**Akhilesh Yadav Gaddam**

**Anil Joshi**

**Credit Card   
Fraud Detection**

**Stat 642**

**Credit Card Fraud Detection**

**Executive Summary**

The main objectives of this project are to predict fraudulent transactions from genuine ones and to understand the various attributes that help determine whether a credit card transaction is a fraudulent one or not. We processed the data using various statistical tools to address the imbalanced nature of our dataset. We used five classifier techniques: logistic regression, k-nearest neighbor, neural network, support vector machine and random forest for accurate predictions. Then, we used ensemble method of majority vote to improve our results by combining several models. Furthermore, we evaluated our ensemble model and further improved the accuracy with an advanced stacking method. Finally, we identified the best indicators of detecting fraudulent credit card transactions.

**Introduction**

The illegal use of credit card or its information without the cardholder’s knowledge is credit card fraud. Activities of this kind takes place all over the world. A recent Nielsen report estimates that in 2016, the losses from credit card fraud topped $24.71 billion worldwide, an all-time high. This is an increase of 12% over the previous year. According to Barclays, 47% of the global credit card fraud occurs here in the United States. Alarmingly, 46% of Americans have fallen victim to these fraud activities[[1]](#footnote-2).

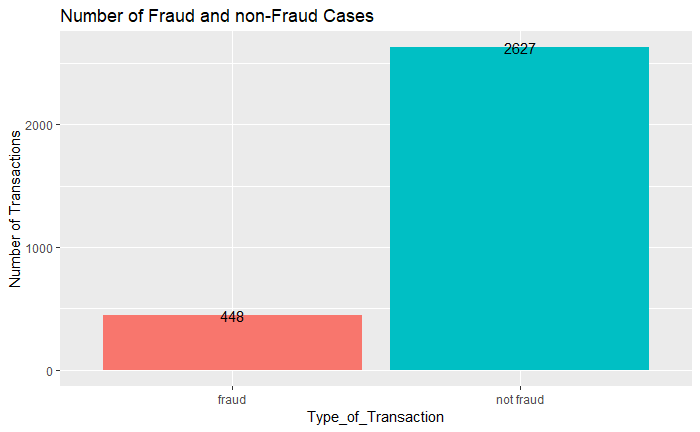
Past research and literature suggest several variables that are most helpful in determining fraudulent transactions. For instance, sudden change in pattern of usage and uninformed use outside of usual geographical area are key pointers of credit card fraud. These mostly relate to historical behavior of card usage. Sometimes, lost or stolen and cancelled cards are also tried to be used. This usually happens when someone who is not the owner of the card tries to use cards for fraud activities.

**Data Overview**

The dataset for this project has been obtained from Kaggle. It has 11 attributes of merchant transaction and 3075 rows of transaction details. The attributes from the dataset are listed below:

|  |  |
| --- | --- |
| Merchant ID | Is it a foreign transaction? |
| Average amount/transaction/day | Is the transaction from a high-risk country? |
| Transaction amount | Daily chargeback amount average |
| Is the transaction declined? | 6-month average chargeback amount |
| Total no. of declines per day | 6-month chargeback frequency |
| Is the transaction fraudulent? |  |

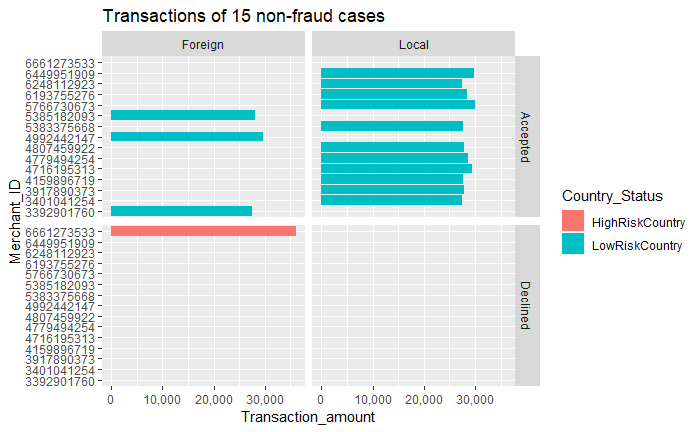
Since merchants are the intermediaries between card service providers and card users, there are instances of multiple transactions from single merchants. It is an imbalanced dataset because only 14.6% (448) of transactions are classified as fraud and 85.4% (2627) are classified as genuine. This holds true in real life too since technological advancements and timely detection of fraudulent activities has limited the number of such instances to low numbers.

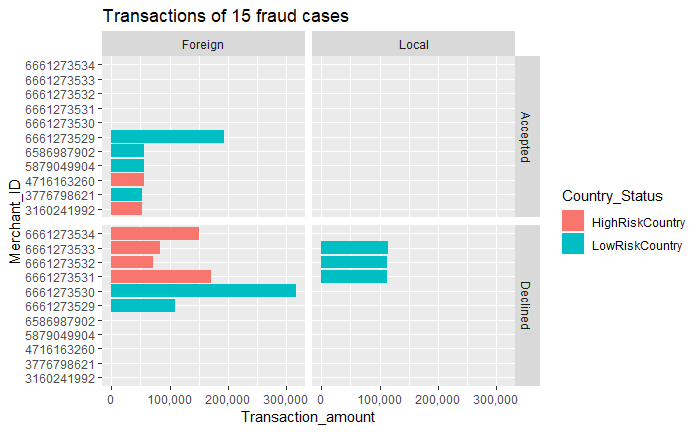


**Understanding fraud transactions**

1.Which factor is highly associated with fraud transactions?

