Team Name: The Analysts

Team Members: Oyindamola Obisesan (oyin.obisesan@ou.edu), Micheal Olubode (mikeolubode@ou.edu), Judah Odiachi (jodiachi@ou.edu), Adesoji Akanji (akanji.adesoji@ou.edu)

Topic: Credit Fraud Detection

Executive Summary

Fraud detection is a series of activities that hinder money from being obtained by illegal means which can include identity theft or stolen credit card. Fraud is a widespread activity that can occur in a large number of industries but the concentration in this work is the credit card companies. With the amount of money at stake, there is a need for a reduction in the amount of credit card frauds, and machine learning algorithms are the methods that have been utilized in the credit card companies to tackle this problem.

The assumptions made in the modeling is that the state of the customer is where they registered with the credit card company. That the customers reported correct gender, age and other personal information.

The analysis of the data shows some features that are important in the determination of whether the transaction was fraudulent or not. Some of these features are the time of the transaction; the data analysis show that a most of the fraudulent transactions were done in the night and the non-fraudulent transaction were during the day. The amount of money of the transaction is also a major factor to consider. Fraudulent transactions were majorly in the between $500 - $1000. This shows that the criminals were careful not to have high transactions to mask these fraudulent transactions making it harder to detect. The state the transaction occurs also had an impact on fraud detection. The type of transaction and age was also a major feature for the modeling of fraudulent transactions. More aged customers were found to be more liable to fraud. We found that the gender of the card owner was not important as it didn't vary also with fraudulent and non-fraudulent transactions.

Credit Card Company should pay attention to older customers who may not have a lot of control over their cards. This would reduce the risk of frauds. Transaction done after 10 pm should be watched closely to avoid frauds as that is the time most of the fraud occurred. The actual place the merchant was based would be used in the modeling approach. The risk of having a fraud is higher if the state of the customer is different from the place the merchant is located.

The random forest models were the most accurate predictive model. The features used in the model are amount (amt), state, age, and category. The hyperparameters of the models were tuned and the best model parameters are selected. The modeling approach would be to cluster the non-fraudulent transaction into three groups based on the amount of non-fraudulent transaction. This is to know the normal spending behavior of the customers. The frauds would be easier to detect this way. If a certain customer is spending more than is perceived from the spending history of the customer, then the transaction can be fraud. The two techniques were considered and the customer grouping had an edge over the ungrouped customer based on the recall, F1 and precision.

While we couldn't reach the goal of 100% accuracy in fraud detection, we created a model that can get very close to that goal with enough time and data. More room for improvement can be found in the dataset. The precision of the model increases when the size of dataset is increased. An increase in the data volume will cause a positive difference in the model by making it more accurate in detecting frauds and reduce the number of false positives.

1.0. Problem Description and Background

The use of credit cards is on the rise and the number of credit cards issued has increased by 2.6% from 2017 to the first quarter of 2018 according to TransUnion insights report. With the increase in the number of credit cards also comes with the growth of credit card frauds. In 2018, losses in fraud were estimated to be above $27 billion and have been growing significantly.

The target in this work would be to determine if a transaction is fraudulent or not by considering certain variables from the dataset provided such as the location the transaction is being initiated, the IP address of the device, the historical transaction pattern or trend of the credit card owner, and the actual information of the transaction (how much?, what was the purchase?). These factors would be analyzed to be able to decide if the transaction should be initiated or declined.

This analysis would be divided into sections for data exploration and visualizations to correctly view the results of the data, data exploration; data missingness, feature importance, feature extraction, feature transformation, feature engineering and data scaling. Under-sampling of data was done to balance the dataset. Data grouping/clustering was also done. Building the predictive model was the last phase of the modeling approach. The models were built using different algorithms and hyperparameter optimization and the best model was picked based on the precision, recall, accuracy, Kappa score and F1 score.

2.0. Data Exploration and Visualization

This shows the bar plots of the fraudulent and non-fraudulent transactions. From this plot, it is observed that the number of fraudulent activities to non-fraudulent activities is very low and it is a very small percentage of the data.

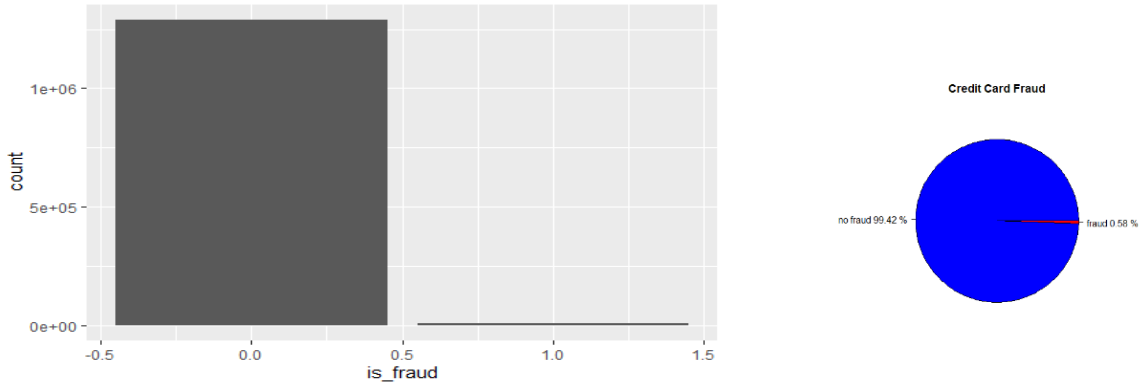


Fig 2.0.1: Count and Percentage of Fraudulent and Non-Fraudulent Transactions in Original Dataset

From the 1,296,675 transactions, there are just 7506 transactions that are frauds. This is a very imbalanced dataset as expected.

0 – Not Fraud	1 – Fraud
1289169	7506

Table 2.1: Number of Fraudulent and Non-Fraudulent Transactions in the Original Dataset

The amount of money for each of the category of fraud is observed on the boxplot below. The boxplot shown represents a very skewed distribution. There are a lot of data points outside the y-axis limits in this plot, and they are classified as outliers.

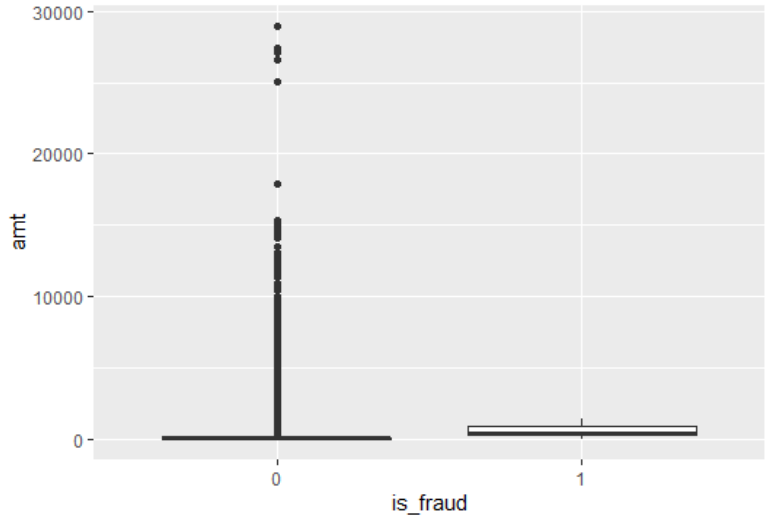


Fig 2.0.2: Transaction Amounts for Fraudulent and Non-Fraudulent Transactions

The expectations when analyzing this data was that the fraudulent activities would have high transaction amount than the non-fraudulent activities. However, this plot refutes that claim as the fraudulent activities have a defined box plot with even fewer outliers and the amount is seen to be lower for the fraudulent activities than for the non-fraudulent activities. The question would be why would the criminals decide to spend the money in smaller amounts? Does this have to do with the blockage put in place by most banks on larger transactions? This behavior looks like a cover that these criminals use to make their activities less obvious.

