Credit Card Fraud Detection

Nirdesh Singh - 20BCE7062 (nirdesh.20bce7062@vitap.ac.in),

Gaurav Sharma - 20BCE7443 (gaurav.20bce7443@vitap.ac.in),

School of Computer Engineering (SCOPE),

Vellore Institute of Technology – AP Campus.

Abstract

The number of online payment options has expanded thanks to e-commerce and several other websites, raising the possibility of online fraud. Due to an increase in fraud rates, academics have begun employing various machine learning techniques to identify and analyse online transaction fraud. The primary purpose of the study is to create and implement unique fraud detection algorithm for streaming transaction data with goal of analysing historical customer transaction information and extracting behavioural patterns, whereby cardholders are grouped according to value of their transactions. Then, using the sliding window method, combine the transactions performed by cardholders from several groups so that each group's unique behaviour can be retrieved. The groupings are then used to train various classifiers individually in future. The classifier with the highest rating score can then be selected as one of the most effective ways to detect fraud. In this study, we used credit card dataset.

Keywords – Credit Card, Amount, AUC-ROC Curve, Logistic Regression, Decision Tree, Artificial Neural Network, Gradient Boosting Model.

Introduction

Detecting credit card fraud entails finding fraudulent purchase attempts and rejecting them rather than executing the sale. Many different methods and strategies are available for identifying fraud, and the majority of merchants use a mix of many of them.

Payment cards are simple to use since identifying your account and authorising the transaction simply need sending a small amount of digits to the bank. They are additionally exposed due of their simplicity. On a few straightforward numbers that must be communicated with the persons you are trading with, it is quite difficult to enforce strict data security.

The cost of credit card theft to the worldwide economy exceeds $24 billion annually, and the amount is rising. The effects of fraud are more severe for smaller retailers, which is why it's crucial to have procedures and tools in place to identify fraud early on.

Literature survey/Related Work

A decision tree-based model that combines Luhn's and Hunt's algorithms has been put out by Prajal Save. To detect if an incoming transaction is fraudulent or not, Luhn's algorithm is employed. The input, which is the credit card number, is used to validate credit card numbers. The degree of outlierness and address mismatch are used to evaluate how far each incoming transaction deviates from the typical profile of the cardholder.

Three machine-learning techniques were described and put into use to find fake transactions. The performance of classifiers or predictors is assessed using a variety of metrics, including the Vector Machine, Random Forest, and Decision Tree. Either these measures depend on or don't depend on prevalence. Additionally, similar methods are employed in mechanisms that identify credit card fraud, and the outcomes of these algorithms have been contrasted.

Algorithms for detecting credit card fraud show which transactions are likely to be fraudulent. To perform prediction, grouping, and outlier identification, we compared machine learning techniques. Shiyang Xuan and others, The Random Forest classifier was employed to train it on the behavioural traits of credit card transactions. The following categories are employed to train the characteristics of legitimate and dishonest behaviour: Random forest based on CART as well as random forest based on random trees. Performance metrics can be used to evaluate the model's performance.

Numerous supervised learning algorithms, from the traditional to the contemporary, have been taken into consideration. These include of Bayesian methods, hybrid algorithms, deep and traditional neural networks, tree-based algorithms, and so forth. It has been evaluated how well machine learning algorithms can spot credit card fraud. Numerous well-known algorithms in the supervised, ensemble, and unsupervised categories were assessed on various criteria. It is concluded that unsupervised algorithms perform better across all metrics both in isolation and in comparison to other approaches because they are better at handling dataset skewness.

The system generated a fraud score for that specific transaction using a variety of criteria and algorithms to forecast the outcome of fraud. An approach for detecting fraud using deep networks has been presented by Xiaohan Yu. The article presented a deep neural network approach for identifying credit card fraud. It has discussed applications for deep neural networks as well as the neural network algorithm method.

Experimental design/ Methodology

Dataset used

We used the Card Transactions dataset, which includes both fraudulent and legitimate transactions, to carry out the credit card fraud detection.

Data Exploration

The datasets including credit card transaction data were first imported. The information in the creditcard data dataframe was then examined. We next explored the various elements of this dataframe after showing the creditcard data using the head() and tail() functions.

