Udacity Machine Learning Nanodegree
Mobile Payments Fraud Detection

Venkat Maddi

July 5th, 2018

I. Definition

Project Overview

Nowadays, people are extensively using mobile devices to handle financial transactions. Banks and Financial industry experts are predicting that customers will utilize mobile devices to initialize payments extensively this year [1]. Financial institutions are constantly working on various methods to improve the customer experience, execute the payments faster and safer [2]. Banks have introduced Reward Points to encourage customers to complete the payments electronically.

Accenture Consulting Study indicates that the Gen Z, new generation adults and young people today, will make up to 40 percent of USA population by 2020. The Gen Z customers are more comfortable with executing transactions from mobile devices [3]. The mobile based payment transactions will grow exponentially in the next few years.

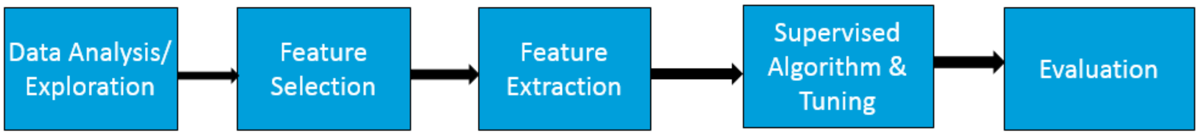
Problem statement

It is very important to detect fraudulent transactions while processing mobile payments. It is not possible to detect the fraudulent transactions manually because of huge volume of transactions banks handle hourly and daily. Researches and Data scientists are creating new algorithms and introducing new processes to detect the fraud as soon as fraudulent transaction hits the financial institutes.

Normally, the financial institutions do not publish mobile money transactions. Kaggle platform [4] has provided a synthetic dataset generated using the simulator called PaySim as an approach to detect the fraudulent transactions. [5]. PaySim uses aggregated data from the "private dataset" to generate a synthetic dataset that resembles the normal operation of transactions and injects malicious behavior to later evaluate the performance of fraud detection methods. The private dataset is based on real transactions from a mobile money services implemented in African country.

I have decided to work on machine learning algorithm to detect the mobile payments' fraud. My project will implement Supervised Learning Classification techniques to detect fraudulent transactions. Also, I will utilize fraud detection dataset available on Kaggle website at https://www.kaggle.com/ntnu-testimon/paysim1.

The high-level design activities and workflow while implementing the Supervised Learning models to predict fraudulent mobile payments are as follows:



Workflow Diagram

1) **Data Analysis/Exploration**: Data Analysis is a very important and key activity of the Machine Learning model creation. The activities within this phase are as follows:
   - Review and understand Data
   - Identify Data thresholds like minimum, maximum and etc
   - Determine Data mean, standard deviation
   - Load required libraries and data files

2) **Feature Selection:** The dataset contains multiple data fields/columns. The data field is considered as a feature.
   - Review all features included in the datafile.
   - Identify Feature Dependency. Some of the features are critical to determine the prediction. If a feature is dependent on another primary feature, then the primary feature must exist in the model.
   - Reduce features to a reasonable number. Eliminate least important features which do not cause major impact to the prediction.
   - Select Best Features from dataset.

3) **Feature Extraction**
   - Review the feature's data distribution and ranges. If the data range (difference between min and maximum values) is wide, then Normalize the data using Logarithm transformation.
   - Normally, Numeric values tend to tune models more effectively. Kaggle Paysim dataset contains features with non-numeric values. I will apply "One-Hot encoding" technique to convert non-numeric values to numeric values.
   - Split data into training and testing
   - Assign a portion of training data to the Validation activity

4) **Supervised Learning Algorithms & Tuning**
   - Create Accuracy and F-score bench marks using Naïve Predictor model.
   - Implement at least five (5) Supervised models.
   - Train the Model with the Training data
   - Tune the Algorithm by modifying Hyper parameters

5) **Evaluation**
   - Execute Model using the Testing data
   - Analyze the results and Model performance
   - Identify a best supervised Model which provides the maximum performance results, high accuracy, and high F-score.

Metrics

Each Supervised Model performance is calculated using twp statistical concepts, Classification Accuracy, and F-Score. The following table provides confusion matrix definitions.