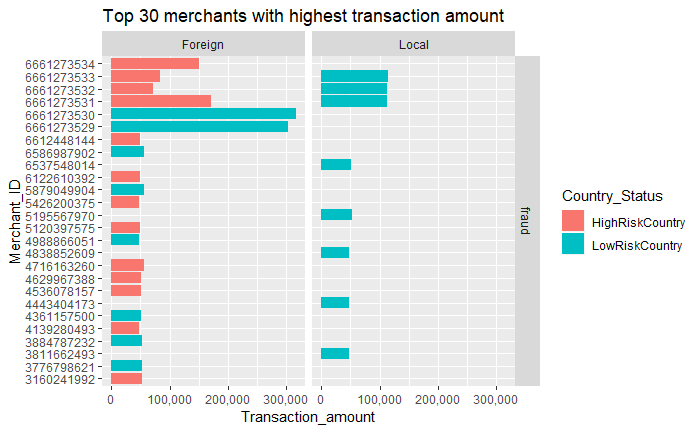
Below graph shows the transactions of 15 fraud and not fraud cases. Most of the transactions done by the non fraud cases are done within the country whereas Fraud cases were international. Moreover, almost all transactions of non fraud cases were accpected but fraud-Intenational transactions has few declines.





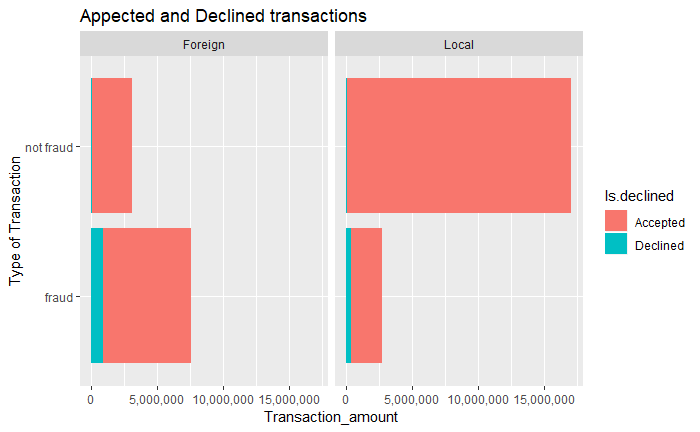
2.Top 30 merchants with high transaction amount

Top 30 merchants by total transaction amount are all fraud transaction. High transaction amount can also be a factor associated with fraud transactions. Most of them were also transacted to High Risk Country which means countries with high cybercrimes.



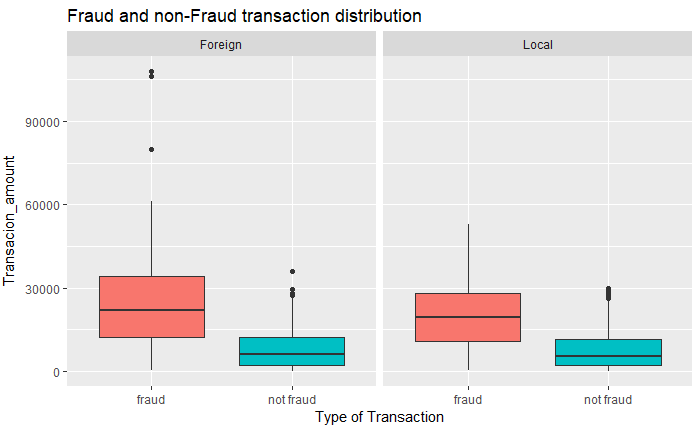
3.Accepted and Declined transactions

Almost all non-fraud transactions were accepted which possibly could be the reason of less transaction amount compared to fraud transaction and also high local transactions than foreign transactions. Fraud transaction faced decline close to 20% which probably could be the case for unusual high amount of international transaction to a high risk country.



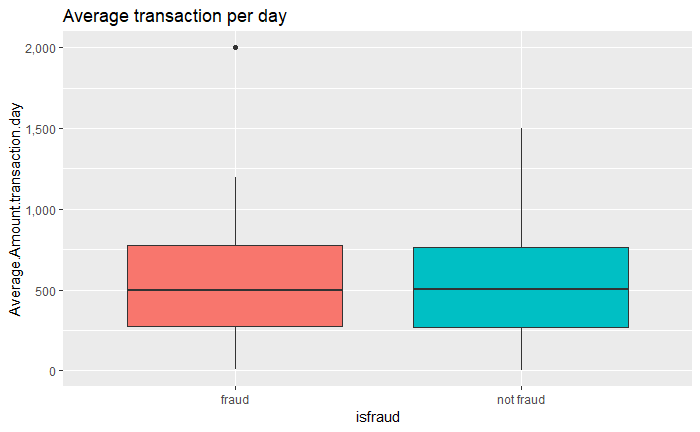
4. Outliers

Maximum transaction amount of non fraud cases is 30,000 in local and 36,000 in international. The maximum foreign transaction for fraud cases goes upto 108000 and the upper limit is 61200 which is very high than the maximum transaction non-fraud cases. Similarly local fraud transactions have maximum transaction higher than non-fraud transactions. So, there is a high possiblity for a transaction to be fraud if it crosses the maximum transaction of non-fraud transaction.