Data Manipulation

We used the scale() method to scale the data in this project portion. This was then applied to our creditcard data amount's amount component. The data is organised according to a defined range with the use of scaling. As a result, the dataset does not contain any extreme values that may prevent the model from working as intended.

Data Modelling

We separated the dataset into a training set and a test set with a split ratio of 0.80 after standardising the entire dataset. This indicates that 80% of the data will be associated with the train data and 20% with the test data. The dim() method was then used to determine the dimensions.

Fitting Logistic Regression Model

We fit the first model in this project segment. With logistic regression, we started. For estimating the likelihood of fraud or not, we employed it. We proceeded to implement this model using the test data. Once we summarised the model, we displayed it using charts. We displayed the ROC curve, also known as the Receiver Optimistic Characteristics, to evaluate the model's performance. In order to do this, we loaded the ROC package first before plotting the ROC curve and evaluating its performance.

Fitting a Decision Tree Model

In order to determine what class the item belongs to, we then constructed a decision tree algorithm to depict the results of a choice. We then developed the decision tree model and plotted it using the rpart.plot() method. To draw the decision tree, we specially employed the recursive partitioning.

Artificial Neural Network

A sort of machine learning algorithm that is based on the human nervous system is called an artificial neural network. The ANN models are able to learn the patterns using the previous data and are able to do classification on the input data. We imported the neuralnet package that allowed me to create the ANNs. Then, we used the plot() method to plot it. Now, there is a range of values for Artificial Neural Networks that is between 1 and 0. I set a threshold of 0.5, that is, numbers over 0.5 will equate to 1 and the rest will be 0.

Gradient Boosting (GBM)

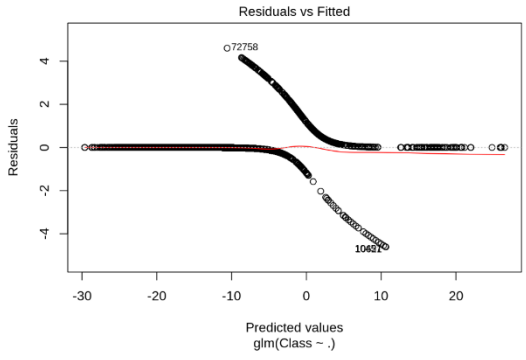
A well-liked machine learning approach called gradient boosting is employed to carry out classification and regression tasks. Weak decision trees are one of the underlying ensemble models that make up this model. An effective gradient boosting model is created by the combination of these decision trees. In the model, we used the gradient descent technique.

AUC-ROC Curve

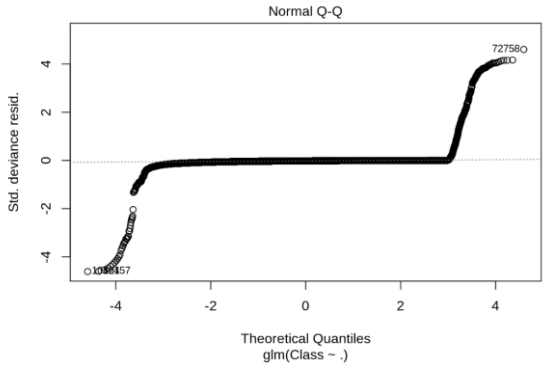
In the concluding stage of the research, we constructed and drew a ROC curve assessing the sensitivity and specificity of the model. The area under the curve is calculated and plotted using the print command. The area of a ROC curve may be used to evaluate a model's sensitivity and accuracy.

