Credit card fraud detection using Supervised Learning Classification

Gopi Selvaraj

Problem Definition

- The objective of this project is to develop a robust and accurate credit card fraud detection model using supervised learning classification techniques in the finance sector. The model should be able to classify transactions or activities into binary classes: "fraudulent" and "non-fraudulent" based on historical available data and the significant features.

- The dataset provided for this project contains historically available transaction data, where each transaction is described by a set of features. These features include customer information, transaction amount, transaction date and time, credit card information and other relevant details. Additionally, each transaction is classified as either "fraudulent" or "non-fraudulent”.

Data dictionary

S.No	Field name	Description	Data type
1	Accountnumber	Unique identifier for the account associated with the transaction.	Int64
2	Customerid	Unique identifier for the customer associated with the account.	Int64
3	Creditlimit	The credit limit assigned to the account.	Float64
4	Availablemoney	The available balance in the account.	Float64
5	Transactiondatetime	Date and time of the transaction.	Object
6	Transactionamount	The amount of money involved in the transaction.	Float64
7	Merchantname	Name of the merchant where the transaction took place.	Object
8	Acqcountry	Country where the acquiring bank is located.	Object
9	Merchantcountrycode	Country code of the merchant's location.	Object
10	Posentrymode	Point of service (POS) entry mode for the transaction.	Float64
11	Posconditioncode	Condition of the POS at the time of the transaction.	Float64
12	Merchantcategorycode	Code indicating the category of the merchant.	Object
13	Currentexpdate	Expiration date of the card at the time of the transaction.	Object
14	Accountopendate	Date when the account was opened.	Object
15	Dateoflastaddresschange	Date of the last address change on the account.	Object
16	Cardcvv	CVV (card verification value) of the card.	Int64
17	Enteredcvv	CVV entered during the transaction.	Int64
18	Cardlast4digits	Last 4 digits of the card number.	Int64
19	Transactiontype	Type of the transaction (e.G., Purchase, cash advance).	Object
20	Echobuffer	An echo buffer associated with the transaction.	Float64
21	Currentbalance	Current balance in the account.	Float64
22	Merchantcity	City where the merchant is located.	Float64
23	Merchantstate	State where the merchant is located.	Float64
24	Merchantzip	ZIP code of the merchant's location.	Float64
25	Cardpresent	Indicator whether the card was present during the transaction.	Bool
26	Posonpremises	Indicator whether the POS was on the merchant's premises.	Float64
27	Recurringauthind	Indicator for recurring authorization.	Float64
28	Expirationdatekeyinmatch	Indicator whether the expiration date matches during key-in transactions.	Bool
29	Isfraud	Indicator whether the transaction is fraudulent.	Bool

Shape and distribution of the target variable

- The dataset has **786363** observation & **29** variables. The complexity is in finding the solution to the problem based on the chosen sampling techniques.

  (**Stratified sampling & Smote Sampling**)

- Due to the huge size of the dataset, have opted for 'Stratified sampling' considering the original dataset as population and the sample dataset contains **150000** observation & **29** variables.

- When calculating the distribution of the target variable, we can see the Majority class is non-fraudulent transactions / 'False' & Minority class is fraudulent transactions / 'True' with ratio of **98.5 : 1.5**

- **Python Version:**

  '3.11.5 | packaged by Anaconda, Inc.

Dropping Variables

- Dropping columns with 100% null values, which includes

echoBuffer	merchantCity
merchantState	merchantZip
posOnPremises	recurringAuthInd

- Excluding the following variables from the dataset, as they are irrelevant to the target variable and further may introduce unwanted confusion during model building:

accountNumber	customerId
transactionDateTime	currentExpDate
accountOpenDate	dateOfLastAddressChange
merchantCountryCode	cardLast4Digits
cardCVV	enteredCVV

Missing Value Imputation

- Due to high imbalance sub-classes in the missing value categorical columns

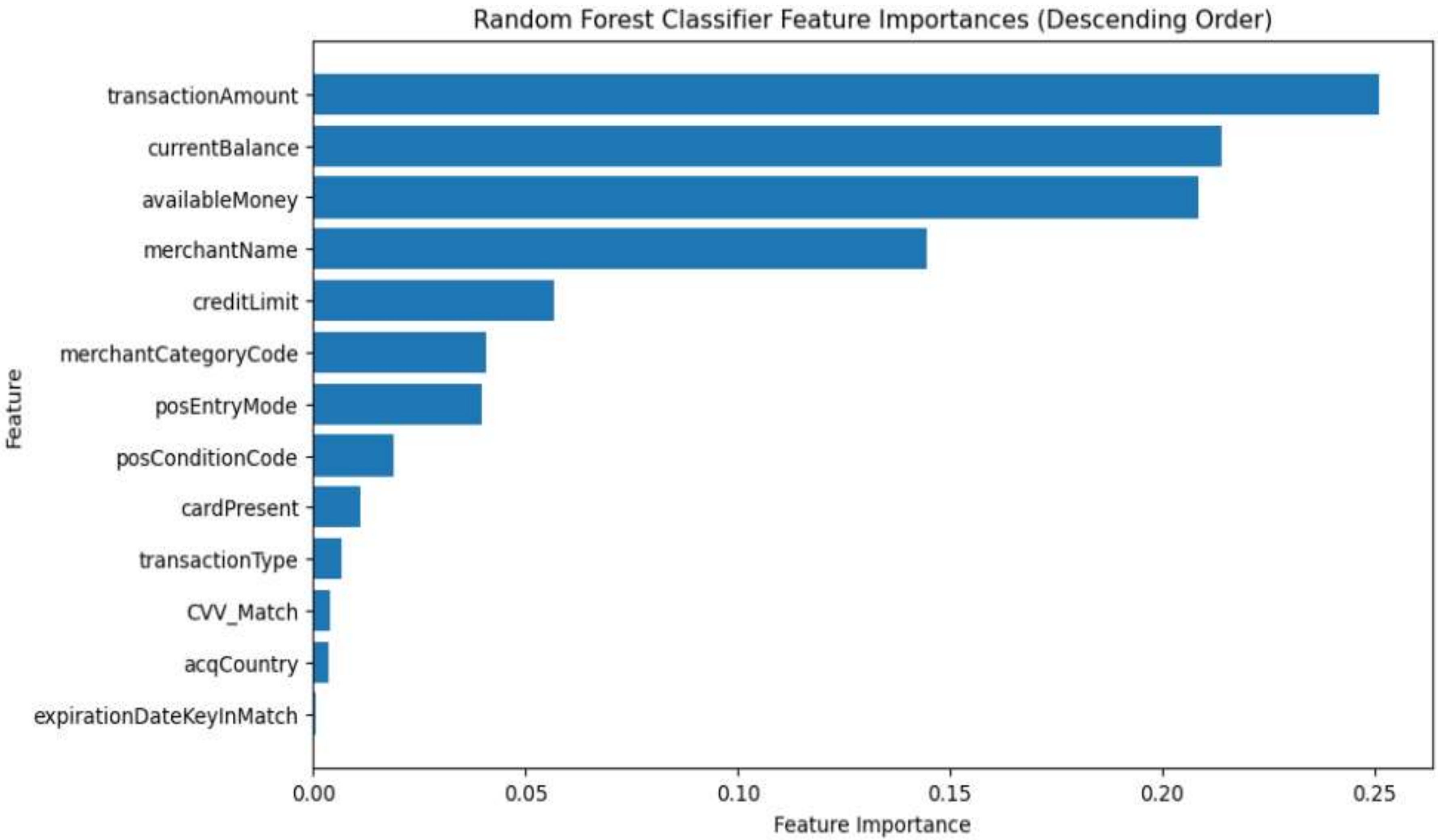
```
acqCountry          merchantCountryCode
US      147783      US      148517
MEX        600      MEX        604
CAN        441      CAN        442
PR         285      PR         287

posEntryMode        posConditionCode
5.0      60295      1.0      119964
9.0      45122      8.0       28540
2.0      37158      99.0       1403
90.0      3745      Name: count, dtype: int64
80.0      2908      transactionType
                    PURCHASE                 142015
                    ADDRESS_VERIFICATION       3951
                    REVERSAL                   3911
```

instead of mode imputation, we have opted for either ‘backward fill’ / ‘forward fill’ to avoid bias towards the most repeated subclass.

Outlier Treatment:

- Even though we can see huge no. of outliers in the numeric columns, have refrained from removing any outliers in the dataset, given its financial nature.