	Predicted as Fraud	**Predicted as Genuine**
Fraud Transaction	True Positive (TP)	False Negative (FN)
Genuine Transaction	False Positive (FP)	True Negative (TN)

Table 1: Confusion Matrix

True Positive (TP): Transaction is Fraud and Model has predicted as Fraud accurately.
False Negative (FN): Transaction is Fraud and Model has predicted as Genuine incorrectly
False Positive (FP): Transaction is Genuine, and Model has predicted as Fraud incorrectly
True Negative (TN): Transaction is Genuine, and Model has predicted as Genuine accurately

The **Accuracy** measures how often the Model makes the correct prediction. It's the ratio of the number of correct predictions to the total number of data points.

$$\textbf{Accuracy} = \frac{TP+TN}{TP+TN+FN+FP}$$

The **Recall(sensitivity)** indicates what proportion of actual fraud transactions is predicted by the Model as fraud. It is a ratio of True Positives to True Positives Plus False Negatives.

$$\textbf{Recall} = \frac{TP}{TP+FN}$$

The **Precision** tells us what proportion of transactions Model predicted as fraud, actually were fraud. It is a ratio of True Positives to True Positives Plus False Positives.

$$\textbf{Precision} = \frac{TP}{TP+FP}$$

The **F-beta** score is a metric that considers both precision and recall:

$$F\beta = (1+\beta^2) * \frac{precision * recall}{(\beta^2 * precision) + recall}$$

II. Analysis

Data Exploration

The input dataset financial mobile based transactions provided by Kaggle platform. The dataset contains input attributes (AKA features) and the Fraud attribute (Target). The file contains more than six million records. Each record consists of both input attributes (features) and output variable. The classification goal is to predict whether mobile payment is a fraudulent or not.

Input Variables (features):

Data Attribute #	Attribute Name	Description
1	Step	It maps a unit of time in the real world. In this case, step 1 represents First hour of transactions
2	Type	Transaction Type, CASH-IN, CASH-OUT, DEBIT, PAYMENT and TRANSFER
3	Amount	Transaction Amount in local currency
4	nameOrig	The customer who initiated the transaction
5	oldbalanceOrg	The initial balance before the transaction
6	newbalanceOrig	The new balance after processing the transaction.
7	nameDest	The customer who is the recipient of the payment
8	oldbalanceDest	The initial balance in the recipient account before the transaction. Note that there is not information for customers that start with M (Merchants).
9	newbalanceDest	The new balance in the recipient account after processing the transaction. Note that there is not information for customers that start with M (Merchants).
11	isFlaggedFraud	If a transfer amount is more than 200,000 then single transaction flags as illegal attempt. The business model flags the transaction as “illegal Attempt” for higher denominations.

Table1: Feature Details

Output Variable (Target)

The 10th attribute, *isFraud*, is an output variable. The output variable valid values are either zero (0) or one (1). If the output variable value is zero, then the data record categorized as a genuine transaction. If the output variable value is one, then the data record is categorized as a Fraudulent transaction.

Data Attribute #	Attribute Name	Description
10	isFraud	Value values are either 0 or 1. The value 1 indicates that this transaction was created by the fraudulent agent inside the simulator

Table2: Target Column Details

Missing Attributes: The recipient account’s old balance and new balance attributes do not have values for all records. If the recipient (destination customer) name starts with M(Merchants), then destination account old balance and destination new balance attributes are zero.

Categorical Features: The second column in the datafile is Type and it explains the transaction category. There are six types of transactions available in the dataset. These transaction Types are CASH-IN, CASH-OUT, DEBIT, PAYMENT and TRANSFER.

The dataset input file contains 6,362,620 rows and 11 columns. It represents 10 features (input variable) and one target column (output). The number of Fraud records count in the dataset is 8,219. It translates 0.1291% of records are fraud payments. The percentage of fraud records are less than one percent.

