Capstone 1 Project : Milestone Report
Fraud Detection in Mobile Payment Data
Springboard Data Science Career Track

Eric Cruz

Problem Statement: Online fraud is increasing and spreading rapidly across geographies and industries, and Mobile Payments represent a significant portion of the growth in both overall transaction and fraud rates. While attempting to detect and prevent fraud, the accuracy of the prediction models can have a significant impact on the ability to strike the right balance between true detection and false positives. The customer experience can be severely compromised by security measures enacted on the basis of a false positive. The fraud rate has a measurable impact on revenue, and new types and methods of fraud are evolving in response to successful detection and prevention efforts.

Dataset: Due to privacy concerns, there is little if any publicly available data for real transactions. PaySim.csv is a simulation of mobile money transactions with the objective to generate a synthetic transactional data set that can be used for research into fraud detection.

The dataset for this project is from the following link on Kaggle:
https://www.kaggle.com/faraz2402/predicting-fraud-for-mobile-payment-services

Clients: The intended clients are financial institutions and merchants who use mobile payments, and will incorporate the findings into a Fraud Detection and Prevention program which covers various types of fraud attacks.

Data Wrangling

See link to the Jupyter Notebook used to explore the data using pandas and matplotlib:

https://github.com/cruzer42/Capstone-Project-1-for-Springboard/blob/master/Capstone%20Project1%20Data%20Wrangling.ipynb

There was no wrangling required with respect to missing values, as determined by the following procedure:

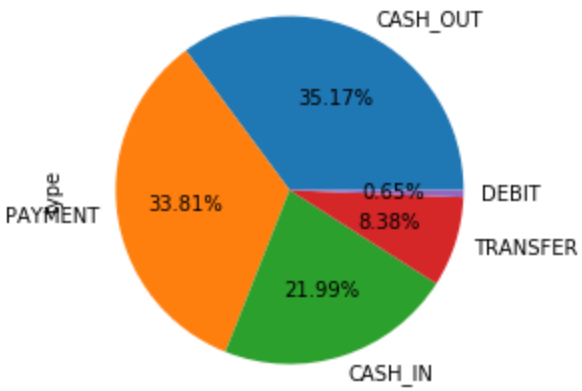
A) The csv file was converted to a pandas dataframe with pd.read_csv(), and examined with functions info(), head(), and describe().
B) A check for missing values was performed with pd.isnull().sum() and none were detected
C) However, zero values were examined and analyzed at a high level, as they may provide insights into the fraud detection patterns. In other words, zero balances have special meaning rather than indicating missing information.
D) A histogram for each column in the dataframe was generated to illustrate the imbalanced nature of the fraud case frequency in the data, and the time series frequency distribution.

Note the following description of each of the 11 columns, as well as observations about zero values and frequency distributions for each column as applicable:

1) step - maps a unit of time in the real world. In this case 1 step is 1 hour of time. Total steps 744 (30 days simulation).
    a) Note that this is the dataset author's description posted on Kaggle, but the actual number of unique values is 743, with values 1 through 743.
    b) The highest frequency is around 50,000 transactions for that hour, and there is a cluster of other intervals with a similar number of transactions. The frequency range gets as low as 2 and the distribution of frequencies appears fairly smooth.
2) type - CASH-IN, CASH-OUT, DEBIT, PAYMENT and TRANSFER.
    a) See frequency of the 5 type values:

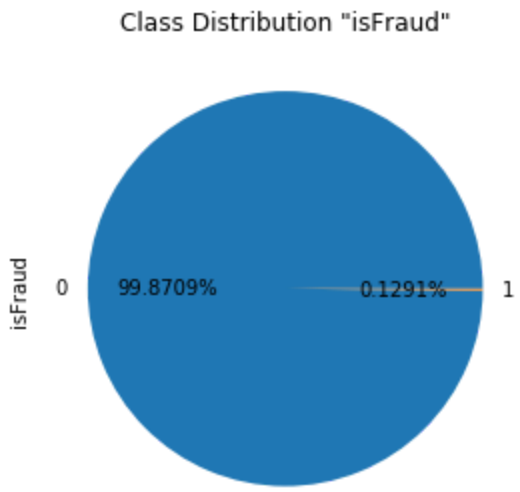
CASH_OUT	2,237,500
PAYMENT	2,151,495
CASH_IN	1,399,284
TRANSFER	532,909
DEBIT	41,432