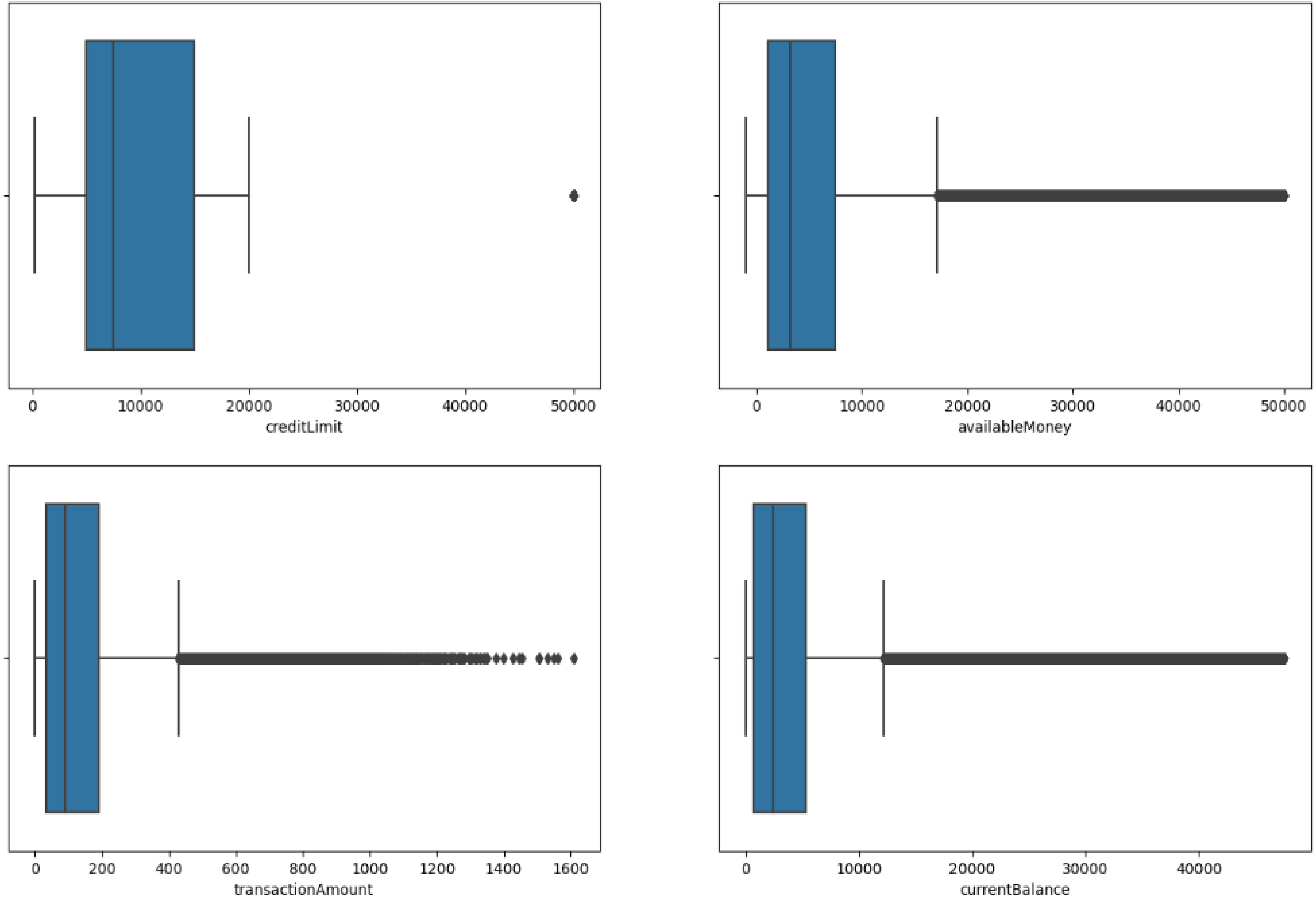
Feature Importance using Random Forest Classification:



Exploratory Data Analysis

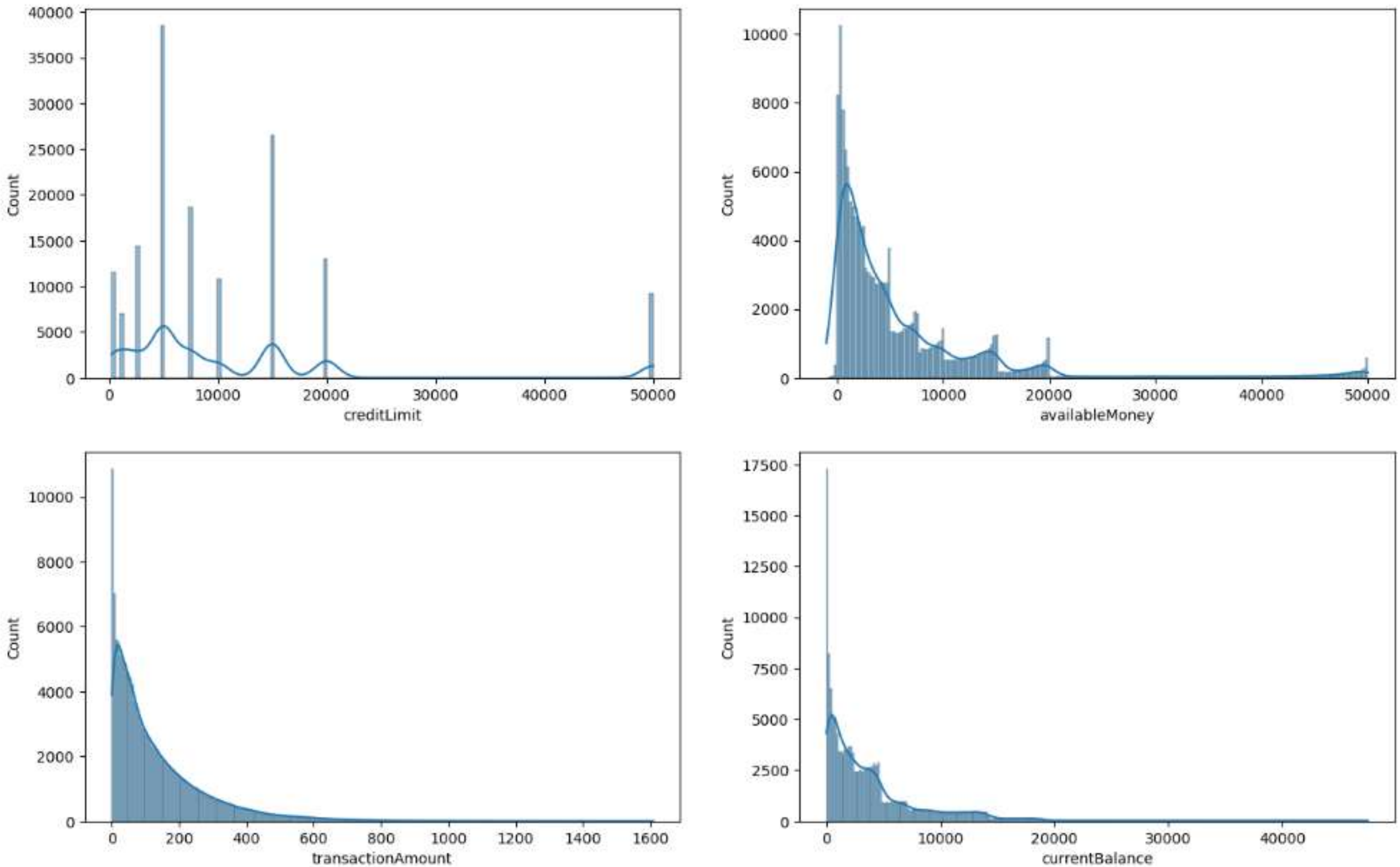
Univariate analysis - Numeric variables

Boxplot:



Numeric variables

Histogram:



<u>Inference</u>:

- Highly skewed(mean>median>mode) data in the numeric variables indicating presence of outliers in the higher scale in all numeric variables.

