UNIZIK Journal of Marketing (UJofM), Vol. 1(3), Nov. 2024

https://journals.unizik.edu.ng/ujofm

Comparative Analysis of Data Science Approaches for credit card Fraud Detection in the USA

Ezenwafor, Ebuka Christian[1†]; Odezi, Jennifer Obuke[2]; and Onwujiobi, Charles[3]

Article Information

Keywords: Fraud detection, data science, credit card fraud, Logistic regression, Neural Network, Decision Tree, Random Forest classifier, Gradient Boosting model

Article History
Received: 26 Jun 2024
Accepted: 26 Oct. 2024
Published: 19 Nov. 2024

Abstract

This study examined the usage of data science approaches to prevent fraud and financial loss in the credit card industry in United states of America. To achieve that, data on credit card fraud was collected from Kaggle which holds over 13000+ observation. The data was cleaned to ensure it usability and ability to fit models without overfitting. Six algorithms were used namely Logistic regression, Neural Network, Decision Tree, Random Forest classifier, Gradient Boosting model and Bagging model was used to identify the best model. However, all these models had accuracy above 0.93(93%) but we will choose Random Forest classifier as the best model with over 0.97(97%) accuracy, and it has the lowest execution speed which is the time needed for computation, data preprocessing, splitting and model evaluation. From the models, no of transaction, IP_address, average transaction and location are the main factor that affect the outcome of a transaction. Fraudsters spend a lot of time and resources looking for loopholes in models and credit card approval process so companies should be updated in various models they use and steadily look out for more opportunities for improving their transaction approval model.

Introduction

Fraud is not a new issue in the financial industry; it has existed for a long time, resulting in the bankruptcy of many companies and the displacement of many individuals from their jobs. This problem has persisted within the industry and is now growing exponentially in recent times. The financial industry, being a highly regulated sector, is constantly exposed to the risk of fraud and monetary loss. The government has instituted regulations such as the Fair Credit Billing Act (FCBA) and the Electronic Fund Transfer Act (EFTA) to help mitigate financial losses within the financial industry, especially in the credit card sector. These regulatory bodies are diligently working to reduce financial fraud in the USA, particularly within the credit card industry. According to security.org (2023), as much as 44% of American credit card users fell victim to credit card fraud in 2022 alone, a significant increase compared to the 35% fraud rate in 2021.

affiliation: [1]Department of Marketing, Nnamdi Azikiwe University, Awka

email: ebuka.ezenwafor@breadfinancial.com; odezi.obuke@chase.com; onwujiobicharles@gmail.com.

† *corresponding author*

This alarming rise has prompted the development of more sophisticated processes and procedures aimed at combating credit card fraud in both the credit card and financial industries. Traditional methods of fraud detection and loss prevention have proven insufficient due to the escalating complexity and sophistication of fraudulent techniques employed by criminals.

Consequently, the adoption of data science, batch analysis, and models designed to identify and prevent fraudulent activities has emerged. Nonetheless, this approach has limitations since it involves waiting until the end of the business day or week for a complete cycle to identify fraudulent transactions. Fraudsters are becoming increasingly adept and relentless in devising new strategies to perpetrate their crimes, which incurs substantial costs for credit card companies and banks, both in terms of human resources and financial resources.

In response to this challenge, leveraging data science and real-time information has appeared as a promising solution. This proposal's primary goal is to investigate the effectiveness of data science models, algorithms, and real-time information in preventing fraud and minimizing monetary loss within the financial industry. Specifically, the paper will delve into how data science techniques can be harnessed for real-time fraud detection and the reduction of monetary loss. This will be achieved through a review of existing literature, including certified fraud examiner bulletins, peer-reviewed academic sources, industry reports, and other pertinent references, focused on the use of machine learning models and real-time data in the detection and prevention of fraud within the credit card industry. Earlier studies have emphasized on the use of singular or multiple machine learning approaches to comb the issues of fraud in credit card industry. However, this study looks to view the machine learning models from the comparative viewpoint to give a wholistic view of different models and their limitations to model performance and accuracy.

Literature Review

Credit card fraud

According to the Association of Certified Fraud Examiners (ACFE), as cited by Sadgali, I., Sael, N., & Benabbou, F. (2018), fraud is a premeditated or deliberate act of depriving another of property or money through cunning, deception, or other unfair acts. Fraud has tremendously grown worldwide and has affected many businesses and institutions across the globe. This has led to a global interest in ways to curtail this heinous activity. According to a report released by the Federal Trade Commission (FTC) in February 2023, consumers lost $8.8 billion (about $27 per person in the US) due to fraud, showing over 30% increase from the previous year.

This raises serious concerns for consumer protection. Fraud encompasses several types, including insurance fraud, financial statement fraud, credit card fraud, mortgage fraud, and money laundering. Fraud rates are at an all-time high in the United States, where 46% of global fraud occurs. It is projected to exceed $12.5 billion (about $38 per person in the US) by 2025 in the USA alone (Credit Card Statistics, 2023).

This paper will focus on credit card fraud, which accounts for the highest fraud losses in the USA. As postulated by Credit Card Statistics (2023), 80% of the credit cards in circulation in the USA are compromised.

According to Askari, S. and Hussain, A. (2017), credit card fraud refers to a financial transaction made by an unauthorized person using a card, and neither the cardholder nor the provider is aware of such a transaction at the time of authorization. It also involves identity theft by using someone else's card or card information to make financial transactions without the cardholder's, card issuers, or merchant's consent. The substantial number of transactions recorded every second makes this possible. Credit card fraud can take various forms, including:

A) Card Not Present (CNP): This involves performing a transaction with a credit card without the merchant physically inspecting or visually checking the card. It accounts for over 65% of the credit card fraud recorded in America. It is commonly perpetrated through phone calls, emails, or websites/internet. As said by Kim Le (2023), this type of fraud is most common in e-commerce and digital shopping and is one of the most challenging frauds to detect and prevent.

B) False Application: This involves applying for a credit card with stolen information. This is another common type of fraud. Credit card applications are usually made online, where you provide your information, and an algorithm verifies your eligibility for the credit card. Fraudsters use others' information like SSN, email addresses, and personal details to apply for a credit card without the victim's knowledge. Often, victims stay unaware of this fraud until the collection team contacts them.

C) Account Takeover (AT): This occurs when fraudsters take control of an account and pretend to be the account owner. The fraudster then contacts the card issuer, posing as the genuine cardholder, to request mail redirection to a new address (Patidar et al., 2011). They then conduct unauthorized transactions in the account before the actual account owner becomes aware. This supports the claims of Credit Card Statistics (2023), which states that 80% of accounts in the USA have been compromised without the account owners' knowledge.

