CREDIT CARD FRAUD DETECTION

AP18110010491- Vani Tadiboyina- CSE-H

Abstract

One of the main and popular payment methods is the use of credit cards. This advent of free transactions has made it easier and easier for users to pay. The growing growth in its use directly affected counterfeit transactions which increased the frequency of illegal activities. Statistics suggest that the losses incurred by these illegal activities amount to millions of dollars each year. The scam is so well done that a normal eye looks like a real transaction. As a result, many banking and economic sectors have begun to rely on technology to combat these illegal agreements. Many machine learning applications are designed to combat deception, but with the improvement of these systems, the techniques used by these technicians are also upgraded. In this project, we are trying to use electronic learning algorithms such as Logistic Regression, ROC, Decision Tree, Artificial Neural Networks, Gradient Boosting, AUC Curve which in these applications are best suited for the purpose of solving the problem of credit card fraud.

Introduction

Financial fraud is a growing concern with far reaching consequences for governments, corporations, the financial industry. As credit card transactions become a widespread payment method, the focus is on the latest computerized accounting systems to deal with the issue of credit card fraud. There are many fraudulent solutions and software to prevent fraud in businesses such as credit card, retail, e-commerce, insurance and industry. The data mining process is one of the most notable and popular methods used to solve the problem of obtaining credit fraud. It is impossible to be absolutely sure of the real purpose and suitability behind a plan or transaction. In fact, looking for existing evidence of fraud from data obtained using mathematical techniques is the best way to work. Credit card fraud is a real process of identifying fraudulent transactions into two phases of legal and fraudulent transactions, a number of strategies designed and used to resolve credit card fraud such as genetic algorithm, neural network architecture, neural algorithm -A migratory bird algorithm, comparative analysis of asset retrieval, SVM, decision tree and random forest is implemented. Credit card fraud is a very common problem but it is also a difficult problem to solve.

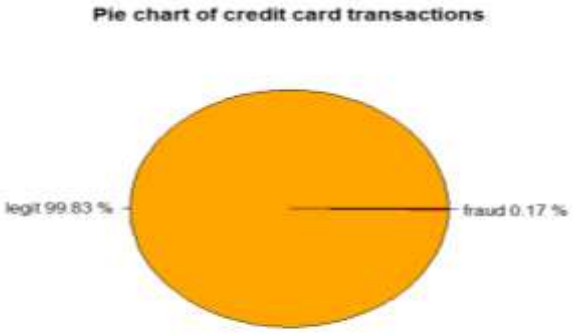
Literature survey:

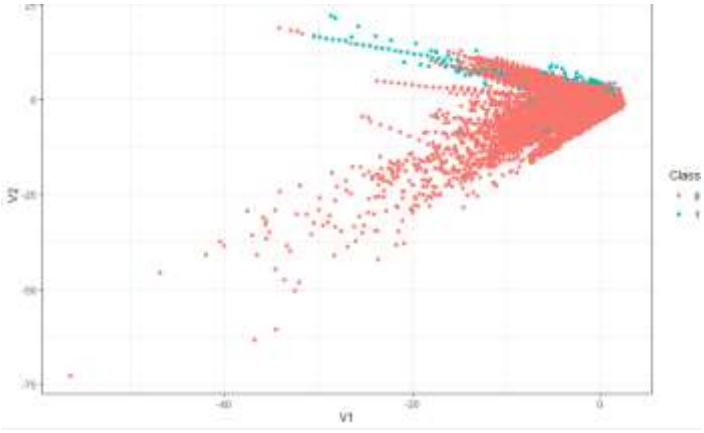
Sahil Dhankhad and three other writers [3] have provided a solution to the perplexing issue of financial services that cost billions of dollars each year. Using surveillance machine learning systems and real-world databases, they have used a number of credit card fraud techniques. In this article, they have used these algorithms to deliver good divisions using aggregation methods. This article provides the differences between the 10 classification algorithms and their accuracy comparisons. The authors used model testing using Accuracy, F1-Score, Recall, Precision, GMean, FPR, TRP techniques. The authors identified the most important factors in the fraudulent discovery of a credit card that could lead to greater testing accuracy models.