A pie chart was generated using matplotlib to visualize the above distribution



3) amount - amount of the transaction in local currency.
    a) Summary stats are included with the describe() output for all columns. The dollar amount ranges from about $92.5MM to zero, although the zero values appear to have special meaning for the 16 cases out of over 6MM observations. All are flagged as isFraud=1, so may need to be excluded or otherwise treated.
4) nameOrig - customer who started the transaction
    a) This column contains an account number, and the frequency of values was obtained using describe() on that column of the dataframe. The highest frequency for an account was 3
    b) The percentage of unique accounts is 99.8536 or 9,313 non-unique accounts out of 6,362,620
5) oldbalanceOrg - initial balance before the transaction
    a) The ratio of zero values is 0.33 or 2,102,449 of 6,362,620

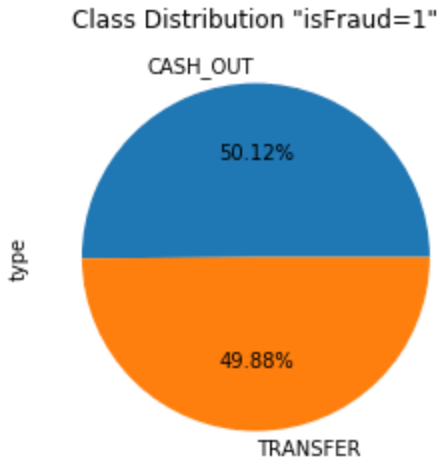
6) newbalanceOrig - new balance after the transaction
    a) The ratio of zero values is 0.57 or 3,609,566 of 6,362,620
7) nameDest - customer who is the recipient of the transaction
    a) The top frequency for recipient accounts is 113, and there is a cluster of frequencies tailing off smoothly.
    b) The percentage of unique accounts is 42.7868 or 3,640,258 non-unique accounts out of 6,362,620
8) oldbalanceDest - initial balance recipient before the transaction. Note that there is not information for customers that start with M (Merchants).
    a) The ratio of zero values is 0.43 or 2,704,388 of 6,362,620
    b) Need to classify orig and dest accounts by “M” and review zero values for each
9) newbalanceDest - new balance recipient after the transaction. Note that there is not information for customers that start with M (Merchants).
    a) The ratio of zero values is 0.38 or 2,439,433 of 6,362,620
    b) Need to classify orig and dest accounts by “M” and review zero values for each
10) isFraud - This is the transactions made by the fraudulent agents inside the simulation. In this specific dataset the fraudulent behavior of the agents aims to profit by taking control or customers accounts and try to empty the funds by transferring to another account and then cashing out of the system.
    a) The ratio of zero values is 0.998709 or 6,354,407 of 6,362,620
    b) The number of isFraud cases is 8,213
    c) A pie plot was generated using matplotlib to visualize the distribution
    d) Note the distribution of isFraud=1 types is only among 2 categories, Cash Out and Transfer. These are split fairly evenly, but recall that only 8.4% of transactions are Transfer type, while 35.2% are Cash Out.
    e) Also note the 16 Amount=0, and the complete set of 16 isFlaggedFraud=1 may need special treatment or exclusion



CASH_OUT 4116

TRANSFER 4097

Name: type, dtype: int64



11) isFlaggedFraud - The business model aims to control massive transfers from one account to another and flags illegal attempts. An illegal attempt in this dataset is an attempt to transfer more than 200.000 in a single transaction.
    a) The ratio of zero values is 0.999997 or 6,362,604 of 6,362,620
    b) The number of isFlaggedFraud cases is 16, which coincidentally is the same number of transactions with Amount=0 and isFraud. Note that for each set of 16 uncommon values, they also have the feature isFraud=1. This may be a reason to exclude or otherwise treat these with special weight.
    c) All are type TRANSFER and the amounts have a pattern of matching “Orig” balances individually or in combination, and all “Dest” balances are zero.
    d) These relationships can be seen by showing head(16) on the 16 flagged values

