iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** XI **Month of publication:** November 2022

DOI: https://doi.org/10.22214/ijraset.2022.47456

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue XI Nov 2022- Available at www.ijraset.com

Credit Card Fraud Detection

Sanskriti Shevgaonkar[1], Priyanka Khadse[2], Omkar Shinde[3], Tanmay Kulkarni[4], Prof. Avinash Gondal[5]
[1, 2, 3, 4, 5]*Department of Information Technology, Watumull College of Engineering Ulhasnagar*

I. INTRODUCTION

'Fraud' in credit card transactions is unauthorized and unwanted usage of an account by someone other than the owner of that account. Necessary prevention measures can be taken to stop this abuse and the behavior of such fraudulent practices can be studied to minimize it and protect against similar occurrences in the future. In other words, Credit Card Fraud can be defined as a case where a person uses someone else's credit card for personal reasons while the owner and the card issuing authorities are unaware of the fact that the card is being used. This problem is particularly challenging from the perspective of learning, as it is characterized by various factors such as class imbalance. The number of valid transactions far outnumber fraudulent ones. Also, the transaction patterns often change their statistical properties over the course of time.

II. SCOPE

Fraud detection involves monitoring the activities of populations of users in order to estimate, perceive or avoid objectionable behavior, which consist of fraud, intrusion, and defaulting. This is a very relevant problem that demands the attention of communities such as machine learning and data science where the solution to this problem can be automated.

III. PLATFORM: GOOGLE COLAB

Google Colab was developed by Google to provide free access to GPU's and TPU's to anyone who needs them to build a machine learning or deep learning model. Google Colab can be defined as an improved version of Jupyter Notebook.
As a programmer, we can perform the following using Google Colab. Write and execute code in Python Document your code that supports mathematical equations Create/Upload/Share notebooks Import/Save notebooks from/to Google Drive Import/Publish notebooks from GitHub Import external datasets e.g. from Kaggle Integrate PyTorch, TensorFlow, Keras, OpenCV Free Cloud service with free GPU.
Colab, or Colaboratory is an interactive notebook provided by Google (primarily) for writing and running Python through a browser. We can perform data analysis, create models, evaluate these models in Colab. The processing is done on Google-owned servers in the cloud. We only need a browser and a fairly stable internet connection. Colab is a great alternative tool to facilitate our work, whether as a student, professional, or researcher. Although Colab is primarily used for coding in Python, apparently we can also use it for R (#Rstats). We can also run R in Google Colab and can mount Google Drive or access BigQuery in R notebook.

A. Software Specifications

1) Google Colaboratory

B. Hardware Specifications

1) Microsoft® Windows® 7/8/10 (32- or 64-bit)
2) 3 GB RAM minimum, 8 GB RAM recommended;
3) 2 GB of available disk space minimum
4) core processor of i3 minimum or above.

C. Dataset

1) Creditcard.csv which is available on Kaggle. (https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud)

D. Packages Requried

1) ranger
2) caret
3) data.table
4) caTools
5) rpart.plot
6) neuralnet
7) gbm
8) pROC

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue XI Nov 2022- Available at www.ijraset.com

IV. LITERATURE REVIEW

A Fraud act as the unlawful or criminal deception intended to result in financial or personal benefit. It is a deliberate act that is against the law, rule or policy with an aim to attain unauthorized financial benefit. Numerous literatures pertaining to anomaly or fraud detection in this domain have been published already and are available for public usage. A comprehensive survey conducted by Clifton Phua and his associates have revealed that techniques employed in this domain include data mining applications, automated fraud detection, adversarial detection.

In another paper, Suman, Research Scholar, GJUS&T at Hisar HCE presented techniques like Supervised and Unsupervised Learning for credit card fraud detection. Even though these methods and algorithms fetched an unexpected success in some areas, they failed to provide a permanent and consistent solution to fraud detection. A similar research domain was presented by Wen-Fang YU and Na Wang where they used Outlier mining, Outlier detection mining and Distance sum algorithms to accurately predict fraudulent transaction in an emulation experiment of credit card transaction data set of one certain commercial bank. Outlier mining is a field of data mining which is basically used in monetary and internet fields. It deals with detecting objects that are detached from the main system i.e. the transactions that aren't genuine. They have taken attributes of customer's behaviour and based on the value of those attributes they've calculated that distance between the observed value of that attribute and its predetermined value.