Samuel A. Oluwadare and three other authors [4] introduced a model for using and improving a credit card fraud model as transaction size and information size grow. In this paper, the database is obtained from European card holders containing 284,807 transactions. Credit card fraud and providing accuracy, sensitivity, clarity, Matthews equity and rating, three Naïve Bayes, a neighborhood of 4 k and structured models are used. In terms of results, they say KNN works better than other methods in terms of results.

Data Exploration and Visualization:

Data visualization and exploration is perhaps the fastest and most useful way to summarize and learn more about our data. The data we use here is available to European cardholders presenting transactions that took place over a two-day transaction, where we have 492 of the 284,807 payment frauds. The database is not very accurate, the good class (fraud) is 0.172% of all transactions. Within this database, there are 31 columns of which 28 are named v1-v28 to protect sensitive data. Some columns represent Time, Value and Category. Time shows the time gap between the first transaction and the next. The total amount of money made. Category 0 represents a legitimate transaction and 1 represents fraud.



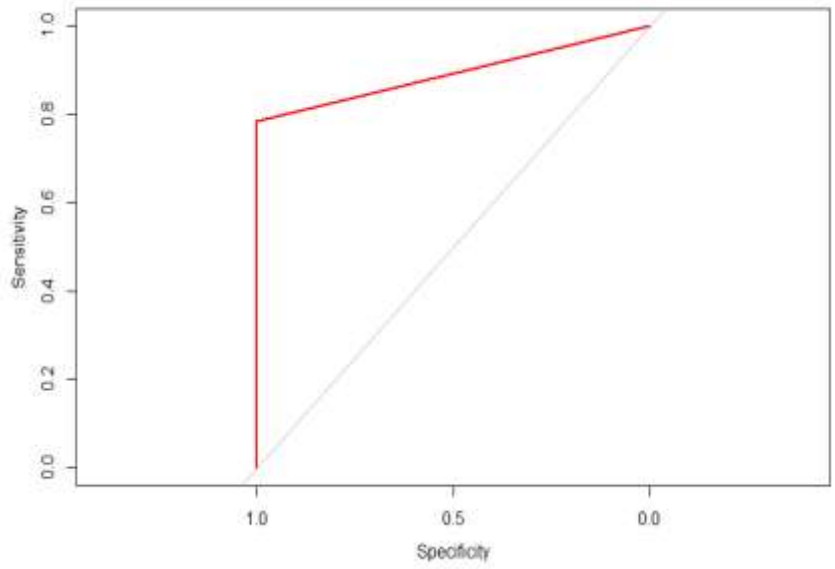


Different Models and Implementation:

Random Forest:

Random Forest Planning is controlled by phases / algorithm integration algorithms. Integrated algorithms are those that combine more than one algorithm of the same or different classification of objects. Random forest planning creates a collection of decision trees from a selected sub-set of training. It then includes votes on the trees of various decisions to determine the final stage of the test item. In short if we have a lot of trees in a strong forest the forest looks good. In the same way in the random forest algorithm where the number of trees in the forest rises gives high accuracy results.

There are four benefits to showing why we use the Random Forest algorithm. The first is that it can be used for partitioning and retrieval activities. Excessive equilibrium is one critical problem that can make the results worse, but with the random Forest algorithm, if there are enough trees in the forest, the separator will not fit well with the model. The third advantage is the subdivision of the random forest can handle the lost values, and the final benefit is that the random subdivision of the Forest can be modeled on the category values.



