

CREDIT CARD FRAUD CLASSIFICATION



**NITTE**  
EDUCATION TRUST

**N.M.A.M. INSTITUTE OF TECHNOLOGY**

(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)

Nitte – 574 110, Karnataka, India

**Department of Computer Science and Engineering**

(B.E. Computer Science & Engineering Program Accredited by NBA from 2018-19 to 2021-25)

Report on Mini Project

# CREDIT CARD FRAUD CLASSIFICATION

Course Code : 21CSA31

Course Name : R Programming

Semester: III SEM

Section: A

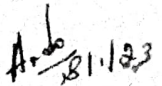
**Submitted To:**

Dr. Anisha P Rodrigues  
Associate Professor  
Department of Computer Science  
and Engineering

**Submitted By:**

A Prashasthi Shetty-4NM21CS001  
Akshata D Bhat-4NM21CS011  
Akshay Prabhu K-4NM21CS013

**Date of submission:**  
18 January 2023

  
Signature of Course Instructor

## ABSTRACT

Our project's primary goal is to use R programming to research and examine the dataset for Credit Card Fraud Classification. The dataset contains transactions made by credit cards in September 2013 by European card holders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284807 transactions. This dataset is highly unbalanced, the positive class (Frauds) account for 0.172% of all transactions. We have used a variety of graphs, such as correlation heatmap and pie charts, for better visualization of the dataset. We used machine learning techniques such as decision trees, logistic regression, artificial neural nets, Receiver Optimistic Characteristics(ROC) curve to illustrate the correlations between different dataset pieces, classify the data and predict which technique can be used for implementation of detection model. We have also implemented a webpage using Shiny R to show some of our graphs. In the end, we are able to examine and comprehend the dataset much more easily utilizing several R concepts, and some algorithms of machine learning.

## TABLE OF CONTENTS

---

<b>Title Page .....</b>	<b>i</b>
<b>Abstract.....</b>	<b>ii</b>
<b>Table of Contents .....</b>	<b>iii</b>
<b>Introduction .....</b>	<b>4</b>
<b>Problem Statement.....</b>	<b>5</b>
<b>Objectives .....</b>	<b>6</b>
<b>Methodology .....</b>	<b>7</b>
<b>Implementation Details .....</b>	<b>10</b>
<b>Results .....</b>	<b>15</b>
<b>Conclusion and Future Scope.....</b>	<b>21</b>
<b>References .....</b>	<b>22</b>
<b>Certificate .....</b>	<b>23</b>

## INTRODUCTION

Credit card fraud is the unauthorized use of a credit card or credit card information to make purchases or withdrawals. This can include using a stolen credit card, using a credit card number obtained through phishing or other forms of identity theft, or making unauthorized charges on a compromised account. Credit card fraud can occur both online and in person, and can result in significant financial losses for both cardholders and financial institutions. To prevent and detect credit card fraud, financial institutions and merchants use a combination of security measures such as chip-enabled cards, fraud detection software, and monitoring of suspicious activities.

Credit card fraud classification is the process of identifying and categorizing fraudulent activity on a credit card account. This is typically done using machine learning algorithms that are trained on historical transaction data to identify patterns or anomalies that are indicative of fraud. The goal of credit card fraud classification is to accurately identify fraudulent transactions while minimizing the number of false positives (legitimate transactions that are incorrectly flagged as fraud).

The dataset here contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions. It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we do not have access to the original features and more background information about the data. Features V1, V2, ... V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependent cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

## **PROBLEM STATEMENT**

The problem statement of this project is “CREDIT CARD FRAUD CLASSIFICATION” to classify and predict legitimate and fraudulent transactions using R Programming and Machine Learning Algorithms.

## OBJECTIVES

- The aim of this project is to build a classifier that can detect credit card fraudulent transactions using R Programming.
- To view, explore, manipulate and visualize the dataset using various features of R for proper prediction of faulty transactions.
- To try and implement our own formulae and algorithms for better functioning of the code.
- To use machine learning algorithms along with R to help classify the data and predict fraudulent transactions and faulty cases.
- To implement a webpage using Shiny R package to show our some of our visualizations.
- To establish a detection model which checks whether class is fraudulent or legitimate.

## METHODOLOGY

The step-by-step implementation process is as follows.

### Step 1: IMPORTING LIBRARIES AND DATASETS

We have read our dataset and imported libraries in this step. The Libraries used in our project are as follows:

- ranger: This library is fast implementation of random forests or recursive partitioning, particularly suited for high dimensional data.
- caret: It's used for classifying and regression training.
- data.table: It's an extension of data.frame in R. It is widely used for fast aggregation of large datasets.
- dplyr: This package provides many tools for the manipulation of data in R.
- corrplot: It provides a visual exploratory tool on correlation matrix.
- caTools: Used for the implementation of set.seed and sample.split for training and testing data.

Amount and class are allocated to each of the 284807 transactions in the dataset. Principal component analysis (PCA) for dimensionality reduction was used to produce this. It is employed to protect users' privacy. The full dataset was then printed using the options(max.print) command.