Unconventional techniques such as hybrid data mining/complex network classification algorithm is able to perceive illegal instances in an actual card transaction data set, based on network reconstruction algorithm that allows creating representations of the deviation of one instance from a reference group have proved efficient typically on medium sized online transaction. There have also been efforts to progress from a completely new aspect. Attempts have been made to improve the alert feedback interaction in case of fraudulent transaction. In case of fraudulent transaction, the authorized system would be alerted and a feedback would be sent to deny the ongoing transaction. Artificial Genetic Algorithm, one of the approaches that shed new light in this domain, countered fraud from a different direction.

In 2015, J. Esmaily and R. Moradinezhad in their paper proposed a hybrid of artificial neural network and decision tree. In their model they used a two-phase approach. In first phase the classification results of Decision tree and Multilayer perceptron were used to generate a new dataset which in second phase is feed into Multilayer perceptron to finally classify the data. This model promises reliability by giving very low false detection rate. Siddhartha Bhattacharyya and 4 others in their paper in 2011 did a detailed comparative study of Support vector machine and random forest along with logistic regression. They concluded through experiments that Random Forest technique shows most accuracy followed by Logistic Regression and Support Vector Machine.

V. IMPLEMENTATION

In the first step of this data science project, we will perform data exploration. We will import the essential packages required for this role and then read our data. Finally, we will go through the input data to gain nec- essary insights about it.

VI. READING EVENTS FROM CREDITCARD.CSV

Before going to ccfd analysis, the first step is to read the data for performing analysis on. The data is saved in dataset named as creditcard.csv. This dataset contains 0.28 million record with various features. The events saved in dataset are unstructured. To perform analysis, reading of data set is done using command "read.csv".

creditcard_data <- read.csv("/content/creditcard.csv")