5.Average transaction per day

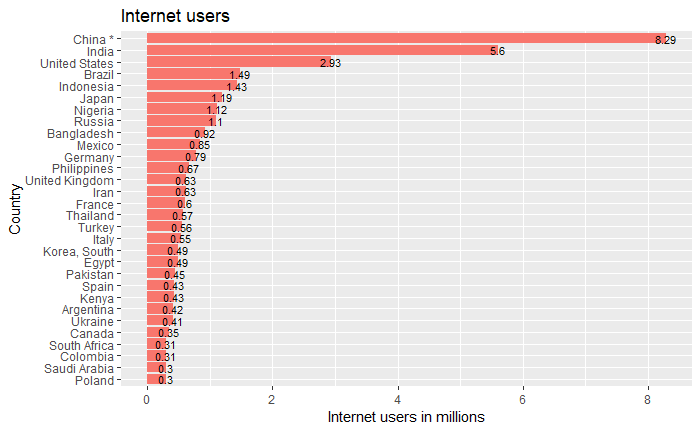
Average transaction per day done by fraud and non-fraud merchants. Assuming this as the threshold transaction of all merchants in the dataset, a mechant id with this threshold transaction per day and if there is an unusually huge amount of transaction might be a fraud transaction

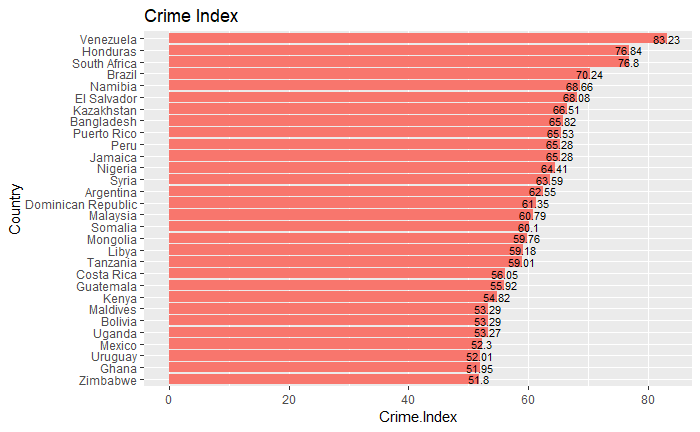


Based on the plots it is evident that foreign transaction with high transaction amount to a high risk country from a merchant id with theshold average transaction per day has a high possibility of being a fraud transction.

High Risk Countries

Below two plots shows the top 30 countries with high internet users and high crime rates. Some of the countries in the top 30 are common in both the plots. These countries have high internet users and high crime rates which means that there is a possibility of high cyber crime rates. Countries like Argentina, Bangladesh, Brazil, Kenya, Mexico, Nigeria and South Africa are there in both top 30 high internert users and crime rates, so these countries can be some of the high risk countries where the international transaction to these countries of a transaction amount more than the threshold amound might be a fraud transaction.





**Data processing**

As all the merchants had done either fraud transactions or genuine transactions, it will be a better way to train a model with a dataset based on transactions. In this way any merchant can be blocked when the machine detects a fraud transaction done by the merchant.

**Over-Sampling / Under-Sampling:** Over-Sampling duplicates the samples with minority class within the dataset whereas under-sampling would remove the samples with majority class from the dataset. They both involve using a bias to select more samples from one class than from another. The number of fraud transactions were oversampled, and the number of non-fraud transactions were under sampled in order to make an equal ratio or balanced class. Moreover, this makes the model not to be biased over one class.

*R Package: ROSE*

**K-Fold Cross Validation:** The K parameter in the cross validation refers to the number of subsets a dataset is to be split into. The dataset is shuffled randomly and split into K subsets where K is the testing dataset and K-1 are the training datasets. The model gets fitted over the K-1 training datasets, predicts the Kth dataset and validates the accuracy. This is repeated K times and the model with the maximum accuracy is chosen. 10-fold cross validation (K=10) was chosen in this case as it gives a smaller bias of the technique and less variance between results.