### Step 2: DATA EXPLORATION

- view: We have used this function to print the dataset in R script.
- dim: we have used this function to find the number of rows and columns in the dataset.
- head: This prints the first 6 rows of the dataset.
- tail: This prints the last 6 rows of the dataset.
- table: Table function shows the frequency of particular values in dataset.
- summary: This function can be used to summarize the values in a vector, dataframe, regression models in R.
- str: This function in R is used for compactly displaying the internal structure of an R object.

### Step 3: DATA VISUALIZATION

We have created various graphs including pie charts, boxplots, histograms, and correlation matrices. We also used our own formulae for the size of the graphs. We have used pie chart for finding percentages of fraud and legitimate and correlation matrix for checking the relationship between different columns.

#### **Step 4: IMPLEMENTATION OF SHINY WEBPAGE IN R**

Shiny is an R package that enables building interactive web applications that can execute R code on the backend. Shiny App requires two inputs: the server for the back end and the user interface for the front end. A side panel with check boxes to choose the necessary graphs is created using the UI parameter. The graphs are pushed into the webpage by the server parameter. We have used the following graphs for fraudulent and legitimate values.

- 1)Boxplot: To show distribution of amount over class
- 2)Histogram: To show distribution of time over class

#### **Step 5: DATA MANIPULATION**

Our dataset already included the PCA technique to reduce its dimensionality. In order to ensure that the machine learning algorithms for classification function properly, we have scaled the data, used `set.seed()` to generate a random number, and based on that random number, distributed the data between training data and testing data using `sample.split()`. To prepare the data and then create a fraud detection model, we have employed a variety of classification models and techniques, which are logistic regression, decision trees, artificial neural networks, and gradient boosting. As the dataset was already labelled, we want to build an detector for unlabeled data and hence these algorithms provide us great platforms for achieving our goal.

#### **Step 6: PREPARATION OF DATA AND BUILDING A PREDICTOR MODEL USING MACHINE LEARNING ALGORITHMS**

Data preparation and classification is achieved as follows:-

- Logistic Regression Model: The data is classified as fraud and legitimate transactions using logistic regression model. This model provides us with some results based on internal workings with the use of graphs. Based on its outcomes, we see that it is the best model for classifying fraudulent and legitimate transactions.
- Decision Trees: For improved classification and predictions of the given class, decision trees are utilized. Recursive plotting was employed in the decision tree, and the tree produced 10 significant classes that aid in determining which class the unlabeled data belongs to.
- Artificial neural network: The ANN models are able to learn the patterns using the historical data and are able to perform classification on the input data. It is bit more efficient than decision trees as for entered data, we can get a proper result.
- Gradient Boosting Model: Gradient Boosting method is the most efficient classification machine learning model for predictions of the data whether it is legitimate or fraudulent. It takes a mixture of weak decision trees and strong decision trees and gives a stronger gradient model, also helping in very accurate predictions for our predictor model apart from logistic regression.



We have implemented Receiver Optimistic Characteristics(ROC) curve and Area Under Curve(AUC) graphs for each of the classification model except Artificial Neural nets and concluded the outcomes of each graph to realize which models are best for creating a accurate detection model.

Fig A) : A part of the dataset we have used.

Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14
0	-1.35980713	-0.07278117	2.53634674	1.37815522	-0.338320770	0.462387778	0.239598554	0.098697901	0.36378697	0.090794172	-0.551599533	-0.617800856	-0.991389847	-0.311169354
0	1.19185711	0.26615071	0.16648011	0.44815408	0.060017649	-0.082360809	-0.078802983	0.085101655	-0.25542513	-0.166974414	1.612726661	1.065235311	0.489095016	-0.143772296
1	-1.35835406	-1.34016307	1.77320934	0.37977959	-0.503198133	1.800499381	0.791460956	0.247675787	-1.51465432	0.207642865	0.624501459	0.066083685	0.717292731	-0.165945923
1	-0.96627171	-0.18522601	1.79299334	-0.86329128	-0.010308880	1.247203168	0.237608940	0.377435875	-1.38702406	-0.054951922	-0.226487264	0.178228226	0.507756870	-0.287923745
2	-1.15823309	0.87773676	1.54871785	0.40303393	-0.407193377	0.095921462	0.592940745	-0.270532677	0.81773931	0.753074432	-0.822842878	0.538195550	1.345851593	-1.119669835
2	-0.42596588	0.96052304	1.14110934	-0.16825208	0.420986881	-0.029272552	0.476200949	0.260314333	-0.56867138	-0.371407197	1.341261980	0.359893837	-0.358090653	-0.137133700
4	1.22965763	0.14100351	0.04537077	1.20261274	0.191880989	0.272708123	-0.005159003	0.081212940	0.46496000	-0.099254321	-1.416907243	-0.153825826	-0.751062716	0.167371963
7	-0.64426944	1.41796355	1.07438038	-0.49219902	0.948934095	0.428118463	1.120631358	-3.807864239	0.61537473	1.249376178	-0.619467796	0.291474353	1.757964214	-1.323865220
7	-0.89428608	0.28615720	-0.11319221	-0.27152613	2.669598660	3.721818061	0.370145128	0.851084443	-0.39204759	-0.410430433	-0.705116587	-0.110452262	-0.286253632	0.074355360
9	-0.33826175	1.11959338	1.04436655	-0.22218728	0.499360806	-0.246761101	0.651583206	0.069538587	-0.73672732	-0.366845639	1.017614468	0.836389570	1.006843514	-0.443522817
10	1.44904378	-1.17633882	0.91385983	-1.37566666	-1.971383165	-0.629152139	-1.423235601	0.048455888	-1.72040839	1.626659058	1.199643950	-0.671439778	-0.513947153	-0.095045045
10	0.38497822	0.61610946	-0.87429970	-0.09401863	2.924584378	3.317027168	0.470454672	0.538247228	-0.55889461	0.309755394	-0.259115564	-0.326143234	-0.090046723	0.362832369
10	1.24999874	-1.22163681	0.38393015	-1.23489869	-1.485419474	-0.753230165	-0.689404975	-0.227487228	-2.09401057	1.323729274	0.227666231	-0.242681999	1.205416808	-0.317630527
11	1.06937359	0.28772213	0.82861273	2.71252043	-0.178398016	0.337543730	-0.096716862	0.115981736	-0.22108257	0.460230444	-0.773656931	0.323387245	-0.011075887	-0.178485175
12	-2.79185477	-0.32777076	1.64175016	1.76747274	-0.136588446	0.807596468	-0.422911390	-1.907107476	0.75571291	1.151086988	0.844555471	0.792943952	0.370448093	-0.734975106
12	-0.75241704	0.34548542	2.05732291	-1.46864330	-1.158393680	-0.077849829	-0.608581418	0.003603484	-0.43616698	0.747730827	-0.793980603	-0.770406729	1.047626997	-1.066603681
12	1.10321544	-0.04029622	1.26733209	1.28909147	-0.735997164	0.288069163	-0.586056786	0.189379714	0.78233289	-0.267975067	-0.450311280	0.936707715	0.708380406	-0.468647288
13	-0.43690507	0.91896621	0.92459077	-0.72721905	0.915678718	-0.127867352	0.707641607	0.087962355	-0.66527135	-0.737979824	0.324097813	0.277192107	0.252624256	-0.291896460
14	-5.40125766	-5.45014783	1.18630463	1.73623880	3.049105878	-1.763405574	-1.559737699	0.160841747	1.23308974	0.345172827	0.917229868	0.970116716	-0.266567765	-0.479129929

Fig B) A function we used to show size and themes of graphs implemented in Shiny R Webpage

```
fig <- function(width, height)
{
  options(repr.plot.width = width, repr.plot.height = height)
}
fig(14, 8)
common_theme <- theme(plot.title = element_text(hjust = 0.5, face = "bold"))
```

## IMPLEMENTATION

```
# Importing Datasets
library(ranger)
library(caret)
library(data.table)
library(dplyr)
library(corrplot)
library(caTools)
creditcard_data <- read.csv("C:/users/akshay/Desktop/MACHINE LEARNING USING
R/creditcard.csv")
options(max.print=999999)

# Data Exploration
View(creditcard_data)
dim(creditcard_data)
head(creditcard_data)
tail(creditcard_data,6)
table(creditcard_data$Time)
summary(creditcard_data$Amount)
str(creditcard_data)

#Data Visualization
table(creditcard_data$Class)
prop.table(table(creditcard_data$Class))
labels <- c("Non-fraud","fraud")
labels <- paste(labels, round(100*prop.table(table(creditcard_data$Class)), 2))
labels <- paste0(labels,"%")
pie(table(creditcard_data$Class), labels, col = c("yellow","black"),
    main = "Pie chart of fraud and legit credit card transactions")

fig <- function(width, height)
{
  options(repr.plot.width = width, repr.plot.height = height)
}
fig(14, 8)
common_theme <- theme(plot.title = element_text(hjust = 0.5, face = "bold"))
ggplot(creditcard_data, aes(x = factor(Class), y = Amount)) + geom_boxplot() +
  labs(x = 'Class', y = 'Amount') +
  ggtitle("Distribution of transaction amount by class") + common_theme
```

```

fig(14, 8)
creditcard_data %>%
  ggplot(aes(x = Time, fill = factor(Class))) + geom_histogram(bins = 100)+
  labs(x = 'Time in seconds since first transaction', y = 'No. of transactions') +
  ggtitle('Distribution of time of transaction by class') +
  facet_grid(Class ~ ., scales = 'free_y') + common_theme
fig(14, 8)
correlations <- cor(creditcard_data[, -c[1]], method = "pearson")
corrplot(correlations, number.cex = .9, method = "circle", type = "full", tl.cex = 0.8, tl.col =
"black")