'creditLimit'

- High positive skewness(2.2) indicating presence of outliers in higher scale.
- 50% of the 'credit limit' values lie between 5000 and 15000.
- We can see outlier value of 50000

'availableMoney'

- High positive skewness(3) indicating presence of outliers in higher scale & we can see huge no. of outliers.
- 50% of the avaiableMoney values lie between 1079 and 7500
- We can see outlier value of 50000 (we have also seen the outlier value in 'creditLimit' as well)

'transactionAmount'

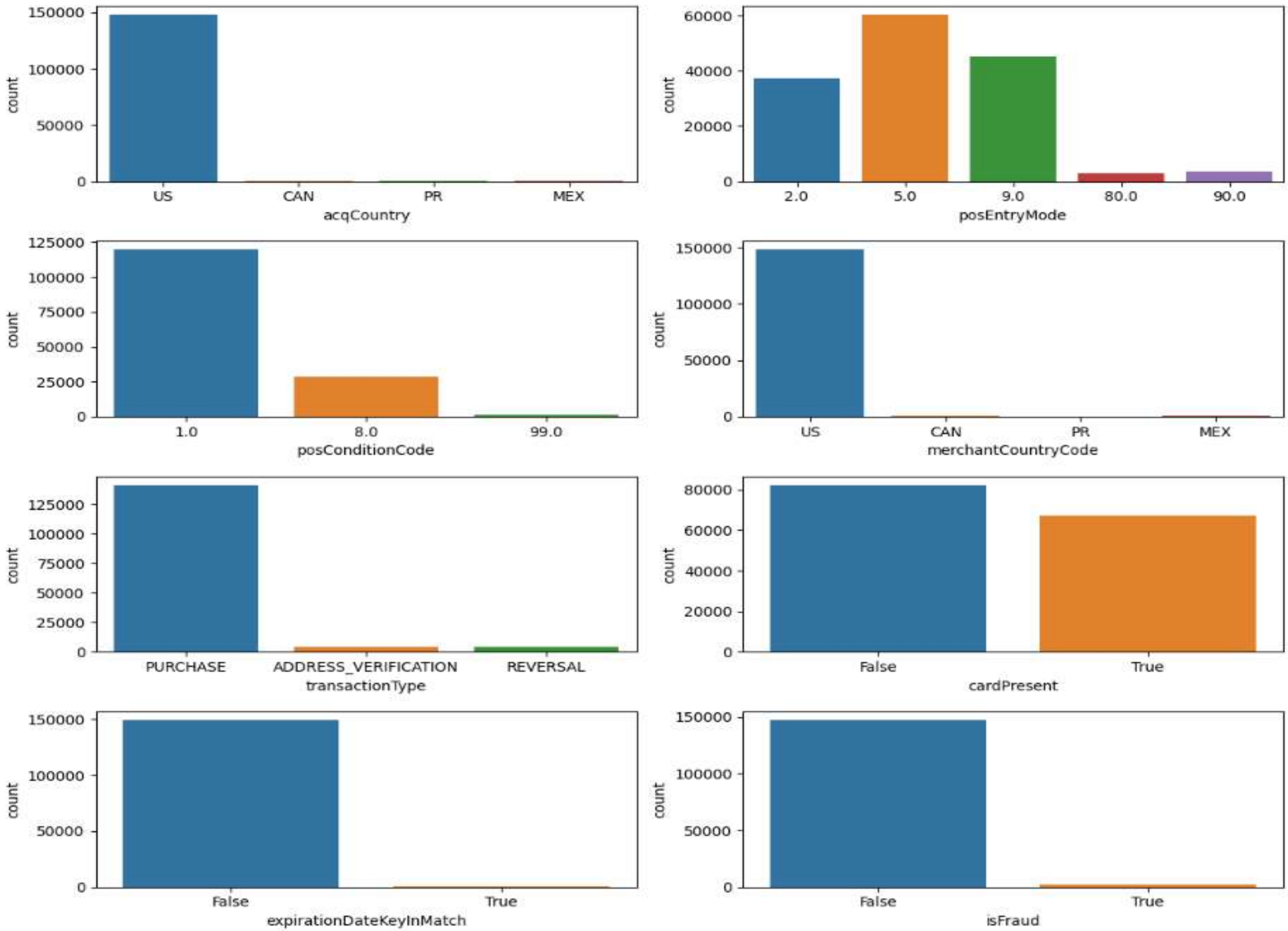
- High positive skewness(2.11) indicating presence of outliers in higher scale & we can see huge no. of outliers.
- 50% of the avaiableMoney values lie between 33.6 and 191.4 indicating most of the transactions are of low value
- We can most outliers lie around 450 too 1300
- We can see extreme outlier value of 1608.3

'currentBalance'

- High positive skewness(3.3) indicating presence of outliers in higher scale & we can see huge no. of outliers
- 50% of the 'currentBalance' values lie between 700.3 and 5291
- Most of the high credit limit customers have not spend most of the money from their credit card.
- We can see an extreme outlier value of 47498.8

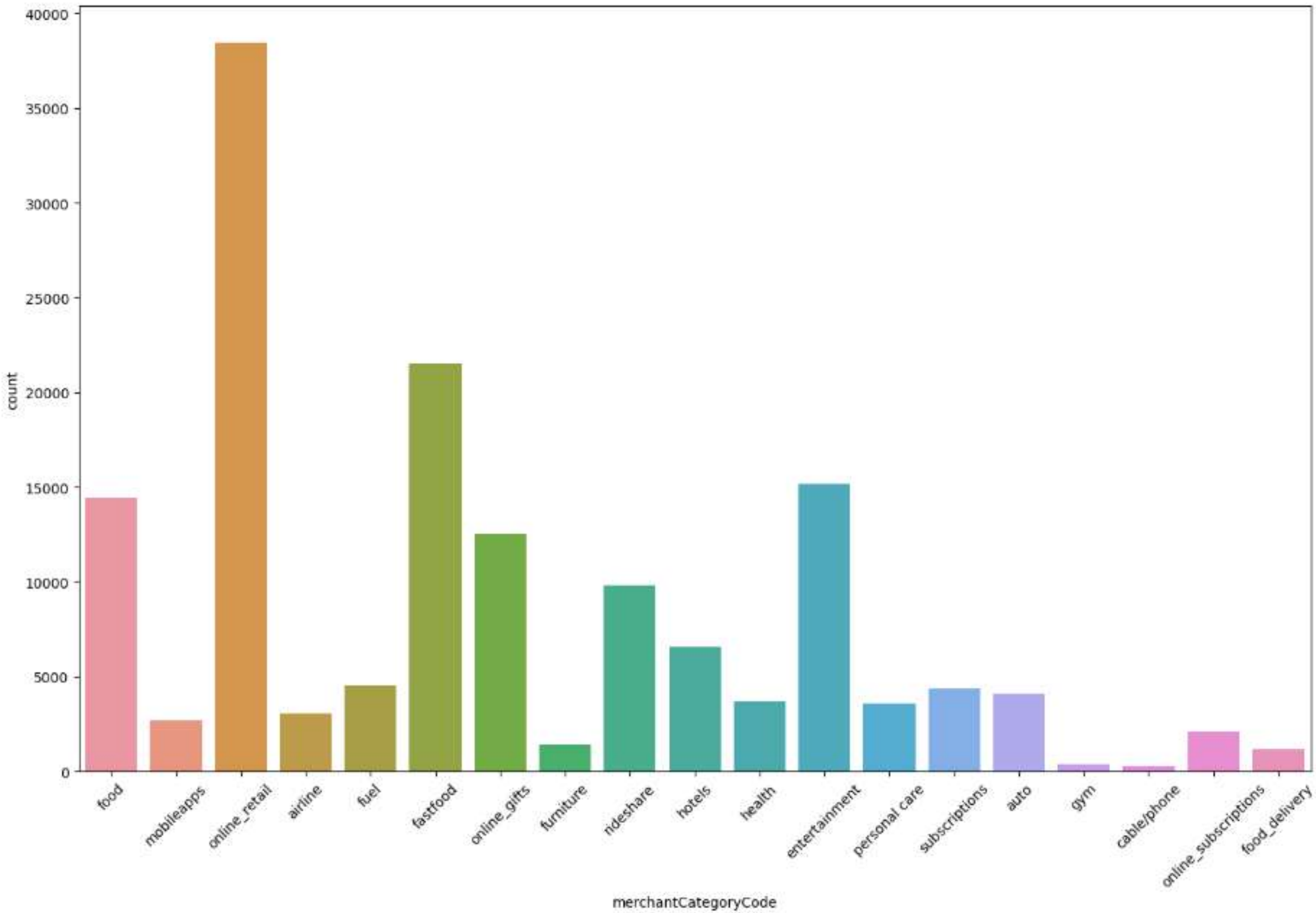
Univariate analysis - Categorical variables

Countplot:



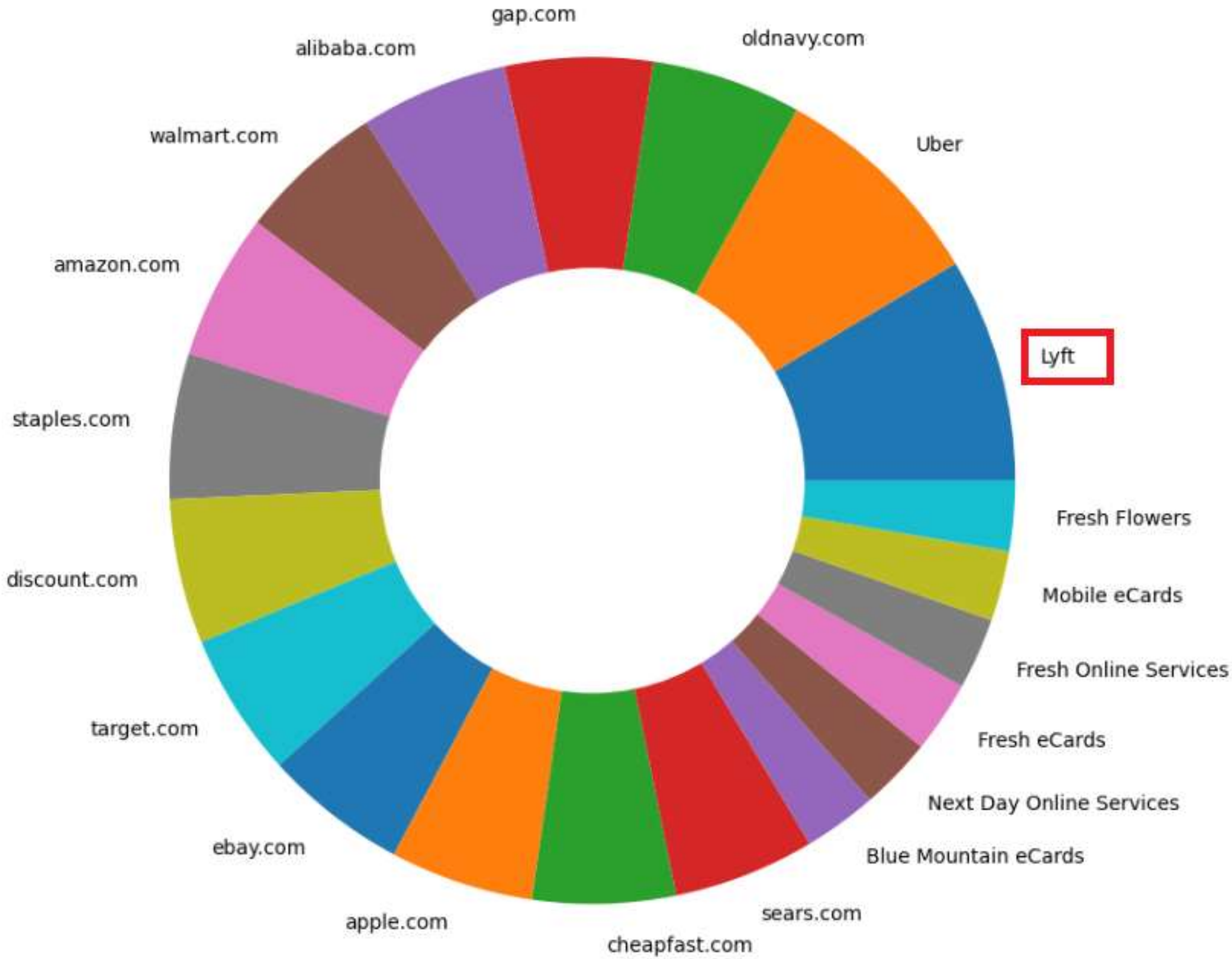
Univariate analysis - Categorical variables

Countplot:



Univariate analysis Categorical variables

Pie-plot:



Inference:

'acqCountry'

- No. of subclasses = 4 & high imbalance in distribution of subclasses.
- Most of the merchants(148286) are from country 'US'.
- We can find data imbalance within subclasses are very few no. of values belong to 'MEX' , 'CAN' & 'PR'.
- Least no. of merchants(290) belong to country Puerto Rico/'PR’.

'merchantCountryCode'

- No. of subclasses = 4 & high imbalance in distribution of subclasses.
- Most of the merchants(148285) belong to country code 'US'.
- We can find data imbalance within subclasses are very few no. of values belong to 'MEX' , 'CAN' & 'PR'.
- Least no. of merchants(288) belong to country code Puerto Rico/'PR'.
- We can observe the distribution of values in 'acqCountry' & 'merchantCountryCode' is very similar. Both columns might be representing the country in which the transaction happened. The small difference in distribution of subclasses might be due to data collection/entry.

  Instead of having both of them in the final model its best we have only one column (Eg.'acqCountry’)

'posEntryMode'

- No. of subclasses = 5 . Values are distributed generally in 5.0, 9.0 & 2.0. Fewer of values in 90.0 & 80.0
- Most of the transactions(60452) used '5'/'PAN auto-entry via chip' followed by transaction(62268) using '9'/'PAN entry via electronic commerce, including remote chip'.
- Least no. of transaction(2907) were from 80. This might happen if there's a problem with the chip reader/ chip on the card is damaged/ if there's some other technical issue. Indicating compromise in security.

'posConditionCode'

- No. of subclasses = 3 & high imbalance in distribution of subclasses.
- Most of the transactions(119734) were successful belonging to '1' followed by voided transactions(28482) belonging to '8'.
- Least no. of transactions(1396) were belonged to 99 indicating refund of money for that transaction.

'transactionType'

- No. of subclasses = 3 & high imbalance in distribution of subclasses.
- Most of the transactions(141755) belonged to type 'PURCHASE'.
- Least no. of transaction(3912) belonged to type 'REVERSAL'.

'cardPresent'

- No. of subclasses = 2 & values seems to be equally distributed among subclasses.
- In most of the transactions(82486) the credit card was not physically present.
- In least of the transactions(67126) the credit card not physically present.

'expirationDateKeyInMatch'

- No. of subclasses = 2 & high imbalance in distribution of subclasses.
- In most of the transactions(149403) there wasn't a match between the actual expiration date and the expiry date entered by the customer.
- Least of the transactions(209) there was a match between the actual expiration date and the expiry date entered by the customer.

'isFraud'

- No. of subclasses = 2 & high imbalance in distribution of subclasses.
- We can see high class imbalance in target variable.
- Most of the transactions(188288) were not fraud & least no. of transactions(2368) were fraud.

'merchantCategoryCode'

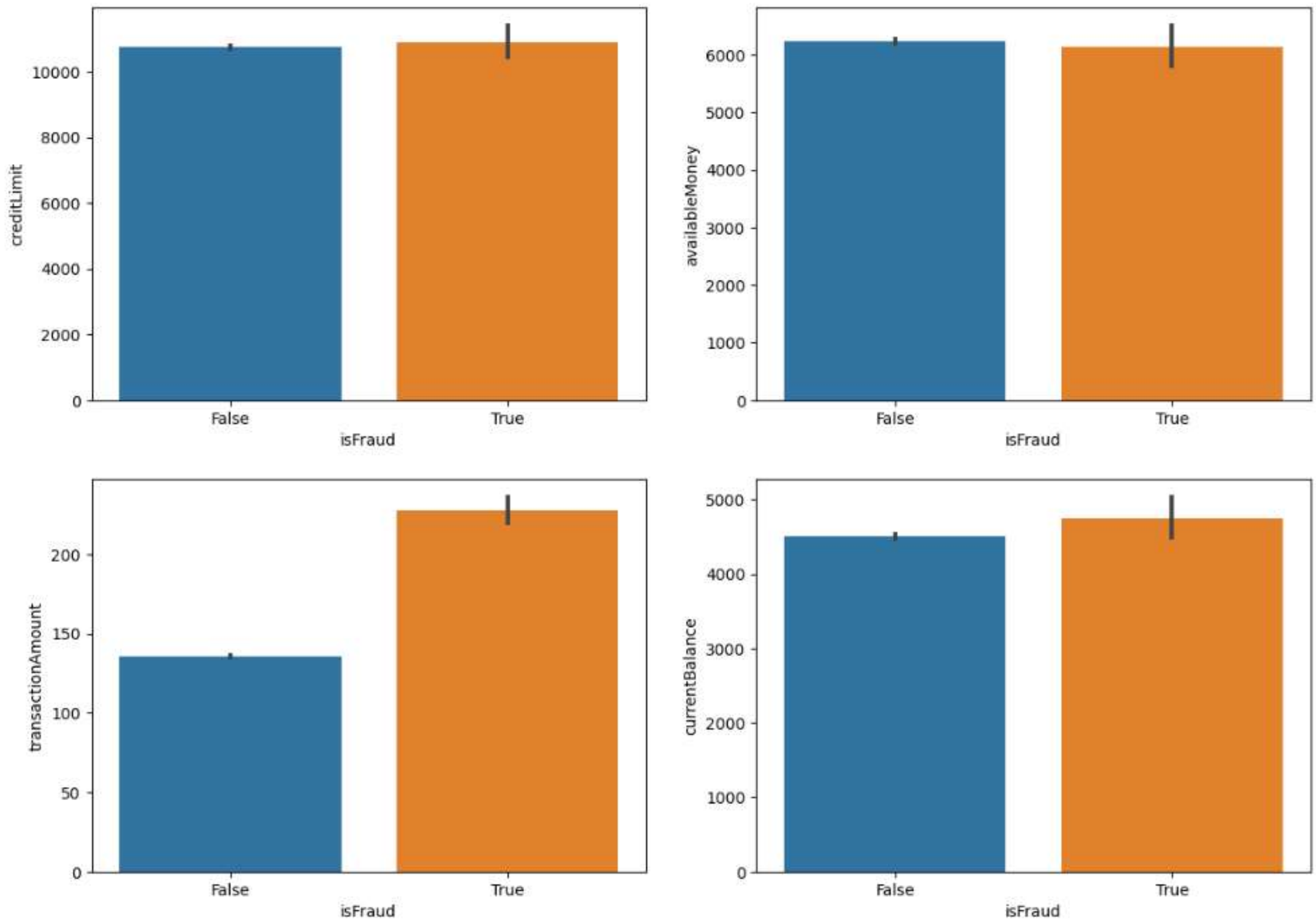
- No. of subclasses = 19 & high imbalance in distribution of subclasses.
- Most of the transactions(38468) were from 'online_retail' transactions.
- Least no. of transactinos(269) were from 'cable/phone' transactions.