Few sample records from the dataset are as follows:

type	**amount**	**nameOrig**	**oldbalance Org**	**nameDest**	**oldbalance Dest**	**isFraud**	**isFlagged Fraud**
PAYMENT	9839.64	C1231006815	170136.00	M1979787155	0.0	0	0
PAYMENT	1864.28	C1666544295	21249.00	M2044282225	0.0	0	0
TRANSFER	181.00	C1305486145	181.00	C553264065	0.0	1	0
CASH_OUT	181.00	C840083671	181.00	C38997010	21182.0	1	0
PAYMENT	11668.14	C2048537720	41554.00	M1230701703	0.0	0	0
PAYMENT	7817.71	C90045638	53860.00	M573487274	0.0	0	0
PAYMENT	7107.77	C154988899	183195.00	M408069119	0.0	0	0
PAYMENT	7861.64	C1912850431	176087.23	M633326333	0.0	0	0
PAYMENT	4024.36	C1265012928	2671.00	M1176932104	0.0	0	0
DEBIT	5337.77	C712410124	41720.00	C195600860	41898.0	0	0

Table 3: Sample Records from Dataset

The following table explains the Statistical info. The amount and balance features contain a higher standard deviation.

step	**amount**	**oldbalanceOrg**	**newbalanceOrig**	**oldbalanceDest**	**isFraud**
count	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06
mean	2.433972e+02	1.798619e+05	8.338831e+05	8.551137e+05	1.224996e+06
std	1.423320e+02	6.038582e+05	2.888243e+06	2.924049e+06	3.674129e+06
min	1.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	1.560000e+02	1.338957e+04	0.000000e+00	0.000000e+00	0.000000e+00
50%	2.390000e+02	7.487194e+04	1.420800e+04	0.000000e+00	2.146614e+05
75%	3.350000e+02	2.087215e+05	1.073152e+05	1.442584e+05	1.111909e+06
max	7.430000e+02	9.244552e+07	5.958504e+07	4.958504e+07	3.561793e+08

Table 4: Dataset Statistical Info

Exploratory Visualization

I have created various graphs to visualize and identify dependency across the features. Figure 1 shows the record count by transaction type. The number of CASH_OUT and PAYMENT records are more than 2 Million each. The CASH_IN records are around 1.5 Million. The TRANSFER records around 500,000. However, The DEBIT records count is much lower less than 45,000 records.

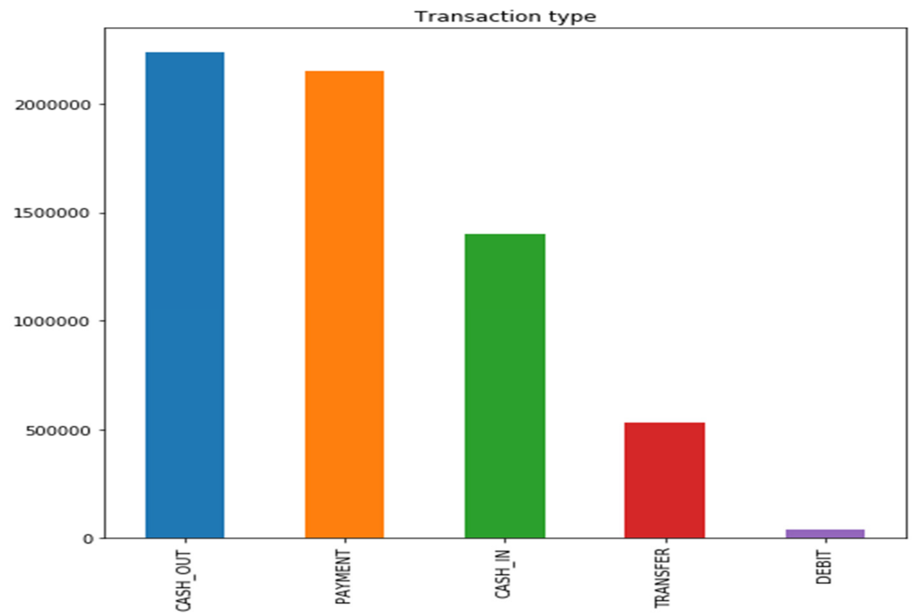


Fig 1: Summary by Transaction Type

I have stared reviewing Fraud transaction types. It seems, the Fraud records have been identified in two types of records, CASH_OUT and TRANSFER. The remaining three types do not have fraud records. Figure 2 shows the Fraud record count by transaction type:

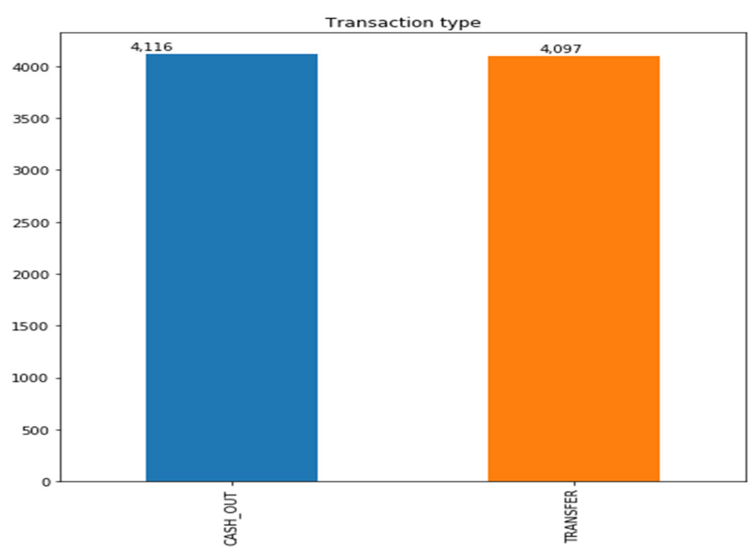


Fig 2: Fraud Transactions Count

The amount values are not evenly distributed. Fig 3 shows that most of the records have amount less value. There are very few records that have more amount more than 1,000,000. The Amount data is skewed towards to the left.

