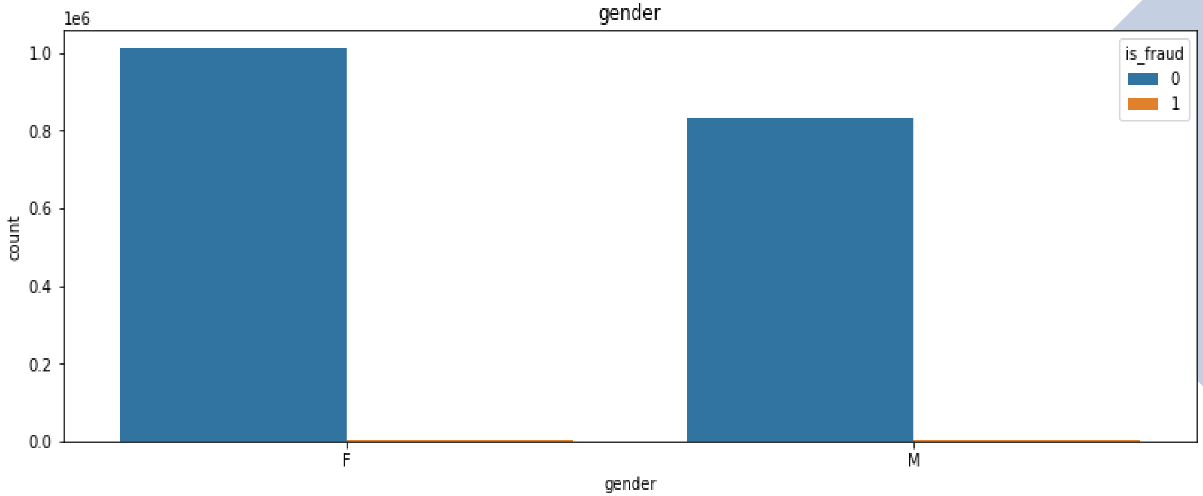
- Grocery plays important role since a lot of fraudulent transactions happens over there.

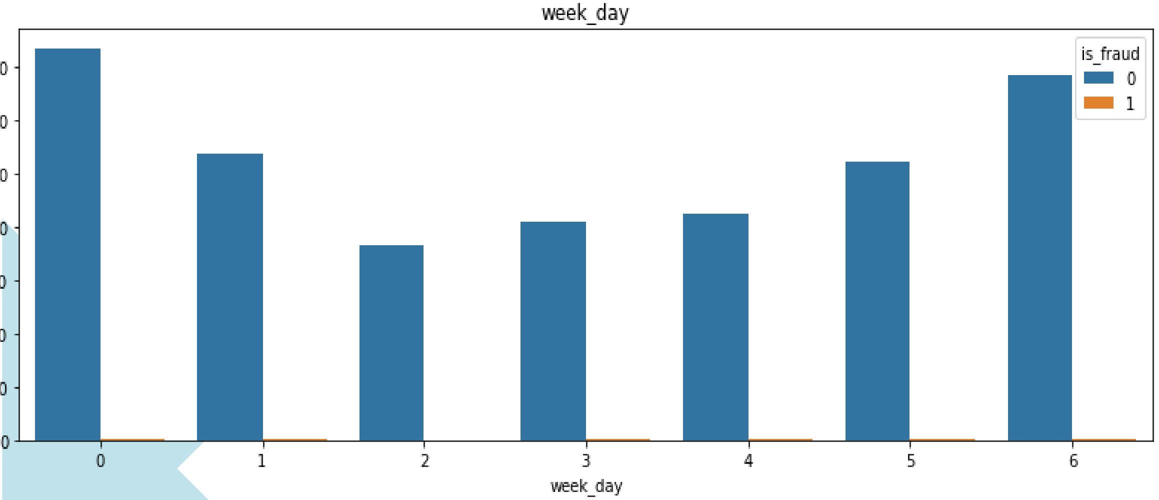
- Hour of the day and minute of the day are the very important factor because fraudsters commit the fraud during the time having more normal transactions.