D) Stolen Cards: This is the least common type of credit card fraud and results in the least monetary loss. This is often because customers are aware when they lose their credit cards and promptly notify the provider to request a new card

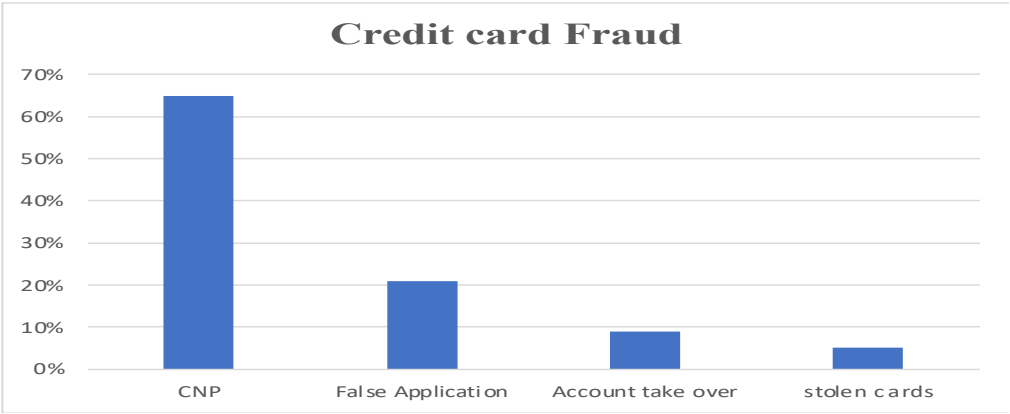


Figure 1
Source: Credit card statistics (2023)

Credit card Fraud: Loss Responsibility

In the case of credit card fraud loss, there is always a discussion about who bears responsibility for the loss. In most cases, it's rarely the customer, and when they do, they bear hardly 5% of the loss. Most of the loss is charged to the issuers or the merchant that authorized the transaction. Furthermore, the issuer will perform due diligence to confirm that fraud occurred on the account and later charge off the loss after 90 days (about 3 months) in compliance with the Fair Credit Billing Act of 1974 amendment.

Empirical review

In the study conducted by Faraji (2022), which centers on the use of supervised learning algorithms to prevent fraud. He made use of logistic regression, decision tree, random forest, KNN, and XGBoost. The study also used the confusion matrix, precision and recall in evaluating the performance of various models. It was found that XGBoost is the fastest and is expected to have the best performance; however, it is only outperforming the random forest in terms of accuracy, precision, recall, and f1-score. In general, the KNN and logistic regression have better performance, which means they better detect fraudulent transactions.

Salekshahrezaee et al 2023 investiagted the role of feature selection in the performance of fraud detection model to contribute to the growing argument on the effect of feature selection. The study found that feature selection effects the performance of model. To do this, they used four ensemble models Random Forest, CatBoost, LightGBM, and XGBoost, and *Principal Component Analysis* (PCA) and *Convolutional Autoencoder* (CAE) as feature extraction methods. This study found that feature extraction influences the performance of a model. More so, convolutional autoencoder is a better extraction method and leads to optimized performance of the model.

According to Ahmed (2022), credit card fraud is one of the biggest threats to financial institutions in the global level. The study investigated various machine learning models (Random Forest, Naïve Bayes, K-Nearest Neighbor, Logistic Regression and Multilayer Perceptron) to find the most effective model in fighting credit card fraud. This model is compared based on accuracy, precision, F1-score. This study shows that random forest has the best accuracy and performance.

Han et. al (2022) developed a credit card fraud detention model using machine learning models. This study made use of logistic regression, artificial intelligence and support vector machine. The performance of the model was measured using accuracy score, F1-score, precision, sensitivity, specificity and recall. From all the index, support vector machine performed best with high execution speed.

Kasasbeh et al (2022) exhaustedly studied the use of artificial neural network to improve the performance of fraud detection model. The model was evaluated based on the precision, sensitivity, specificity, accuracy, F-measure, area under curve (AUC) and root mean square error (RMSE). The study showed that ANN with 4 hidden layers improves the performance of a model and increases the execution speed.

Aburbeian el al (2023) postulated that imbalance data is one of the issues of using machine learning algorithms to prevent fraud. In this study, they studies ways of handling imbalance data and agreed that synthetic minority over-sampling technique (SMOTE) is the best way of handling imbalance data.

Fraud detection Models
This study made use of Logistic regression, Neural network, Random Forest classifier, decision deeper tree, bagging classifier and gradient boosting model.

Logistic regression
Logistic regression is a supervised learning technique that uses independent variable to predict the outcome of a dependent variable (classification). This is commonly used to present the outcome of the binary model (0,1) in a supervised model. According to the study by Sadgali et al., (2018), logistic regression is a generalized linear model that maximizes the log likelihood function to decide the beta coefficients of the model. In a study conducted by Zeager et al., (2017) a game theory learning approach was developed to replicate fraudsters best strategy using logistic regression as fraud detection algorithm which was implemented in a credit card company. However, the shortfall of this model is the vulnerability to overfitting which invalidates the prediction of the model.

Neural Network
Just as the human brain is intricately connected through neurons, artificial neural networks run using a similar network of simulated nodes. A neural network consists of an input layer, an output layer, and hidden layers in between. This model functions through simulating nodes and undergoes many iterations to achieve the highest model accuracy. However, one challenge with neural networks is their inherent complexity, making it difficult to explain the calculations and mathematical processes that underlie the fitted model.

Decision Deeper Tree

This is a non-parametric supervised data mining technique that partitions data into leaf, root and nodes, and decision of best optimized split is made at each internal node of the tree, using impurity measures, splitting and pruning (Jain et al, 2016). This algorithm requires little preparation, it's easy to explain and handles both numeric and categorical variable.

Random Forest-Classifier
Random forest classifier is an ensemble learning method that functions by constructing multiple decision tree. The random forecast classifier can be said to be a more complex model that combines a lot of decision tree to increase prediction accuracy and reduces overfitting. This algorithm is best used when there is a high variation in the data and when the model is prone to overfitting.

Gradient Boosting Model-classifier
Gradient boosting model is an ensemble learning method that relies on weaker models to build a stronger model by minimizing prediction error in a sequential order. This complex algorithm works on iterating simulation aims are developing a better model than the last iteration using minimal loss function.

Bagging Classifier
Bagging model is an ensemble learning model where multiple models are trained independently on parallel on different subset and the final model is made by

aggregating/averaging the predictions of all fitted models. like its name bagging implies, the objective of bagging is to get a lesser variance than any model has independently by averaging the outcome of other models. Bagging model is represented by this formula. This algorithm works perfectly when the variables are unstable and tends to overfit. This model has over 97% accuracy and stands as the best model in this study.

$$\widehat{f_{bag}} = \widehat{f_1}(X) + \widehat{f_2}(X) + \cdots + \widehat{f_b}(X)$$

Research Methodology

Sources of data

Data collection

The data is sourced from Kaggle, and it holds 14 Features and 13650 rows of which 11 have been turned into numerical input values using principal component analysis (PCA) due to a confidentiality concern. These features are named as V1, V2, and V14. According to [31], the feature contains general information such as gender, age in months, identification (ID), credit limit, past month bills, past month payments, account status, IP_address, time, no, of_transaction, amount, credit history, purpose, saving account, credit amount, and frequency.