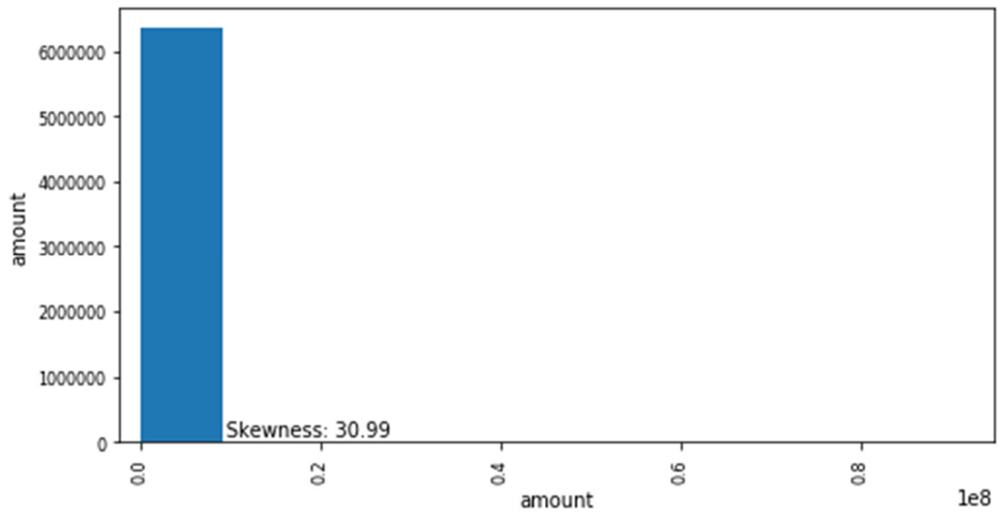


Fig 3: Amount distribution

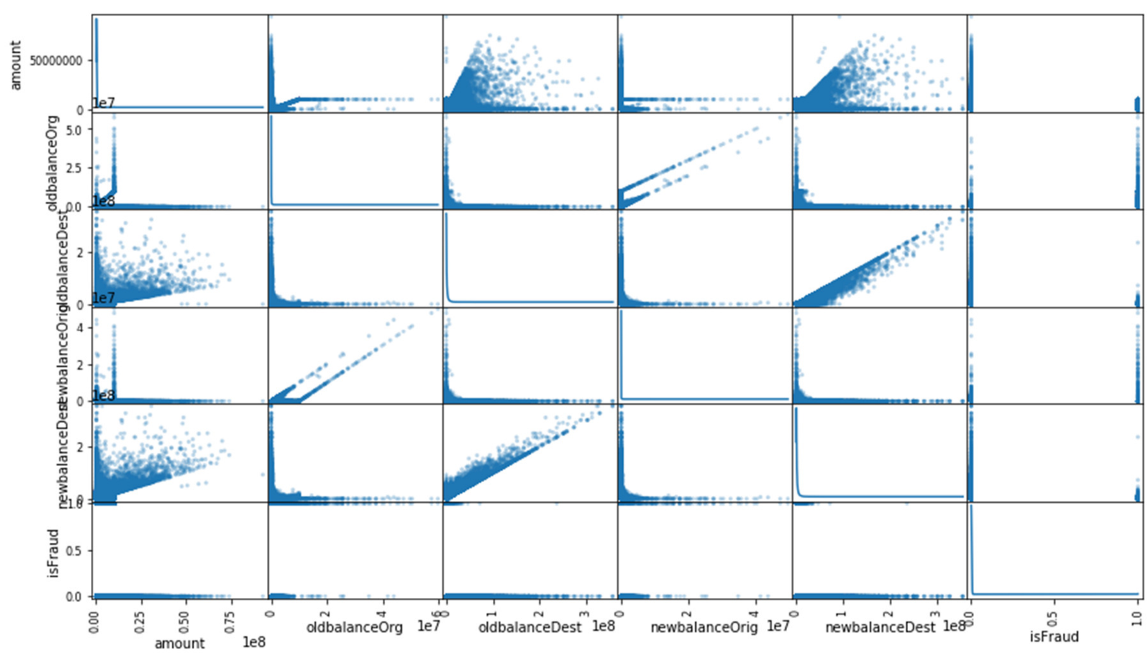


Figure 4 shows that there is no correlation between the features. It means, I must include all the features to predict the fraudulent status:

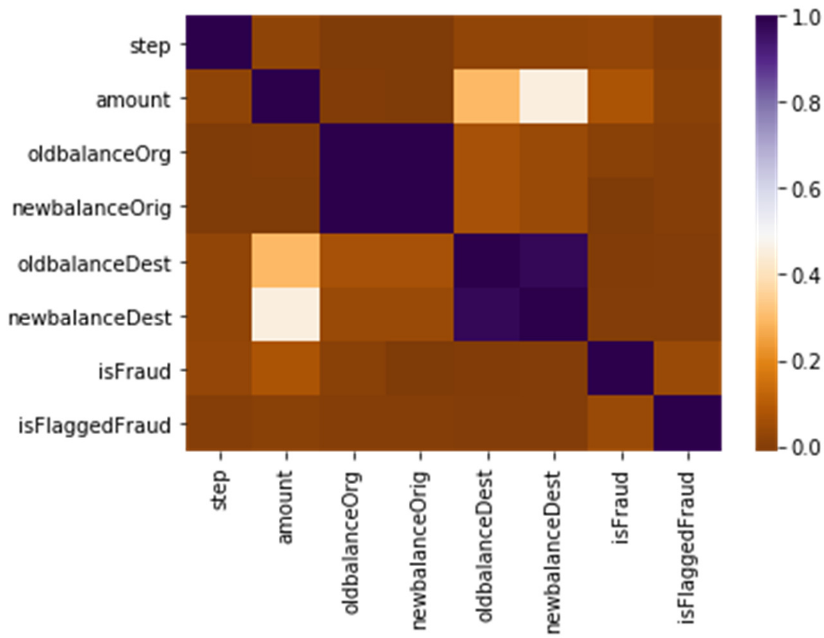


Fig 4: Features Correlation

Algorithms & Techniques

The Decision Tree algorithms are vastly utilized for Supervised Learning models. The dataset contains millions of records, and each record consist of input variables (features) and output variable (target). I will execute multiple Supervised Learning models with various hyper parameters to predict the fraudulent transactions. Here are the proposed model details:

- **Naïve Predictor**: I will start initial model with the assumption that all payment records are Genuine and none of the records belongs to fraud category. In this model, there are Zero fraud records exist in the prediction outcome. I will notice False Positives (FP) count equal to the Fraud records in the original dataset. The F-beta score is skewed in this model because of low number of False Positives compare to the total number of records.
- **Naïve Bayes Classifier**: It is a simple probabilistic classifier, which is based on applying Bayes' theorem [6]. The Bayes theorem depends on the conditional probability. The model used by a naive Bayes classifier makes strong independence assumptions. This means that the existence of a particular feature of a class is independent or unrelated to the existence of every other feature. I could not find relationship between features as per above scatter matrix diagrams.
- **Logistic Regression as a classifier**: This model is appropriate when Target (Dependent) variable is a dichotomous (binary). In our dataset, the Target variable values are either 0 or 1.
- **K-Nearest Neighbors Classifier (KNN):** The entire dataset is divided into K number of classes. KNN model calculates Euclidean distance between target variable and each test

variable. Based on the distance, Model predicts the suitability of a Class target variable belongs to. Here is KNN diagram from Wikipedia, each color represents a class:

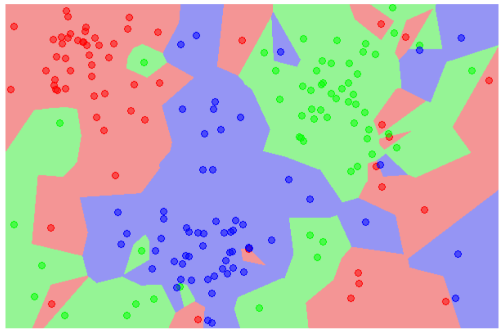


Fig 5: KNN class representation

- **Decision Tree Classifier**: The decision tree classifiers organize a series of test questions and conditions in a tree structure. In the decision tree, the root and internal nodes contain attribute test conditions to separate the records that have different characteristics. All the terminal node is assigned a class label either Yes or No. The following figure shows identify person's credit rating after verifying the age.

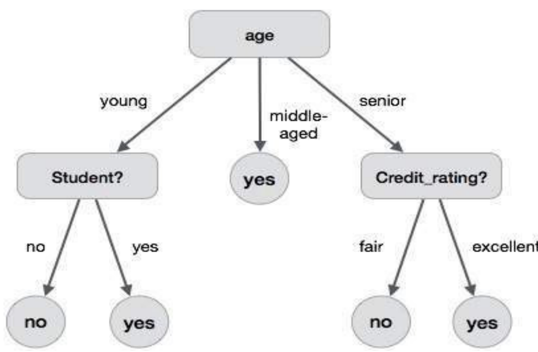


Fig 6: Decision Tree representation

- **Random Forest Classifier:** This model creates set of small decision trees from a randomly selected subset of training data. Then, it aggregates them into "Forest of Trees". Each tree provides a weak predictor because the tree is handling the subset of data. Combining each weaker predictor will potentially generate a stronger predictor model. Here is diagram [8] illustrating Random Forest example, each Tree generated a predictor class (Class-A, Class-B,... Class-N) and combining all classes generates an aggregated Final predictor class.

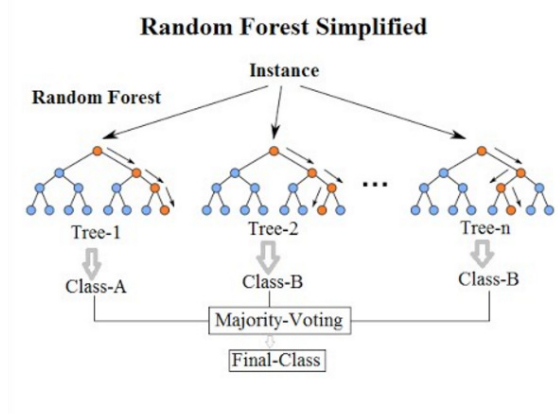


Fig 7: Random Forest Classifier Illustration

- **Ensemble – Voting Classifier:** Ensemble methods are techniques which create multiple models and combine them to produce better results. The Voting classifier is one of the Ensemble methods widely used in the Supervised Learning. There major difference between Random Forest and Voting Classifier is, number of models utilized. In the Random Forest Classifier, we create multiple Decision Tree classifiers and generate a combined class. In the Voting Classifier, we create multiple supervised learning models and generate a combined class. The Ensemble technique would provide more accurate results compare to the individual model. Here is diagram [9] illustrating Voting Classifer Ensemble method.