Area under the curve: 0.8923

Logistic Regression:

Logistic regression is a well-established statistical method for predicting binomial or multinomial outcomes. Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. Logistic Function:

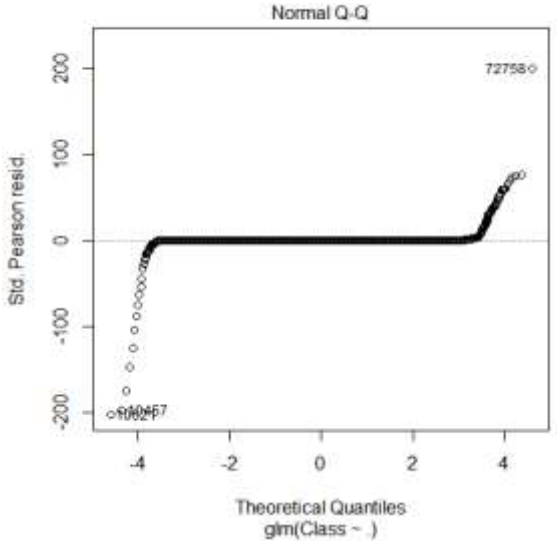
$P(y = 1) = 1/(1 + e - \beta 0 + \beta 1x1 + \beta 2x2 + ... + \beta kxk)$

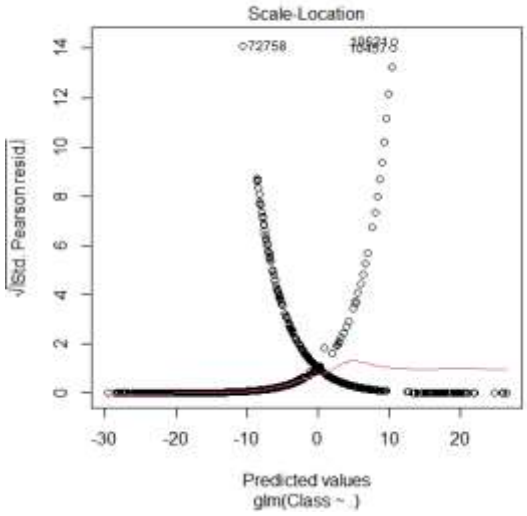
where, x1, x2..., xk are independent variables. P(y) = probability prediction. Although logistic regression is a classification algorithm, we predict probabilities which must be transformed into a binary values (0 or 1) in order to actually make a probability prediction.

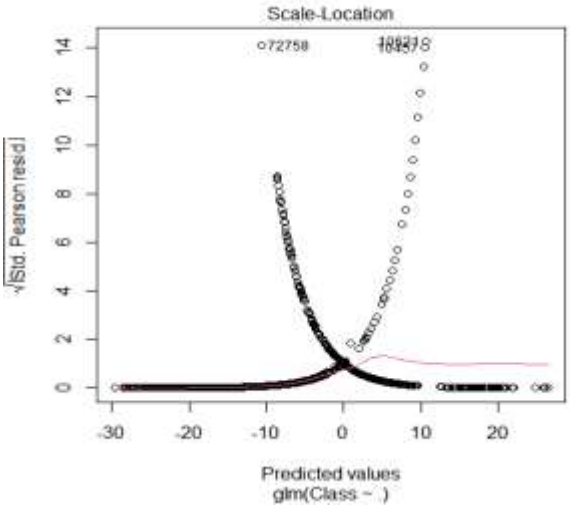
- summary(LogisticModel)