Data Preprocessing

In machine learning model development, data comes in different form and shape thus needs data cleaning to ensure adequacy of the data which includes this process is called that cleaning and preprocessing which includes,

1) Imbalanced information/classification: this occurs where a high percentage of the data fall towards one of the classes e.g. (99.7% of the data are recorded as not fraudulent transaction). this hampers the accuracy of the model thus lead to model inaccuracy. To handle these issues, I employed the use of Synthetic Minority Oversampling Technique **(**SMOTE) in R which creates a dataset by oversampling the observation from the minority class to balance out the whole dataset.

2) Plot Dataset Correlation Matrix for the whole dataset. This is done to take care of multicollinearity where multiple independent variables are highly correlated amongst each other if not taken care of leads to overfitting.

3) Other data cleaning: this includes, removing duplicated value, missing values, outliers and selecting features needed for the model.

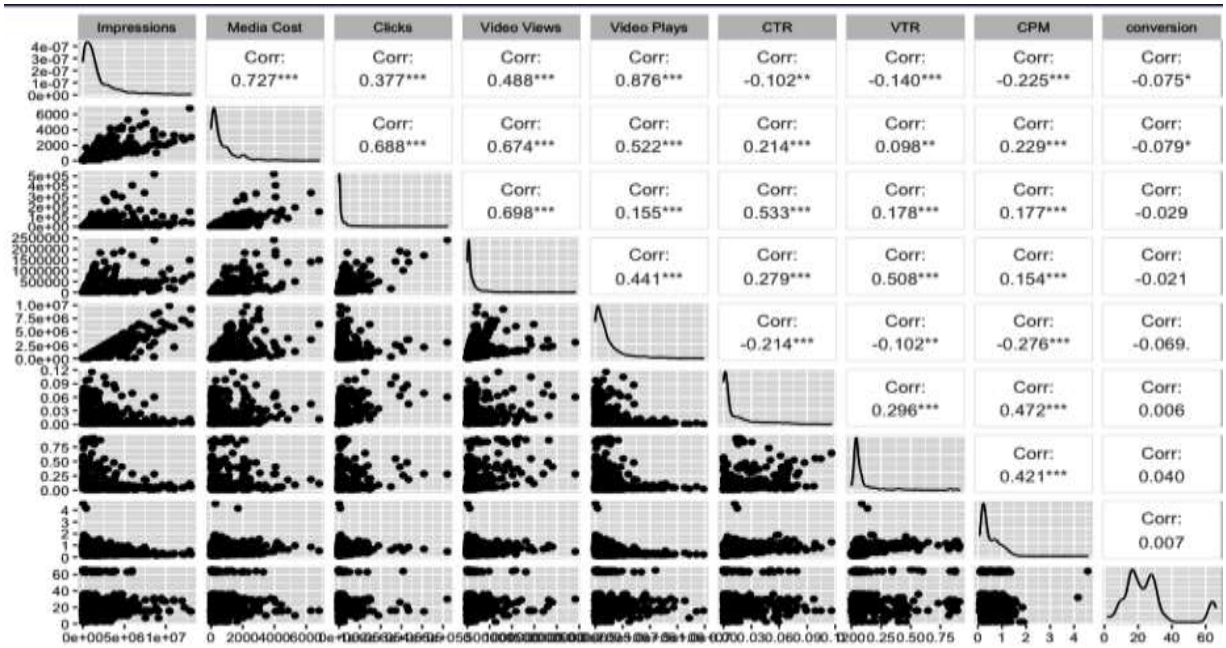


Figure 2. Correlation table
Source: researchers' analysis

Fraud detection model outcomes

This study made use of Logistic regression, Neural network, Random Forest classifier, decision deeper tree, bagging classifier and gradient boosting model.

Logistic regression

Logistic regression is a supervised learning technique that uses independent variable to predict the outcome of a dependent variable (classification). In this study, Logistic regression has a model accuracy While on the validation dataset. The model has 0.935 accuracy, 0.065 error, sensitivity 0.91, 0.92 specificity, precision of 0.93, F1-score of 0.92 and success class of 1. More so, from the confusion matrix, you will see that true negative is around 13000, true positive is 520, false positive 70 while false negative is 60.

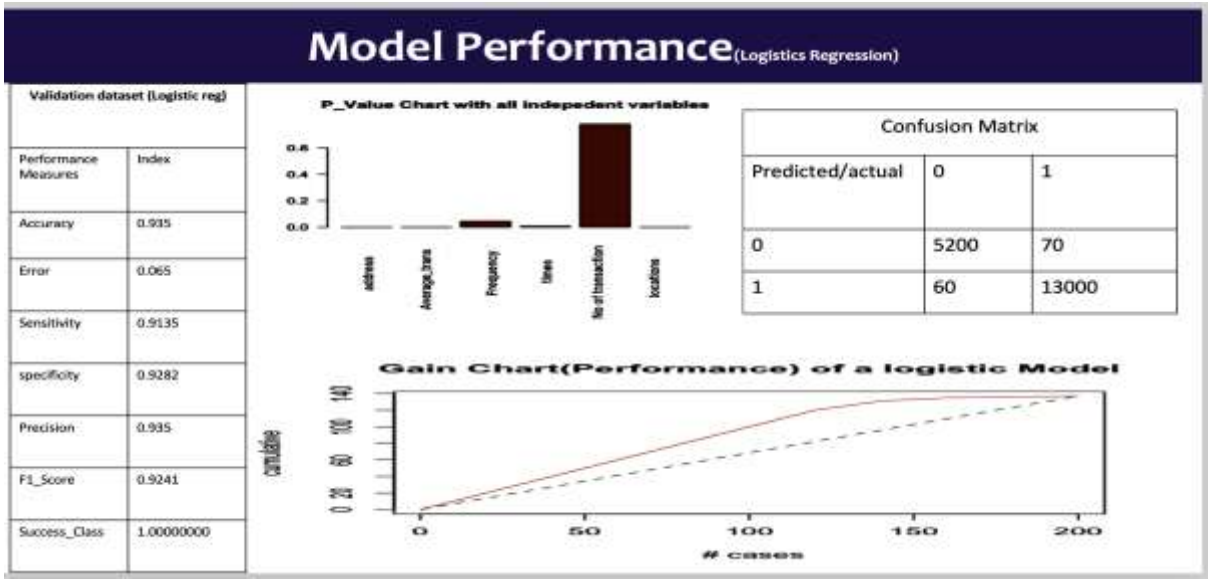


Figure 3. logistic regression Model Performance
Source: researchers' analysis

1. Neural Network

This model functions through simulating nodes and undergoes numerous iterations to achieve the highest model accuracy. Based on the fitted model, the neural network shows an accuracy on the validation dataset. The model has 0.965 accuracy, 0.035 error, sensitivity 0.96, 0.963 specificity, precision of 0.98, F1-score of 0.97, success_class of 1and success_prob of 0.58.