3.0. Exploratory Data Analysis

Transactions by Category

Here we analyze the data based on their categories to determine which category/ type of transactions do the fraud perpetrators target. As seen from the pictorial representation in Appendix A1, the credit card fraudsters happen to perpetrate fraudulent activities mostly towards grocery_ pos and shopping_net categories, this could be a pointer as to the kind of merchants that more attention should be paid to during transactions.

In exploring the transaction activities, we dig deeper into grocery_pos and shopping_net transactions which have been shown to have a higher degree of fraudulent activity associated with them. In the chart in Appendix A2, it is seen that the average amount spent by fraudulent transactions is greater than the average amount spent on actual non-fraudulent transactions, and is not specific to a certain Merchant showing that the fraud perpetrators do well no to leave traces or show alliance with any particular merchant and that the fraudsters relatively spend more on internet shopping than grocery_pos fraudulent activities.

Transactions by State

Comparing the mean amounts of fraud transactions by state shown in Appendix A3, there is only 1 distinct state that shows that all activities from the state of Delaware are fraudulent. The reason for this might be attributed to this being simulated data.

From the analysis of the maximum and minimum amount of fraudulent and non-fraudulent transactions, it is seen in Appendix A4 that the fraudsters are clever in masking their activities, by making the maximum amount of their transactions on a state-wise level less than the maximum amounts of non-fraudulent transactions making fraudulent activities a bit more difficult to detect. It is seen that the minimum amount of transactions are greater than the

minimum transaction amounts from non -fraudulent activities. This is understandable as a fraudster would not necessarily involve in high risk for minimum rewards.

Analysis of Transaction Time

An analysis of data based on the time of transactions shows that grouping transactions by morning and afternoon for both fraudulent and non-fraudulent activities does not give us clear insights as to the timeline of fraudulent activities for quick detection.

Fig. 3.0.1 Percentage timeline of fraudulent and non-fraudulent transaction activities

A closer look as to transaction behavior post noon shows a behavioral trend in the data, 86% of fraudulent transactions and 33% of non-fraudulent transactions occur between the hours of 12:00 – 8:00 pm, giving an inkling to the possibility of taking into consideration, time of transaction in the model, as shown in the plots below.

Fig. 3.0.2 Percentage timeline of fraudulent and non-fraudulent transaction activities post noon

Further exploration reveals that 97% of the 14% fraudulent activities in the hours of 8 – 12 am occur within the hours of 10:00 pm – 11:59 pm, while 50% of the 67% non-fraudulent transactions post noon occur between 10:00 – 11:59 pm, further indicating a time preference for fraudulent activities.

Fig. 3.0.3 Percentage timeline of fraudulent and non-fraudulent transaction activities after 10 pm.

4.0. Data Preparation

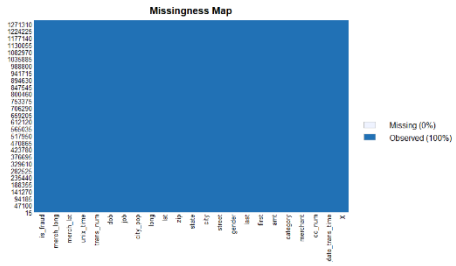


Fig. 4.0.1 Exploring Missingness in the Dataset

From the chart above, it is seen that this dataset does not contain missing data.

Feature Transformation: Looking at the distribution amounts in the fraudulent transactions in the charts below, we see a good distribution, but in the non-fraudulent transaction; the chart pairs below, we see a highly skewed distribution. Running a box-cox transformation on both the fraudulent and non-fraudulent data, we see that the boxcox transformation spreads the non-fraudulent data, but does not seem to be good when dealing with fraudulent amounts in spreading the data.

Fraudulent Transactions Amount

Fig. 4.0.2 Distribution of fraudulent transaction amounts and its boxcox transformation

Non-Fraudulent Transactions Amount

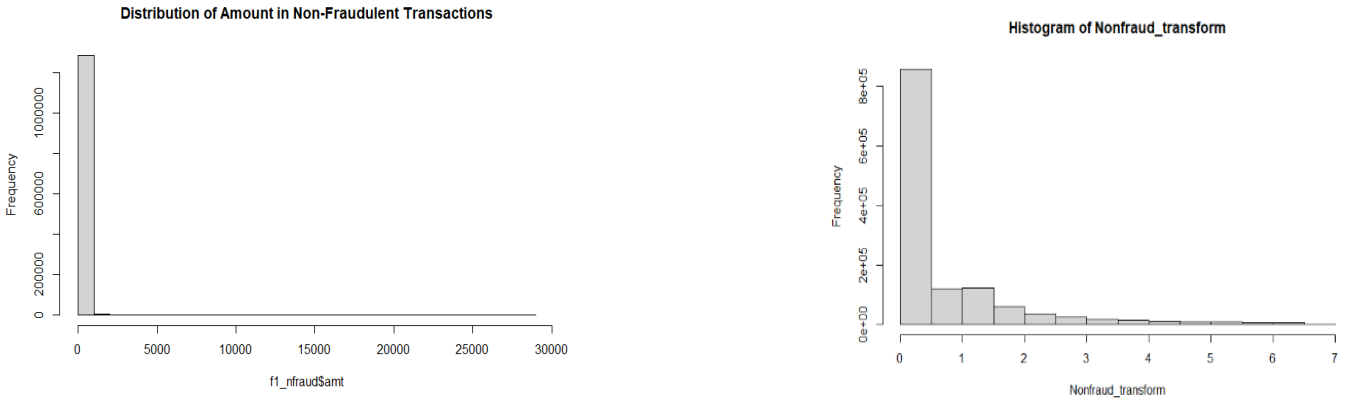


Fig. 4.0.3 Distribution of Non-fraudulent transaction amounts and its boxcox transformation

- **Scaling data:** The numeric data was scaled using the min-max scaler to centralize the data.
- **Feature Engineering:** The feature; date of birth was changed into the age of each of the transactions. The time of the purchase was also converted into a new feature.
- **Feature selection:** The parameters in the model were ran into a logistic regression model. The model gave the following parameters the highest importance; **amt, time, category, city_pop and age.** The significance of the models was determined from the p-value of the feature. If the p-value is less than 0.05, we can say that there is no enough evidence to statistically reject the hypothesis and the features are categorized as useful in the prediction of the output variable. These features were also shown to have a high variability for the fraudulent label. These were the features used in the modeling and prediction.

5.0. Balancing of the Dataset

Imbalance dataset: The major challenge with imbalanced data is that the classifier would mostly label the test data as non-fraudulent. This causes a large classification error on the fraud cases as the classifier has a very high accuracy but a very low recall. In this particular case, as shown above, the cost of the fraud is very high and it is very costly to classify a fraudulent transaction as non-fraudulent. The classifiers learn better with a balanced dataset. The solution to this would be to change the distribution of the dataset (Saputra & Suharjito, 2019).

There are various methods of sampling to change the distribution of the dataset.

Under-sampling: This works with the majority dataset by reducing the percentage of this label in the data to balance the data. This method is best when there is a large amount of data available. Reducing the number of the training sample helps to improve run time and storage. There are two methods of under-sampling; Random and Informative.

The major drawback of this method is the amount of information lost by reducing some observations from the majority class. Some distinct and important information might be lost.

Oversampling: This involves working on the minority label. It increases the observations of the minority class by replicating the minority class observations. This can also be done by informative and random method. The advantage of this method is that it does not lead to any information loss as we still have all the data.

The drawback of the method is that it can lead to overfitting as we have replicated data. This would give a high accuracy on the train data set but a low accuracy on the test dataset.