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18
0	-1.35981	-0.07278	2.536347	1.378155	-0.33832	0.462388	0.239599	0.098698	0.363787	0.090794	-0.5516	-0.6178	-0.99139	-0.31117	1.468177	-0.4704	0.207971	0.025791
0	1.191857	0.266151	0.16648	0.448154	0.060018	-0.08236	-0.0788	0.085102	-0.25543	-0.16697	1.612727	1.065235	0.489095	-0.14377	0.635558	0.463917	-0.1148	-0.18336
1	-1.35835	-1.34016	1.773209	0.37978	-0.5032	1.800499	0.791461	0.247676	-1.51465	0.207643	0.624501	0.066084	0.717293	-0.16595	2.345865	-2.89008	1.109969	-0.12136
1	-0.96627	-0.18523	1.792993	-0.86329	-0.01031	1.247203	0.237609	0.377436	-1.38702	-0.05495	-0.22649	0.178228	0.507757	-0.28792	-0.63142	-1.05965	-0.68409	1.965775
2	-1.15823	0.877737	1.548718	0.403034	-0.40719	0.095921	0.592941	-0.27053	0.817739	0.753074	-0.82284	0.538196	1.345852	-1.11967	0.175121	-0.45145	-0.23703	-0.03819
2	-0.42597	0.960523	1.141109	-0.16825	0.420987	-0.02973	0.476201	0.260314	-0.56867	-0.37141	1.341262	0.359894	-0.35809	-0.13713	0.517617	0.401726	-0.05813	0.068653
4	1.229658	0.141004	0.045371	1.202613	0.191881	0.272708	-0.00516	0.081213	0.46496	-0.09925	-1.41691	-0.15383	-0.75106	0.167372	0.050144	-0.44359	0.002821	-0.61199
7	-0.64427	1.417964	1.07438	-0.4922	0.948934	0.428118	1.120631	-3.80786	0.615375	1.249376	-0.61947	0.291474	1.757964	-1.32387	0.686133	-0.07613	-1.22213	-0.35822
7	-0.89429	0.286157	-0.11319	-0.27153	2.669599	3.721818	0.370145	0.851084	-0.39205	-0.41043	-0.70512	-0.11045	-0.28625	0.074355	-0.32878	-0.21008	-0.49977	0.118765
9	-0.33826	1.119593	1.044367	-0.22219	0.499361	-0.24676	0.651583	0.069539	-0.73673	-0.36685	1.017614	0.83639	1.006844	-0.44352	0.150219	0.739453	-0.54098	0.476677
10	1.449044	-1.17634	0.91386	-1.37567	-1.97138	-0.62915	-1.42324	0.048456	-1.72041	1.626659	1.199644	-0.67144	-0.51395	-0.09505	0.23093	0.031967	0.253415	0.854344
10	0.384978	0.616109	-0.8743	-0.09402	2.924584	3.317027	0.470455	0.538247	-0.55889	0.309755	-0.25912	-0.32614	-0.09005	0.362832	0.928904	-0.12949	-0.80998	0.359985
10	1.249999	-1.22164	0.38393	-1.2349	-1.48542	-0.75323	-0.6894	-0.22749	-2.09401	1.323729	0.227666	-0.24268	1.205417	-0.31763	0.725675	-0.81561	0.873936	-0.84779
11	1.069374	0.287722	0.828613	2.71252	-0.1784	0.337544	-0.09672	0.115982	-0.22108	0.46023	-0.77366	0.323387	-0.01108	-0.17849	-0.65556	-0.19993	0.124005	-0.9805
12	-2.79185	-0.32777	1.64175	1.767473	-0.13659	0.807596	-0.42291	-1.90711	0.755713	1.151087	0.844555	0.792944	0.370448	-0.73498	0.406796	-0.30306	-0.15587	0.778265
12	-0.75242	0.345485	2.057323	-1.46864	-1.15839	-0.07785	-0.60858	0.003603	-0.43617	0.747731	-0.79398	-0.77041	1.047627	-1.0666	1.106953	1.660114	-0.27927	-0.41999
12	1.103215	-0.0403	1.267332	1.289091	-0.736	0.288069	-0.58606	0.18938	0.782333	-0.26798	-0.45031	0.936708	0.70838	-0.46865	0.354574	-0.24663	-0.00921	-0.59591
13	-0.43691	0.918966	0.924591	-0.72722	0.915679	-0.12787	0.707642	0.087962	-0.66527	-0.73798	0.324098	0.277192	0.252624	-0.2919	-0.18452	1.143174	-0.92871	0.68047
14	-5.40126	-5.45015	1.186305	1.736239	3.049106	-1.76341	-1.55974	0.160842	1.23309	0.345173	0.91723	0.970117	-0.26657	-0.47913	-0.52661	0.472004	-0.72548	0.075081
15	1.492936	-1.02935	0.454795	-1.43803	-1.55543	-0.72096	-1.08066	-0.05313	-1.97868	1.638076	1.077542	-0.63205	-0.41696	0.052011	-0.04298	-0.16643	0.304241	0.554432
16	0.694885	-1.36182	1.029221	0.834159	-1.19121	1.309109	-0.87859	0.44529	-0.4462	0.568521	1.019151	1.298329	0.42048	-0.37265	-0.80798	-2.04456	0.515663	0.625847

Figure1. Credit Card data CSV (1)