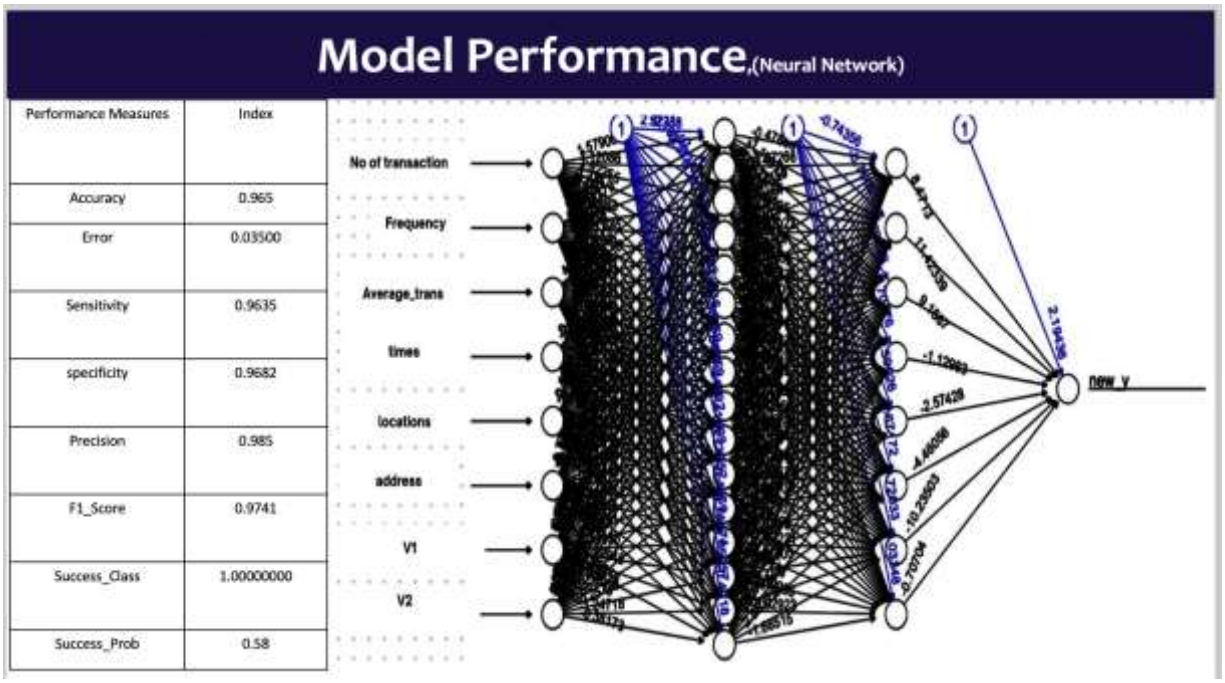


Figure 4. Neural Network Model Performance
Source: researchers' analysis

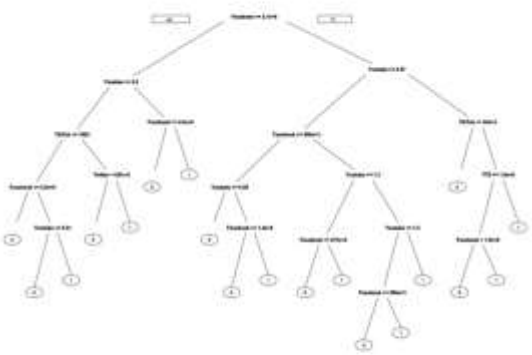
Decision Deeper Tree

This is a non-parametric supervised data mining technique that partitions data into leaf, root and nodes, and decision of best optimized split is made at each internal node of the tree, using impurity measures, splitting, and pruning (Jain et al, 2016). On the training model the model has 0.97 accuracy, 0.028 error, sensitivity 0.97, 0.963 specificity, precision of 0.98, F1-score of 0.97 and success class of 1. While on the validation dataset. The model has 0.955 accuracy, 0.045 error, sensitivity 0.86, 0.963 specificity, precision of 0.93, F1-score of 0.98 and success class of from the confusion matrix, true negative correctly predicted 12670, true positive predicted 470, false positive 60 while false negative 40.

.

Model Performance (Decision Deeper Tree)

Training dataset		Validation dataset	
Performance Measures	Index	Performance Measures	Index
Accuracy	0.97155361	Accuracy	0.9550000
Error	0.02844639	Error	0.0450000
Sensitivity	0.97151899	Sensitivity	1.0000000
specificity	0.97163121	specificity	0.8571429
Precision	0.98713826	Precision	0.9383562
F1_Score	0.97926635	F1_Score	0.9681979
Success_Class	1.00000000	Success_Class	1.0000000



Confusion Matrix		
Predicted/actual	0	1
0	12670	60
1	40	470

Figure 5. Decision Tree Model Performance
Source: researchers' analysis

Random Forest-Classifier

This algorithm is best used when there is a high variation in the data and when the model is prone to overfitting. This model has accuracy of 97% which is one of the best accuracy measures in this study.

While on the validation dataset, the model has 0.965 accuracy, 0.035 error, 0.99 sensitivity, 0.9 specificity, precision of 0.99, F1-score of 0.97 and success class of 1.

Model Performance (Random Forest-classifier)

Training dataset		Validation dataset	
Performance Measures	Index	Performance Measures	Index
Accuracy	1	Accuracy	0.9650000
Error	0	Error	0.0350000
Sensitivity	1	Sensitivity	0.9927007
specificity	1	specificity	0.9047619
Precision	1	Precision	0.9577465
F1_Score	1	F1_Score	0.9749104
Success_Class	1.00000000	Success_Class	1.0000000

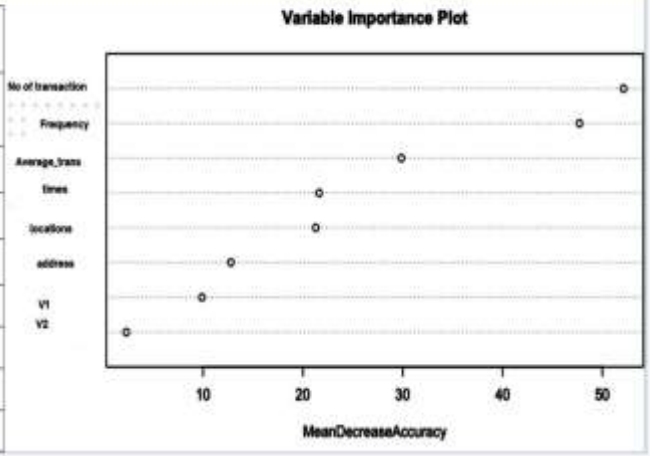


Figure 6. Random forest classifier Model Performance
Source: researchers' analysis

Gradient Boosting Model-classifier

Gradient boosting model is an ensemble learning method that relies on weaker models to build a stronger model by minimizing prediction error in a sequential order. On the training model the model has 0.97 accuracy, 0.028 error, 0.963 specificity, precision

of 0.99, F1-score of 0.97 and success class of 1. While on the validation dataset The model has 0.96 accuracy, 0.04 error, 0.963 specificity, precision of 0.99, F1-score of 0.97 and success class of 1.