'merchantName'

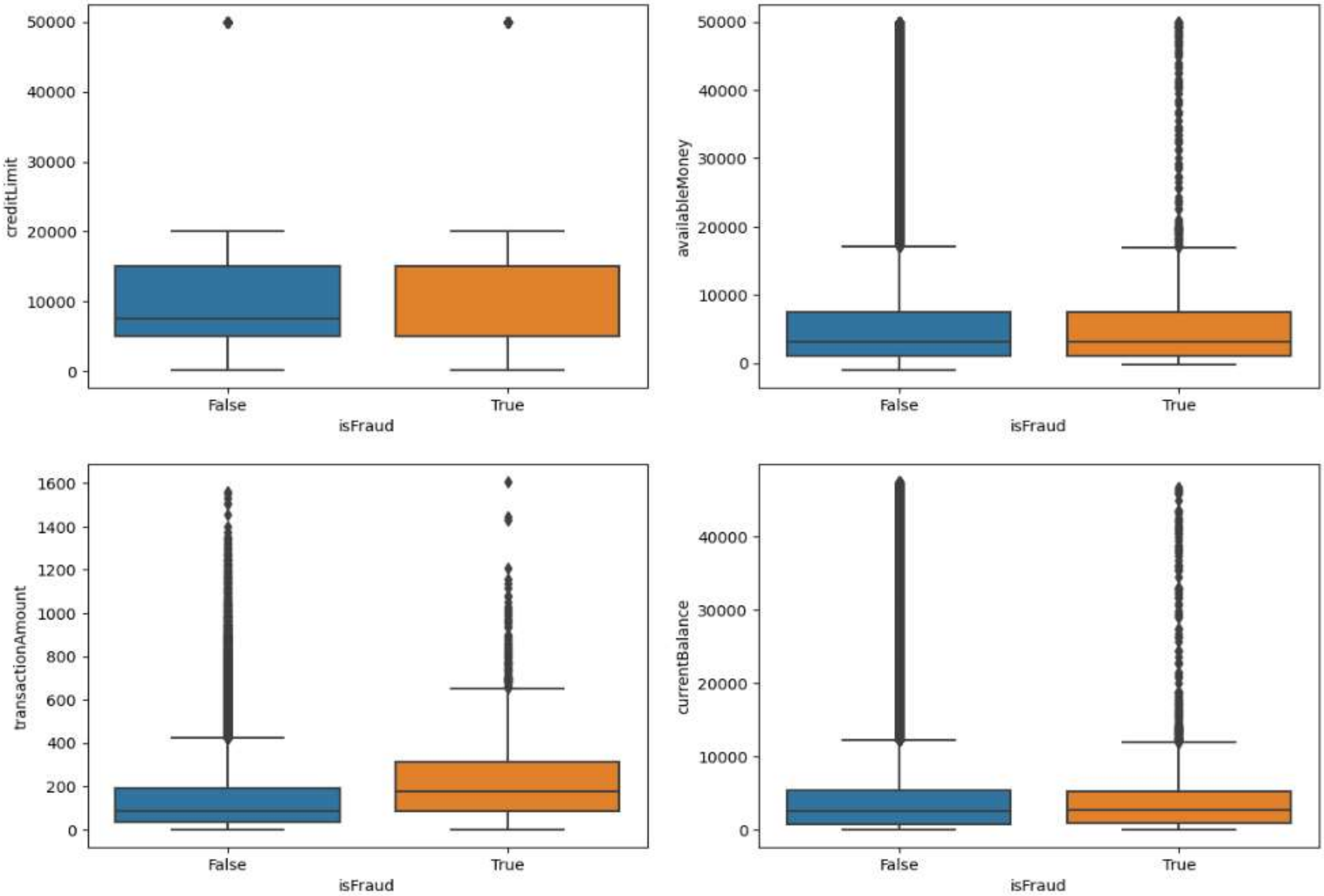
- Most of the transactions(6785) are from 'Lyft'.
- Followed by 6612 transactions from 'Uber'.

Bivariate analysis – Numerical VS Categorical

<u>Barplot</u>



Boxplot



creditLimit
availableMoney
transactionAmount
currentBalance
isFraud
False
True

Inference:

'creditLimit' vs ‘isFraud’

- We can there is no significant difference in the distribution of ‘creditLimit’ for fraudulent and non-fraudulent transactions.
- 50% of the data of ‘creditLimit’ for the fraudulent transactions & non-fraudulent transactions lie around 5000 to 15000.

'availableMoney' vs ‘isFraud’

- We can there is no significant difference in the distribution of ‘availableMoney’ for fraudulent and non-fraudulent transactions.
- 50% of the data of ‘availableMoney’ for the fraudulent transactions lies around 1057.7 and 7425.1
- 50% of the data of ‘availableMoney’ for the non fraudulent fraudulent transactions lies around 1079.8 and 7500
- There are huge no. of outliers in both categories.

'transactionAmount' vs ‘isFraud’

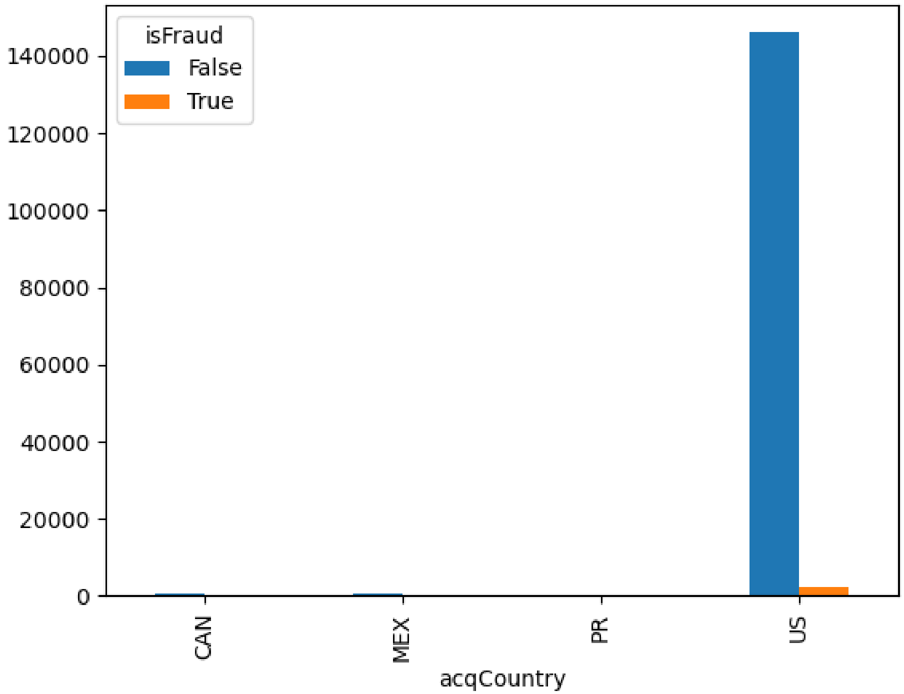
- We can there is a significant difference in the distribution of ‘transactionAmount’ for fraudulent and non-fraudulent transactions.
- 50% of the data of ‘transactionAmount’ for the fraudulent transactions lies around 86.6 and 313.3
- 50% of the data of ‘transactionAmount’ for the non fraudulent fraudulent transactions lies around 33.1 and 189.
- There are huge no. of outliers in both categories.