Synthetic Data Generation: The method creates artificial data of the minority class to correct the imbalance in the data. The SMOTE package is a very common method of doing this. It creates new data using feature space similarities from the minority class. It uses bootstrapping and K-nearest Neighbors.

In this case, we have a high volume of data set and we can under-sample the data. This is to reduce the proportion of the non-fraudulent transactions to equal the amount of the fraudulent transactions we have.

5.1. Data Processing

Under-sampled Data

After the data was under-sampled, the following analysis was derived from the data. The data now contains equal proportions of fraudulent and non-fraudulent transactions.

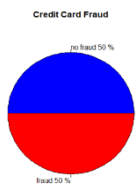


Fig 5.1.1 Distribution of new sampled dataset of fraudulent and non-fraudulent cases.

The boxplot for the ages show that more fraudulent transactions were recorded for people of higher age group. The mean of the fraud transaction for age is higher than that of no fraud. This was the evidence we used to include age as one of the parameters to model fraudulent transactions.

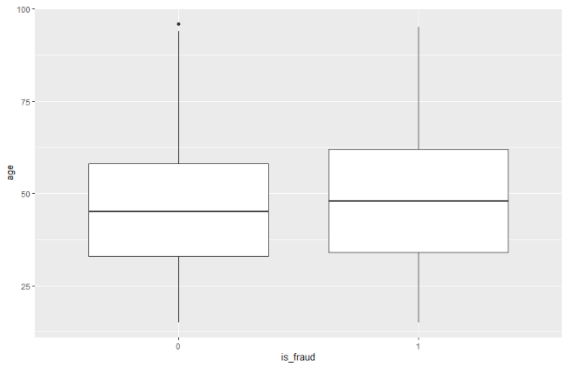


Fig. 5.1.2 Boxplot of age distribution for fraud and non-fraud

For the undersampled data, we notice a slightly different trend. The amount of money from the fraud is higher than that of without fraud. This clearly shows the disadvantage of under-sampling data, it reduces some of the information that would have been derived from the data.

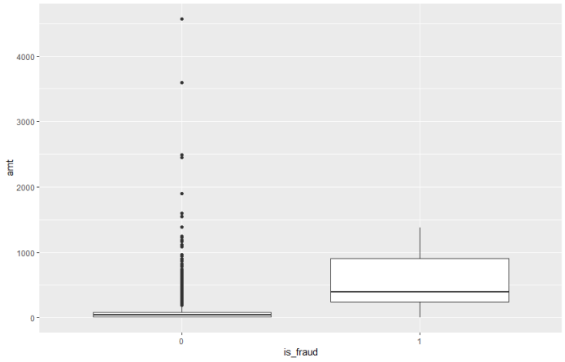


Fig. 5.1.3 Boxplot of amt distribution for fraud and non-fraud

6.0. Data Grouping/Clustering

The analysis of the boxplot shows that there are points that are outside the max line. These points can be regarded as outliers. In this case, no outlier would be removed because though these values are outside the zone of other values, they might be useful in helping us understand the customer behavior.

This leads to the concept of splitting the customers into three parts based on the amount of the transaction.

Group 1: The number of transactions less than $100. From the plot, we can see that most of the transactions below $100 are non-fraudulent transactions. The mean amount of the transactions are about $35 for the no-fraud and even lower at $25 for frauds. This shows that most of the frauds are not below $100. Most of the transactions below $100 are non-fraud transactions.

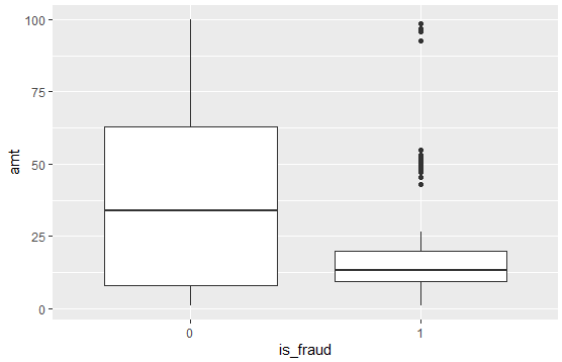


Fig. 6.0.1. Boxplot of amt distribution for fraud and non-fraud (group 1)

Group 2: This shows the amount of transaction between $100 and $1000. From the plot, most of the transactions that take place between $100 and $1000 are fraudulent transactions with the mean at $700.

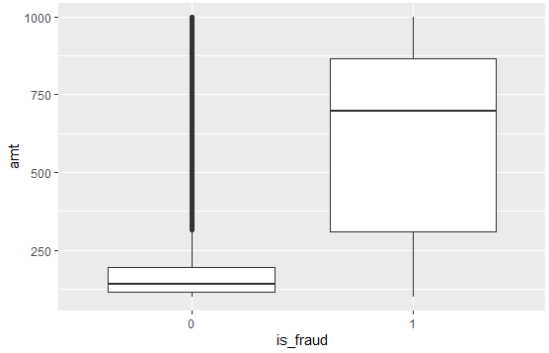


Fig. 6.0.2 Boxplot of amt distribution for fraud and non-fraud (group 2)

Group 3: This shows the amount of transactions above $1000. From here, it can be seen that the amount of fraud in this region is very small compared to the amount of non-fraud. The criminals who do the fraud can be said to be watchful of the limits that might have been set in place by the customer and therefore avoid large transactions and instead, withdraw the money in transactions between $100 and $1000.

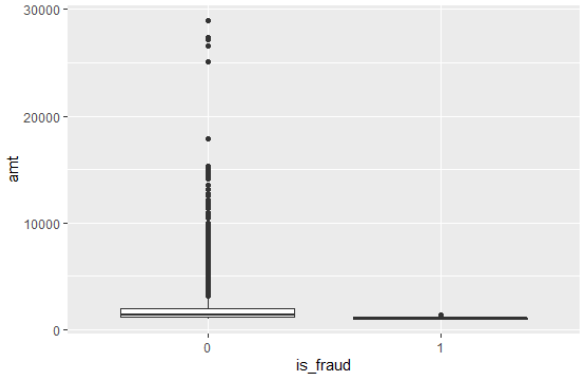


Fig. 6.0.3 Boxplot of amt distribution for fraud and non-fraud (group 3)

7.0. The Cost Sensitivity of Credit Card Detection

For the case of most machine learning algorithms, there is no difference between right and wrong classifications. In this case, there is a difference between the cost of having a false negative and having a false positive. If the model predicts a transaction as a fraud but in the actual sense it is not a fraud (false positive), the cost of the misclassification is an inconvenience to the customer and bad customer representation. This can be corrected by analyzation of the transaction by human staff and contacting the cardholder. On the other hand, classifying a transaction as no-fraudulent when the case is actually fraudulent (false negative) has a higher cost. The cost of this misclassification is the actual amount of the fraudulent activity.

From the analysis of this dataset, from the 7506 fraudulent transactions, the total amount of money lost is $3,988,089 and this is a huge amount of money. For this case, false negatives are actually very costly to both the customers and the bank. We could analyze the amount of money lost for the customers with high amount of fraudulent transactions.

name	sum_amt
Jennifer Scott	14319.39
Scott Martin	13656.63
Micheal Walters	12011.59
Susan Garcia	11457.11
Charles Preston	11061.27
Jasmine Wade	10994.76

Table 7.1 : Name of the customers and the sum of amount lost to fraud

We can see the amount the highest customer lost during the two years of this study. "**Jennifer Scott**" lost **$14,319.39** to fraudulent transactions. This is a huge amount of money.

A classification metric that utilizes cost-sensitive classification can be used to put a higher cost on false positives as against false negatives. Since the cost of transaction is different for each transaction, it is not realistic to assign a cost value for the same class (fraud or no fraud). Cost-sensitive algorithms can be applied before training the model, while training the model (cost-sensitive logistic regression or cost-sensitive decision trees) or after training the model.