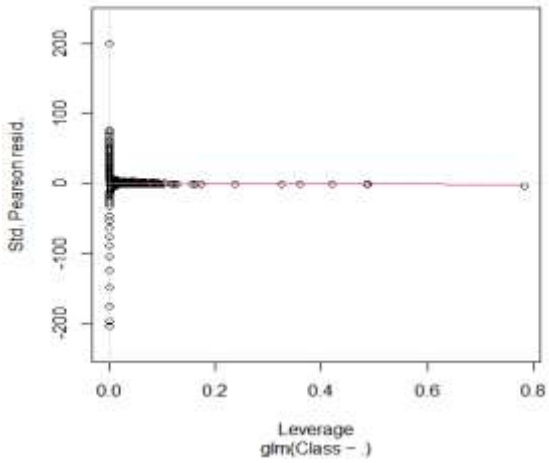
```
Call:
glm(formula = Class ~ ., family = binomial(), data = trainingdata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.6108  -0.0292  -0.0194  -0.0125   4.6021

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -8.651305   0.160212 -53.999  < 2e-16 ***
V1           0.072540   0.044144   1.643 0.100332
V2           0.014818   0.059777   0.248 0.804220
V3           0.026109   0.049776   0.525 0.599906
V4           0.681286   0.078071   8.726  < 2e-16 ***
V5           0.087938   0.071553   1.229 0.219079
V6          -0.148083   0.085192  -1.738 0.082170 .
V7          -0.117344   0.068940  -1.702 0.088731 .
V8          -0.146045   0.035667  -4.095 4.23e-05 ***
V9          -0.339828   0.117595  -2.890 0.003855 **
V10         -0.785462   0.098486  -7.975 1.52e-15 ***
V11          0.001492   0.085147   0.018 0.986018
V12          0.087106   0.094869   0.918 0.358532
V13         -0.343792   0.092381  -3.721 0.000198 ***
V14         -0.526828   0.067084  -7.853 4.05e-15 ***
V15         -0.095471   0.094037  -1.015 0.309991
V16         -0.130225   0.138629  -0.939 0.347537
V17          0.032463   0.074471   0.436 0.662900
V18         -0.100964   0.140985  -0.716 0.473909
V19          0.083711   0.105134   0.796 0.425897
V20         -0.463946   0.081871  -5.667 1.46e-08 ***
V21          0.381206   0.065880   5.786 7.19e-09 ***
V22          0.610874   0.142086   4.299 1.71e-05 ***
V23         -0.071406   0.058799  -1.214 0.224589
V24          0.255791   0.170568   1.500 0.133706
V25         -0.073955   0.142634  -0.519 0.604109
V26          0.120841   0.202553   0.597 0.550783
```

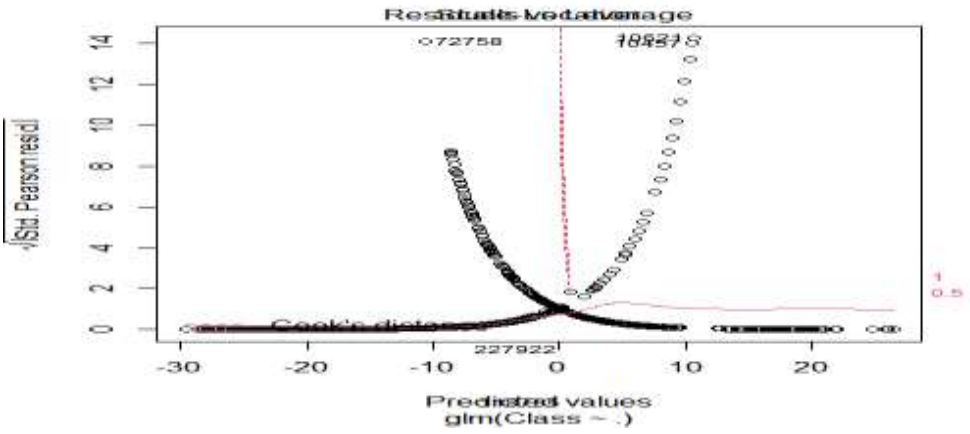
```
V27         -0.852018   0.118391  -7.197 6.17e-13 ***
V28         -0.323854   0.090075  -3.595 0.000324 ***
Amount       0.292477   0.092075   3.177 0.001491 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 5799.1  on 227845  degrees of freedom
Residual deviance: 1790.9  on 227816  degrees of freedom
AIC: 1850.9
Number of Fisher Scoring iterations: 12
```