Data Storytelling

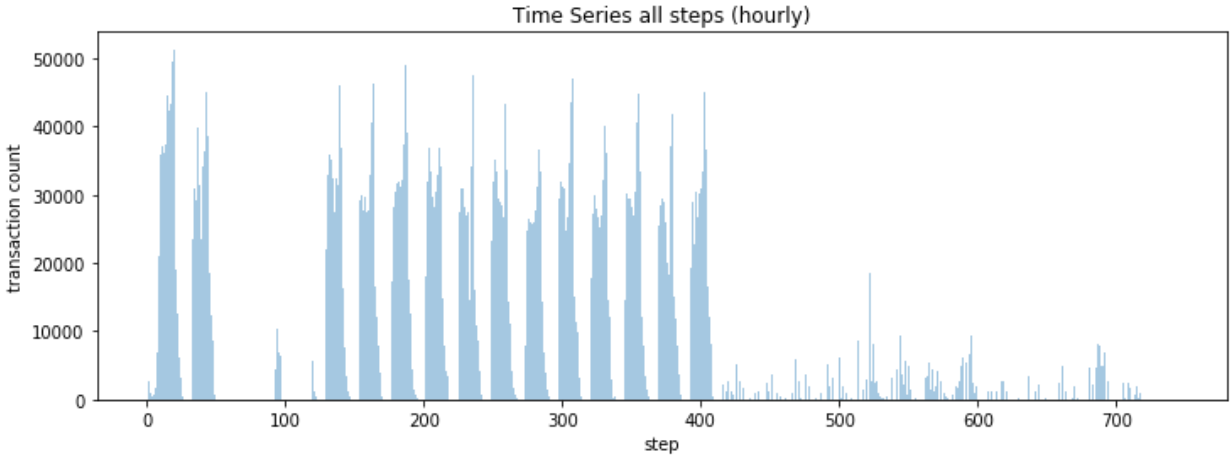
There are some data stories which emerge from Exploratory Data Analysis, which lead to a few predictions about features which could help detect fraud. See Jupyter notebook link:

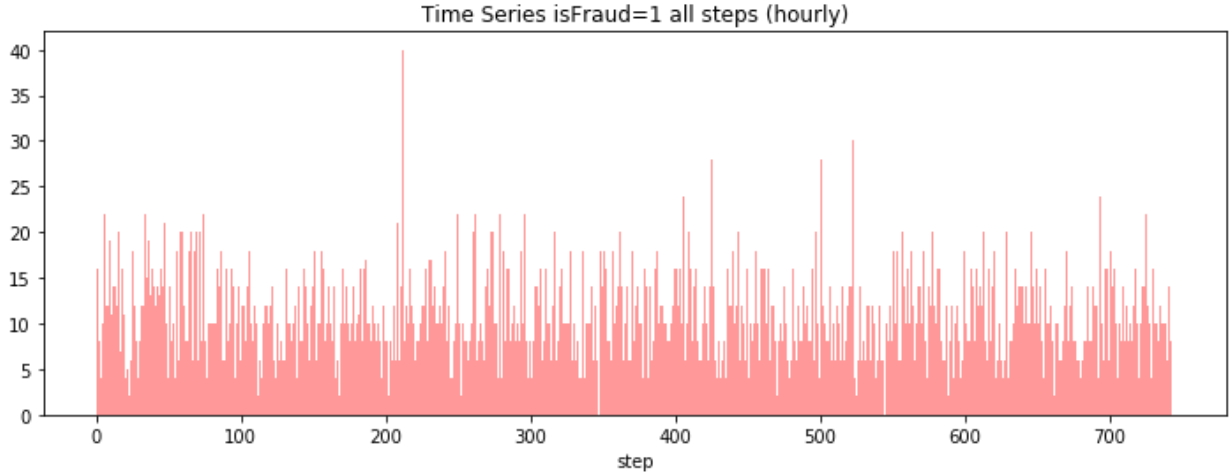
https://github.com/cruzer42/Capstone-Project-1-for-Springboard/blob/master/Capstone%20Project1%20Data%20Storytelling.ipynb

1. **The timing of fraud transactions appears to be steady and consistent, contrary to other activity which contains gaps of varying sizes between active time slots.**
2. **The transaction “Amount” is highly correlated with the “OldBalanceOrig” for fraud transactions but not correlated for other transactions**
3. **The Amount on fraud transactions is in a smaller, tighter range than the distribution of amounts for all transactions.**