*R Package: caret*

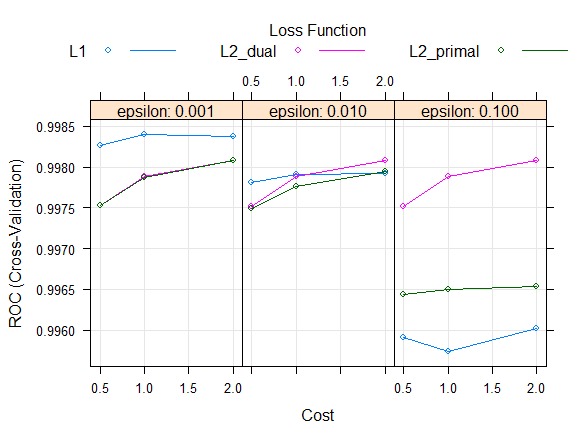
**Data Analysis**

Classification is a supervised learning approach in which the machine learns from the patterns of an input labelled dataset given to it and the uses this learning to classify new observations. The categories of the labels are the classes which can be a bi-class, multi-class or ordinal class. The dataset in this project has a bi-class – Fraud or not-Fraud and is dependent on other variables. The dataset was partitioned into 75% and 25% where the former was the training dataset and the later was testing dataset. Five different classification algorithms along with 10-fold cross validation were used to train five models with the training dataset. Finally, an Ensemble model was generated by majority voting from the five models.

**Logistic Regression:** Logistic Regression is both regression and classification algorithm which does regression analysis (examining the relationship between one dependent binary variable and two or more other variables of any type) when the dependent variable has binary class and predicts the class for new observations based on this relationship. The ‘train’ function in ‘caret’ package was used to perform 10-fold cross validation. High performance model is chosen by optimizing the cost parameter using,

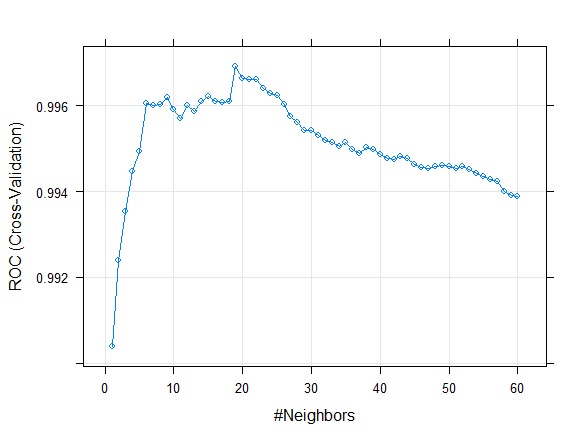
* Method = “regLogistic” - L2 Regularized Logistic Regression. L2 regularization is Ridge regularization that optimizes the model using primal and dual methods. L2 would add a squared magnitude to the cost function to compensate the loss function and optimizes it. L1 regularization is LASSO which adds an absolute value of magnitude to the cost function to compensate the loss function. L2 is used for overfitting issue and L1 is used for feature selection. The below graph shows the optimization of cost function by L1, L2 dual and L2 primal methods for different epsilon (error distribution) values.
* Metric = “ROC”. This is used to determine the Area Under Curve (AUC) which represents the model’s ability to discriminate between true positive rate (Sensitivity) and true negative rate (Specificity).
* Preprocessing: All the independent variables were standardized to avoid a single or group of variables being dominant because of high variance between the values.

The model had the highest ROC at cost = 1, L1 Regularization and epsilon = 0.001



**K Nearest Neighbors (KNN):** K Nearest Neighbors is a supervised classification model that first calculates the Euclidean distances from a target object to all other objects, and then performs classification based on the k nearest neighbors’ labels. The target object will be labelled according to the majority of its nearest neighbors’ class. During this process, data is scaled so that the calculation of distance is weighted equally across all variables. In addition, we tune our KNN model by finding the best k—how many neighbors the target object should rely on.

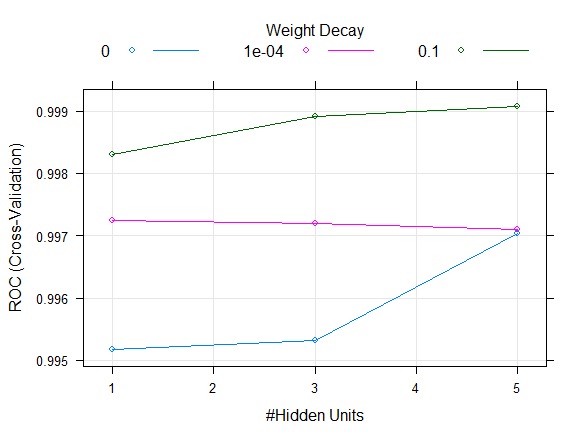
When k is too small, the model is not learning all the patterns in the training dataset, whilst when k is too large, the model is not made general therefore creates an overfitting issue. Thus, we want to choose a k that does not only give the best accuracy but also takes sensitivity and specificity into consideration. For this purpose, we use 10-fold cross validation to train the model with 10 different calibration and validation subsets as well as k from 1 to 60, during which ROC is the metric to select the best k that gives that largest AUC. The chart below shows the ROC values fluctuate as the value of k changes. We can see that when k = 19, the ROC reaches to the highest point. In conclusion, we select k = 19 to classify the target label based on Euclidean distance.



**Neural Network:** Neural Network is a classification technique that classifies through one or more layers. An Artificial Neural Network has one input layer, one output layer and one or more hidden layers between them. Each input layer carries a weight to each neuron in the next layer, so the activation function in each neuron in the sum of product of the input weights with their coefficients. The activation function also takes in the bias weight (it prevents the function from deviating away from the categories) and is standardized to form the sigmoid function. In this way the neural network classifies at each node until the output layer.