Model Performance (GBM, Classifier)

Training dataset	
Performance Measures	Index
Accuracy	0.97155361
Error	0.02844639
Sensitivity	0.96202532
specificity	0.99290780
Precision	0.99672131
F1_Score	0.97906602
Success_Clas	1.00000000

Validation dataset	
Performance Measures	Index
Accuracy	0.9600000
Error	0.0400000
Sensitivity	0.9708029
specificity	0.9365079
Precision	0.9708029
F1_Score	0.9708029
Success_Class	1.0000000

Training dataset		
Predicted/actual	0	1
0	140	12
1	1	304

Validation dataset		
Predicted/actual	0	1
0	590	40
1	30	12900

Figure 7. Gradient boosting Model Performance
Source: researchers' analysis

Bagging Classifier

Bagging model is an ensemble learning model where multiple models are trained independently on parallel on different subset and the final model is made by aggregating/averaging the predictions of all fitted models. like its name bagging implies, the objective of bagging is to get a lesser variance than any model has independently by averaging the outcome of other models. Bagging model is represented by this formula. On the training model the model has 0.97 accuracy, 0.026 error, sensitivity 0.97, 0.99 specificity, precision of 0.99, F1-score of 0.98 and success class of 1. While on the validation dataset the model has 0.965 accuracy, 0.035 error, sensitivity 0.98, 0.923 specificity, precision of 0.96, F1-score of 0.97 and success class of 1.

$$\widehat{f_{bag}} = \widehat{f_1}(X) + \widehat{f_2}(X) + \cdots + \widehat{f_b}(X)$$

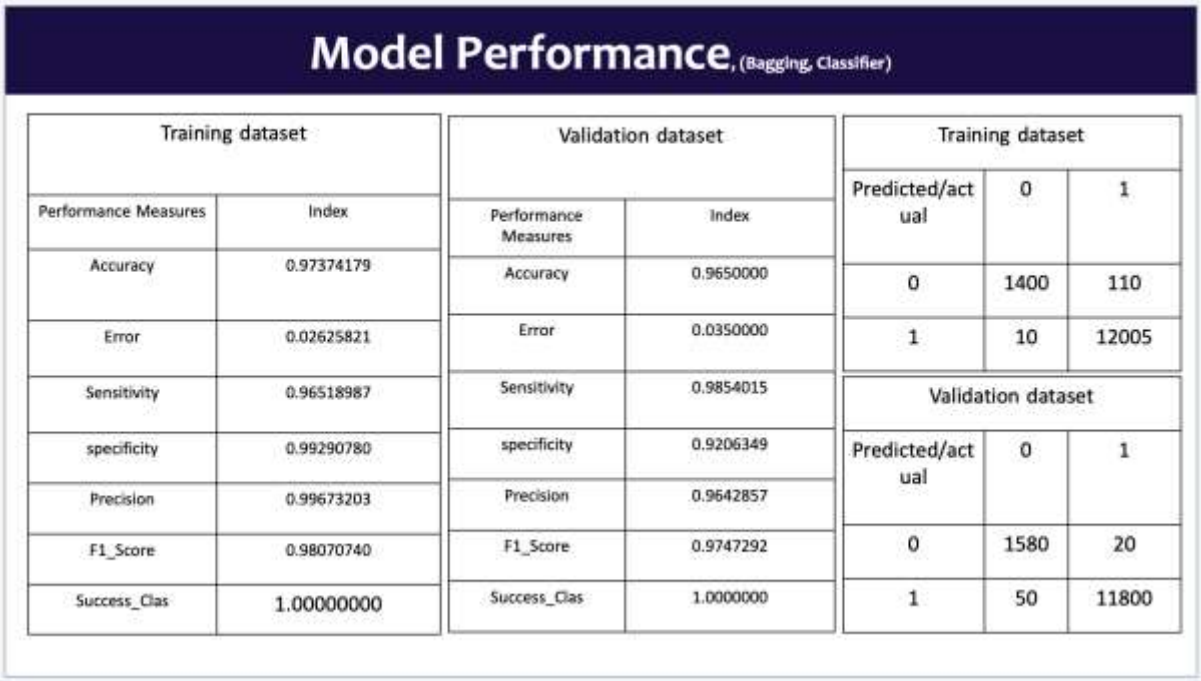


Figure 8. Bagging classifier Model Performance
Source: researchers' analysis

Result and Discussion

Overall, all these models have over 93% accuracy on the validation dataset which proves the usability of this models. However, various algorithms perform best in certain conditions. In this present condition, random forest, bagging, and neural network performed best with 97% accuracy followed by gradient boosting model and decision tree with 96% accuracy and lastly Logistic regression with 94% accuracy. In this study, we will adopt random forest classifier because it has the best performance evaluation like precision, F1- score, sensitivity, specificity, recall, and performance speed compared with the other algorithms tested. From the adopted model, No_of_transaction has most effect on Fraud(ȳ) seconded by IP_address, frequency and average_trans.. Moreso, these algorithms can be fine-tuned to get similar accuracy like random forest with the use of hyperparameter tuning. Hyperparameter tuning entails continuous iteration of parameters to see which gives more overall accuracy of the model.