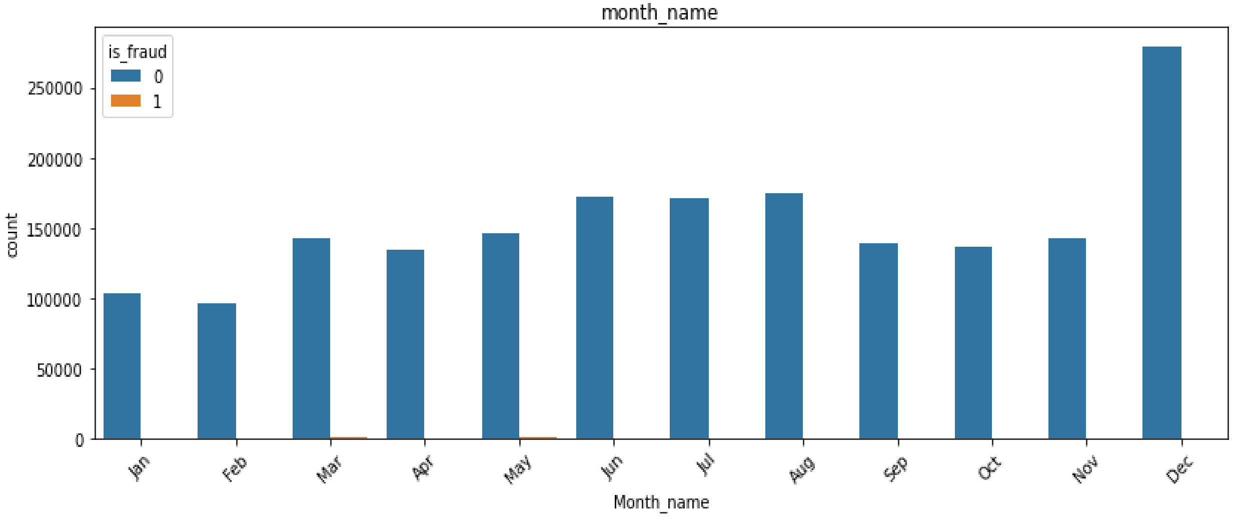
- Gender (male) plays important role since there are lot of male fraudsters as per the data.

Important variables and their Impact









Inferences:

- Accuracy cannot be counted on when dealing with unbalanced datasets since it cannot detect all fraudulent transactions.
- Recall says out of fraudulent transactions. what percentage were correctly identified. Recall is very effective in imbalanced data sets. We got best Recall score from DT model as 77%. Our model has correctly predicted 77% of all fraudulent transactions though RF predicted 79%.
- Precision says out of all fraudulent transactions predicted to be fraudulent, how many were actually fraudulent. Our Decision tree model gave 34% precision though Random forest model gave 49%.
- F1-Score is the weighted average of Precision and Recall and F1 score takes both false positives and false negatives into consideration and is very effective in imbalanced datasets. DT and RF precited 47% and 60% respectively.
- We got Area under curve in ROC as 98% in Random Forest model. Higher the AUC score better is the model at predicting fraudulent and non-fraudulent transactions.
- AUC of Precision Recall curve we got is 87% from Decision Tree model.

.

Business Impact:

- The imbalanced was taken and Data Processing was done. Data was scaled to remove any Skewness.
- Model was trained using normal data (unsampled/imbalanced) and trained again using sampled data.
- After training model, the trained algorithm was tested on Test data provided.
- I got the recall value of 0.79 which depicts the model correctly predicts 79% of fraudulent transactions.
- The cost incurred before model deployment is 426784.44 dollars per month and cost incurred after the model deployment is 30838.12 dollars and hence the final cost saved accounts to 395946 dollars after model Deployment.

.