```

### #App Implementation

```

library(shiny)
shinyApp(ui = ui, server = server)
ui <- fluidPage(
  titlePanel("Credit Card Fraud Detection"),
  sidebarLayout(
    sidebarPanel(
      checkboxInput("show_fraud", "Show Fraud", TRUE),
      checkboxInput("show_legit", "Show Legit", TRUE),
    ),
    mainPanel(
      plotOutput("boxplot"),
      plotOutput("histogram")
    )
  )
)
library(shiny)
server <- function(input, output) {
  filtered_data <- reactive({
    if (input$show_fraud && input$show_legit) {
      return(creditcard_data)
    } else if (input$show_fraud) {
      return(creditcard_data[creditcard_data$Class == 1,])
    } else if (input$show_legit) {
      return(creditcard_data[creditcard_data$Class == 0,])
    } else {
      return(NULL)
    }
  })
}

```

```

output$boxplot <- renderPlot({
  if (is.null(filtered_data())) return(NULL)
  ggplot(filtered_data(), aes(x = factor(Class), y = Amount)) +
    geom_boxplot() +
    labs(x = 'Class', y = 'Amount') +
    ggtitle("Distribution of transaction amount by class")
})
output$histogram <- renderPlot({
  if (is.null(filtered_data())) return(NULL)
  ggplot(filtered_data(), aes(x = Time, fill = factor(Class))) +
    geom_histogram(bins = 100) +
    labs(x = 'Time in seconds since first transaction', y = 'No. of transactions') +
    ggtitle('Distribution of time of transaction by class') +
    facet_grid(Class ~ ., scales = 'free_y') + common_theme
})
}

```

#### # Data Manipulation

```

creditcard_data$Amount=scale(creditcard_data$Amount)
NewData=creditcard_data[,-c(1)]
head(NewData)
set.seed(123)
data_sample = sample.split(NewData$Class,SplitRatio=0.80)
train_data = subset(NewData,data_sample==TRUE)
test_data = subset(NewData,data_sample==FALSE)
View(train_data)
dim(test_data)

```

#### # Fitting Logistic Regression Model

```

Logistic_Model=glm(Class~.,test_data,family=binomial())
summary(Logistic_Model)
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")

```

#### # 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)
library(pROC)
roc_obj <- roc(creditcard_data$Class, probability[,2])
plot(roc_obj)
auc(roc_obj)