8.0. Pre-Modeling Analysis

The initial dataset has 23 data sets, this was reduced to 5 features by data exploration and feature importance. These 5 features are those that are used in the prediction of the model.

In the exploratory phase, feature engineering, data scaling, outlier analysis, boxcox transformation and missing data analysis has been done.

The various algorithms considered for the work are(S.P Maniraj et al., 2019);

1. Logistic Regression:

Strength

- This was chosen because it has the p-value feature that can help in the feature reduction by recognizing features that are not statistically significant. This feature can be removed due to the results and this prevents overfitting of the other models.
- The results of the logistic regression are easily interpretable. This helps in recognizing how change in a feature can affect the model output.
- Logistic regression can be regularized to avoid overfitting.

Weakness

- The major drawback of this model is that it is most useful when the relationship between the predictors and output is linear. They are not robust enough to handle complex relationships.

2. Random Forest

Strength

- These models are robust enough to handle outliers and also model non-linear relationships in a data.

Weakness

- They are prone to overfitting, but this was constrained by managing the number of splits and trees.

3. Support Vector Machines (SVM)

Strength

- These models can be readily used to model non-linear relationships by picking form the number of kernels available.
- They also robust enough to avoid overfitting.

Weakness

- These models are memory intensive.
- They are overly sensitive to scaled data. The data must be scaled to be used in the model.

Due to the "No Free Lunch" theorem which states that no one algorithm works best for every problem, we tried out the above algorithms to see what worked best for our problem. Each of the models would be validated to get the more useful and accurate model.

Validation Technique

The following metrics would be used to validate our model as being good enough or if we need to do more hyperparameter optimization.

1. **Recall** – This is the most important validation means. This is because we are dealing with a fraud data and it is more important the model predicts the maximum number of positive fraud labels. The loss associated with this is high if a fraud goes undetected.

2. **Precision** - The number of positive outcomes predicted correctly is also important. Though we want to detect as many frauds as possible, we do not want to have too many "false alarms". This might make the company lose customers.
3. **F1 Score:** This is a more robust metric as it considers both the precision and recall. This is because the higher the recall, the lower the precision. The F1 is a balanced metric.
4. **Kappa score:** This would show how much our model does in relation to the accuracy of the model without any model prediction. This shows how the model does compared to random chance.
5. **Accuracy**

9.0. Modeling

Random Forest: The 10-fold cross-validation is done on the model and also 3 repeats. This is to reduce the tendency of overfitting the model on the training set. The model with accurate value for mtry with an accuracy of 0.967 (96.7%).

mtry	Accuracy	Kappa	AccuracySD	KappaSD
1	0.8450788	0.6901573	0.009816423	0.019634272
2	0.9394497	0.8788992	0.009393838	0.018788577
3	0.9644957	0.9289915	0.004454989	0.008910064

Table 9.1: Hyper paramter Optimization of Random Forest

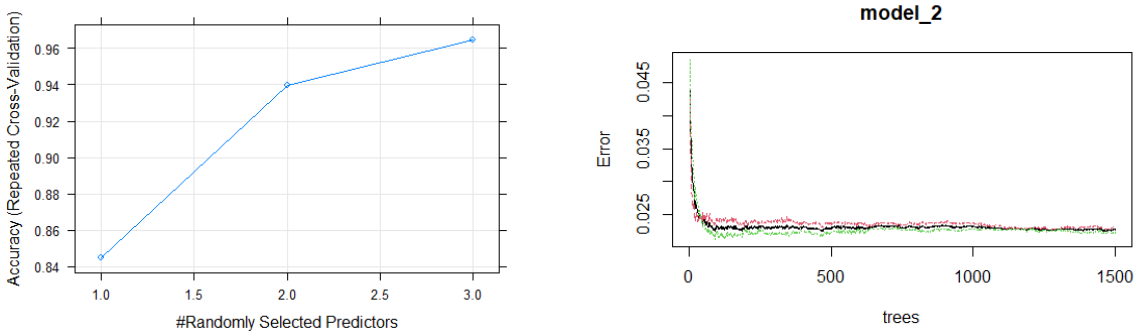


Fig. 9.0.1 Hyper paramter Optimization of Random Forest

The highest accuracy was obtained when the nodes was split with 3 features at each node.

After executing the code, the output is produced that shows the number of decision trees developed using the classification model for random forest algorithms, i.e. 1500 decision trees. The confusion matrix is also known as the error matrix that shows the visualization of the performance of the classification model.

```
Call:
 randomForest(formula = is_fraud ~ ., data = fraud_new, ntree = 1500,      mtry = 3, trControl = control)
               Type of random forest: classification
                     Number of trees: 1500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 2.28%
Confusion matrix:
     0    1 class.error
0 7332  174  0.02318145
1  168 7338  0.02238209
```

The random forest model used the Gini decrease to split the nodes of the tree and from the random forest model, we extracted the feature importance of all the parameters in the model. The most important parameter is the amount and the least important parameter is "city_pop". The iteration over the feature extraction and model evaluation shows that if city pop is removed, the accuracy of the model decreases.

Feature	Mean Decrease Gini
Amount	4904.1009
Time	1376.6162
Category	855.4762
Age	211.5520
City_pop	157.6537

Table 9.2: Feature Importance from Random Forest

Logistic Regression Model

Logistic regression was run using some selected variables and the summary of the fit indicated variable importance. The final model after iteration is as shown below;

$$fit2 = glm(data = under\_sample, is\_fraud \sim amt + city\_pop + Time + age + category, family = "binomial")$$

9.1. Result and Validation Analysis

- Random Forest
- SVM
- Logistic Regression Summary

```
Call:
glm(formula = is_fraud ~ amt + city_pop + Time + age + category,
    family = "binomial", data = under_over)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-8.4904  -0.6881  -0.0799   0.4809   2.0292

Coefficients:
                         Estimate Std. Error z value Pr(>|z|)
(Intercept)            -3.485e+00  4.476e-02 -77.849  < 2e-16 ***
amt                     1.072e-02  6.638e-05 161.488  < 2e-16 ***
city_pop               -1.810e-08  2.163e-08  -0.837    0.403
Time                   -7.220e-09  4.893e-10 -14.756  < 2e-16 ***
age                     2.480e-03  3.655e-04   6.785 1.16e-11 ***
categoryfood_dining     1.504e+00  4.582e-02  32.820  < 2e-16 ***
categorygas_transport   2.841e+00  4.158e-02  68.336  < 2e-16 ***
categorygrocery_net     2.428e+00  4.859e-02  49.974  < 2e-16 ***
categorygrocery_pos     1.858e+00  3.816e-02  48.685  < 2e-16 ***
categoryhealth_fitness  1.646e+00  4.734e-02  34.773  < 2e-16 ***
categoryhome            9.201e-01  4.409e-02  20.868  < 2e-16 ***
categorykids_pets       1.953e+00  4.403e-02  44.361  < 2e-16 ***
categorymisc_net       -9.465e-01  6.349e-02 -14.907  < 2e-16 ***
categorymisc_pos        2.152e+00  4.648e-02  46.293  < 2e-16 ***
categorypersonal_care   2.160e+00  4.487e-02  48.131  < 2e-16 ***
categoryshopping_net   -2.109e+00  7.247e-02 -29.098  < 2e-16 ***
categoryshopping_pos   -1.772e+00  6.318e-02 -28.049  < 2e-16 ***
categorytravel          2.421e+00  5.269e-02  45.961  < 2e-16 ***
```

From the summary of the model, we see that city_pop is not relevant to the prediction of fraud cases in the transactions. This is reflected in its p-value, there is a probability of 0.403 that its contribution is as a result of random chance. Also, when compared with other variables with p-values considerably less than the 0.05 standard, we confirm that the feature city_pop is not a relevant feature.

