Capital One data science challenge

Version id: (v20.01) Candidate ID: (C1882695) Initials: SM

1 Required Questions & Answers:

Question 1: Load

Programmatically download and load into your favorite analytical tool the transactions data.

Q1.1: Please describe the structure of the data. Number of records and fields in each record?

Given dataset has 786,363 rows and 29 columns. Among those 29 columns, 3 has boolean, 11 has float64, 6 has int64 and 9 has object type data.

Q1.2: Please provide some additional basic summary statistics for each field. Be sure to include a count of null, minimum, maximum, and unique values where appropriate.

There are 6 columns (echoBuffer, merchantCity, recurringAuthInd, posOnPremises, merchantZip and merchantState) that has 100% nan value. 5 columns (acqCountry, posEntryMode, merchantCountryCode, transactionType and posConditionCode) has $< 1\%$ nan values. Other columns don't hold any nan value.

Table 4 shows all other statistics including null% count, minimum, maximum and unique value etc.

Question 2: Plot

Q2.1: Plot a histogram of the processed amounts of each transaction, the transactionAmount column.

Transaction amounts are split into 20 bins to draw the histogram plot against their frequency and fraud frequency. Results are shown in Figure 1

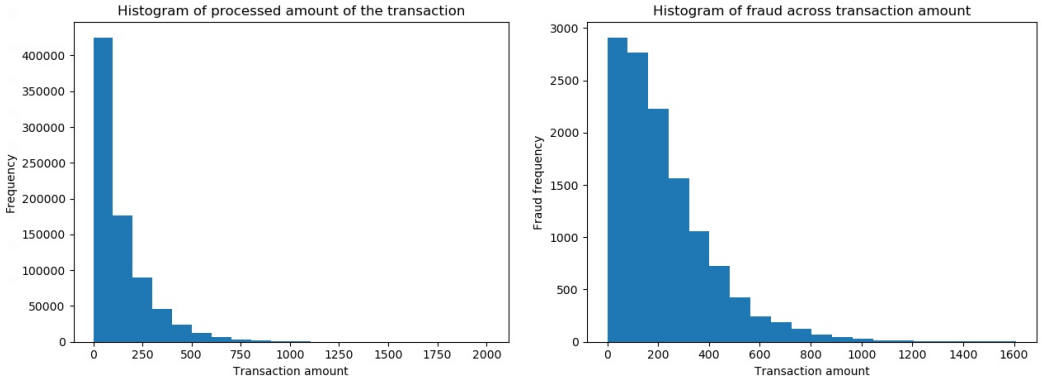

Figure 1: Histogram of transaction processed amount by count and fraud frequency

Q2.2: Report any structure you find and any hypotheses you have about that structure.

Figure 1 shows that most of the transactions are happening between 0-250 transaction amount. And the frequency decreases rapidly when amount increases. Most of frauds also happen when the amount is in range of 0-200. Fraud is less likely to happen when transaction amount increases.

Question 3: Data Wrangling - Duplicate Transactions You will notice a number of what look like duplicated transactions in the data set. One type of duplicated transaction is a reversed transaction, where a purchase is followed by a reversal. Another example is a multi-swipe, where a vendor accidentally charges a customer's card multiple times within a short time span.

Q3.1: Can you programmatically identify reversed and multi-swipe transactions?

By identifying reverse, If the question means to find how many *REVERSAL* category are there in *transactionType* column then there are 20,303 data samples in the dataset.

I define Multi swipe category as, the transactions are happening within the same merchant for same customer with same transaction amount within a time interval. The intuition for this definition is, generally multi swipe happens due to swiping card multiple times quickly for the same transaction. I use *Time gap* as a filter to filter out multiple swipes. Time gap is calculated by finding the time difference of two back to back transactions of same multi swipe category. Table 1 shows number of samples for different time gap in multi swipe category.

Q3.2: What total number of transactions and total dollar amount do you estimate for the reversed transactions? For the multi-swipe transactions? (please consider the first transaction to be "normal" and exclude it from the number of transaction and dollar amount counts)

There are 20,303 data samples of reversed transactions and total dollar amount for that category is 2,821,792.5. Table 1 shows number of samples and total dollar amount for multiple swipe category within different time gap.

Time gap	No. samples	Total dollar amount
1 min	4,430	652,692.03
2 min	8,879	1,304,345.41
5 min	13,403	1,933,949.11

Table 1: Multi swipe category filter by back to back transaction's time gap

Q3.3: Did you find anything interesting about either kind of transaction?