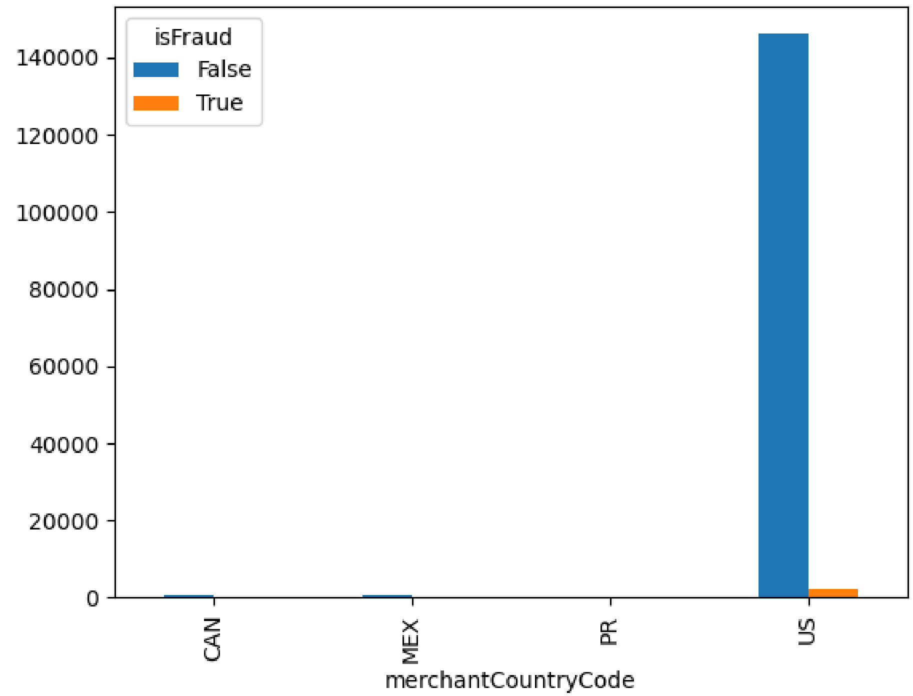
'currentBalance' vs ‘isFraud’

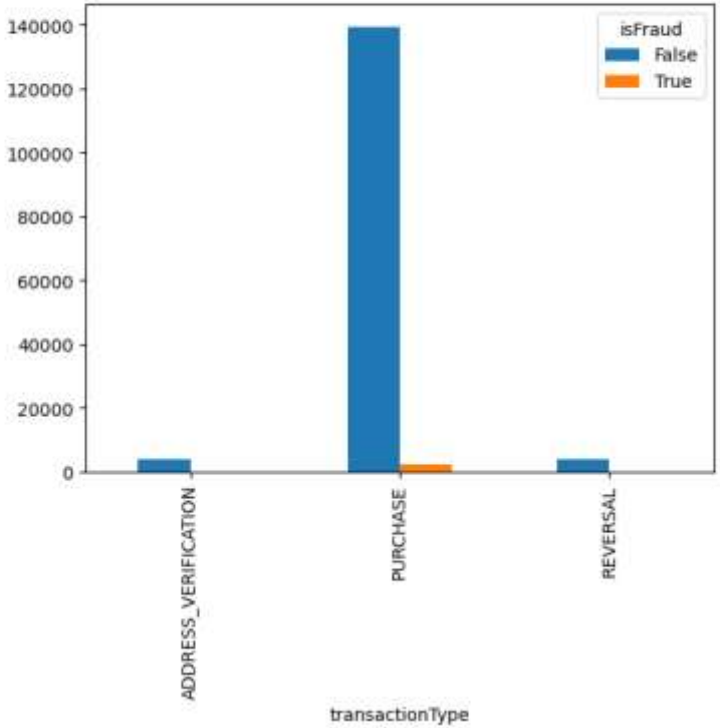
- We can there is a very small variation in the distribution of ‘currentBalance’ for fraudulent and non-fraudulent transactions.
- 50% of the data of ‘currentBalance’ for the fraudulent transactions lies around 794.2 and 5215.8
- 50% of the data of ‘currentBalance’ for the non fraudulent fraudulent transactions lies around 699.205 and 5292.615
- There are huge no. of outliers in both categories.

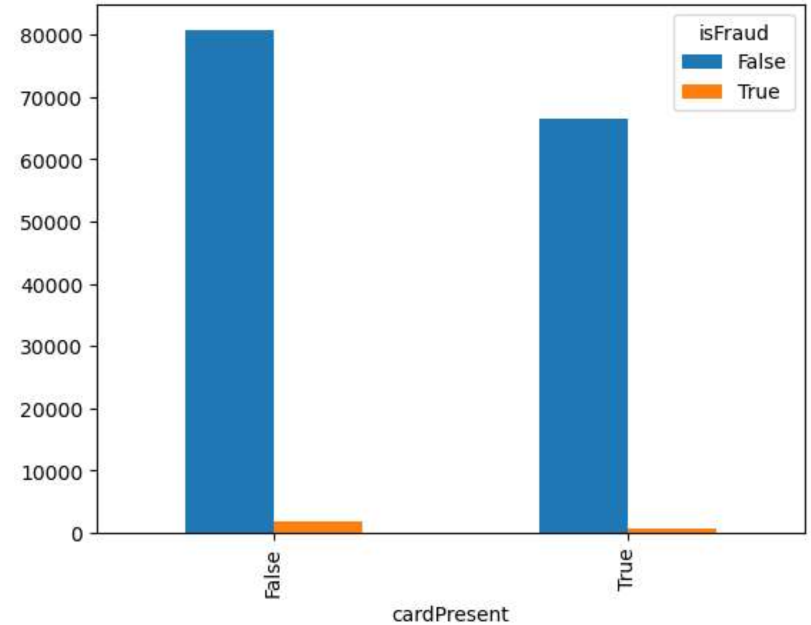
Bivariate analysis – Categorical VS Categorical