- Amt: The estimate of the variable amt, amount spent in a transaction, is 0.001077 which indicates the change in the log of odds for a unit change in Amt while other predictors are held constant. This is equivalent to an odds value of 1.001, which doesn't show a significant change in odds.
- Food_dining: The estimates of the category food_dining indicates an odds ratio of 4.49 which indicates that there is an odds increase of 349% of the transaction to be a fraud if the credit card is used to buy food items.
- Shopping_pos: It is also interesting to note the estimate value of the category shopping_pos, it has a log odds of -1.772 which is equivalent to an odds of 0.17. This implies that there is an 83% reduction in the odds of being a fraudulent transaction is the category of the transaction is shopping_pos while other predictors are held constant

		Logistic Regression model				
		Recall	Precision	Accuracy	Kappa	F1
undersample model	Undersampled data	0.8674	0.7503	0.8178	0.6356	0.804612
	Imbalanced Train data	0.9695	0.1281	0.9695	0.2105	0.226299
	Imbalanced Test data	**0.7273**	**0.0708**	**0.9621**	**0.1229**	**0.129039**
oversample model	Oversampled data	0.8814	0.751	0.8256	0.651	0.810992
	Imbalanced Train data	0.7359	0.1324	0.9706	0.2168	0.224423
	Imbalanced Test data	**0.7254**	**0.0772**	**0.9655**	**0.1335**	**0.139549**
		Decision Tree Model				
		Recall	Precision	Accuracy	Kappa	F1
undersample model	Undersampled data	0.8987	0.9572	0.9247	0.8493	0.927028
	Imbalanced Train data	0.9572	0.0505	0.8957	0.086	0.095938
	Imbalanced Test data	**0.9473**	**0.0338**	**0.8954**	**0.0583**	**0.065271**
oversample model	Oversampled data	0.9534	0.9186	0.9371	0.8742	0.935677
	Imbalanced Train data	0.9187	0.1064	0.9549	0.1822	0.190712
	Imbalanced Test data	**0.9082**	**0.07279**	**0.955**	**0.1285**	**0.134778**
		Random Forest Model				
		Recall	Precision	Accuracy	Kappa	F1
undersample model	Undersampled data	1	0.9991	0.9995	0.9991	0.99955
	Imbalanced Train data	1	0.1533	0.968	0.2585	0.265846
	Imbalanced Test data	**0.9636**	**0.0983**	**0.9657**	**0.1726**	**0.178401**
oversample model	Oversampled data	1	1	1	1	1
	Imbalanced Train data	1	0.4575	0.9931	0.6248	0.627787
	Imbalanced Test data	**0.8597**	**0.3032**	**0.9918**	**0.4452**	**0.448295**

Table 9.3: Summary of Model Metrics

- One basic fact to point out is that accuracy is useless when working with imbalanced data because the misclassification rate is very high. This means we can assign the negative class to all the instances and get an accuracy value of 99%. Nevertheless, we reported our accuracy values just for visualization. Except for the

decision tree model, the models have the highest precision and recall values for the balanced sampled (under-sampled and over-sampled) training data than for the imbalanced training and testing data.

- After building the models on the sampled data, the models were applied on the imbalanced training data to have a feel of how well fraud cases are detected before applying them on the skewed test data. However, we will focus our discussion on the imbalanced test data since we don't care as much about the training data. In addition, over-sampling gave better results than under-sampling except for the decision tree model where recall is higher for the under-sampling model. Nevertheless, precision is higher for oversampling in all the three models reported in Table 9.3. Over-sampling was restricted to 200,000 instances of the original 1.3 million instances in the training data because of the available computational power for training the models. Hence, what we call oversampling is actually an under-sampling of the negative cases and oversampling of the positive cases (fraud cases).
- F1 score and Kappa both take precision and recall into consideration. As seen in Table 9.3 the metrics of the under-sampling models on the skewed test data, the best model is the Random Forest and it gave an F1 score of 0.1784 while Decision Tree gave the worst F1 value of 0.0653 and a Kappa value of 0.0583, this clearly shows that the prediction of the model is very close to a random guess.
- On the other hand, Random Forest on the over-sampling data gave the best F1 score and Kappa on the skewed test data, values of 0.4483 and 0.4452 respectively, while Decision Tree again gave the lowest F1 and Kappa scores of 0.1348 and 0.1285 respectively. Since over-sampling generally produced better results, the following explanation of results will be based only on the models developed on the oversampled data.
- The lowest recall (0.7254) was obtained from logistic regression. It follows that 73% of the actual fraud cases were detected by the logistic regression model. This is not very good as 27% of fraud cases can go undetected and can make the customers lose a lot of money while ruining the reputation of the credit card company.
- Decision Tree gave a recall of 0.9082 on the test data while random forest gave 0.8597. However, precision values are terribly low for both logistic regression and random forest, 0.0772 and 0.0728 respectively. This means about 7% of the fraud cases detected by the model are actually fraud cases while the other 93% are false positives. This will surely annoy customers when most of their normal transactions are flagged as fraud.
- Random forest gave the best precision of 0.3032, it is not a great value but surely the best of the three. Although it is important to maximize recall in our application so as to minimize undetected or eliminate fraud cases nevertheless, we need a trade-off between recall and precision so as not to annoy and lose valuable customers. Considering the tradeoff, random forest on the over-sampled data gave the best metrics. This is also seen in the Kappa value of 0.4452.

9.2. Group Based on amt

	Recall	Precision	F1
Group 1	0.8917	0.057	0.11
Group 2	0.99	0.2896	0.448
Group 3	1	0.33	0.4999

Table 9.4: Summary of Model Metrics for Group 1, 2 & 3

- The random forest model was built on each of the customer group. This result in the table shows that this method worked best with amy above $1000 as it gives recall, precison and F1 score more than that gotten from having one model on the data set. The Group 2 (100-1000) also performed well and has a higher F1 score on the group but a slightly lower prescion score. The Recall for group 2 was also better than the first general model. The group 1 though it also have a higher recall, it has a very low precison score and also F1 score. The model would be best recommended for use in group 2 and group 3.