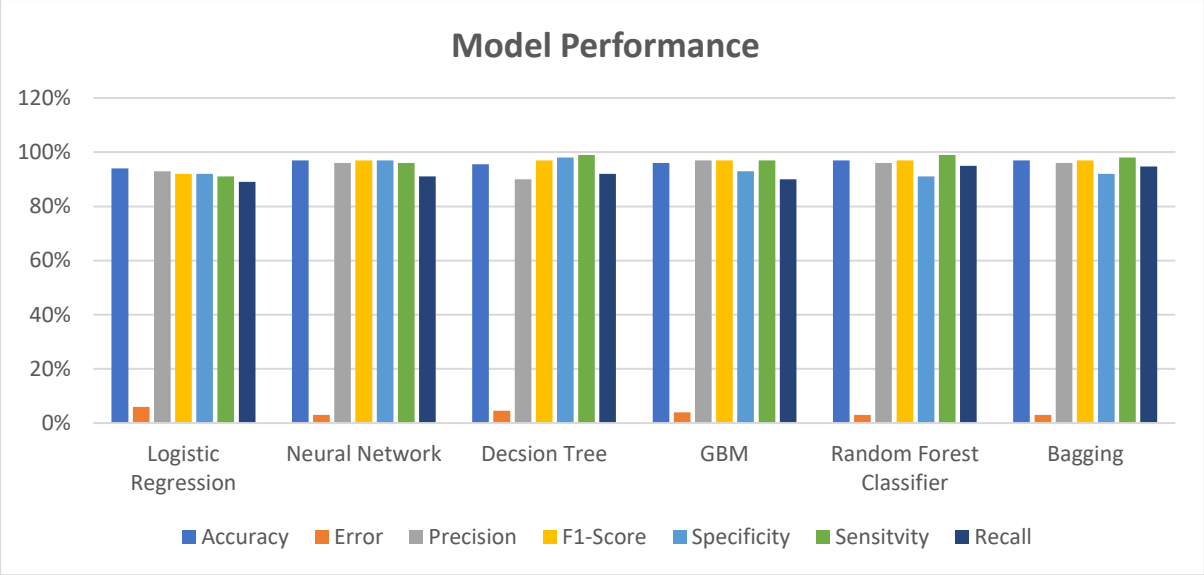


Fig 9. multiple Model Performance
Source: researchers' analysis

Evaluation Parameters	Machine Learning Classifier					
	Logistic Regression	Neural Network	Decsion Tree	GBM	Random Forest Classifier	Bagging
Accuracy	94%	97%	96%	96%	97%	97%
Error	6%	3%	5%	4%	3%	3%
Precision	93%	96%	90%	97%	96%	96%
F1-Score	92%	97%	97%	97%	97%	97%
Specificity	92%	97%	98%	93%	91%	92%
Sensitvity	91%	96%	99%	97%	99%	98%
Recall	89%	91%	92%	90%	95%	95%

Fig 10.

Recommendation and Conclusion:

Overall, fraud loss has presented a significant threat to the financial industry, especially for credit card companies, as they are held accountable for fraudulent transactions carried out within their customers' (cardholders') accounts. These firms have invested a substantial amount of resources in curtailing credit card fraud, employing roles such as fraud analysts, fraud examiners, and fraud investigators. However, all these approaches are focused on identifying fraud after it has already occurred. Credit card companies are now using the potential of machine learning and artificial intelligence to further reduce credit card losses by predicting whether credit card transactions are fraudulent or not. This study clearly shows that machine learning models can indeed assist in predicting credit card transactions.

Furthermore, fraudsters spend a significant amount of time searching for vulnerabilities in fraud prevention models and systems. Organizations should dedicate more time to identifying these vulnerabilities and implementing necessary corrections before fraudsters have the chance to exploit them.

Finally, companies must ensure that the machine learning models they deploy are not overfitting, as this could lead to issues of false positives or true negatives, which are undesirable in credit card transactions. Credit card companies should design machine

learning models that align with their specific business requirements and ensure regular updates of these models, as fraudsters are continuously advancing their tactics in committing these crimes. It's important for credit card companies to educate their customers about the possibilities of credit card fraud and the best approach to reporting any abnormalities noticed in their accounts."

Limitation

This research was performed with dummy data collected from Kaggle with a lot of missing and variance in the data. The data has limited features that was used to fit the model. Such study can be replicated with original data from a company that has a lot of features which can give room to other factors that can improve fraud prevention models. We used just six algorithms. Studies can be conducted to explore other algorithms that can be effective in preventing fraud. Opportunities in fraud prevention is unending; studies can be performed in other states or countries to see the factors that influences fraud the most.

Reference

Aburbeian, A. M., & Ashqar, H. I. (2023, May). Credit Card Fraud Detection Using Enhanced Random Forest Classifier for Imbalanced Data. In *International Conference on Advances in Computing Research* (pp. 605-616). Cham: Springer Nature Switzerland.

Ahmed, M. H. (2022). Credit Card Fraud Detection Techniques: A Survey. *ScienceOpen Preprints*.

Askari.S and Hussain.A, 2017, Credit Card Fraud Detection Using Fuzzy ID3, IEEE, Computing, Communication and Automation (ICCCA), 446-452.

Credit card fraud statistics (2023). (2023, June 20). Merchant Cost Consulting. https://merchantcostconsulting.com/lower-credit-card-processing-fees/credit-card-fraud.

Faraji, Z. (2022). A Review of Machine Learning Applications for Credit Card Fraud Detection with A Case study. *SEISENSE Journal of Management*, *5*(1), 49-59. https://doi.org/10.33215/sjom.v5i1.770

Jain, R., Gour, B., & Dubey, S. (2016). A hybrid approach for credit card fraud detection using rough set and decision tree technique. *International Journal of Computer Applications*, *139*(10), 1-6.

Kasasbeh, B., Aldabaybah, B., & Ahmad, H. (2022). Multilayer perceptron artificial neural networks-based model for credit card fraud detection. *Indonesian Journal of Electrical Engineering and Computer Science*, *26*(1), 362-373.

Khan, S., Alourani, A., Mishra, B., Ali, A., & Kamal, M. (2022). Developing a Credit Card Fraud Detection Model using Machine Learning Approaches. *International Journal of Advanced Computer Science and Applications*, *13*(3).

Kim, L.(2023). Card Not Present Fraud and False Declines: Understanding Risks and Ways to Protect Your Business. Experian. https://www.experian.com/blogs/insights/2023/01/card-not-present-fraud.

New FTC data show consumers reported losing nearly $8.8 billion to scams in 2022. (2023, February 23). Federal Trade Commission. https://www.ftc.gov/news-

events/news/press-releases/2023/02/new-ftc-data-show-consumers-reported-losing-nearly-88-billion-scams-2022

Patidar, R., & Sharma, L. (2011). Credit card fraud detection using neural network. *International Journal of Soft Computing and Engineering (IJSCE)*, *1*(32-38).

Sadgali, I., Sael, N., & Benabbou, F. (2018). Detection of credit card fraud: State of art. *Int. J. Computer. Sci. Network Security*, *18*(11), 76-83.

Salekshahrezaee, Z., Leevy, J. L., & Khoshgoftaar, T. M. (2023). The effect of feature extraction and data sampling on credit card fraud detection. *Journal of Big Data*, *10*(1), 6.

Trivedi, N. K., Simaiya, S., Lilhore, U. K., & Sharma, S. K. (2020). An efficient credit card fraud detection model based on machine learning methods. *International Journal of Advanced Science and Technology*, *29*(5), 3414-3424.