Results & discussion

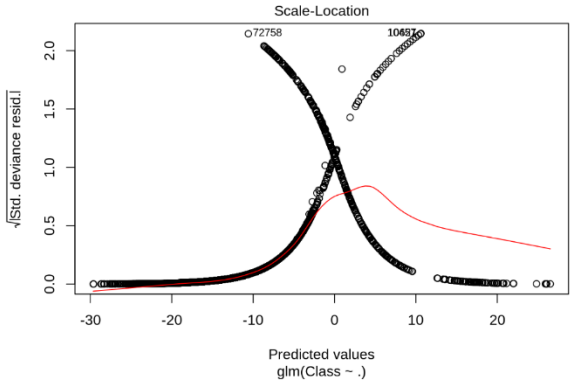
First, logistic regression was fitted. For estimating the likelihood of a result in a class, such as pass/fail, positive/negative, and in our instance, fraud/not fraud, logistic regression is utilised. We obtained the following results:

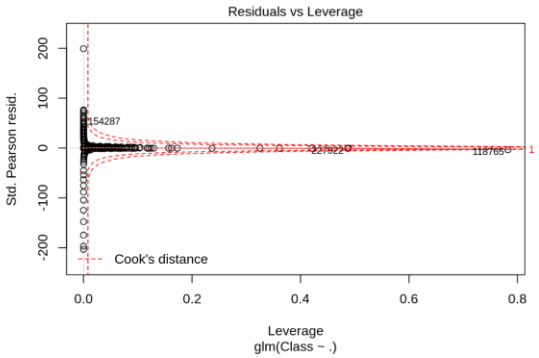


Residuals vs Fitted



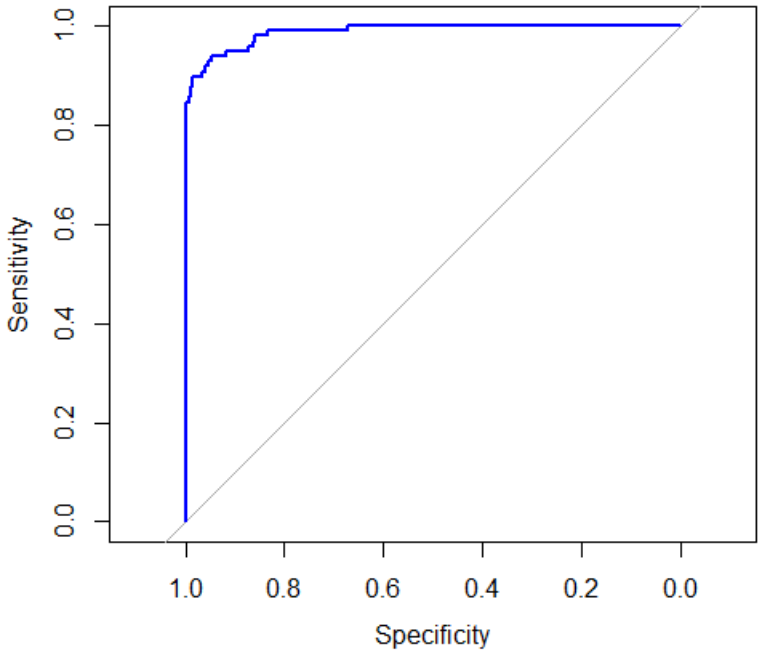
Normal Q-Q





Scale-Location

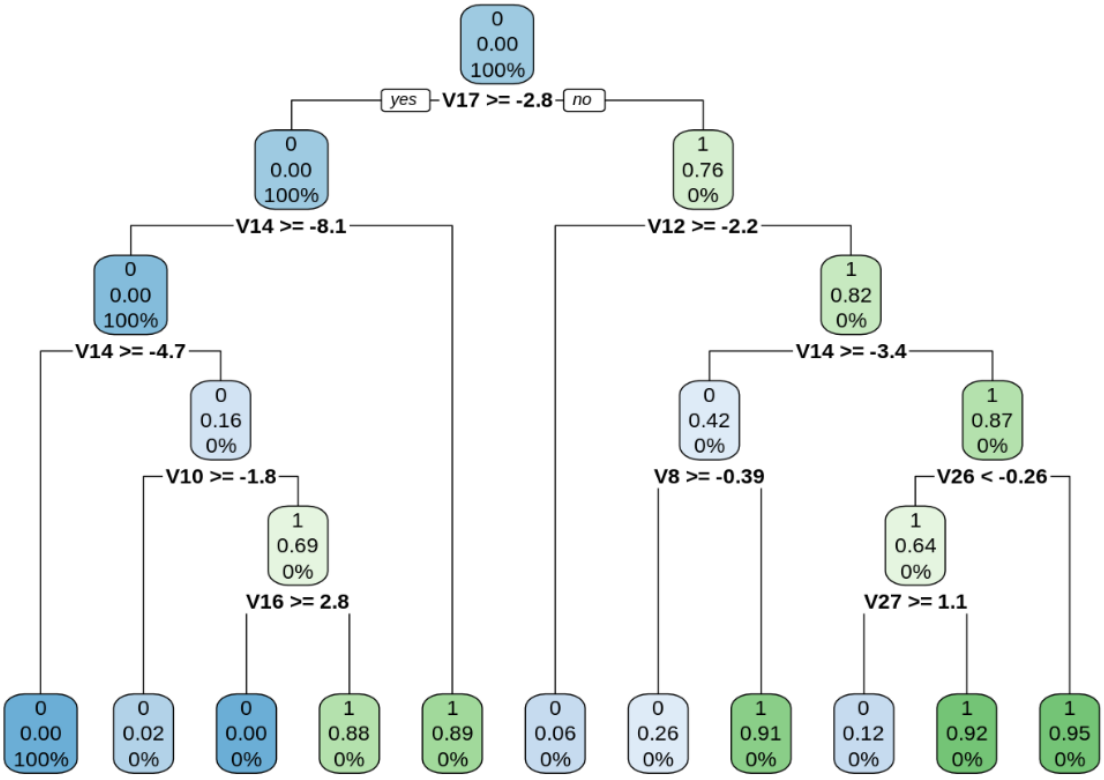
Residual vs Leverage



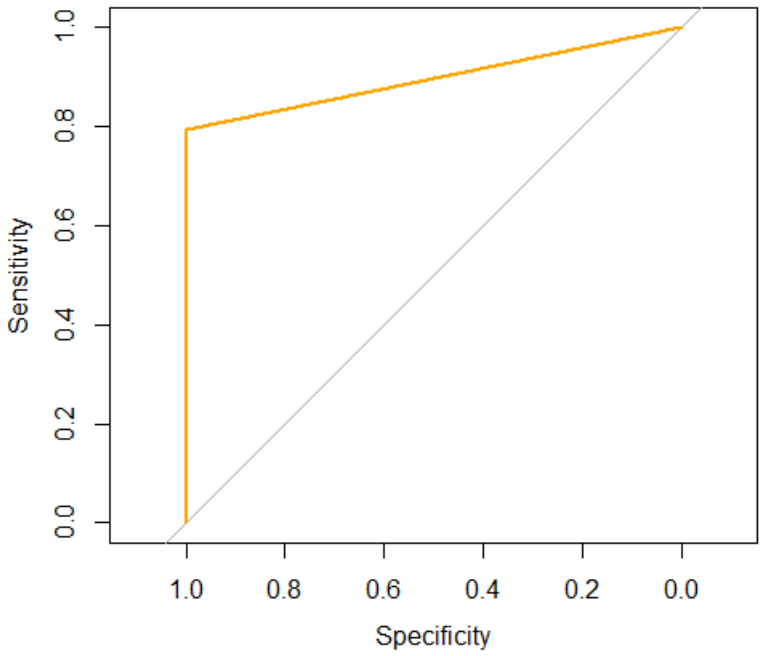
Specificity vs Sensitivity for Logistic Regression Model

Using the logistic regression model, we obtained a precision of 0.9875 and an accuracy of 0.9993855.

Next, we put a decision tree algorithm into practise. To visualise the results of a choice, we employ decision trees. These results are essentially a consequence that allows us to determine what class the object belongs to. Now that our decision tree model has been implemented, we will plot it via the rpart.plot() method. To draw the decision tree, we will explicitly employ the recursive partitioning.