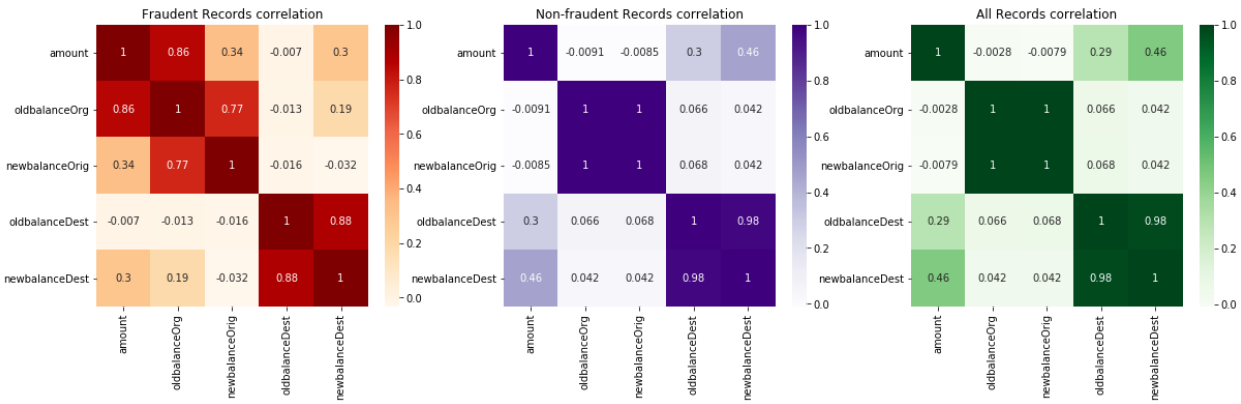
Note the following descriptions of the visualizations for the summary points above:

- The frequency distribution of transactions over time (743 steps, which is 1 hour less than 24 hours * 31 days) shows no more activity after step 718 (the final 24 hours or full last day of the data period). The frequency distribution where isFraud=1 is pretty even across the entire timeframe, which does not correspond to the non-fraud transaction distribution which contains gaps and tails off after about 400 out of 743 steps.

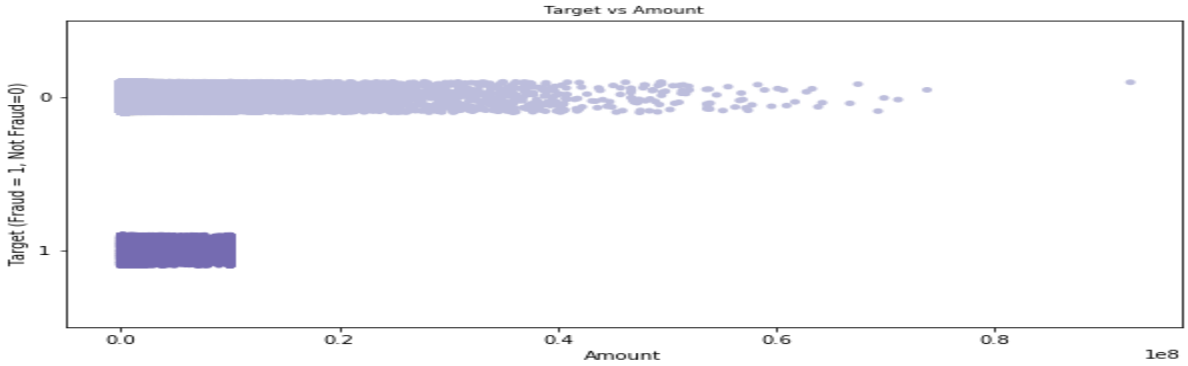




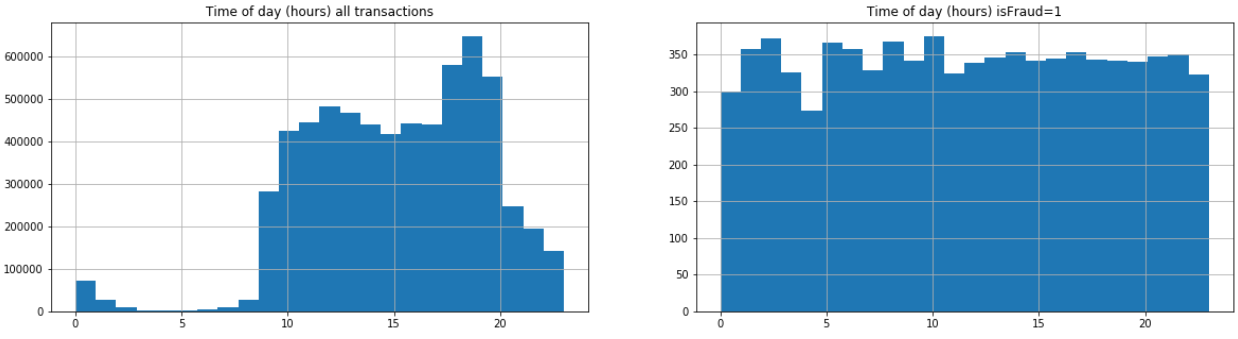
- The correlation heatmaps showing the fraud transactions, non-fraud, and combined results are plotted side-by-side. Note that fraud transactions show significantly higher correlation between “Amount” and “OldBalanceOrig” and slightly higher correlations between other features, as highlighted by the grid shading.