©IJRASET: All Rights are Reserved

M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE
V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
-0.6178	-0.99139	-0.31117	1.468177	-0.4704	0.207971	0.025791	0.403993	0.251412	-0.01831	0.277838	-0.11047	0.066928	0.128539	-0.18911	0.133558	-0.02105	149.62	0
1.065235	0.489095	-0.14377	0.635558	0.463917	-0.1148	-0.18336	-0.14578	-0.06908	-0.22578	-0.63867	0.101288	-0.33985	0.16717	0.125895	-0.00898	0.014724	2.69	0
0.066084	0.717293	-0.16595	2.345865	-2.89008	1.109969	-0.12136	-2.26186	0.52498	0.247998	0.771679	0.909412	-0.68928	-0.32764	-0.1391	-0.05535	-0.05975	378.66	0
0.178228	0.507757	-0.28792	-0.63142	-1.05965	-0.68409	1.965775	-1.23262	-0.20804	-0.1083	0.005274	-0.19032	-1.17558	0.647376	-0.22193	0.062723	0.061458	123.5	0
0.538196	1.345852	-1.11967	0.175121	-0.45145	-0.23703	-0.03819	0.803487	0.408542	-0.00943	0.798278	-0.13746	0.141267	-0.20601	0.502292	0.219422	0.215153	69.99	0
0.359894	-0.35809	-0.13713	0.517617	0.401726	-0.05813	0.068653	-0.03319	0.084968	-0.20825	-0.55982	-0.0264	-0.37143	-0.23279	0.105915	0.253844	0.08108	3.67	0
-0.15383	-0.75106	0.167372	0.050144	-0.44359	0.002821	-0.61199	-0.04558	-0.21963	-0.16772	-0.27071	-0.1541	-0.78006	0.750137	-0.25724	0.034507	0.005168	4.99	0
0.291474	1.757964	-1.32387	0.686133	-0.07613	-1.22213	-0.35822	0.324505	-0.15674	1.943465	-1.01545	0.057504	-0.64971	-0.41527	-0.05163	-1.20692	-1.08534	40.8	0
-0.11045	-0.28625	0.074355	-0.32878	-0.21008	-0.49977	0.118765	0.570328	0.052736	-0.07343	-0.26809	-0.20423	1.011592	0.373205	-0.38416	0.011747	0.142404	93.2	0
0.83639	1.006844	-0.44352	0.150219	0.739453	-0.54098	0.476677	0.451773	0.203711	-0.24691	-0.63375	-0.12079	-0.38505	-0.06973	0.094199	0.246219	0.083076	3.68	0
-0.67144	-0.51395	-0.09505	0.23093	0.031967	0.253415	0.854344	-0.22137	-0.38723	-0.0093	0.313894	0.02774	0.500512	0.251367	-0.12948	0.04285	0.016253	7.8	0
-0.32614	-0.09005	0.362832	0.928904	-0.12949	-0.80998	0.359985	0.707664	0.125992	0.049924	0.238422	0.00913	0.99671	-0.76731	-0.49221	0.042472	-0.05434	9.99	0
-0.24268	1.205417	-0.31763	0.725675	-0.81561	0.873936	-0.84779	-0.68319	-0.10276	-0.23181	-0.48329	0.084668	0.392831	0.161135	-0.35499	0.026416	0.042422	121.5	0
0.323387	-0.01108	-0.17849	-0.65556	-0.19993	0.124005	-0.9805	-0.98292	-0.1532	-0.03688	0.074412	-0.07141	0.104744	0.548265	0.104094	0.021491	0.021293	27.5	0
0.792944	0.370448	-0.73498	0.406796	-0.30306	-0.15587	0.778265	2.221868	-1.58212	1.151663	0.222182	1.020586	0.028317	-0.23275	-0.23556	-0.16478	-0.03015	58.8	0
-0.77041	1.047627	-1.0666	1.106953	1.660114	-0.27927	-0.41999	0.432535	0.263451	0.499625	1.35365	-0.25657	-0.06508	-0.03912	-0.08709	-0.181	0.129394	15.99	0
0.936708	0.70838	-0.46865	0.354574	-0.24663	-0.00921	-0.59591	-0.57568	-0.11391	-0.02461	0.196002	0.013802	0.103758	0.364298	-0.38226	0.092809	0.037051	12.99	0
0.277192	0.252624	-0.2919	-0.18452	1.143174	-0.92871	0.68047	0.025436	-0.04702	-0.1948	-0.67264	-0.15686	-0.88839	-0.34241	-0.04903	0.079692	0.131024	0.89	0
0.970117	-0.26657	-0.47913	-0.52661	0.472004	-0.72548	0.075081	-0.40687	-2.19685	-0.5036	0.98446	2.458589	0.042119	-0.48163	-0.62127	0.392053	0.949594	46.8	0
-0.63205	-0.41696	0.052011	-0.04298	-0.16643	0.304241	0.554432	0.05423	-0.38791	-0.17765	-0.17507	0.040002	0.295814	0.332931	-0.22038	0.022298	0.007602	5	0
1.298329	0.42048	-0.37265	-0.80798	-2.04456	0.515663	0.625847	-1.30041	-0.13833	-0.29558	-0.57196	-0.05088	-0.30421	0.072001	-0.42223	0.086553	0.063499	231.71	0