Zeager. M.F, Sridhar.A, Fogal.N, Adams.S, Brown. D.E, and Beling. P.A, 2017, Adversarial Learning in Credit Card Fraud Detection, IEEE, *Systems and Information Engineering Design Symposium (SIEDS)*, 112-116.

Fraud_Tree Model

```
install.packages("caret")
install.packages("MASS")
install.packages("forecast", dependencies = TRUE)
install.packages("MLmetrics")
install.packages("leaps")
install.packages("DALEX")
install.packages("InformationValue")
install.packages("ROCR")
install.packages("gains")
install.packages("neuralnet")
install.packages("rpart.plot")
install.packages("randomForest")
install.packages("gbm")
install.packages("adabag")
install.packages("ipred")
library(adabag)
library(ipred)
library(gbm)
library(randomForest)
library(neuralnet)
library(nnet)
library(ROCR)
library(InformationValue)
library(DALEX)
library(leaps)
library(MLmetrics)
library(forecast)
library(caret)
library(readxl)
Fraud_Mix <- read_excel("~/Desktop/myproject/Fraud_Mix.xlsx")
```

```
View(Fraud_Mix)
summary(Fraud_Mix)
mean(Fraud_Mix$Sales)
Fraud_Mix$new_y <- ifelse(Fraud_Mix$Sales > 697.1303, 0, 1)
Fraud_nn<- Fraud_Mix[-c(1,2)] Fraud_nn
# create detect outlier function detect_outlier <- function(x) {
QuanCle1 <- quanCle(x, probs=.25) QuanCle3 <- quanCle(x, probs=.75)
IQR = QuanCle3-QuanCle1
x > QuanCle3 + (IQR*1.5) | x < QuanCle1 - (IQR*1.5) }
remove_outlier <- function(dataframe, columns=names(dataframe)) {
for (col in columns) {
dataframe <- dataframe[!detect_outlier(dataframe[[col]]), ] }
# return dataframe print("Remove outliers") print(dataframe) }
Fraud_nn_outlier <- remove_outlier(Fraud_nn, c(6)) nrow(Fraud_nn_outlier)
##### Data preprocessing using standardization Fraud_nn_outlier$one <-
Fraud_nn_outlier$new_y ==1 Fraud_nn_outlier$zero <- Fraud_nn_outlier$new_y
==0
set.seed(1)
sample_data <- sample(c(1:647), 457) Fraud_train_data <-
Fraud_nn_outlier[sample_data, ] Fraud_test_data <- Fraud_nn_outlier[-
sample_data, ]
# The Classification Tree Model
library(rpart)
library(rpart.plot)
###_____
_____
#first Tree
set.seed(1)
Frd.rpart.1 <- rpart(new_y ~ No_transaction + frequency + Average_trans + Time +
location + IP_address
+ V1 + V2, data = Fraud_train_data, method = "class")
prp(Frd.rpart.1, type = 1, extra = 1, under = TRUE, split.font = 1, varlen = -10)
prp(Frd.rpart.1)
# prediction
Frd.rpart.1.pred <- predict(Frd.rpart.1, Fraud_train_data, type = "class")
# Confusion Matrix
Frd.rpart.Metrics <- table("Predicted" = Frd.rpart.1.pred, "Actual" =
Fraud_train_data$new_y)
# Perfromance Metrics
Tree_1 <- t(data.frame("Accuracy" =
sum(diag(Frd.rpart.Metrics))/sum(Frd.rpart.MeCcs),
"Error" = 1 - (sum(diag(Frd.rpart.Metrics))/sum(Frd.rpart.Metrics)), "SensiCvity" =
Frd.rpart.MeCcs[2,2] / colSums(Frd.rpart.MeCcs)[2], "Specificity" =
Frd.rpart.Metrics[1,1] / colSums(Frd.rpart.Metrics)[1], "Precision" =
Frd.rpart.Metrics[2,2] / rowSums(Frd.rpart.MeCcs)[2], "F1_Score" = 2 *
((Frd.rpart.Metrics[2,2] /
colSums(Frd.rpart.Metrics)[2])*(Frd.rpart.Metrics[2,2] /
rowSums(Frd.rpart.Metrics)[2]))/ ((Frd.rpart.MeCcs[2,2] /
colSums(Frd.rpart.Metrics)[2])+(Frd.rpart.Metrics[2,2] /
rowSums(Frd.rpart.Metrics)[2])),
))
```

```
"Success_Class" = 1
#_____
_____
```

##Deeper Tree

```
set.seed(1)
Frd.rpart.2 <- rpart(new_y ~ No_transaction + frequency + Average_trans + Time +
location + IP_address
+ V1 + V2, data = Fraud_train_data, method = "class", cp = 0, minsplit = 1)
prp(Frd.rpart.2, type = 1, extra = 1, under = TRUE, split.font = 1, varlen = -10)
prp(Frd.rpart.2)
# prediction on training data
Frd.rpart.2.pred <- predict(Frd.rpart.2, Fraud_train_data, type = "class")
# Confusion Matrix
Frd.rpart.MeCcs.2 <- table("Predicted" = Frd.rpart.1.pred, "Actual" =
Fraud_train_data$new_y)
# Perfromance Metrics
Tree_2 <- t(data.frame("Accuracy" =
sum(diag(Frd.rpart.MeCcs.2))/sum(Frd.rpart.MeCcs.2),
"Error" = 1 - (sum(diag(Frd.rpart.MeCcs.2))/sum(Frd.rpart.MeCcs.2)), "Sensitvity" =
Frd.rpart.MeCcs.2[2,2] / colSums(Frd.rpart.MeCcs.2)[2], "Specificity" =
Frd.rpart.MeCcs.2[1,1] / colSums(Frd.rpart.MeCcs.2)[1], "Precision" =
Frd.rpart.MeCcs.2[2,2] / rowSums(Frd.rpart.MeCcs.2)[2], "F1_Score" = 2 *
((Frd.rpart.MeCcs.2[2,2] /
colSums(Frd.rpart.MeCcs.2)[2])*(Frd.rpart.Metrics[2,2] /
rowSums(Frd.rpart.MeCcs.2)[2]))/ ((Frd.rpart.MeCcs.2[2,2] /
colSums(Frd.rpart.MeCcs.2)[2])+(Frd.rpart.MeCcs.2[2,2] /
rowSums(Frd.rpart.Metrics.2)[2])), "Success_Class" = 1 ))
#Validation on the test dataset
Frd.rpart.2.test <- predict(Frd.rpart.2, Fraud_test_data, type = "class")
length(Frd.rpart.2.test)
length(Fraud_test_data$new_y)
```