Conclusion

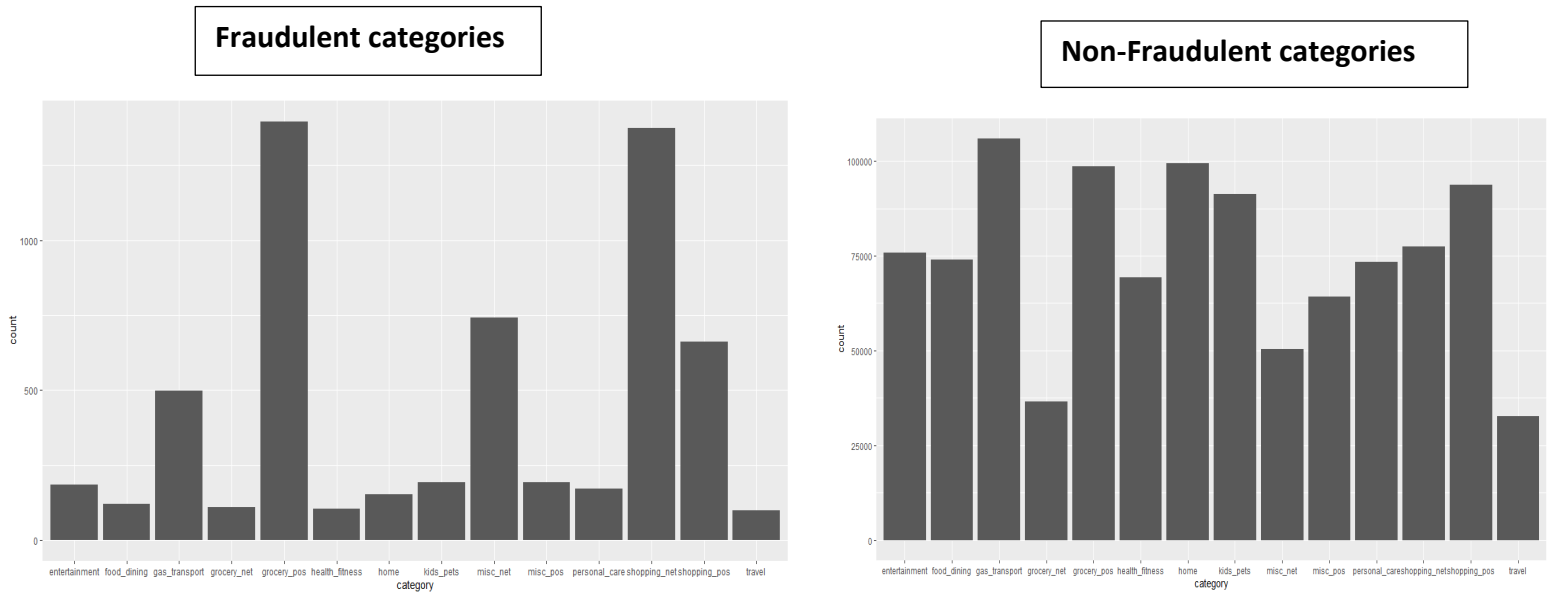
- The model developed for this study can be useful in the analysis of fraud transactions. The most accurate model is the Random Forest model. This model shows the highest recall, precision and F1.
- The alternative plan conducted in this work was dividing the dataset into three based on the amount of the transaction. The new analysis gave better result for the F1, recall and precision. This is because the new group is more representative of the customer spending habits.
- The variable amt of the data set is the most important feature. The chances of a transaction being fraudulent does not actually increase with the increase in the amount. Most fraud occurs between $500-$1000. This masked behavior was accurately depicted in our model.
- The time the transaction occurs is also very important as most of the frauds occur at night after 10 pm.
- Machine learning algorithms are very useful in a lot of real-life applications and this study has shown how it can help us detect fraud and reduce the amount lost to fraud daily.
- More room for improvement can be found in the dataset. The precision of the model increases when the size of the dataset is increased. An increase in the data volume will cause a positive difference in the model by making it more accurate in detecting frauds and reduce the number of false positives.
- However, we were limited by computational power in the amount of the available instances we could use in developing our models. Only 15% of the available instances could be used in developing the model on a balanced data. This means we could only use 200,000 instances which is a balanced random sample of the positive and negative cases.

References

S P Maniraj, Aditya Saini, Shadab Ahmed, & Swarna Deep Sarkar. (2019). Credit Card Fraud Detection using Machine Learning and Data Science. *International Journal of Engineering Research And*, *08*(09), 110–115.

Saputra, A., & Suharjito. (2019). Fraud detection using machine learning in e-commerce. *International Journal of Advanced Computer Science and Applications*, *10*(9), 332–339.

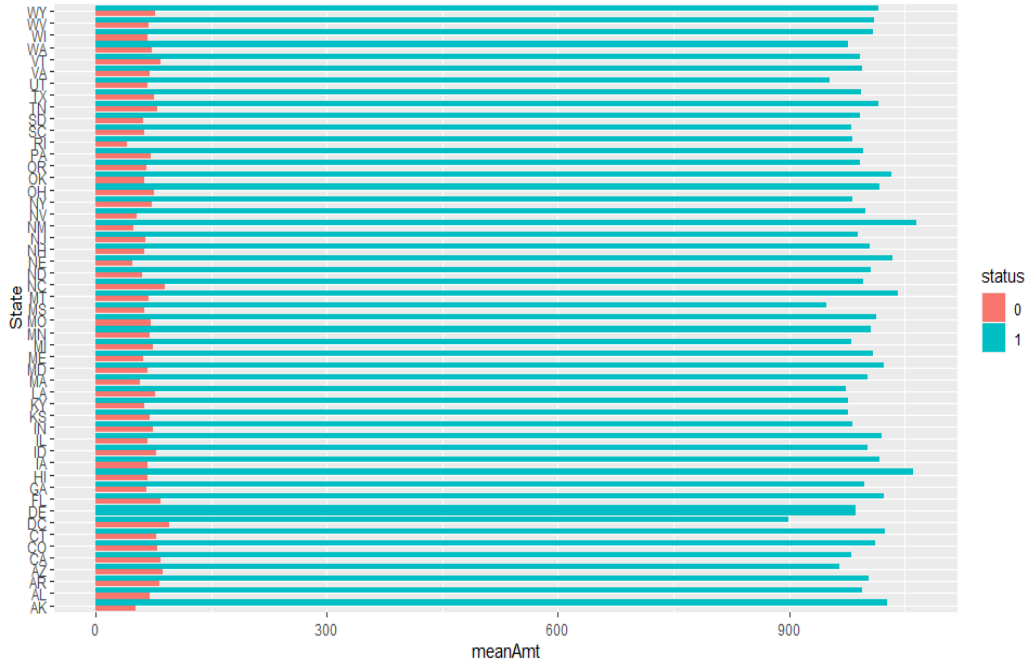
Appendix A



A1. Figure showing the categories of fraudulent and non-fraudulent transaction occurrences in the dataset. Fraudulent transaction occur mostly in Grocery_POS and Shopping_Net categories.

A2. Figure showing the merchants related to Grocery_POS and Shopping_Net categories. All merchants happen to experience fraudulent activities in both categories.