* Method = “nnet”. It follows gradient descent where the error function is optimized to reach its local minimum. It uses a decay value (less than 1) to multiply it with the weights so that the function goes through gradient descent. Size is the number of neurons in the hidden layer. The backpropagation algorithm calculates the gradient of the error function with respect to the weights until it reaches the local minimum. Below graph show the variation of ROC for different Size and three different Decay weights. This prevents the model from being overfit.
* Metric = “ROC”. This is used to determine the Area Under Curve (AUC) which represents the model’s ability to discriminate between true positive rate (Sensitivity) and true negative rate (Specificity).
* Preprocessing: All the independent variables were standardized.

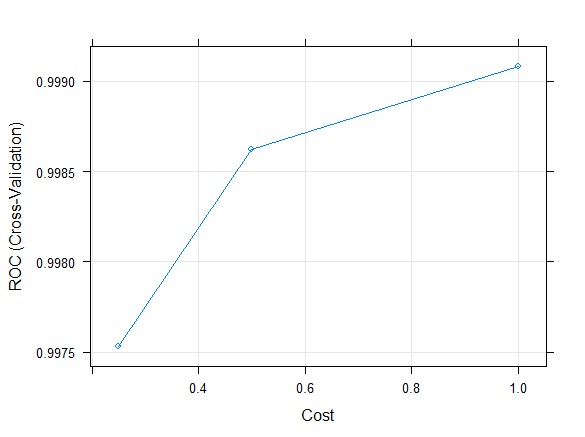
The final values used for the model were size = 5 and decay = 0.1



**Support Vector Classifier:** SVMstands for Support Vector Machines. It is aSupervised learning algorithmwhich can be used for classification and regression problems as Support vector classification and Support vector regression. SVM is based on the idea of finding a hyperplane that best separates the features into different domains. The best ROC for the support vector classifier is chosen by,

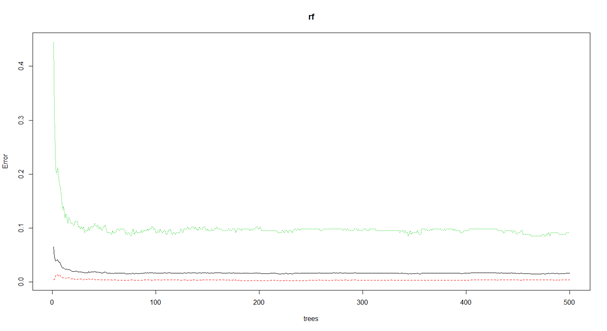
* Method = “svmRadial”. This is a support vector classifier with Radial Basis Function (RBF) kernel. A Radian kernel function has a gamma parameter where gamma is the tuning parameter which accounts for the smoothness of the decision boundary (hyperplane) and controls the variance. If gamma is small, then the variance is high, far reach by the margin and low bias, vice versa. The C parameter controls the cost of misclassification in the dataset. So, the C parameter must be optimized to its minimum by adjusting the gamma parameter. Below graph shows that the gamma value turned out to be constant at 0.417 after 0.5 of cost. The highest ROC at gamma 0.417 is taken.
* The metric used here is also “ROC” and the values were standardized.

The final values used for the model were gamma = 0.417436 and C = 1



**Random Forest Classifier:** Random forests or random decision forests are an [ensemble learning](https://en.wikipedia.org/wiki/Ensemble_learning) method for [classification](https://en.wikipedia.org/wiki/Statistical_classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis) and other tasks that operates by constructing a multitude of [decision trees](https://en.wikipedia.org/wiki/Decision_tree_learning). Ensemble methods use multiple learning models to gain better predictive results. In the case of a random forest, the model creates an entire forest of random uncorrelated decision trees to arrive at the best possible answer.

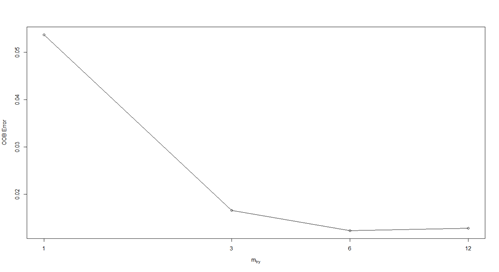
To demonstrate the basic implementation the “randomForest” package has been used. A natural benefit of the bootstrap resampling process is that random forests have an out-of-bag (OOB) sample that provides an efficient approximation of the test error. This provides a built-in validation set and makes identifying the number of trees required to stabilize the error rate during tuning more efficient. However, as illustrated below some difference between the OOB error and test error are expected.