#ConfusionMatrix

```
Frd.rpart.MeCcs.3 <- table(Frd.rpart.2.test, Fraud_test_data$new_y)
Tree_3 <- t(data.frame("Accuracy" =
sum(diag(Frd.rpart.MeCcs.3))/sum(Frd.rpart.MeCcs.3), "Error" = 1 -
(sum(diag(Frd.rpart.Metrics.3))/sum(Frd.rpart.Metrics.3)), "Sensitivity" =
Frd.rpart.MeCcs.3[2,2] / colSums(Frd.rpart.MeCcs.3)[2], "Specificity" =
Frd.rpart.Metrcs.3[1,1] / colSums(Frd.rpart.Metrics.3)[1], "Precision" =
Frd.rpart.Metrics.3[2,2] / rowSums(Frd.rpart.Metrics.3)[2], "F1_Score" = 2 *
((Frd.rpart.Metrics.3[2,2] /
colSums(Frd.rpart.Metrics.3)[2])*(Frd.rpart.MeCcs.3[2,2] /
rowSums(Frd.rpart.Metrics.3)[2]))/ ((Frd.rpart.MeCcs.3[2,2] /
colSums(Frd.rpart.MeCcs.3)[2])+(Frd.rpart.MeCcs.3[2,2] /
rowSums(Frd.rpart.MeCcs.3)[2])),
))
"Success_Class" = 1

#_____
_____
```

 # Random Forecast

```
rf.Frd.1 <- randomForest(as.factor(new_y) ~ No_transaction + frequency +
Average_trans + Time + location + IP_address
+ V1 + V2, data = Fraud_train_data,ntree = 500, mtry = 4, nodesize = 5, importance
= TRUE)
## variable importance plot
varImpPlot(rf.Frd.1, type = 1, main = "Variable Importance Plot")
# confusion matrix (Training data)
rf.Frd.1.pred <- predict(rf.Frd.1, Fraud_train_data)
rf.Frd.tab1 <- table("Predicted" =rf.Frd.1.pred, "Actual" = Fraud_train_data$new_y)
rf.Frd.tab1
###validation dataset
rf.Frd.2.pred <- predict(rf.Frd.1, Fraud_test_data)
rf.Frd.tab2 <- table("Predicted" =rf.Frd.2.pred, "Actual" = Fraud_test_data$new_y)
rf.Frd.tab2
RF_perform <- t(data.frame("Accuracy" = sum(diag(rf.Frd.tab2))/sum(rf.Frd.tab2),
"Error" = 1 - (sum(diag(rf.Frd.tab2))/sum(rf.Frd.tab2)),
"Sensitvity" = rf.Frd.tab2[2,2] / colSums(rf.Frd.tab2)[2],
"Specificity" = rf.Frd.tab2[1,1] / colSums(rf.Frd.tab2)[1],
"Precision" = rf.Frd.tab2[2,2] / rowSums(rf.Frd.tab2)[2],
"F1_Score" = 2 * ((rf.Frd.tab2[2,2] / colSums(rf.Frd.tab2)[2])*(rf.Frd.tab2[2,2] /
rowSums(rf.Frd.tab2)[2]))/
((rf.Frd.tab2[2,2] / colSums(rf.Frd.tab2)[2])+(rf.Frd.tab2[2,2] /
rowSums(rf.Frd.tab2)[2])),
))
"Success_Class" = 1
#_____
_____
```

 # Boosting Method

```
install.packages("gbm") # gradient boosting classification
library(gbm)
set.seed(1)
Frd.gbm.1 <- gbm(new_y ~ No_transaction + frequency + Average_trans + Time +
location + IP_address
+ V1 + V2, data = Fraud_train_data) # Predictions (Training Data)
gbm.pred <- predict(Frd.gbm.1 , Fraud_train_data, type="response") pred.gbm.1 =
ifelse(data.frame(gbm.pred) > 0.6, 1, 0)
# Confusion Matrix training
gbm.table.1 <- table("Predicted" = pred.gbm.1 , "Actual" =
Fraud_train_data$new_y) gbm.table.1
##validation with test data
gbm.pred.test <- predict(Frd.gbm.1 , Fraud_test_data, type="response") pred.gbm.2
= ifelse(data.frame(gbm.pred.test ) > 0.6, 1, 0)
# Confusion Matrix training
gbm.table.2 <- table("Predicted" = pred.gbm.2 , "Actual" = Fraud_test_data$new_y)
gbm.table.2
GBM_perform <- t(data.frame("Accuracy" =
sum(diag(gbm.table.2))/sum(gbm.table.2), "Error" = 1 -
(sum(diag(gbm.table.2))/sum(gbm.table.2)),
"Sensitivity" = gbm.table.2[2,2] / colSums(gbm.table.2)[2], "Specificity" =
gbm.table.2[1,1] / colSums(gbm.table.2)[1],
"Precision" = gbm.table.2[2,2] / rowSums(gbm.table.2)[2],
```

```
"F1_Score" = 2 * ((gbm.table.2[2,2] / colSums(gbm.table.2)[2])*(gbm.table.2[2,2] /
rowSums(gbm.table.2)[2]))/
((gbm.table.2[2,2] / colSums(gbm.table.2)[2])+(gbm.table.2[2,2] /
rowSums(gbm.table.2)[2])),
))
"Success_Class" = 1
#_____
_____
```

 # Bagging Method

```
set.seed(1)
Frd.bb.1 <- bagging(new_y ~ No_transaction + frequency + Average_trans + Time +
location + IP_address
+ V1 + V2, data = Fraud_train_data) # Predictions (Training Data)
bbm.pred <- predict(Frd.bb.1 , Fraud_train_data, type="response") pred.bbm.1 =
ifelse(data.frame(bbm.pred) > 0.6, 1, 0)
# Confusion Matrix training
bbm.table.1 <- table("Predicted" = pred.bbm.1 , "Actual" =
Fraud_train_data$new_y) bbm.table.1
##validation with test data
bbm.pred.test <- predict(Frd.bb.1 , Fraud_test_data, type="response") pred.bbm.2 =
ifelse(data.frame(bbm.pred.test ) > 0.6, 1, 0)
# Confusion Matrix training
bbm.table.2 <- table("Predicted" = pred.bbm.2 , "Actual" = Fraud_test_data$new_y)
bbm.table.2
BBM_perform <- t(data.frame("Accuracy" =
sum(diag(bbm.table.2))/sum(bbm.table.2), "Error" = 1 -
(sum(diag(bbm.table.2))/sum(bbm.table.2)),
"Sensitivity" = bbm.table.2[2,2] / colSums(bbm.table.2)[2], "Specificity" =
bbm.table.2[1,1] / colSums(bbm.table.2)[1], "Precision" = bbm.table.2[2,2] /
rowSums(bbm.table.2)[2], "F1_Score" = 2 * ((bbm.table.2[2,2] /
colSums(bbm.table.2)[2])*(bbm.table.2[2,2] / rowSums(bbm.table.2)[2]))/
((bbm.table.2[2,2] / colSums(bbm.table.2)[2])+(bbm.table.2[2,2] /
rowSums(bbm.table.2)[2])), "Success_Class" = 1
))
```