# 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)

# 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))
  )
)

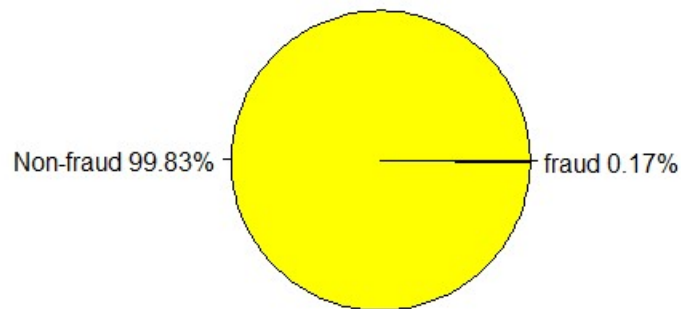
# Determine best iteration based on test data
gbm.iter = gbm.perf(model_gbm, method = "test")
model.influence = relative.influence(model_gbm, n.trees = gbm.iter, sort. = TRUE)
#Plot the gbm model
plot(model_gbm)

```

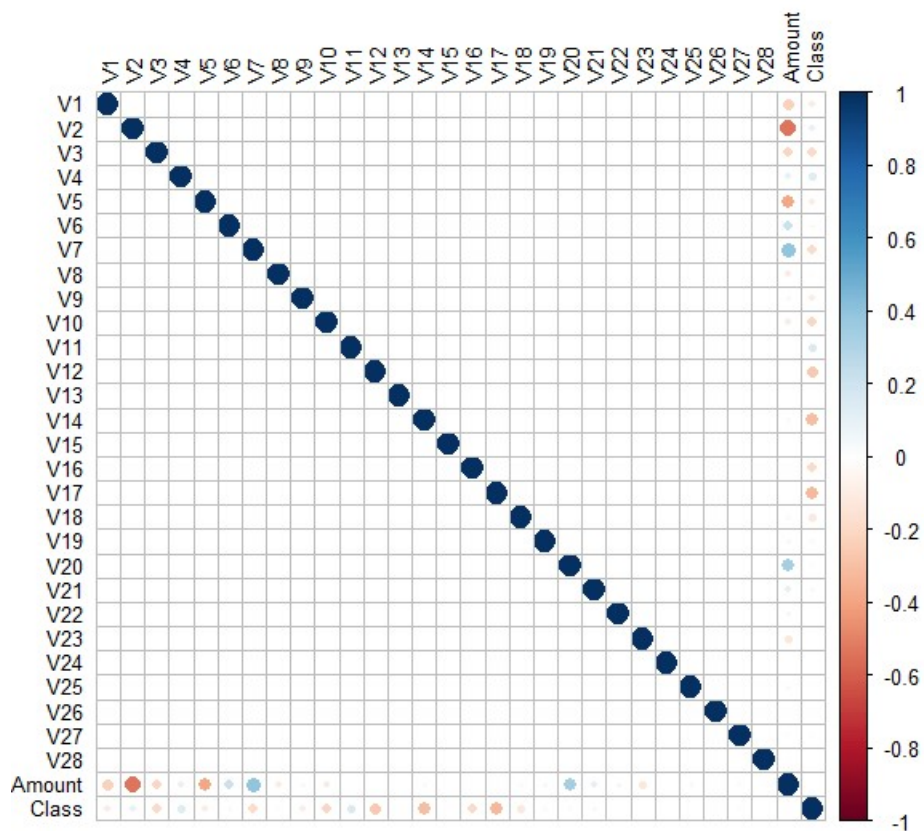
```
# Plot and calculate AUC on test data
library(pROC)
gbm_test = predict(model_gbm, newdata = test_data, n.trees = gbm.iter)
gbm_auc = roc(test_data$Class, gbm_test, plot = TRUE, col = "red")
print(gbm_auc)
```

## RESULTS AND DISCUSSIONS

Pie chart of fraud and legit credit card transactions

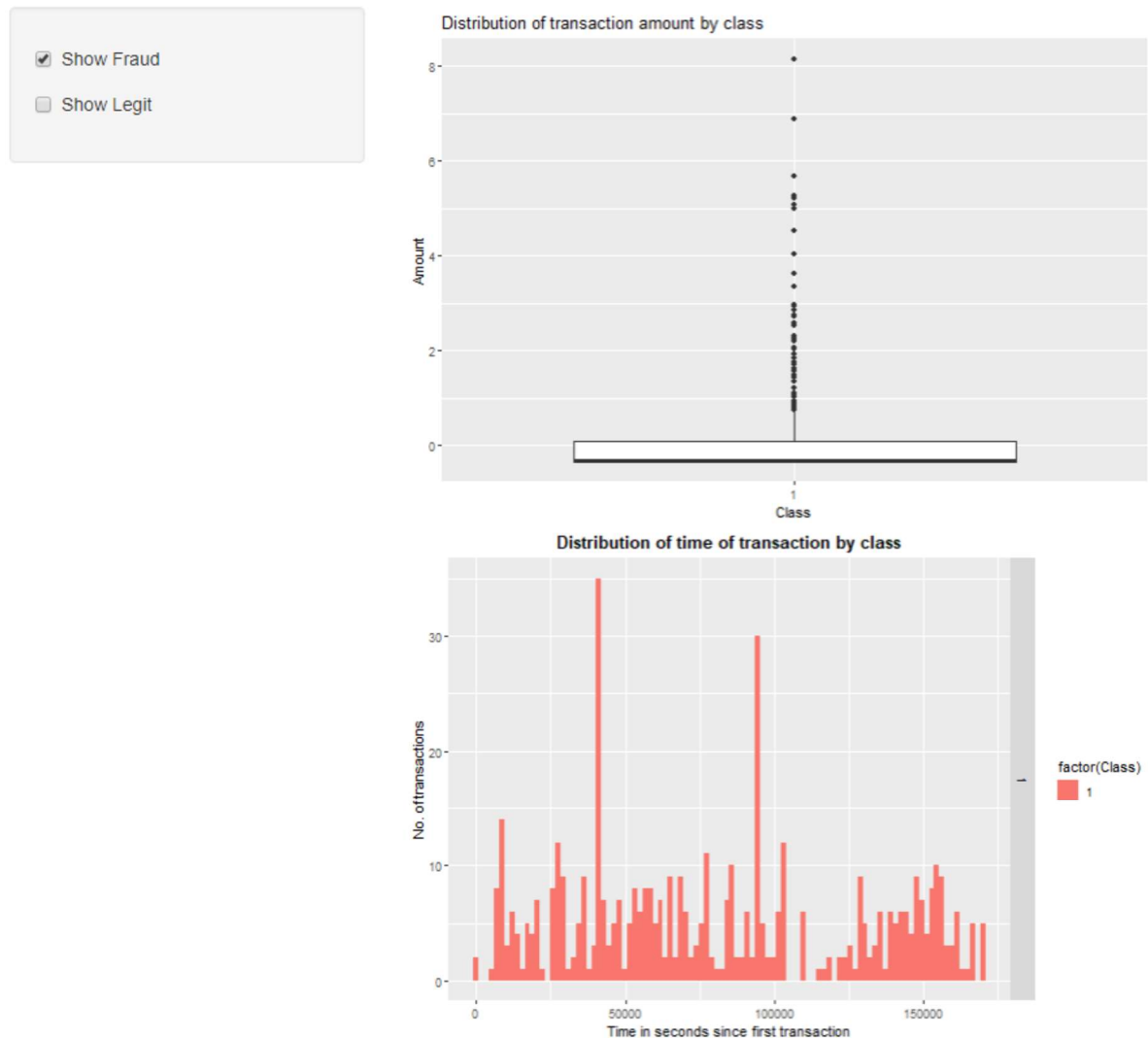


**Fig 1:** This is an pie chart representing fraud and non-fraud transactions where fraud accounts to only 0.17% . This is done for checking the distribution of classes.



**Fig 2:** This is an correlation matrix showing the relationships between the different columns of the dataset and presents which class has better hand in contributing to the classification.

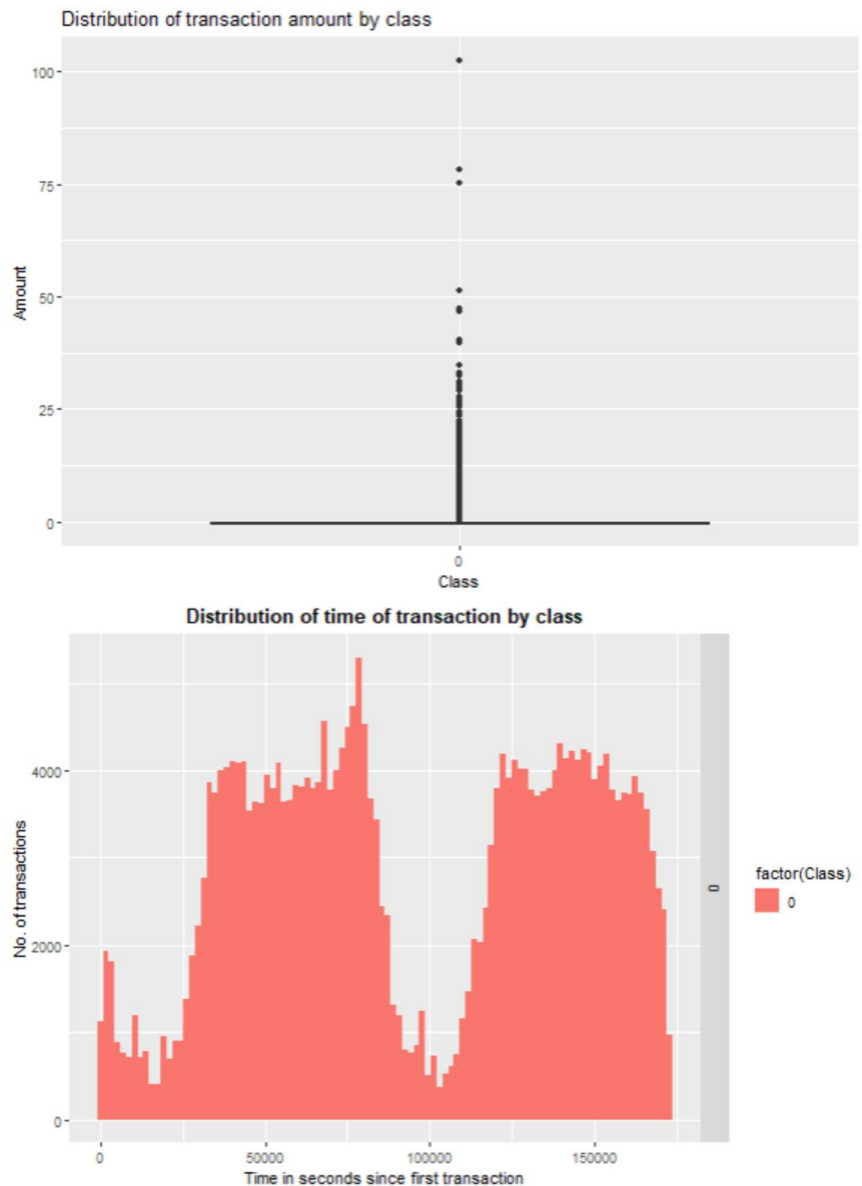