There are 12,417 fraud samples in the dataset. Table 2 shows how many frauds are happening in different category. It shows Reversed category has $\sim 2x$ more fraud than Multi swipe category. Although, in multi swipe category number of frauds increase as time gap increases.

Category	Time gap	Fraud(%)
Reversed	-	2.71
Multi swipe	1 min	0.70
Multi swipe	2 min	1.19
Multi swipe	5 min	1.89

Table 2: Percentage of fraud within different duplicated transactions category

Question 4: Model Fraud is a problem for any bank. Fraud can take many forms, whether it is someone stealing a single credit card, to large batches of stolen credit card numbers being used on the web, or even a mass compromise of credit card numbers stolen from a merchant via tools like credit card skimming devices.

Q4.1 Each of the transactions in the dataset has a field called isFraud. Please build a predictive model to determine whether a given transaction will be fraudulent or not. Use as much of the data as you like (or all of it).

Random forest[1] is used primarily as the predictive model to determine fraud. Random forest uses multiple decision trees [3] for prediction. A decision tree uses multiple algorithms to decide splitting up a node into two or more sub-nodes. Decision of splitting nodes are determined by calculating the entropy of different features. Provided dataset is well suited for a random forest model. Because, it holds different feature sets on which classifying fraudulent data is more convenient for the model.

First, I process the dataset by dropping columns where all the values are null. Then, the columns that hold $< 1\%$ null values are replaced with more frequent data from the column. After processing, the dataset is split into training and testing set with 70:30 ratio. With this dataset decision tree give $\sim 98\%$ accuracy on the testing set.

Cross validation: To test model accuracy across different dataset split I used k-cross[2] validation. For $k = 10$ random forest model gave mean accuracy 97% with 0.00843 standard deviation across k-cross.

Q4.2 Provide an estimate of performance using an appropriate sample, and show your work.

Codes are written in a modular way such that any dataset can be integrated very easily. Although given dataset is very small, for this experiment I randomly choose 10% data from the given dataset with a uniform distribution to make a smaller version. Distribution of isFraud is almost same between smaller and large dataset. I train a decision tree model with this new smaller dataset version. And the model gives same accuracy $(\sim 97\%)$ as the larger dataset.

Q4.3 Please explain your methodology (modeling algorithm/method used and why, what features/data you found useful, what questions you have, and what you would do next with more time)

Models analysis: I experimented with some other models to get a better view of the chosen model. Table 3 shows prediction accuracy, training time and prediction time for different models. All models have been run with same dataset and train test split ratio. It shows, prediction accuracy are almost same for all models. Although, training and prediction time varies significantly. Time wise NaiveBayes gives best time performance whereas SVM is worst. Random forest takes more time to train but the prediction is faster. Whereas, KNN training time is 4 times faster than prediction time. NaiveBayes accuracy is very competitive with random forest and also take less time. However, as the primary model I chose Random forest because historically it shows more prominent result with complex dataset than NaiveBayes. My intuition is, if the dataset changes, Random forest will give better performance than NaiveBayes.

Model	Pred acc (%)	Training time (sec)	Prediction time (sec)
Random forest	98.56	13.54	0.52
KNN	98.36	3.94	11.28
NaiveBayes	98.43	0.32	0.08
SVM	98.4	348.85	131.59

Table 3: Prediction accuracy, training time and prediction time across different models

Data analysis: Figure 2 shows correlation heat map between different features columns. For examples there are very high correlation between isFraud and transactionAmount features whereas low correlation with merchentCategory features. Correlation plots give very insightful information sometime to find and produce better feature sets.

What would I do next with more time:

- **Data perspective:** Would do more analysis on data by finding skewness, kurtosis, z-score, PCA etc. of different data distribution and to engineer more features.
- **Model perspective:** Would write more complex models such as neural network, xgboost etc. to find their performance.
- **Software engineering perspective:** Would write git commit messages, comments, unit testing etc. more rigorously across different modules.