- See the stripplot of Amount for fraud vs non-fraud transactions. It suggests that amounts for fraud are generally on a smaller scale than other transactions.



- The step value was converted to date/time format, setting the first step to the default value of the first hour of 1/1/1970. Although the assumption that the step starts on a specific day or hour may be wrong, any patterns detected could easily be shifted to different starting point. It turns out that day of week does not appear to reveal a different pattern for fraud transactions, but the hourly activity of fraudulent transaction remains steady througout the period.



Statistical Data Analysis

We can perform some statistical analysis to confirm the differences which appear obvious from the visualizations.

In particular, the “Type” and “Amount” come forward in the data visualizations for reasons discussed below. See link to the statistical tests on these variable as described below:

https://github.com/cruzer42/Capstone-Project-1-for-Springboard/blob/master/Capstone%20Project1%20Statistical%20Data%20Analysis.ipynb

There is a significant subgroup in the “Type” variable because only 2 of the 5 types are represented for the target cases (e.g. isFraud=1). See the crosstab frequency table, or contingency table of Type/isFraud:

isFraud	0	1
type		
CASH_IN	1399284	0
CASH_OUT	2233384	4116
DEBIT	41432	0
PAYMENT	2151495	0
TRANSFER	528812	4097

We can perform the Pearson’s Chi-squared test of the null hypothesis that the frequency of categorical variable “Type” is independent across the “isFraud” categories. As expected, the p-value is virtually zero, so the null hypothesis is rejected. This indicates dependence of the categorical variable on the category.

stats.chi2_contingency()

(22082.53571319108, 0.0, 4, array([[1.39747778e+06, 1.80622440e+03],

[2.23461179e+06, 2.88821075e+03],

[4.13785187e+04, 5.34812728e+01],

[2.14871781e+06, 2.77719374e+03],

[5.32221110e+05, 6.87889834e+02]]))

The more important subgroup is the target variable itself, the “isFraud” boolean indicator. The dataset is highly imbalanced with only 0.13%, or about 8 thousand of approximately 6 million transactions. As shown in the Data Storytelling analysis, the visualization of “Amount” shows the fraud transactions in a smaller, tighther range than the nonfraud transactions. We can test the null hypothesis that the mean “Amount” is the same for both categories, using a t-test. See the mean and std calculated by category, along with the corresponding p-value calculated from the t-test.

The number of nonfraud amounts is 6354407

The number of fraud amounts is 8213

The std amount nonfraud is 596236.9813471739

The std amount with fraud is 2404252.9472401612

The mean amount nonfraud is 178197.04172739814

The mean amount with fraud is 1467967.299140387

The calculated stdev_diff is 602079.9804398556

The mean fraud minus nonfraud amount is 1289770.257412989

The calculated t-stat is 194.01200466038233

ttest_ind(fraud_amount, nonfraud_amount)
Ttest_indResult(statistic=194.01200466037974, pvalue=0.0)

The resulting p-value of virtually zero indicates that the null hypothesis that the Amount is the same for both categories is rejected, or the Amount is statistically different.

Also shown in the Data Storytelling analysis, there is a strong correlation between “Amount” and “Old Balance Orig” which can be seen in the fraud transactions but not the non-fraud transactions, and higher but not quite as significant between “Amount” and “New Balance Orig”. This is plausible given the provided description “In this specific dataset the fraudulent behavior of the agents aims to profit by taking control or customers accounts and try to empty the funds by transferring to another account and then cashing out of the system.”

The Paysim dataset is a synthetic simulation of mobile payment data, because actual data is generally not available due to proprietary concerns. This means the creation of underlying data can be considered a bootstrap approach. The test to determine fraud is inferential, but not exactly along the lines of traditional hypothesis testing due to the imbalance observed in the target boolean variable. Rather than a linear regression analysis, a more appropriate approach for binary classification is logistic regression. It also seems plausible to consider a Bayesian approach, such as the GaussianNB model. However, there are several other models that may be appropriate given the binary classification problem, which will be considered as we learn about models in upcoming sections of the course. Some potential candidates are Random Forest Classifier, Support Vector Machine, and KNN just to name a few.