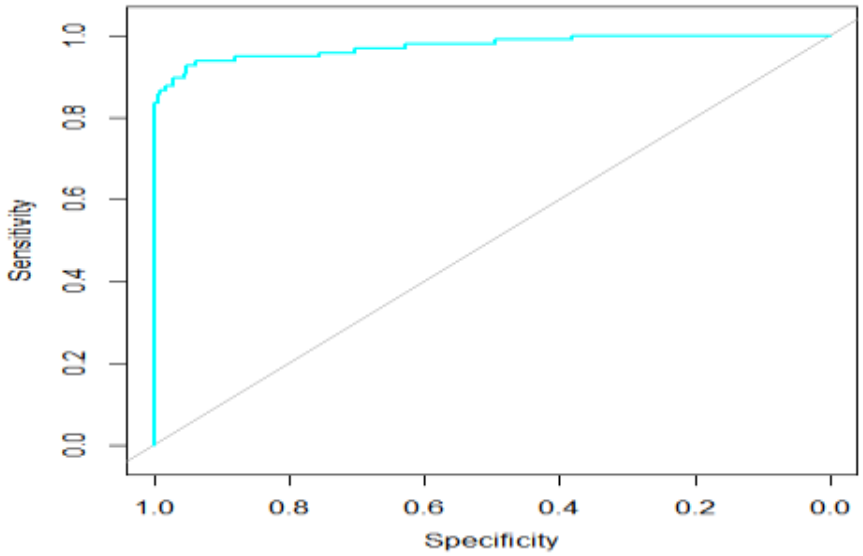








Performance of the model using ROC curve



>print(auc.gbm)

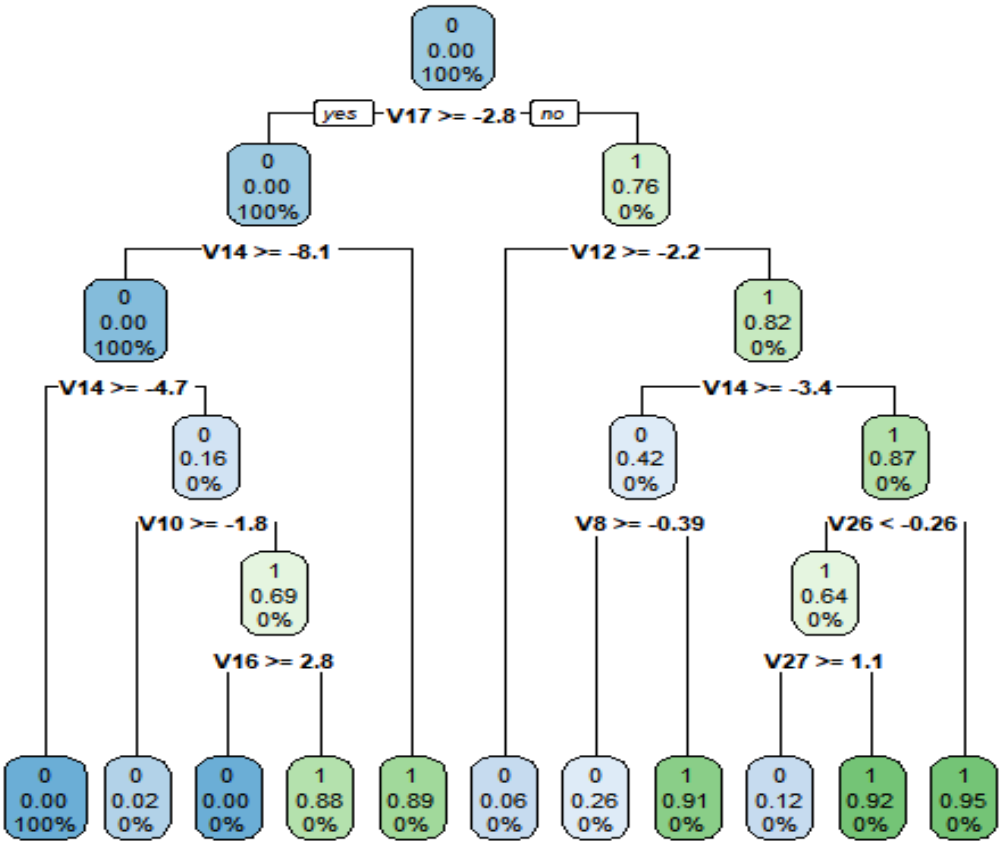
Call:
roc.default(response = testdata$Class, predictor = LogiReg.Prediction, plot = T, col = "cyan")

Data: LogiReg.Prediction in 56863 controls (testdata$Class 0) < 98 cases (testdata$Class 1).