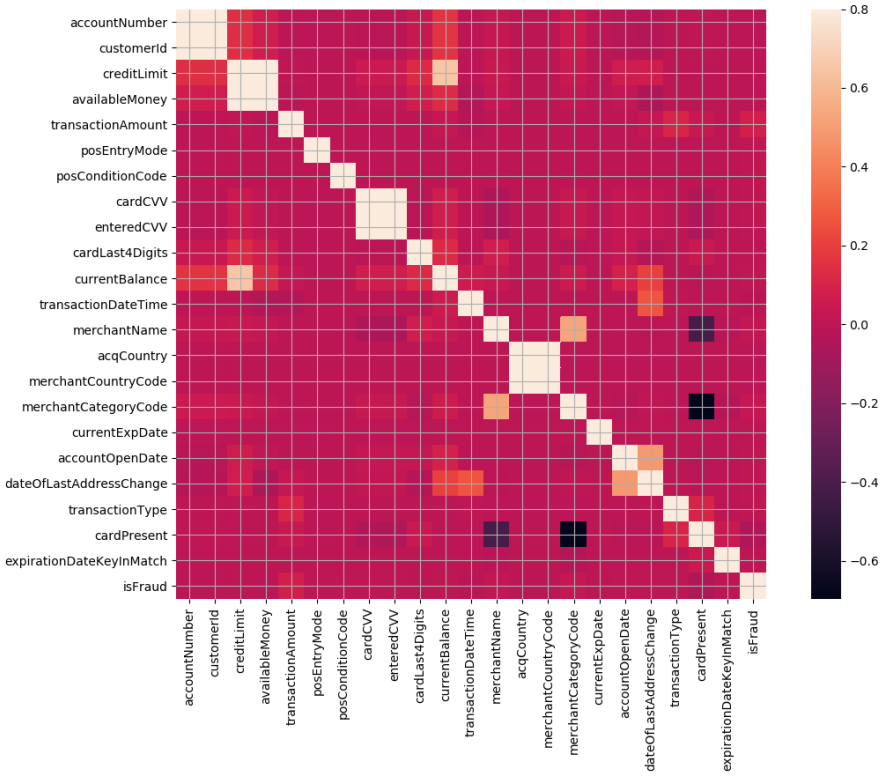

Figure 2: Correlation heatmap between different features

Columns	count	unique	freq	mean	std	min	25%	50%	75%	max	dtype	size	∼null(%)
accountNumber	786363			537232599.5	255421092.3	100088067	330133277	507456073	767620004	999389635	int64	786363	0
customerId	786363			537232599.5	255421092.3	100088067	330133277	507456073	767620004	999389635	int64	786363	0
creditLimit	786363			10759.46446	11636.17489	250	5000	7500	15000	50000	int64	786363	0
availableMoney	786363			6250.725369	8880.783989	-1005.63	1077.42	3184.86	7500	50000	float64	786363	0
transactionDateTime	786363	776637	4								object	786363	0
transactionAmount	786363			136.985791	147.725569	0	33.65	87.9	191.48	2011.54	float64	786363	0
merchantName	786363	2490	25613								object	786363	0
acqCountry	781801	4	774709								object	786363	1
merchantCountryCode	785639	4	778511								object	786363	0
posEntryMode	782309			9.049943181	16.77412952	2	2	5	9	90	float64	786363	1
posConditionCode	785954			3.271980294	9.809022992	1	1	1	1	99	float64	786363	0
merchantCategoryCode	786363	19	202156								object	786363	0
currentExpDate	786363	165	5103								object	786363	0
accountOpenDate	786363	1820	33623								object	786363	0
dateOfLastAddressChange	786363	2184	3819								object	786363	0
cardCVV	786363			544.4673376	261.5242203	100	310	535	785	998	int64	786363	0
enteredCVV	786363			544.1838566	261.5512537	0	310	535	785	998	int64	786363	0
cardLast4Digits	786363			4757.417799	2996.58381	0	2178	4733	7338	9998	int64	786363	0
transactionType	785665	3	745193								object	786363	0
echoBuffer	0										float64	786363	100
currentBalance	786363			4508.739089	6457.442068	0	689.91	2451.76	5291.095	47498.81	float64	786363	0
merchantCity	0										float64	786363	100
merchantState	0										float64	786363	100
merchantZip	0										float64	786363	100
cardPresent	786363	2	433495								bool	786363	0
posOnPremises	0										float64	786363	100
recurringAuthInd	0										float64	786363	100
expirationDateKeyInMatch	786363	2	785320								bool	786363	0
isFraud	786363	2	773946								bool	786363	0

Table 4: Different statistics of the dataset

References

[1] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[2] R. Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[3] J. F. Magee. *Decision trees for decision making*. Harvard Business Review, 1964.