### Credit Card Fraud Detection



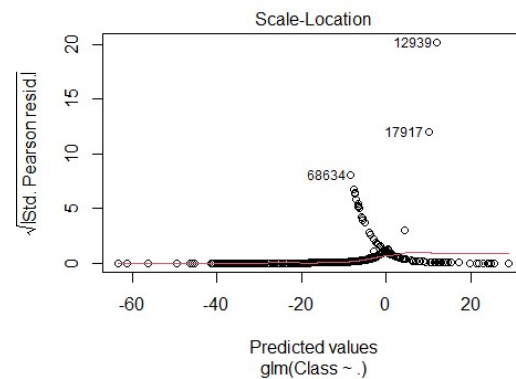
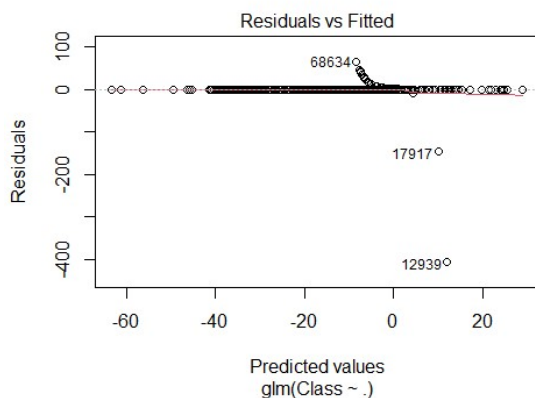
**Fig 3:** This is the boxplot showing distribution of transaction amount by class and histogram for distribution of time of transaction by class for fraudulent values in the Shiny Webpage we have implemented.

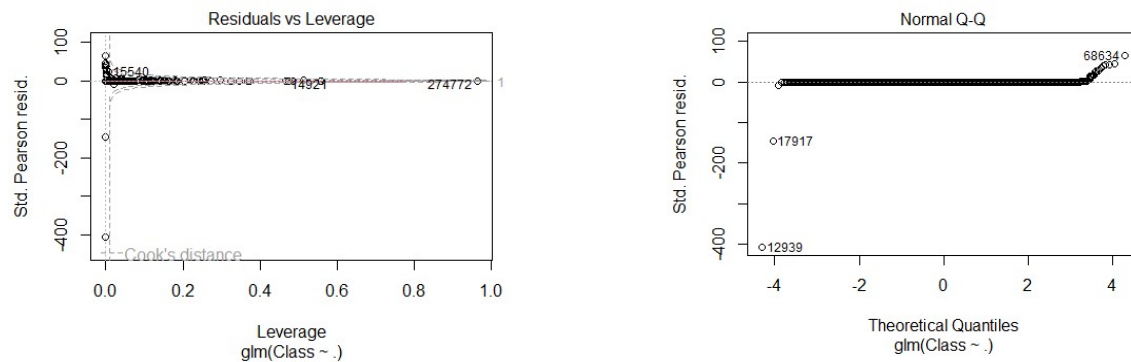


## Credit Card Fraud Detection

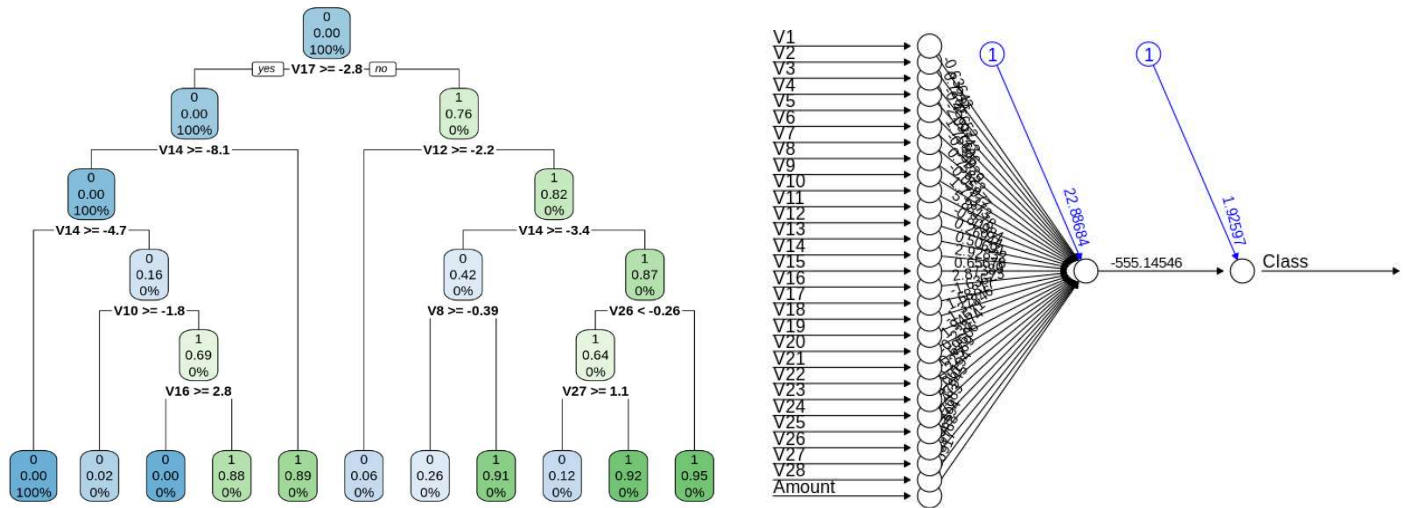
☐ Show Fraud☒ Show Legit

**Fig 4:** In this figure, we have shown distributions for legitimate values in the Shiny Webpage.

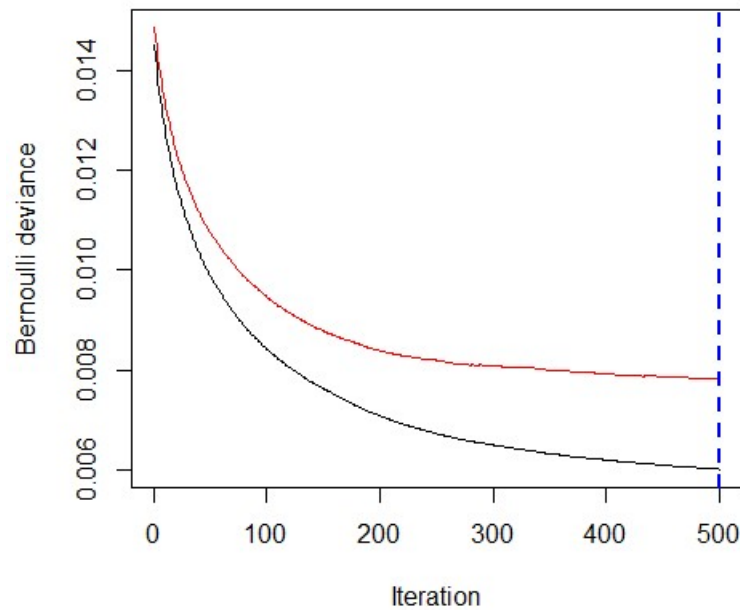




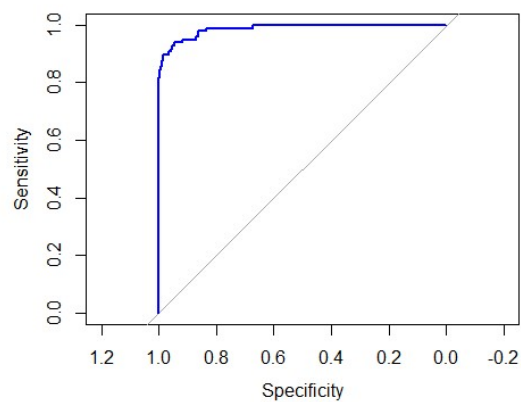
**Fig 5:** These are the graphs we have obtained through the logistic regression algorithm. They explain the relationship between different outputs obtained through logistic regression. A logistic regression is used for modeling the outcome probability of a class such as pass/fail, positive/negative and in our case – fraud/not fraud.



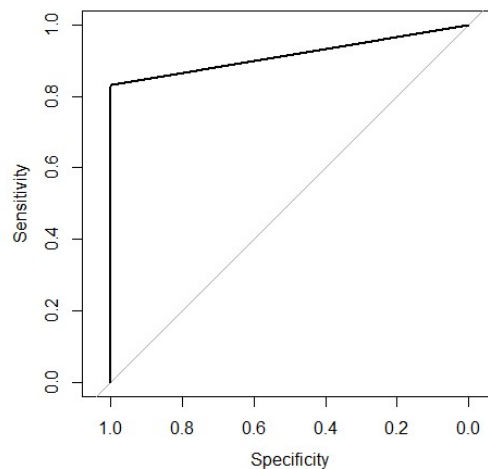