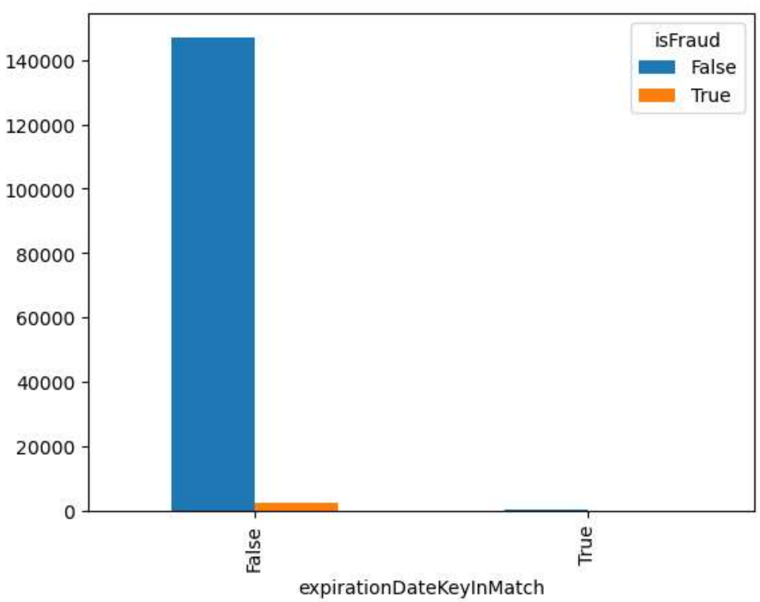
Crosstab

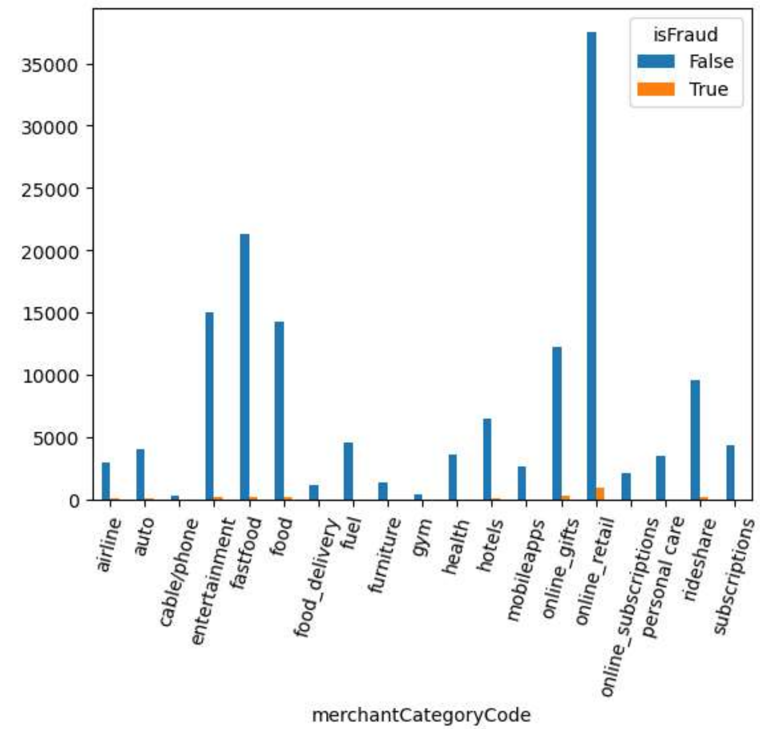





isFraud
False
True
ADDRESS_VERIFICATION
PURCHASE
REVERSAL
transactionType


isFraud
False
True
False
True
cardPresent


isFraud
False
True
False
True
expirationDateKeyInMatch


isFraud
False
True
airline
auto
cable/phone
entertainment
fastfood
food
food_delivery
fuel
furniture
gym
health
hotels
mobileapps
online_gifts
online_retail
online_subscriptions
personal care
rideshare
subscriptions
merchantCategoryCode

Inference:

'acqCountry' vs ‘isFraud’

- Most fraudulent transactions(2342) are from ‘US’ followed by ‘MEX’/Mexico with 14 transactions & the least no. of fraudulent transactions(25) are from ‘PR’/Puerto-Rico.
- Most non fraudulent transactions(145944) are from ‘US’ followed by ‘MEX’/Mexico with 579 transactions & the least no. of non fraudulent transactions(285) are from ‘PR’/Puerto-Rico.

'merchantCountryCode' vs ‘isFraud’

- Most fraudulent transactions(2341) are from merchant country code ‘US’ followed by ‘Mex’ with 14 transactions & the least no. of fraudulent transactions(5) are from ‘PR’/Puerto-Rico.
- Most non fraudulent transactions(145944) are from merchant country code ‘US’ followed by ‘Mex’ with 583 transactions & the least no. of non fraudulent transactions(283) are from ‘PR’/Puerto-Rico.

'transactionType' vs ‘isFraud’

- Most fraudulent transactions(2285) are belong to purchase type ‘PURCHASE’ & the least no. of fraudulent transactions(25) are from ‘ADDRESS_VERIFICATION’.
- Most non fraudulent transactions(139470) are belong to purchase type ‘PURCHASE’ & the least no. of non fraudulent transactions(3854) are from ‘REVERSAL’.

'cardPresent' vs ‘isFraud’

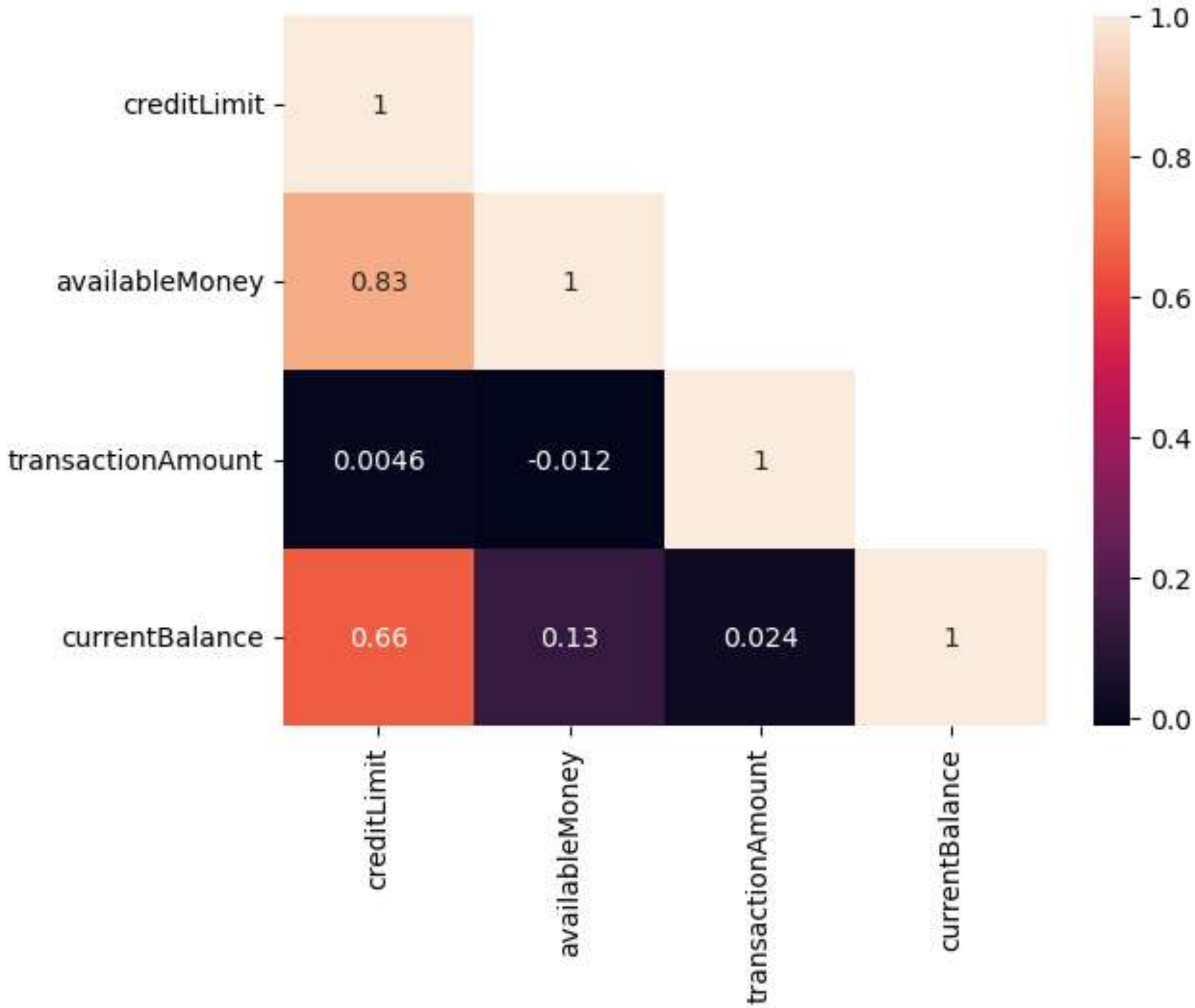
- Most fraudulent transactions(1717) are where the card was physically not present & the least no. of fraudulent transactions(651) are where the card was physically present.
- Most non fraudulent transactions(80769) are where the card was physically not present & the least no. of non fraudulent transactions(66475) are where the card was physically present.

'expirationDateKeyInMatch' vs ‘isFraud’

- Most fraudulent transactions(2364) are where the expiration date didn’t match with the one entered by the customer & the least no. of fraudulent transactions(4) are where the expiration date matched with the one entered by the customer.
- Most non fraudulent transactions(147039) are where the expiration date didn’t match with the one entered by the customer & the least no. of non fraudulent transactions(205) are where the expiration date matched with the one entered by the customer.

Multivariate analysis

Correlation plot



Inference:

- We can infer the presence of Multi-collinearity from the above plot.
- We can see high positive correlation(0.83) between 'creditLimit' & 'availableMoney'.
- We can see high positive correlation(0.66) between 'creditLimit' & 'currentBalance'.
- We can see very weak positive correlation(0.13) between 'currentBalance' & 'availableMoney'.