Area under the curve: 0.9748

Decision Tree:

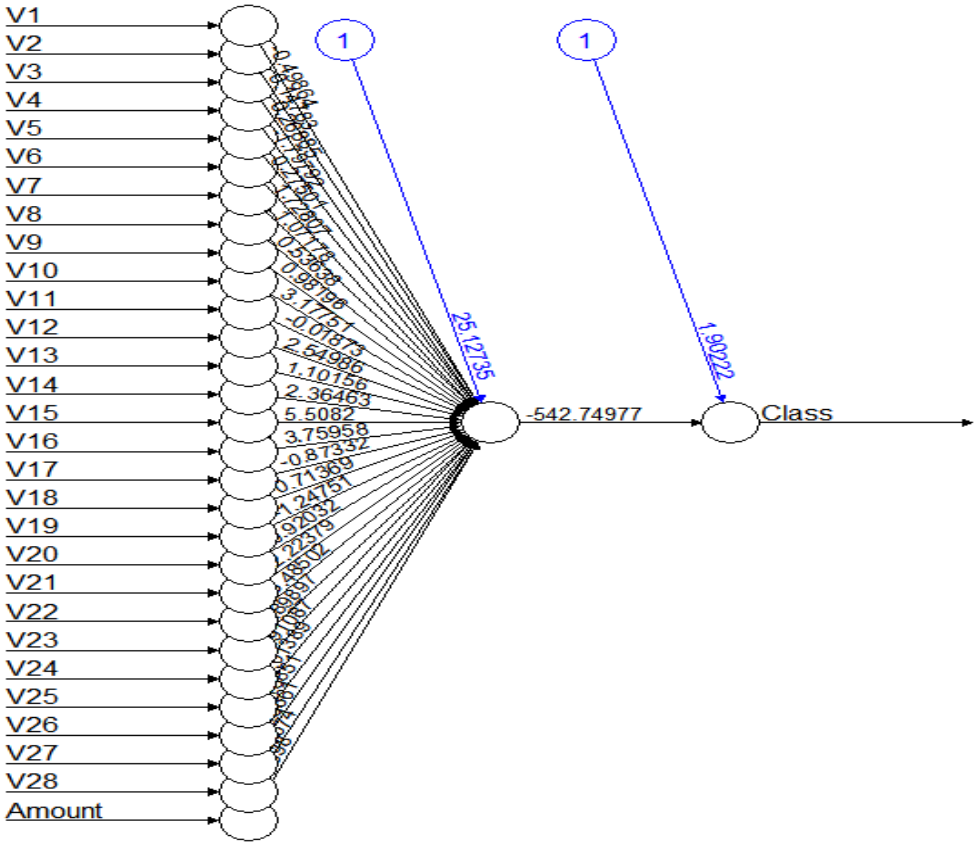
Decision trees is the most intuitive one among all the other machine learning algorithms. It is a supervised learning algorithm that can be used for solving both regression and classification problems. It solves the problem by representing the given data and the attributes as a tree. The internal nodes of the tree are the attributes, the branches are the conditional statements and the leaf nodes are the target classes. The order of attributes in the tree is decided by calculating the contribution of that attribute using methods like Information Gain .



> mean(predictedValue == NewData$Class)
[1] 0.9995471

Artificial Neural Networks:

We are importing the neural net kit which helps us to implement our ANNs. Using historical data, the ANN models can learn the patterns, and can identify the input data. We then plotted it using the plot () function. In the case of ANN, we have a range of numbers that is between 1 and 0. We set a threshold as 'X' and values above 'X' will correspond to 1 and the rest will be 0.