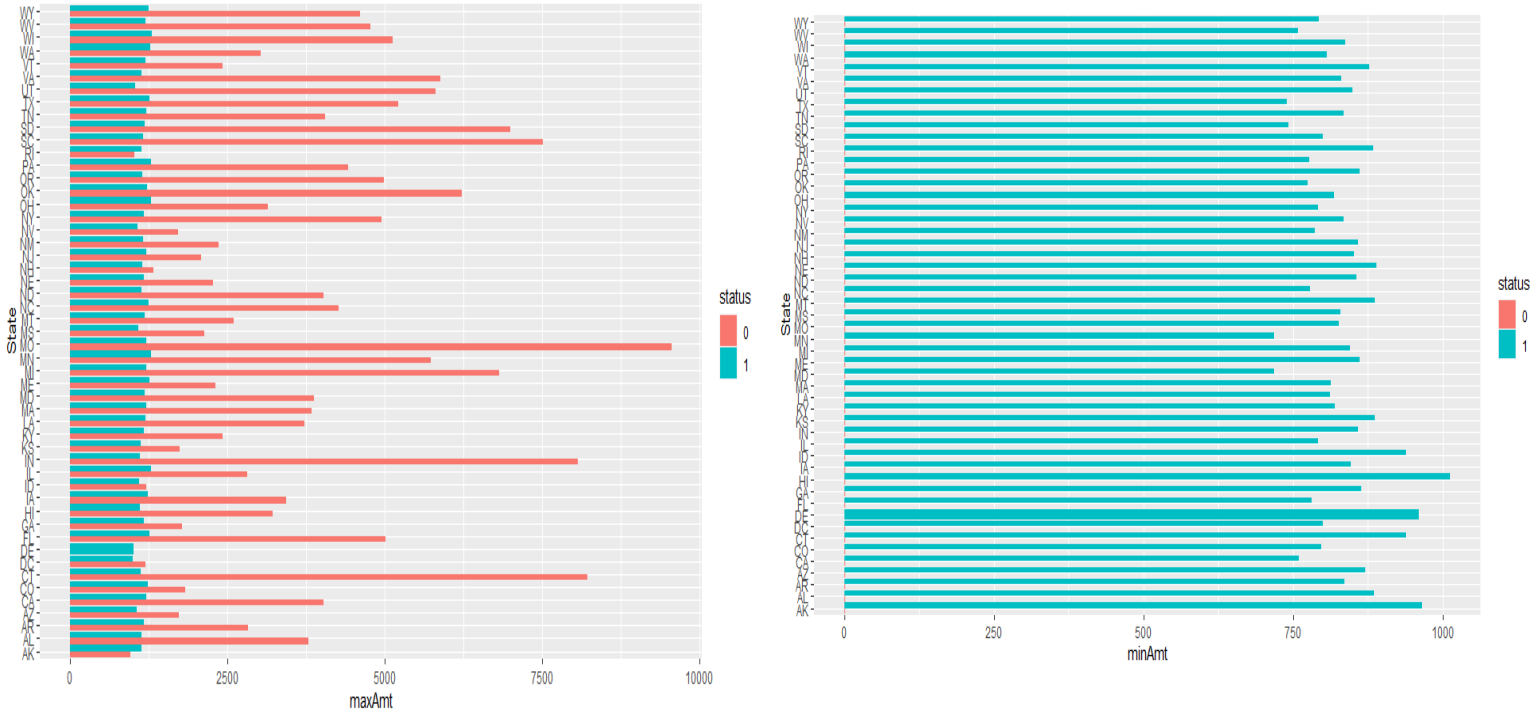
Mean Amount in Transactions



A3. Mean amounts in transactions by state. "1" referring to fraudulent and "0" referring to Non-fraudulent transactions.

Max. Amount in Transactions

Min. Amount in Transactions



A4. (a) Maximum amounts in transactions by state. (b) Minimum amounts in transactions by state "1" referring to fraudulent and "0" referring to Non-fraudulent transactions

APPENDIX B. Machine Learning Algorithm Results

SVM: Hyperparameter Optimization.

```
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 gamma cost
   0.1  100

- best performance: 0.04323172

   gamma cost      error  dispersion
1    0.1  100 0.04323172 0.005601052
2    0.5  100 0.04329852 0.004152385
3    1.0  100 0.04563021 0.005698543
4    2.0  100 0.05029310 0.006430013
5    3.0  100 0.05182528 0.006609330
6    4.0  100 0.05369039 0.004702655
```

9.3. Model Evaluation

1. **SVM**

- Train

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 7211  250
         1  295 7256

               Accuracy : 0.9637
                 95% CI : (0.9606, 0.9666)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.9274

 Mcnemar's Test P-Value : 0.05946

            Sensitivity : 0.9667
            Specificity : 0.9607
         Pos Pred Value : 0.9609
         Neg Pred Value : 0.9665
             Prevalence : 0.5000
         Detection Rate : 0.4833
   Detection Prevalence : 0.5030
      Balanced Accuracy : 0.9637

       'Positive' Class : 1

     Sensitivity          Specificity       Pos Pred Value       Neg Pred Value            Precision
       0.9666933           0.9606981            0.9609323            0.9664924            0.9609323
          Recall                   F1           Prevalence       Detection Rate Detection Prevalence
       0.9666933           0.9638042            0.5000000            0.4833467            0.5029976
Balanced Accuracy
       0.9636957
```

- Test

```
Confusion Matrix and Statistics

          Reference
Prediction      0      1
         0 530891     99
         1  22683   2046

               Accuracy : 0.959
                 95% CI : (0.9585, 0.9595)
    No Information Rate : 0.9961
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1462

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.953846
            Specificity : 0.959024
         Pos Pred Value : 0.082737
         Neg Pred Value : 0.999814
             Prevalence : 0.003860
         Detection Rate : 0.003682
   Detection Prevalence : 0.044499
      Balanced Accuracy : 0.956435

       'Positive' Class : 1
```

Sensitivity	Specificity	Pos Pred Value	Neg Pred Value	Precision
0.953846154	0.959024448	0.082736868	0.999813556	0.082736868
Recall	**F1**	**Prevalence**	**Detection Rate**	**Detection Prevalence**
0.953846154	0.152266131	0.003859864	0.003681717	0.044499108
Balanced Accuracy				
0.956435301				

2. Random Forest
   - **Train**

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 7332  168
         1  174 7338

               Accuracy : 0.9772
                 95% CI : (0.9747, 0.9795)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9544

 Mcnemar's Test P-Value : 0.7869

            Sensitivity : 0.9776
            Specificity : 0.9768
         Pos Pred Value : 0.9768
         Neg Pred Value : 0.9776
             Prevalence : 0.5000
         Detection Rate : 0.4888
   Detection Prevalence : 0.5004
      Balanced Accuracy : 0.9772

       'Positive' Class : 1
```

Sensitivity	Specificity	Pos Pred Value	Neg Pred Value
0.9776179	0.9768185	0.9768371	0.9776000
Precision	**Recall**	**F1**	**Prevalence**
0.9768371	0.9776179	0.9772273	0.5000000
Detection Rate	**Detection Prevalence**	**Balanced Accuracy**	
0.4888090	0.5003997	0.9772182	

- **Test**

```
Confusion Matrix and Statistics

          Reference
Prediction      0      1
         0 542804     70
         1  10770   2075

               Accuracy : 0.9805
                 95% CI : (0.9801, 0.9809)
    No Information Rate : 0.9961
    P-Value [Acc > NIR] : 1

                  Kappa : 0.272

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.967366
            Specificity : 0.980545
         Pos Pred Value : 0.161541
         Neg Pred Value : 0.999871
             Prevalence : 0.003860
         Detection Rate : 0.003734
   Detection Prevalence : 0.023114
      Balanced Accuracy : 0.973955

       'Positive' Class : 1

     Sensitivity          Specificity       Pos Pred Value       Neg Pred Value
     0.967365967          0.980544607          0.161541456          0.999871057
       Precision               Recall                   F1           Prevalence
     0.161541456          0.967365967          0.276851234          0.003859864
  Detection Rate Detection Prevalence    Balanced Accuracy
     0.003733901          0.023114200          0.973955287
```

3. Logistic Regression Model

Logistic regression was run using some selected variables and the summary of the fit gave an indication of variable importance. This model was used for part of the feature selection.

```
Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
(Intercept)         -2.090e+00  5.129e-01  -4.075 4.60e-05 ***
amt                  1.081e-02  2.477e-04  43.666  < 2e-16 ***
genderM             -8.785e-02  5.015e-02  -1.752 0.079805 .
stateAL             -1.384e+00  5.005e-01  -2.764 0.005704 **
stateAR             -1.297e+00  5.063e-01  -2.562 0.010420 *
stateAZ             -2.183e+00  5.616e-01  -3.887 0.000101 ***
stateCA             -1.503e+00  4.945e-01  -3.040 0.002367 **
stateCO             -1.303e+00  5.212e-01  -2.500 0.012424 *
stateCT             -1.942e+00  7.162e-01  -2.712 0.006691 **
stateDC             -1.563e+00  6.352e-01  -2.461 0.013866 *
stateDE              1.159e+01  1.432e+02   0.081 0.935478
```

The table above was obtained from the summary of the model fit. From the z value, we see that variable genderM is not as important as amt, and the probability that stateDE's contribution is the same as random chance is very high, 0.935. We ran a second model where the variables genderM and state were dropped.