Statistical test of significance:

Numeric VS Categorical – Independent T test
Categorical VS Categorical – Chi square contingency test

	p_value	Relationship exists(p_value<0.05)
creditLimit	0.52	No
availableMoney	0.61	No
transactionAmount	5.51988874420541e-201	Yes
currentBalance	0.066197	No
merchantName	0.02	Yes
acqCountry	0.5	No
merchantCountryCode	0.48	No
posEntryMode	2.466790695981825e-85	Yes
posConditionCode	1.647234284880733e-05	Yes
merchantCategoryCode	1.1225135028669137e-142	Yes
transactionType	6.6658021229242185e-06	Yes
cardPresent	1.1392665521580453e-65	Yes
expirationDateKeyInMatch	0.91	No
CCV_Match	5.386291691519736e-06	Yes

Feature engineering:

- We have two features ''cardCVV' & 'enteredCVV' which represent the actual credit card CVV and the one entered by the customer. We can't infer information from them as individual columns when they exist separately.
- Hence we can create a new column 'CVV_match' that contains records of correct(1) or wrong(0) CVV entered by customer.

Data preprocessing

Scaling the data

- Scaling is performed in the dataset for the numeric variables after train test split as we can hide the mean standard deviation of training data from the test data.

Outlier treatment:

- The popular methods of handling the outliers are Trimming/removing the outlier based on IQR or z-Score or capping them. We would like to consider the outliers in this dataset as a pattern and prefer not to treat them but to handle them differently since it is important for the model to get trained based on some extreme values in order to predict in an efficient way consider their nature in the financial sector.

Transformation technique:

- We prefer transformation of the numeric variables(using Yeo-Johnson transformation) so that we can try to convert them to near normal distribution instead of opting for any outlier treatment.

Encoding the Categorical Variables:

- The performance of a machine learning model not only depends on the model and the hyperparameters but also on how we process and feed different types of variables to the model. Since most machine learning models only accept numerical variables, pre-processing the categorical variables becomes a necessary step. Converting these categorical variables to numbers such that the model is able to understand and extract valuable information is known as Encoding. There are various encoding techniques available like Dummies, One Hot Encoder, Label Encoder, Ordinal Encoder etc., We have used Label Encoder for encoding our categorical variables considering the no. of categorical variables and we have also seen there is not a significant improvement in model performance when dummy encoding was utilized.

- We use Label encoding technique when the categorical feature is ordinal. In this case, retaining the order is important. Hence encoding should reflect the sequence. In Label encoding, each label is converted into an integer value. We will create a variable that contains the categories representing the education qualification of a person likewise.

- The final processed data which we would be using in building various Classification Models like Logistic Regression, Decision Tree, Random Forest etc. We, with the help of the built models we would infer on the significance and effects of each independent variable on our target variable for predicting the patterns and rate of successful conversion to give some insightful ideas for effective marketing.

Model Building & Evaluation

- The Modeling is the core of any machine learning project. This step is responsible for the results that should satisfy or help satisfied the project goals.

- Building a model in machine learning is creating a mathematical representation by generalizing and learning from training data.

- Our problem statement come under the Classification thus we have decided to use various models namely

1. Logistic Regression Model
2. Decision Tree Model
3. Random Forest Model
4. Naïve Bayes Model
5. Ada Boost Technique
6. Gradient Boosting Technique
7. eXtreme Gradient Boosting Technique

- **Evaluation**: Recall, Precision, Accuracy & F1 Score (wieghted)

Stratified Sampling & SMOTE Techniques

- Since our Data Set is Very Large classification data is an imbalanced data, it is desirable to sample the dataset into Sampling Datasets in a way that preserves the same proportions of examples in each class as observed in the Original Dataset. **This is called a Stratified Sampling**.
- We can achieve this by setting the **"Stratify"** argument to the **Merchant Category Code** component of the original dataset. & we Have **150000** Rows after Sampling
- Once the Sample is taken from various random state of 4 this Sample will be used for the various model with various combination, various ensemble and Stacking Techniques
- Smote Has Been Done on Train Data Only and Test is Passed for Evaluation Metrics. Although we don't prefer SMOTE as its not representative of the real world data.

Population Target Variable Proportion		**Sample Target Variable Proportion**	
False	98.42096	False	98.42096
True	1.579042	True	1.579042

Value Counts of Train and Test
Target Variable Value Counts Train on SMOTE: 235590
Target Variable Value Counts Test: 29923

Hyper tuning the model

Model: Decision Tree Classifier Without SMOTE

Parameter Used:

GridSearchCV

Criterion	**Entropy, Gini**
Max Depth	30, 35, 40
Min Samples Split	50, 60, 70, 80, 90
Min Samples Leaf	10, 20, 30, 40

Best Parameters

Criterion	Entropy
Max Depth	30
Min Samples Split	80
Min Samples Leaf	10

Model performance - Training scores

Model	Accuracy	Recall	Precision	F1 Score	TN	FN	FP	TP
Logistic Regression [Base model] - Train	0.984176	0.984176	0.984426	0.976327	117795	1894	0	0
Decision Tree [base model] - Train	1.000000	1.000000	1.000000	1.000000	117795	0	0	1894
Decision Tree [significant columns] - Train	0.997067	0.997067	0.997070	0.996923	117792	348	3	1546
Random forest [significant columns] - Train	0.996992	0.996992	0.996972	0.996852	117779	344	16	1550
Adaboost [significant columns] - Train	0.997067	0.997067	0.997065	0.996925	117789	345	6	1549
Gradientboost [significant columns] - Train	0.984318	0.984318	0.982064	0.976743	117791	1873	4	21
XGboost [significant columns] - Train	0.984176	0.984176	0.984426	0.976327	117795	1894	0	0
Gaussian NB - Train	0.975821	0.975821	0.969269	0.972502	116760	1859	1035	35
Decision Tree [SC] [SMOTE] - Train	0.997742	0.997742	0.997752	0.997742	117791	528	4	117267
Random forest [SC] [SMOTE] - Train	0.997423	0.997423	0.997424	0.997423	117559	371	236	117424

Model performance - Test scores

Model	Accuracy	Recall	Precision	F1 Score	TN	FN	FP	TP
Logistic Regression [Base model] - Test	0.984159	0.984159	0.984410	0.976302	29449	474	0	0
Decision Tree [base model] - Test	0.968085	0.968085	0.969989	0.969033	28941	447	508	27
Decision Tree [significant columns] - Test	0.971260	0.971260	0.970464	0.970861	29032	443	417	31
Random forest [significant columns] - Test	0.972362	0.972362	0.970632	0.971492	29064	442	385	32
Adaboost [significant columns] - Test	0.972162	0.972162	0.970548	0.971350	29059	443	390	31
Gradientboost [significant columns] - Test	0.983992	0.983992	0.968567	0.976219	29444	474	5	0
XGboost [significant columns] - Test	0.984126	0.984126	0.968569	0.976286	29448	474	1	0
Gaussian NB - Test	0.976172	0.976172	0.969107	0.972595	29203	467	246	7
Decision Tree [SC] [SMOTE] - Test	0.662099	0.662099	0.974010	0.782732	19550	212	9899	262
Random forest [SC] [SMOTE] - Test	0.660395	0.660395	0.974132	0.781481	19496	209	9953	265

Best models that have been achieved without SMOTE

1. Random forest
2. Ada boosting

.

Top 5 important Features from best Model:

These are the most significant columns which we have found after comparing the pvalue<0.05 after building a stats logistic regression model

- 'transactionAmount'
- 'posEntryMode'
- 'posConditionCode'
- 'cardPresent'
- 'CVV_Match'

Business Interpretation:

- Implement proactive measures for customer information updates and security enhancements.
- Utilize dynamic credit limits based on transaction history.
- Scrutinize larger transactions for fraud prevention.
- Prioritize monitoring for accounts with low available funds.
- Allocate extra resources for fraud prevention during seasonal peaks.
- Implement real-time monitoring for transaction volume changes.
- Emphasize security for accounts with higher available funds.
- Adjust security measures with changes in transaction entry modes.
- Tailor security to account for varying vulnerability levels.
- Refine fraud detection strategies based on transaction characteristics.

References

- https://www.kaggle.com/datasets/iabhishekbhardwaj/fraud-detection(Dataset)
- https://www.fraud.com/post/the-history-and-evolution-of-fraud (Background Research)
- https://www.amygb.ai/blog/how-fraud-detection-works-in-banking (Application)
- https://www.sciencedirect.com/science/article/pii/S0957417421017164 (Past Reasearch)
- https://ieeexplore.ieee.org/document/10085493 (Ongoing Research)