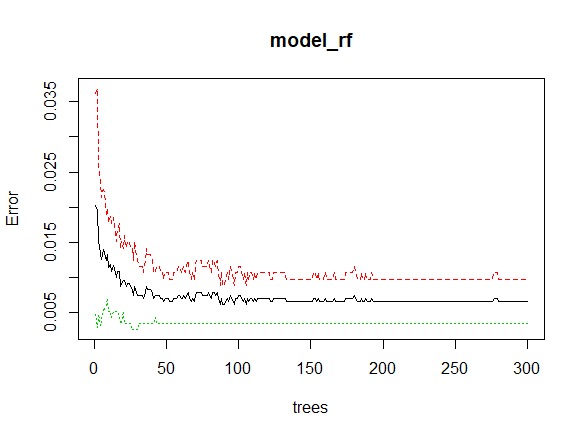
The default random forest performed 500 trees and 3 (= Number of features/3 i.e. 10/3) randomly selected predictor variables at each split. Then the confusion matrix and other classification metrics of the testing dataset were noted. There was no need for cross-validation, or a separate test set to get an unbiased estimate of the test set error because the OOB error calculated during model training is an enough indicator of test set performance. We noticed that we can still seek improvement by tuning our random forest model.

We tuned the Random Forest model using different sets of parameters. From the previous illustration, it can be inferred that the error rate remains constant after 300 trees. Mtry is the number of variables to randomly sample as candidates at each split.

Below graph depicts the OOB errors for different values of mtry:



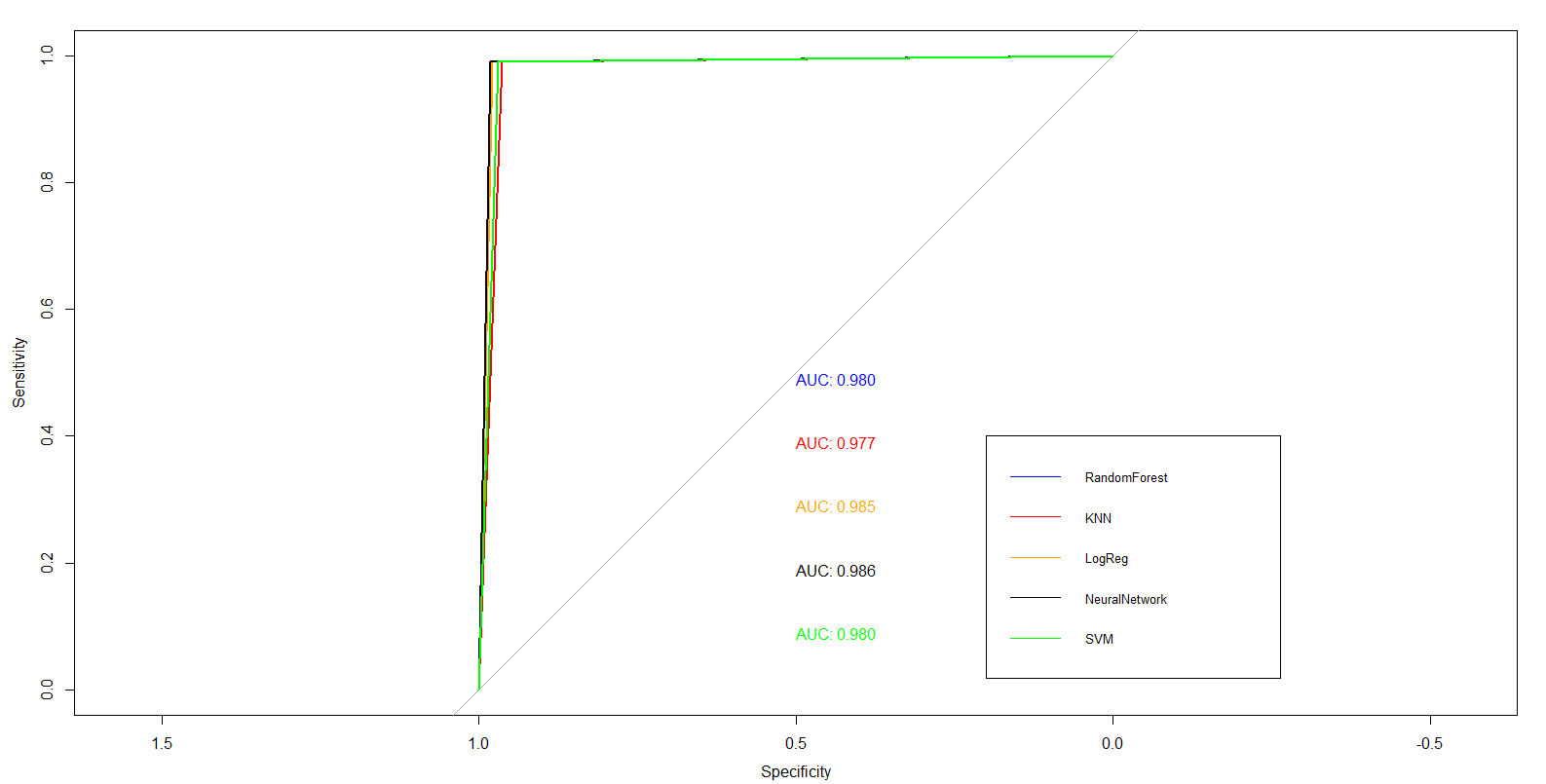
6 is the optimum value of mtry since that is the value which produces the least OOB error. Then the random forest was run again with the new ntree and mtry values.