**Fig 6:** We have implemented decision trees algorithm to plot the outcomes of our model. Based on the outcomes presented by the model, we can conclude to which class the object belongs to. Here, the 10 most important classes which decide the class of an object are shown. Then we have implemented a artificial neural net. The neural net is assigned by some values by a inner algorithm and based on the testing data entered, we conclude to which class it belongs to.



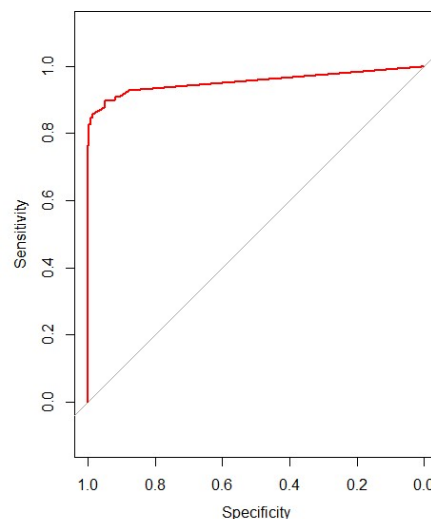
**Fig 7:** Based on the gradient boosting algorithm, a deviance model is built for giving proper prediction of the data given.



**Fig 8:** This Receiver Optimistic Characteristics graph was obtained through the outcomes of logistic regression and based on this graph; we are able to conclude that the values closer to the top-left of the sensitivity-specificity are some of the values which will help us understand how they may work for prediction model.



**Fig 9:** This ROC curve was plotted using the outcomes of Decision Tree Model and we have found out that the AUC of this curve is 0.9165 indicating that its accuracy is very less compared to the accuracy we want.