The final model is as shown below

$$fit2 = glm(data = under\_sample, is\_fraud \sim amt + city\_pop + Time + age + category, family = "binomial")$$

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 6684  822
         1 1842 5664

               Accuracy : 0.8225
                 95% CI : (0.8163, 0.8286)
    No Information Rate : 0.5679
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6451

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.8733
            Specificity : 0.7840
         Pos Pred Value : 0.7546
         Neg Pred Value : 0.8905
             Prevalence : 0.4321
         Detection Rate : 0.3773
   Detection Prevalence : 0.5000
      Balanced Accuracy : 0.8286

       'Positive' Class : 1
```

Train data (under-sampled, balanced)

```
Confusion Matrix and Statistics

          Reference
Prediction      0      1
         0 533102    585
         1  20472   1560

               Accuracy : 0.9621
                 95% CI : (0.9616, 0.9626)
    No Information Rate : 0.9961
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1229

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.727273
            Specificity : 0.963018
         Pos Pred Value : 0.070806
         Neg Pred Value : 0.998904
             Prevalence : 0.003860
         Detection Rate : 0.002807
   Detection Prevalence : 0.039646
      Balanced Accuracy : 0.845146

       'Positive' Class : 1
```

Test data (imbalanced)

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 90287 10073
         1 24812 74828

               Accuracy : 0.8256
                 95% CI : (0.8239, 0.8272)
    No Information Rate : 0.5755
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.651

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.8814
            Specificity : 0.7844
         Pos Pred Value : 0.7510
         Neg Pred Value : 0.8996
             Prevalence : 0.4245
         Detection Rate : 0.3741
   Detection Prevalence : 0.4982
      Balanced Accuracy : 0.8329

       'Positive' Class : 1
```

Train data (over-sampled, balanced)

```
Confusion Matrix and Statistics

          Reference
Prediction      0      1
         0 534968    589
         1  18606   1556

               Accuracy : 0.9655
                 95% CI : (0.965, 0.9659)
    No Information Rate : 0.9961
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1335

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.72541
            Specificity : 0.96639
         Pos Pred Value : 0.07717
         Neg Pred Value : 0.99890
             Prevalence : 0.00386
         Detection Rate : 0.00280
   Detection Prevalence : 0.03628
      Balanced Accuracy : 0.84590

       'Positive' Class : 1
```

Test data (imbalanced)

4. Decision Tree Model

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 6696  810
         1  321 7185

               Accuracy : 0.9247
                 95% CI : (0.9203, 0.9288)
    No Information Rate : 0.5326
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8493

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.8987
            Specificity : 0.9543
         Pos Pred Value : 0.9572
         Neg Pred Value : 0.8921
             Prevalence : 0.5326
         Detection Rate : 0.4786
   Detection Prevalence : 0.5000
      Balanced Accuracy : 0.9265

       'Positive' Class : 1
```

Train data (under-sampled, balanced)

```
Confusion Matrix and Statistics

          Reference
Prediction       0       1
         0 1154208     321
         1  134961    7185

               Accuracy : 0.8957
                 95% CI : (0.8951, 0.8962)
    No Information Rate : 0.9942
    P-Value [Acc > NIR] : 1

                  Kappa : 0.086

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.957234
            Specificity : 0.895312
         Pos Pred Value : 0.050547
         Neg Pred Value : 0.999722
             Prevalence : 0.005789
         Detection Rate : 0.005541
   Detection Prevalence : 0.109623
      Balanced Accuracy : 0.926273

       'Positive' Class : 1
```

Test data (imbalanced)

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 95884  4476
         1  8106 91534

               Accuracy : 0.9371
                 95% CI : (0.936, 0.9382)
    No Information Rate : 0.52
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8742

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9534
            Specificity : 0.9221
         Pos Pred Value : 0.9186
         Neg Pred Value : 0.9554
             Prevalence : 0.4800
         Detection Rate : 0.4577
   Detection Prevalence : 0.4982
      Balanced Accuracy : 0.9377

       'Positive' Class : 1
```

Train data (over-sampled, balanced)

```
Confusion Matrix and Statistics

          Reference
Prediction      0      1
         0 528760    197
         1  24814   1948

               Accuracy : 0.955
                 95% CI : (0.9544, 0.9555)
    No Information Rate : 0.9961
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1285

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.908159
            Specificity : 0.955175
         Pos Pred Value : 0.072790
         Neg Pred Value : 0.999628
             Prevalence : 0.003860
         Detection Rate : 0.003505
   Detection Prevalence : 0.048157
      Balanced Accuracy : 0.931667

       'Positive' Class : 1
```

Test data (imbalanced)

5. Random Forest

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 7499    0
         1    7 7506

               Accuracy : 0.9995
                 95% CI : (0.999, 0.9998)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.9991

 Mcnemar's Test P-Value : 0.02334

            Sensitivity : 1.0000
            Specificity : 0.9991
         Pos Pred Value : 0.9991
         Neg Pred Value : 1.0000
             Prevalence : 0.5000
         Detection Rate : 0.5000
   Detection Prevalence : 0.5005
      Balanced Accuracy : 0.9995

       'Positive' Class : 1
```

Train data (under-sampled, balanced)

```
Confusion Matrix and Statistics

          Reference
Prediction      0      1
         0 534617     78
         1  18957   2067

               Accuracy : 0.9657
                 95% CI : (0.9653, 0.9662)
    No Information Rate : 0.9961
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1726

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.96364
            Specificity : 0.96576
         Pos Pred Value : 0.09832
         Neg Pred Value : 0.99985
             Prevalence : 0.00386
         Detection Rate : 0.00372
   Detection Prevalence : 0.03783
      Balanced Accuracy : 0.96470

       'Positive' Class : 1
```

Test data (imbalanced)

```
Confusion Matrix and Statistics

          Reference
Prediction      0      1
         0 100360      0
         1      0  99640

               Accuracy : 1
                 95% CI : (1, 1)
    No Information Rate : 0.5018
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1

 Mcnemar's Test P-Value : NA

            Sensitivity : 1.0000
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 1.0000
             Prevalence : 0.4982
         Detection Rate : 0.4982
   Detection Prevalence : 0.4982
      Balanced Accuracy : 1.0000

       'Positive' Class : 1
```

Train data (over-sampled, balanced)

```
Confusion Matrix and Statistics

          Reference
Prediction      0      1
         0 549337    301
         1   4237   1844

               Accuracy : 0.9918
                 95% CI : (0.9916, 0.9921)
    No Information Rate : 0.9961
    P-Value [Acc > NIR] : 1

                  Kappa : 0.4452

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.859674
            Specificity : 0.992346
         Pos Pred Value : 0.303240
         Neg Pred Value : 0.999452
             Prevalence : 0.003860
         Detection Rate : 0.003318
   Detection Prevalence : 0.010943
      Balanced Accuracy : 0.926010

       'Positive' Class : 1
```

Test data (imbalanced)

Grouped Data

```
> test_y_RF_below$byClass
         Sensitivity          Specificity       Pos Pred Value       Neg Pred Value
        0.8916666667         0.9844191390         0.0569073262         0.9998839796
           Precision               Recall                   F1           Prevalence
        0.0569073262         0.8916666667         0.1069866267         0.0010532830
      Detection Rate Detection Prevalence    Balanced Accuracy
        0.0009391773         0.0165036272         0.9380429028

> test_y_RF_btw$byClass
         Sensitivity          Specificity       Pos Pred Value       Neg Pred Value
          0.98776098           0.96532423           0.28969595           0.99981850
           Precision               Recall                   F1           Prevalence
          0.28969595           0.98776098           0.44800000           0.01411557
      Detection Rate Detection Prevalence    Balanced Accuracy
          0.01394281           0.04812910           0.97654261
> test_y_RF_above = caret::confusionMatrix(RF.test_above, amt_above_test$is_fraud, positive =
 "1")
> test_y_RF_above$byClass
         Sensitivity          Specificity       Pos Pred Value       Neg Pred Value
           1.0000000            0.5761285            0.3325301            1.0000000
           Precision               Recall                   F1           Prevalence
           0.3325301            1.0000000            0.4990958            0.1743525
      Detection Rate Detection Prevalence    Balanced Accuracy
           0.1743525            0.5243209            0.7880643
```