Figure2. Credit Card data CSV (2)

A. Data Exploration

First we imported the datasets that contain transactions made by credit cards. we then explored the data that is contained in the creditcard_data dataframe. After displaying the creditcard_data using the head() function as well as the tail() function, we proceeded to explore the other components of this dataframe.

B. Data Manipulation

In this section of the project, we scaled the data using the scale() function. We applied this to the amount component of our creditcard_data amount. With the help of scaling, the data is structured according to a specified range. Therefore, there are no extreme values in the dataset that might interfere with the functioning of the model.

C. Data Modelling

After standardizing the entire dataset, I split the dataset into training set as well as test set with a split ratio of 0.80. This means that 80% of the data will be attributed to the train_data whereas 20% will be attributed to the test_data. I then found the dimensions using the dim() function.

VII. FITTING LOGISTIC REGRESSION MODEL

In this section of the project, we fit the first model. we began with logistic regression. we used it for modeling the outcome probability of fraud/not fraud. we proceeded to implement this model on the test data. Once I summarized the model, we visualized it through plots. In order to assess the performance of the model, we portrayed the Receiver Optimistic Characteristics or ROC curve. For this, we first imported the ROC package and then plotted the ROC curve to analyze its performance.

Code:

```
# Fitting Logistic Regression Model
Logistic_Model=glm(Class~.,test_data,family=binomial())
summary(Logistic_Model)
# Visualizing summarized model through the following plots
plot(Logistic_Model)
# ROC Curve to assess the performance of the model
library(pROC)
lr.predict <- predict(Logistic_Model,test_data, probability = TRUE)
auc.gbm = roc(test_data$Class, lr.predict, plot = TRUE, col = "blue")
```

A. Fitting a Decision Tree Model

Next, we implemented a decision tree algorithm to plot the outcomes of a decision through which we could conclude as to what class the object belongs to. we then implemented the decision tree model and plotted it using the rpart.plot() function. we specifically used the recursive parting to plot the decision tree.

Code:

```
# Fitting a Decision Tree Model
library(rpart)
```

```
library(rpart.plot)
decisionTree_model <- rpart(Class ~ . , creditcard_data, method = 'class')
predicted_val <- predict(decisionTree_model, creditcard_data, type = 'class')
probability <- predict(decisionTree_model, creditcard_data, type = 'prob')
rpart.plot(decisionTree_model
```

B. Artificial Neural Network

Artificial Neural Networks are a type of machine learning algorithm that are modeled after the human nervous system. The ANN models are able to learn the patterns using the historical data and are able to perform classification on the input data. We imported the neuralnet package that allowed me to implement the ANNs. Then we proceeded to plot it using the plot() function. Now, in the case of Artificial Neural Networks, there is a range of values that is between 1 and 0. I set a threshold of 0.5, that is, values above 0.5 will correspond to 1 and the rest will be 0.

Code:

```
# Artificial Neural Network
library(neuralnet)
ANN_model =neuralnet (Class~.,train_data,linear.output=FALSE)
plot(ANN_model)
predANN=compute(ANN_model,test_data)
resultANN=predANN$net.result
resultANN=ifelse(resultANN>0.5,1,0)
```

C. Gradient Boosting (GBM)

Gradient Boosting is a popular machine learning algorithm that is used to perform classification and regression tasks. This model comprises of several underlying ensemble models like weak decision trees. These decision trees combine together to form a strong model of gradient boosting. We implemented gradient descent algorithm in the model.