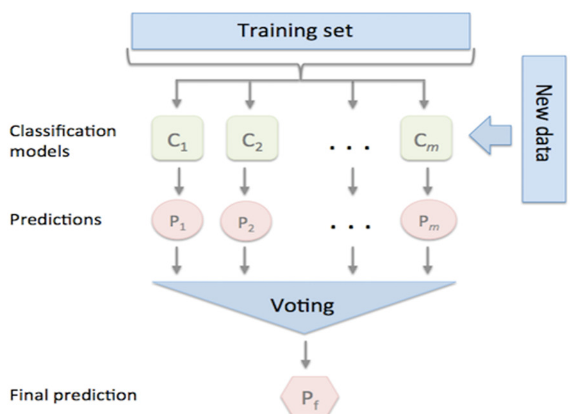


Fig 8: Ensemble Voting Classifier

Benchmark

I will the Supervised Learning model with Naïve Bayes algorithm and make it as a benchmark score. I will slowly implement other Supervised models and tune the hyper parameters to improve the score. The Naïve Bayes algorithm accuracy is 99.16% and F-score is much lower, 0.2157. Here are Accuracy and F-score for Naïve Bayes algorithm:

Model Name	**Accuracy**	**F-Score**
Naive Bayes	0.9916	0.2157

III. Methodology

Data Processing

I have executed various pre-processing steps to normalize data and to delete less-useful columns. The Data preprocessing details are as follows:

- **Delete Three Types of Records**: The data analysis and bar chart graphs clearly indicate that two types of data records (TRANSFER and CASH_OUT) contain Fraud transactions. The remaining three types (PAYMENT, CASH_IN and Debit) of records are Genuine transactions. Therefore, we can safely delete the remaining three types of records from the dataset and keep the first two types in the dataset for data processing.
- **Convert String to Integer:** The Type column contains two types of string values either TRANSFER and CASH_OUT. Therefore, we can convert column from string to either 1 or 0. I have created a new column (c_type) to store the converted value. If the transaction type is "TRANSFER", then assign 1 to c_type. If the transaction type is "CASH_OUT", then assign 0 to c_type.
- **Logarithmic Normalization:** The amount and balance column values are distributed between zero and millions. These columns should be normalized for accurate and effective prediction. Therefore, I have applied Logarithmic to the amount and balance columns. The Log of zero is infinite. I have changed amount vales from zero to 0.01 before applying Logarithmic function.
- **Drop Columns:** The source account name and destination name columns do not add much value to the Machine Learning Model. Therefore, I have decided to drop Name columns along with the pre-normalized data columns.

Here amount and balance histogram diagram after normalizing the data. The skewness after normalization is much lower compare to the post-normalization