The accuracy and Kappa value had a slight increase after tuning the model.

**Summary of 5 base classifiers**

**ROC Curve:** The graph below represents the model’s ability to discriminate between true positive rate (Sensitivity) and true negative rate (Specificity). Sensitivity is the positive classes identified by the model out of total actual positive classes aka true positive rate. Specificity is the negative classes identified by the model out of total actual negative classes aka true negative rate. The ROC curve can also be plotted with Sensitivity against 1-Specificity aka false positive rate. Closer the edge of the curve is to 1, more accurate the model is. Clearly, all models have almost equal and are high in accuracy.



**Best Indicators:**

From our study, we found out that some of the variables better predict fraudulent activities than other. We looked at the variable importance in all five of our models and averaged the ranks of these significant predictor variables. The tables below show the top 5 variables based on the variable importance in each model:

|  |  |
| --- | --- |
| **KNN** | **Rank** |
| Transaction\_amount | 1 |
| Total.Number.of.declines.day | 2 |
| isForeignTransaction | 3 |
| isHighRiskCountry | 4 |
| X6.month\_chbk\_freq | 5 |

|  |  |
| --- | --- |
| **Neural Network** | **Rank** |
| Transaction\_amount | 1 |
| Average.Amount.transaction.day | 2 |
| isForeignTransaction | 3 |
| Total.Number.of.declines.day | 4 |
| X6.month\_chbk\_freq | 5 |
| **Logistic Regression** | **Rank** |
| Total.Number.of.declines.day | 1 |
| X6.month\_chbk\_freq | 2 |
| Transaction\_amount | 3 |
| Average.Amount.transaction.day | 4 |
| isForeignTransaction | 5 |

|  |  |
| --- | --- |
| **Support Vector Machine** | **Rank** |
| Transaction\_amount | 1 |
| Total.Number.of.declines.day | 2 |
| isForeignTransaction | 3 |
| isHighRiskCountry | 4 |
| X6.month\_chbk\_freq | 5 |

|  |  |
| --- | --- |
| **Random Forest** | **Rank** |
| Total.Number.of.declines.day | 1 |
| Transaction\_amount | 2 |
| isHighRiskCountry | 3 |
| Average.Amount.transaction.day | 4 |
| X6.month\_chbk\_freq | 5 |

We then identified the best 4 indicators based on the average ranking of variable importance in all 5 models:

1. **Transaction amount:** Out of pattern, large amount of transaction amount is usually a fraud activity. For instance, a customer using an average of $200 weekly for a while using $2500 at once is highly suspicious.
2. **Total number of declines per day:** Someone trying to use a card that is declined a lot of times in a single day, without proper communication with the card issuer is also a suspicious activity.
3. **Is foreign transaction:** A lot of times, an unauthorized or pre-scheduled foreign transaction for a customer is always a high-alert transaction for card issuers and merchants who approve the transaction.
4. **Is high risk country:** Research suggests that there are certain high-risk countries where a lot of fraudulent activities originate. Mexico and Brazil are few of the countries that have a very high chargeback risk.

In many cases that are not online, there are other ways to say if a transaction might be fraudulent. For instance, customers trying to do a card not present transaction, with just card numbers is a high-alert activity.

**Results**

This section shows the prediction results for the 769 testing data using an initial ensemble model and an improved ensemble model.

**Ensemble (Majority Vote):** Initially, we used the majority ensemble method, without using any packages in R. This is a very basic way of looking at the outputs of our five models and pick the class for which at least three of the models predicted the same class. In order to make sure that our results were accurate, we first tuned the models, cross-validated it and predicted the majority test result.

We found that neural network resulted in the highest median accuracy of 98.2%. However, since our data has class imbalance, we focused on the f-measure since provides a better insight into our data because f-measure takes recall and precision into consideration. Still, neural network resulted in the highest f-measure value. The table below shows all other metrics of model performances.

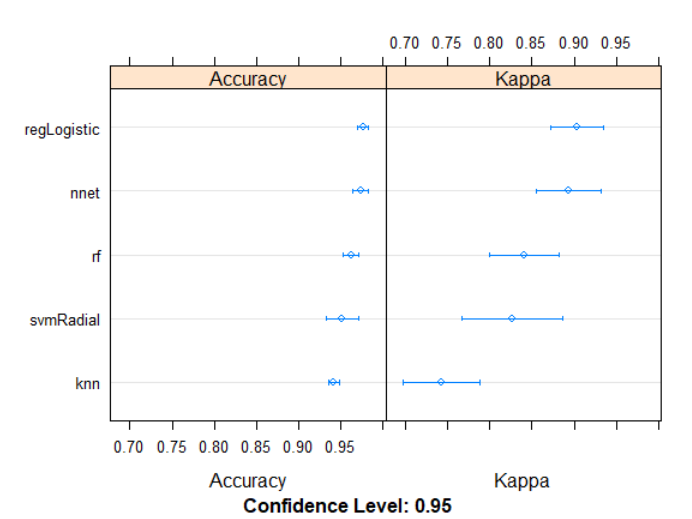
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Metric** | **Random Forest** | **K-Nearest Neighbors** | **Logistic Regression** | **Neural Network** | **SVM** | **Ensemble** |
| Recall/Sensitivity | 0.9910 | 0.9820 | 0.9910 | 0.9910 | 0.9910 | 0.9910 |
| Precision | 0.8462 | 0.8134 | 0.8871 | 0.8943 | 0.8397 | 0.8594 |
| Accuracy | 0.9727 | 0.9649 | 0.9805 | 0.9818 | 0.9714 | 0.9753 |
| Specificity | 0.0304 | 0.0380 | 0.0213 | 0.0198 | 0.0319 | 0.0274 |
| FMeasure | 0.9129 | 0.8898 | 0.9362 | 0.9402 | 0.9091 | 0.9205 |
| TP | 110 | 109 | 110 | 110 | 110 | 110 |
| TN | 638 | 633 | 644 | 645 | 637 | 640 |
| FP | 20 | 25 | 14 | 13 | 21 | 18 |
| FN | 1 | 2 | 1 | 1 | 1 | 1 |