Code:

```
# Gradient Boosting (GBM)
library(gbm, quietly=TRUE)
# Get the time to train the GBM model
system.time(
model_gbm <- gbm(Class ~ .
                 , distribution = "bernoulli"
                 , data = rbind(train_data, test_data)
                 , n.trees = 500
                 , interaction.depth = 3
                 , n.minobsinnode = 100
                 , shrinkage = 0.01
                 , bag.fraction = 0.5
                 , train.fraction = nrow(train_data) / (nrow(train_data) + nrow(test_data))
    )
  )
# Determine best iteration based on test data
gbm.iter = gbm.perf(model_gbm, method = "test")
model.influence = relative.influence(model_gbm, n.trees = gbm.iter, sort. = TRUE)
#Plot the gbm model
plot(model_gbm)
```

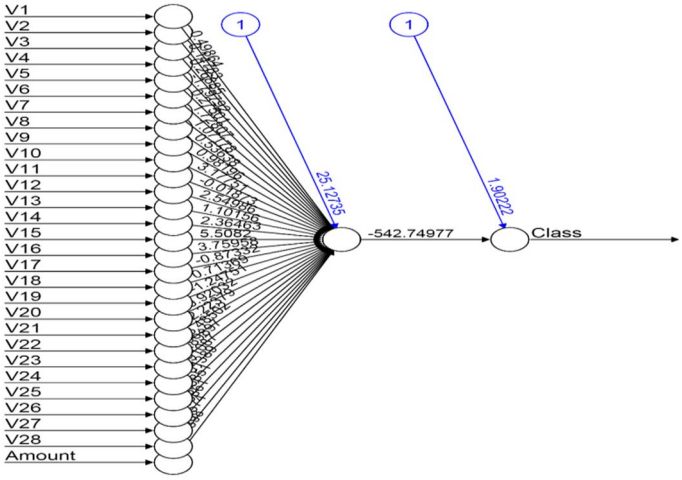
D. AUC-ROC Curve

In the last section of the project, we calculated and plotted an ROC curve measuring the sensitivity and specificity of the model. The print command plots the curve and calculates the area under the curve. The area of a ROC curve can be a test of the sensivity and accuracy of a model.

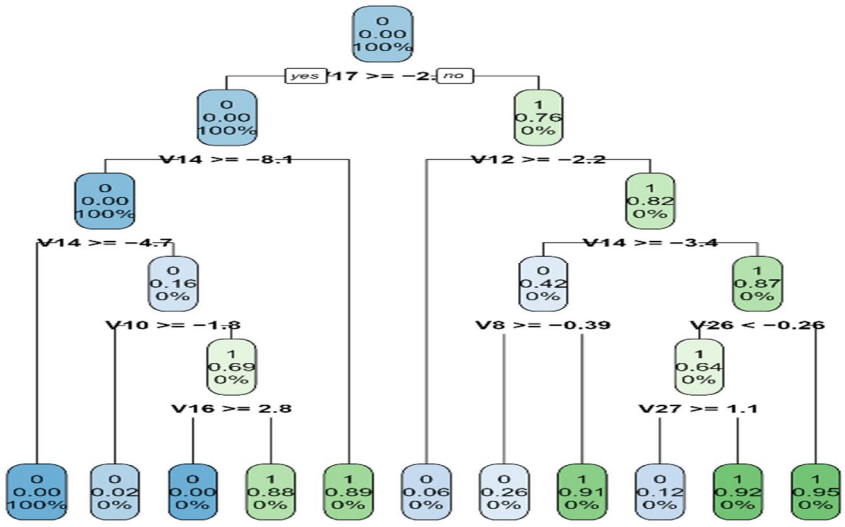
Code:

```
# Plot and calculate AUC on test data
library(pROC)
gbm_test = predict(model_gbm, newdata = test_data, n.trees = gbm.iter)
gbm_auc = roc(test_data$Class, gbm_test, plot = TRUE, col = "red")
print(gbm_auc)
```