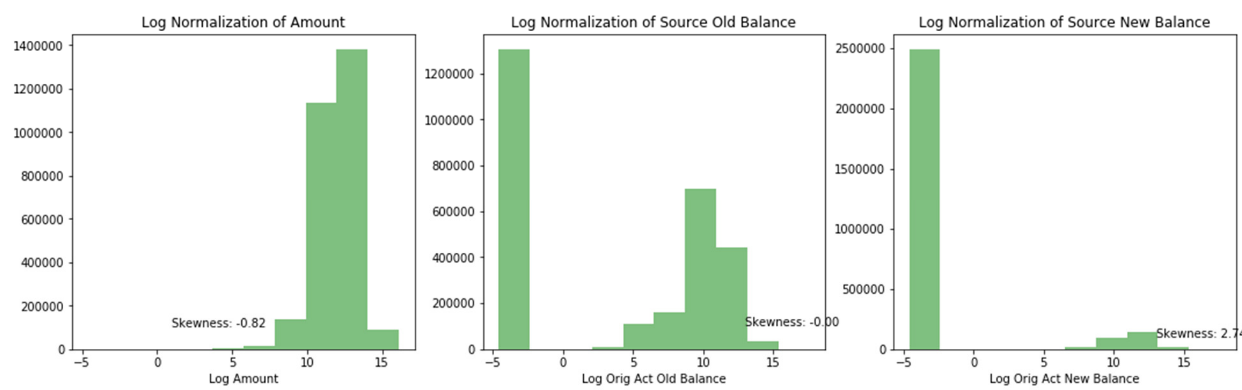


Fig 9: Amount, Original account old balance and new balance histograms

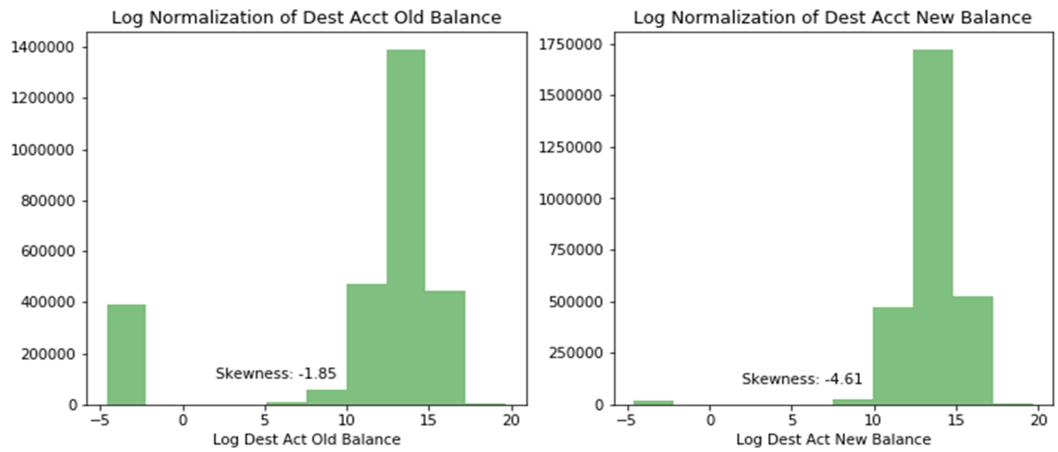


Fig 10: Dest account old balance and new balance histograms after normalization

Implementation

The next task is Model implementation. The features and target data has been split into training and testing datasets. I have allocated 80% of datasets to the training data and remaining 20% of datasets to the testing to verify the accuracy of the model.

ShuffleI will set aside at least 30% of Training data for the Data validation.

I will also verify the quality of the data. I will verify which features which cause the major impact on detecting the fraud. Some of the features may not cause major impact to the fraud detection. I will eliminate the least important features that cause minor impact to the fraud detection.

Refinement

IV. Results

Model Evaluation and Validation

Justification

V. Conclusion

Free-Form Visualization

Reflection

Solution statement

The dataset contains multiple non-numeric columns like transaction type. The non-numeric feature columns will be converted to 1/0 binary values. As described in above section, there are several non-numeric columns that need to be converted. The amount and balance columns will be normalized using scaling technique to have a reasonable data range.

I am not sure which algorithms would be fit for this problem or what hyper parameters configurations are needed. Here are some of the Supervised Learning Algorithms I will try to apply during the implementation:

I will implement above Supervised models and identify a best Supervised Model applicable to the fraud detection.

Benchmark model

Project design

After executing these tasks, I will prepare a conclusion and document observations at the end.

References

[1] Mobile Payment Trends in 2018: https://www.paymentvision.com/blog/2017/12/26/7-trends-that-prove-mobile-payments-are-here-to-stay-in-2018

[2] Mobile Payments Safer and Faster: https://www.mobilepaymentstoday.com/blogs/3-trends-for-2018-safer-data-faster-payments-better-experiences/

[3] Banking Future Payments- Accenture Consulting Study : https://www.accenture.com/us-en/insight-banking-future-payments-ten-trends

[4] Kaggle Financial Fraud Detection dataset: https://www.kaggle.com/ntnu-testimon/paysim1

[5] PaySim Simulator: E. A. Lopez-Rojas , A. Elmir, and S. Axelsson. "PaySim: A financial mobile money simulator for fraud detection". In: The 28th European Modeling and Simulation Symposium-EMSS, Larnaca, Cyprus. 2016

[6] How to fix Skewed dataset in Machine Learning: https://becominghuman.ai/how-to-deal-with-skewed-dataset-in-machine-learning-afd2928011cc

[7] Naïve Bayes theorem : https://www.python-course.eu/naive_bayes_classifier_introduction.php

[8] Random Forest Diagram: https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d

[9] Voting Classifier sample diagram: https://rasbt.github.io/mlxtend/user_guide/classifier/EnsembleVoteClassifier/