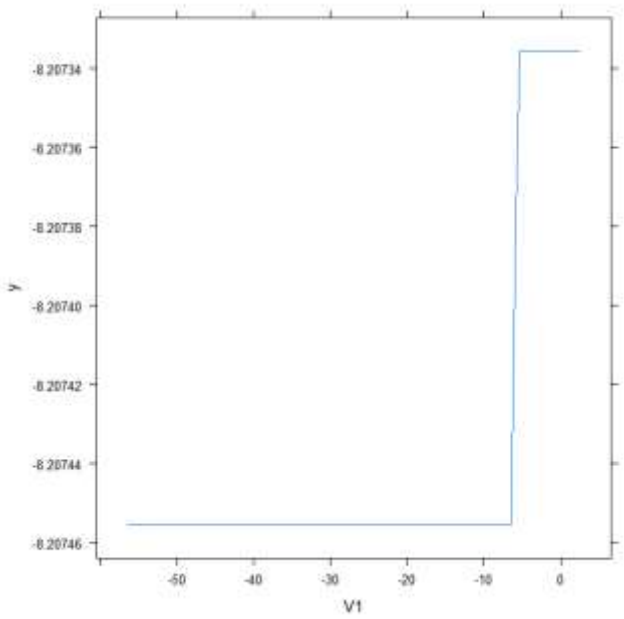


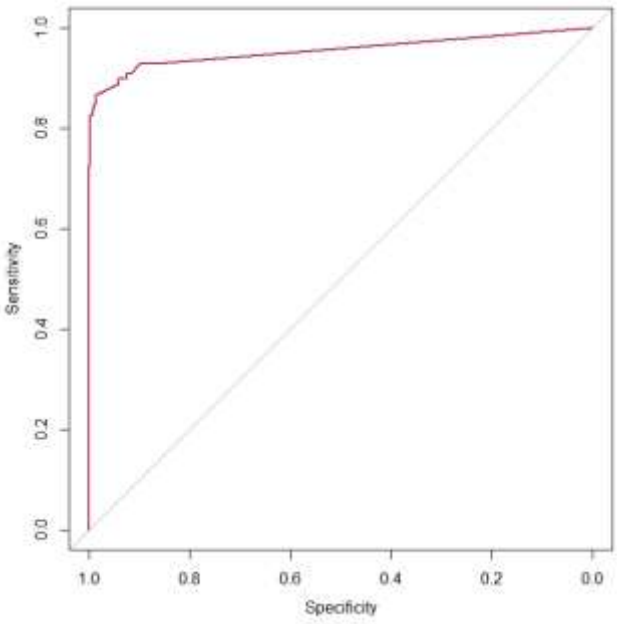
>mean(NNResult==testdata$Class)
[1] 0.9993504
> table(NNResult, testdata$Class)

NNResult	0	1
0	56847	21
1	16	77

Gradient Boosting:

Gradient Boosting is a common algorithm for machine learning used to perform classification and regression tasks. This model consists of different underlying models such as weak decision trees. These decision trees combine to form a powerful gradient boosting model. In this method, each new tree is a fit on a modified version of the original data set.





>print(gbm_auc)

Call:
roc.default(response = testdata$Class, predictor = gbm_test, plot = T, col = "maroon")

Data: gbm_test in 56863 controls (testdata$Class 0) < 98 cases (testdata$Class 1).

Area under the curve: 0.9555

CONCLUSION:

- Logistic regression has the best performance. Random forest is not preferred as it overfits the training data and GBM is preferred as it improves upon both models.

- ANN is not preferred as alternative algorithms such as Decision Tree, Regression which are available and are simple, fast, easy to train. They also provide better performance.

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.

References

[1] Sahil Dhankhad ; Emad Mohammed ; Behrouz
Far “Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative Study” 2018 IEEE International Conference on Information Reuse
and Integration (IRI)

[2] John O. Awoyemi ; Adebayo O. Adetunmbi ;
Samuel A. Oluwadare “Credit card fraud detection using machine learning techniques: A comparative analysis” 2017 International Conference on Computing Networking and
Informatics (ICCNI)

THANK YOU!