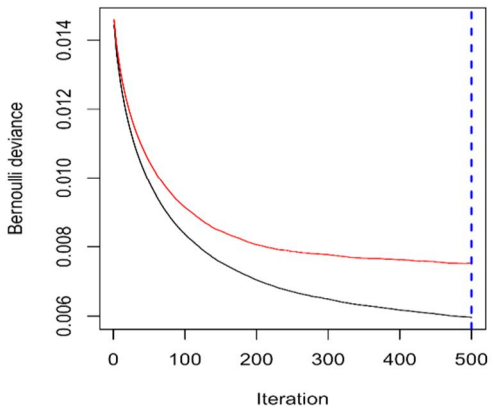
VIII. VISUALIZING THE RESULTS IMPLEMENTED USING VARIOUS SET OF ALGORITHMS



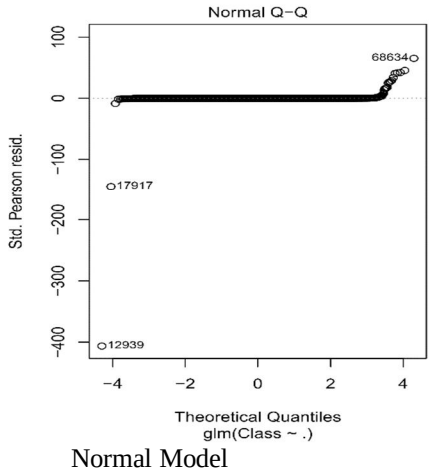
ANN Model



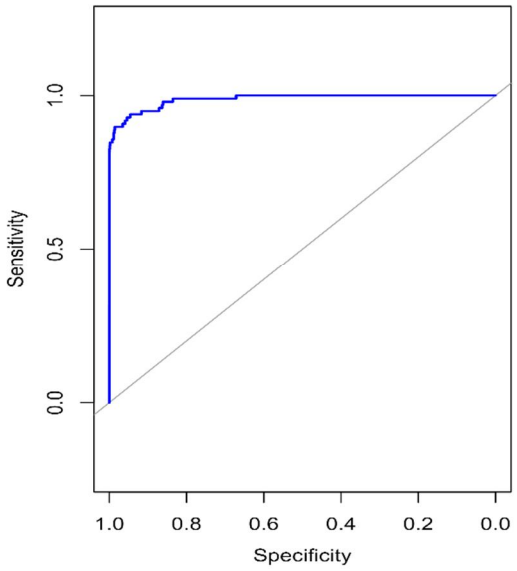
Decision Tree model



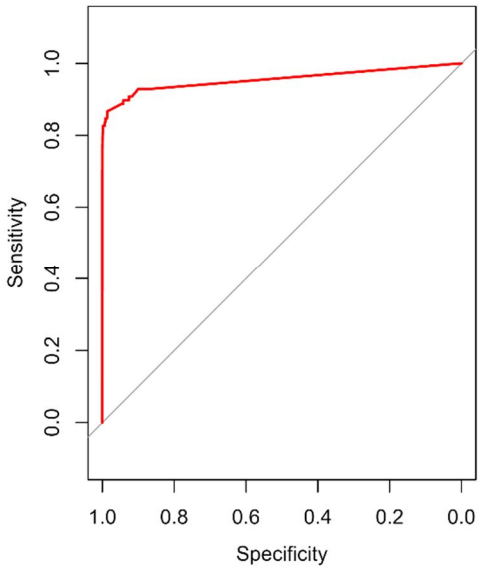
GBM Mode



Normal Model



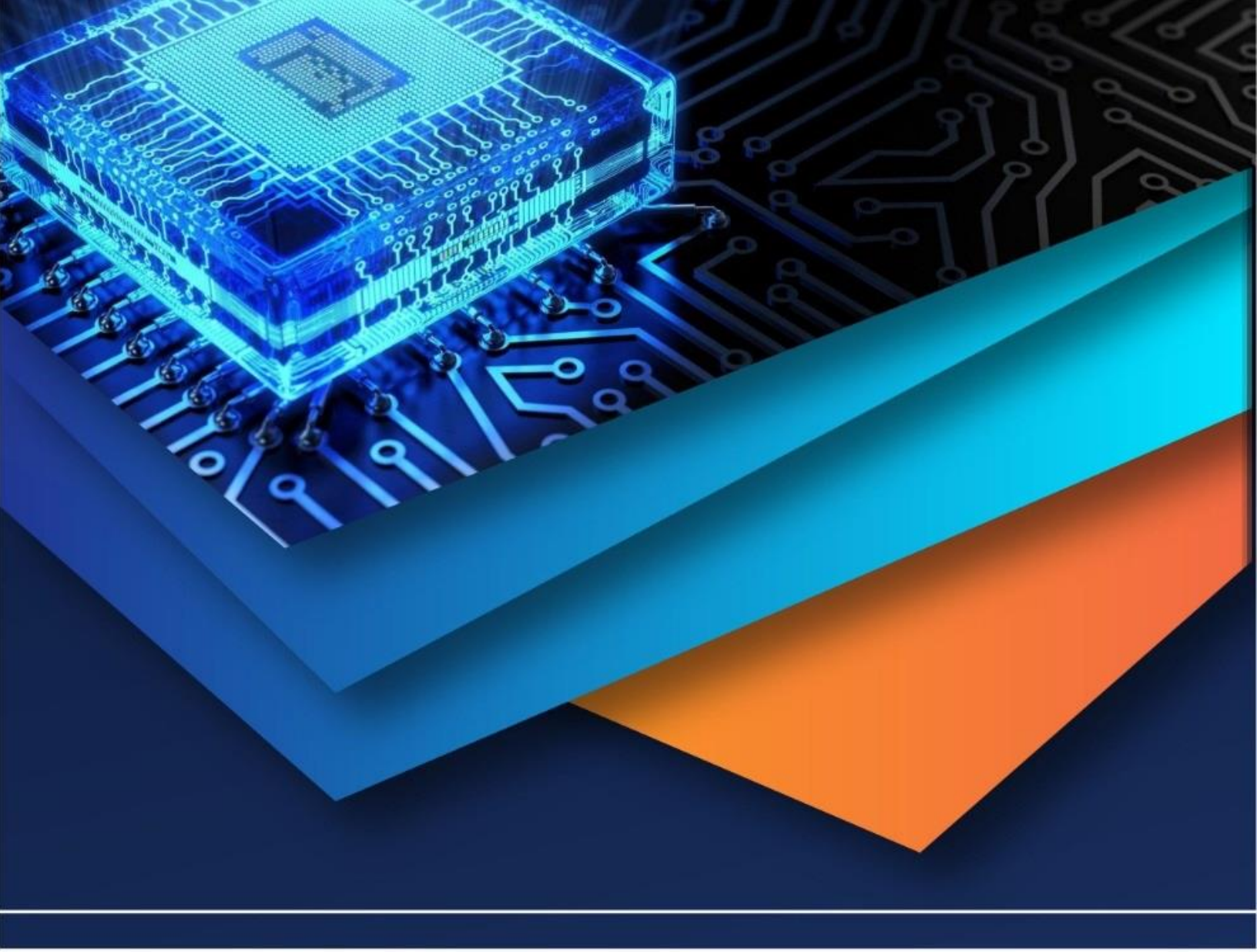
ROC Curve



AUC-ROCCurve

IX. CONCLUSION

Concluding our R Data Science project, we learnt how to develop a credit card fraud detection model using machine learning. We used a variety of ML algorithms to implement this model and also plotted the respective performance curves for the models. We also learnt how data can be analyzed and visualized to discern fraudulent transactions from other types of data. Hope you enjoyed this credit card fraud detection project of machine learning using R.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)