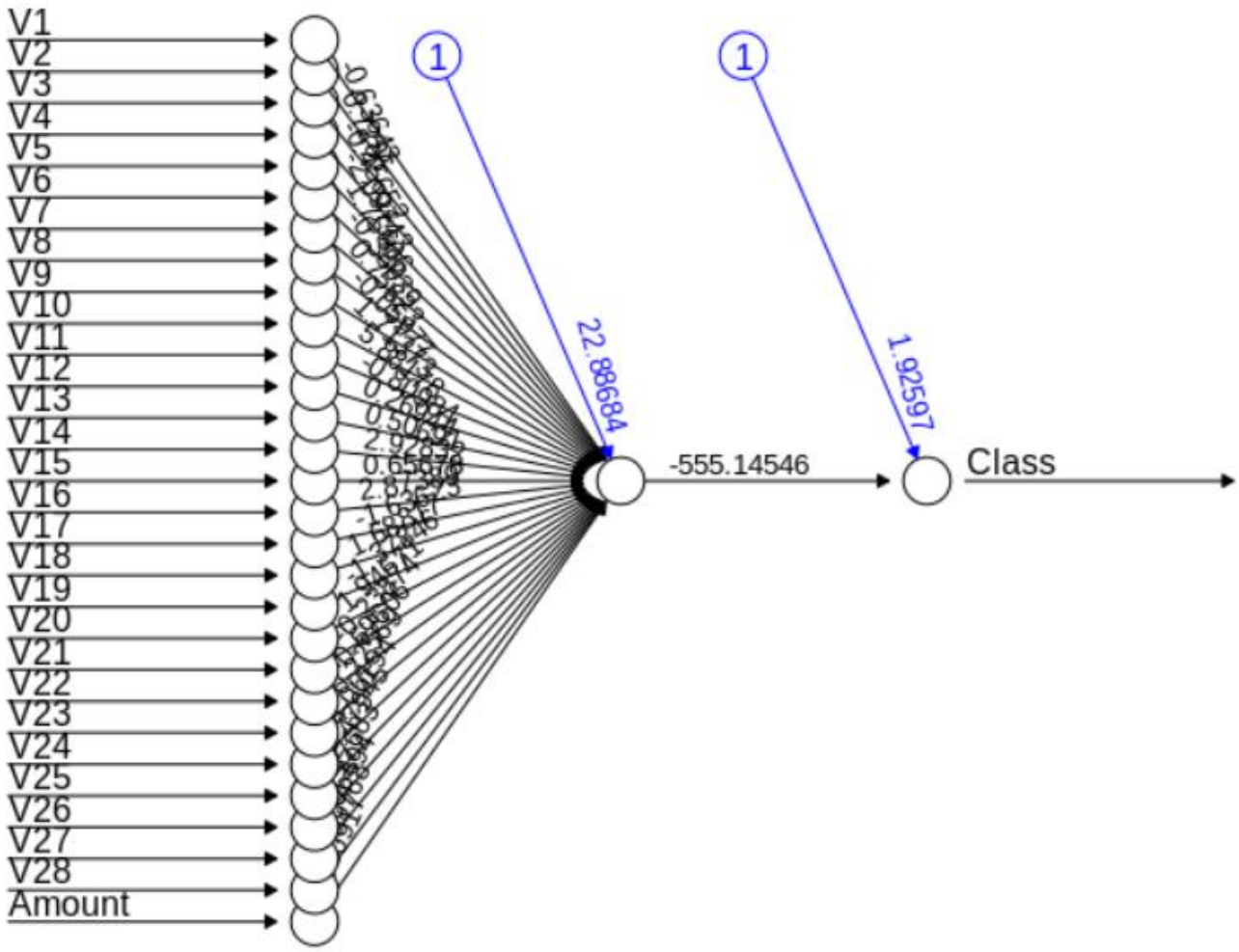
Decision Tree



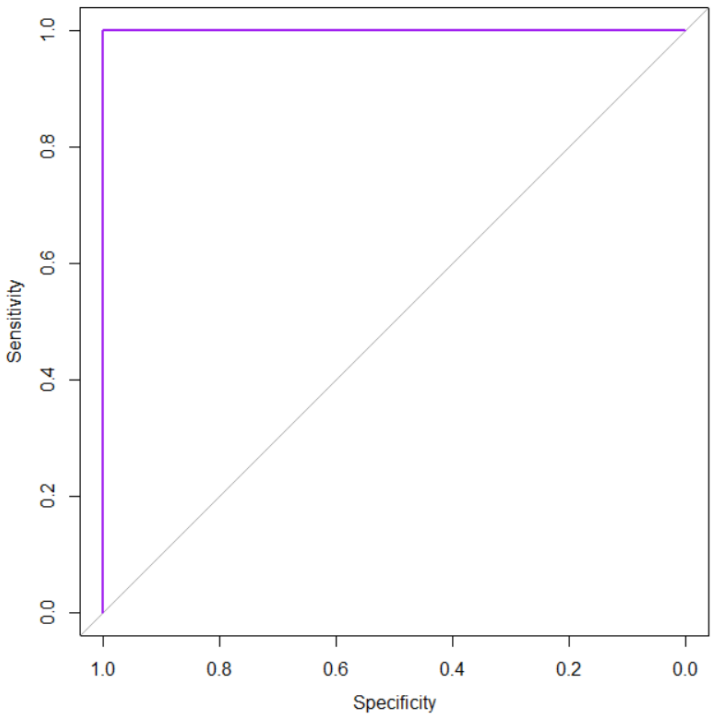
Specificity vs Sensitivity for Decision Tree Model

Using the decision tree model, we obtained a precision of 0.8973 and an accuracy of 0.7412339.

A sort of machine learning algorithm that is based on the human nervous system is called an artificial neural network. The ANN models may do categorization on the input data and can learn patterns from past data. The neuralnet package that would enable us to use our ANNs is imported. Then, we used the plot() method to plot it. Now, there is a range of values for Artificial Neural Networks that is between 1 and 0. We established a threshold of 0.5, meaning that numbers above this value correspond to 1, and values below this value to 0. We put this into practise as follows:



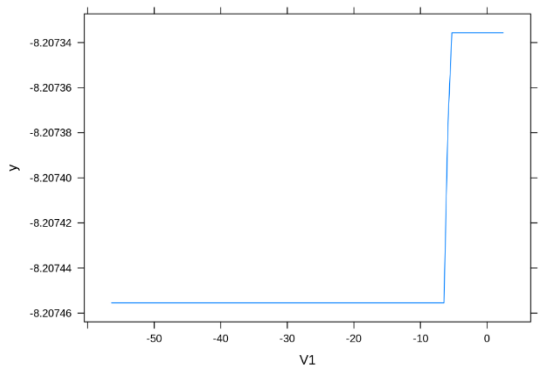
Artificial Neural Network (ANN)



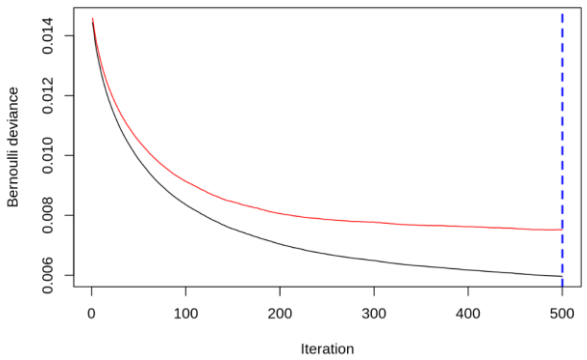
Specificity vs Sensitivity for Artificial Neural Network Model

Using the ANN model, we obtained a precision of 0.9999 and an accuracy of 0.9998947.

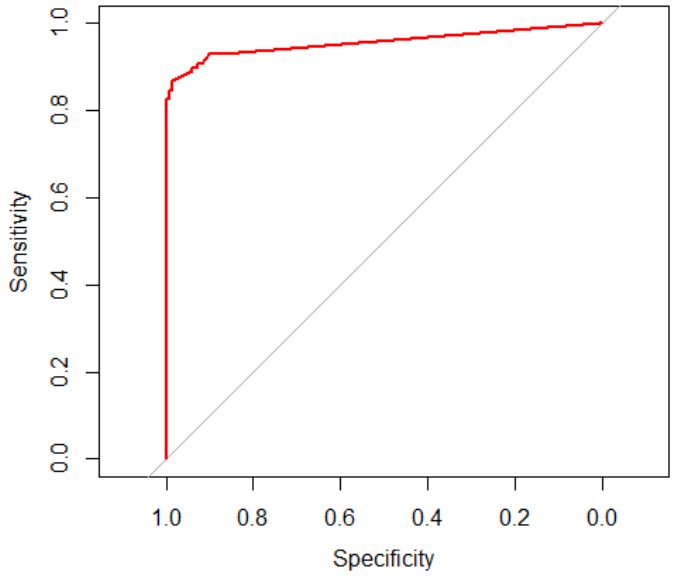
A well-liked machine learning approach called gradient boosting is employed to carry out classification and regression tasks. Weak decision trees are one of the underlying ensemble models that make up this model. An effective gradient boosting model is created by the combination of these decision trees. The following is how we'll include the gradient descent technique into our model:



V1 vs y



Iteration vs Bernoulli deviance



Specificity vs Sensitivity for Gradient Boosting Model

Using the Gradient Boosting Model, we obtained a precision of 0.9555 and an accuracy of 0.9993504.

Following is a summary of the four models' precision and accuracy:

Sl. No.	Model Used	Precision	Accuracy
1	Logistic Regression Model	0.9875	0.9993855
2	Decision Tree Model	0.8973	0.7412339
3	Artificial Neural Network Model	0.9999	0.9998947
4	Gradient Boosting Model	0.9555	0.9993504

Table of Models' Accuracy and Precision

We can see that the Artificial Neural Network (ANN) model has the greatest accuracy and precision, respectively, of 0.9998947 and 0.9999. Contrarily, the Decision Tree Model has the lowest accuracy (0.7412339) and precision (0.8973), respectively.

Future scope and Limitation

Even though we fell short of our objective of 100% accuracy in fraud detection, we did manage to develop a system that, given enough time and data, may get very near to that objective. There

is some potential for improvement here, as with any effort of this nature. Due to the nature of the project, it is possible to integrate many algorithms as modules and combine their outputs to improve the final result's accuracy.

The absence of credit card databases is one of this's drawbacks. The dataset has more opportunities for development. When was previously shown, as the dataset size grows, the algorithms' accuracy rises. Consequently, additional data will undoubtedly improve the model's ability to identify frauds and decrease the amount of false positives. However, the banks themselves must formally approve this.

Conclusion

Unquestionably, using a credit card fraudulently is a criminal act of dishonesty. The most popular fraud schemes, as well as how to spot them, are listed in this article, which also reviews current research in the area. Along with the method, pseudocode, description of how it is implemented, and results of experimentation, this work has also provided a detailed explanation of how machine learning may be used to improve fraud detection. The algorithm's accuracy is greater than 99.98%, and its precision is 99.99%. This high proportion of precision and accuracy is predicted given the stark disparity between the amount of transactions that are legitimate and those that are real.

References

[1] Vaishnavi Nath Dornadula, “Credit Card Fraud Detection using Machine Learning Algorithms”, doi : 10.1016/j.procs.2020.01.057.

[2] Aisha Fayyomi, Derar Eleyan, “A Survey Paper On Credit Card Fraud Detection Techniques”, ISSN 2277-8616.

[3] S P Maniraj, “Credit Card Fraud Detection using Machine Learning and Data Science”, doi : 10.17577/IJERTV8IS090031.

[4] P. Save, P. Tiwarekar, K. N., and N. Mahyavanshi, “A Novel Idea for Credit Card Fraud Detection using Decision Tree”, Int. J. Comput. Appl., vol. 161, no. 13, pp. 6–9, 2017, doi: 10.5120/ijca2017913413.

[5] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang, and C. Jiang, “Random forest for credit card fraud detection”, ICNSC 2018 - 15th IEEE Int. Conf. Networking, Sens. Control, pp. 1–6, 2018, doi: 10.1109/ICNSC.2018.8361343.

[6] X. Yu, X. Li, Y. Dong, and R. Zheng, “A Deep Neural Network Algorithm for Detecting Credit Card Fraud”, Proc. - 2020 Int. Conf. Big Data, Artif. Intell. Internet Things Eng. ICBAIE 2020, pp. 181–183, 2020, doi: 10.1109/ICBAIE49996.2020.00045.