**Ensemble (Stacking):** The majority voting ensemble method averages the performances of all 5 classifiers. Therefore, our overall prediction is not the best performance. We continued improving our outcome by applying a more advanced ensemble method called stacking. Stacking builds multiple models and a second-layer supervisor model that learns how to best combine the predictions of the primary models.

We used R packages “caretEnsemble” to perform the stacking ensemble method. The second-layer supervisor model is random forest. This turned out to be the best model to combine all 5 base classifiers and yielded the best results and has lifted the accuracy from previously 97.53% to 99.87% at mtry = 2. The table below shows all other metrics of model performances.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Metric** | **Random Forest** | **K-Nearest Neighbors** | **Logistic Regression** | **Neural Network** | **SVM** | **Ensemble** |
| Recall/ Sensitivity | 0.9910 | 0.9910 | 0.9910 | 0.9910 | 0.9910 | 0.9820 |
| Precision | 0.8462 | 0.8209 | 0.8871 | 0.9016 | 0.8462 | 1.0000 |
| Accuracy | 0.9727 | 0.9675 | 0.9805 | 0.9831 | 0.9727 | 0.9974 |
| Specificity | 0.0304 | 0.0365 | 0.0213 | 0.0182 | 0.0304 | 0.0000 |
| FMeasure | 0.9129 | 0.8980 | 0.9362 | 0.9442 | 0.9129 | 0.9909 |
| TP | 110 | 110 | 110 | 110 | 110 | 109 |
| TN | 638 | 634 | 644 | 646 | 638 | 658 |
| FP | 20 | 24 | 14 | 12 | 20 | 0 |
| FN | 1 | 1 | 1 | 1 | 1 | 2 |

This shows the different accuracy and kappa values during model tuning and cross validation process:



Overall, neural network is still the model with the highest accuracy. However, it should be noted that the improved ensemble modeling using stacking resulted in a better accuracy, even better than any other classifier models we used.

**Limitations**

Our models predicted fraudulent activities with high accuracy and f-measure values. However, the dataset and our analysis does have a few limitations as discussed below:

* **Small sample size:** This dataset with 3075 transactions has small sample size. In machine learning algorithms, a larger dataset means more stances for the algorithm to learn from. So, a larger dataset would have resulted better predictive accuracy.
* **Imbalance in data:** The small percentage of transactions in our dataset and in real-life situations are fraudulent making the detection difficult and inaccurate at times. Our usage of statistical tools helped us balance the data, but they aren’t perfect ways to solve this issue.
* **Metrics of measurement:** Fraudulent activities can occur at different times, locations, amounts and over similar looking transactions. There aren’t any specific variables that are sure to result in fraud. Hence making prediction a difficult task.
* **Noise/ bias and variances:** As in other datasets and machine learning, there are instances where the data is not easily understandable and might be meaningless. A variable pointing to fraudulent transaction in one situation might not be the same in another.
* **Efficiency of the models:** Popular methods of total accuracy check and confusion matrix are not applicable since the data is very skewed and there is no clear understanding of the performance of models. Hence, advanced understanding of the models is required.

**Conclusion**

Based on the plots, external research and the model output, Transaction amount, Total number of declines per day, foreign transaction to high risk countries are the important factors that is highly correlated with fraud transactions. So when there is a international transaction of an amount higher than the average transaction amount per day by a merchant to a high risk country, then it has a high possibility of being a fraud transaction.

Our research helped us understand the various models to accurately predict the fraudulent credit card transactions. We built various statistical models that helps prevent suspicious activities and found variable indicators that raises red flag when fraud transactions are tried. However, it must be noted that in this modern day and age, fraudsters keep trying newer ways to carry on these activities. Hence, it is very important to constantly include additional attributes into models that might help predict these activities. Also, using advanced classifier techniques and statistical models and tools ensures reduction and prevention of potential fraud activities in a timely manner.

1. According to reports cited from Nielson, Barclay’s and ACI Worldwide.   
   Source: <https://www.creditdonkey.com/credit-card-fraud-statistics.html> [↑](#footnote-ref-2)