```
Call:
roc.default(response = test_data$Class, predictor = gbm_test,      plot = TRUE, col = "red")

Data: gbm_test in 56863 controls (test_data$Class 0) < 98 cases (test_data$Class 1).
Area under the curve: 0.9551
```

**Fig 10:** This graph was obtained using Gradient Boosting Algorithm and this graph is an area under curve graph. Based on its output, we were able to conclude that this is the best model for creating a nearly accurate detection model with the area under curve of 0.9551 which indicates it has the best accuracy compared to other comparison models.

## CONCLUSION AND FUTURE SCOPE

Through this project, we have learnt the features of R programming and some algorithms of Machine learning to implement different algorithms and formulae in R to classify and detect fraudulent transactions in the given dataset. We have established a model using machine learning algorithms to properly detect a given transaction and showcase it as fraud or non-fraud. Based on the number of legitimate values given in the dataset we have consider, we were expecting an accuracy of nearly 99.83%.

Through the curves we have plotted for each algorithm, we are able to conclude that Gradient Boosting Algorithm is the best method for predicting accurate results for our model. We have obtained an value of 0.9551 for GBM model which is nearly accurate, based on our analysis.

The value of 0.9551 or 95.51% accuracy instead of 99.83% may have an indication that some of the legitimate values displayed in the dataset are false positive values.

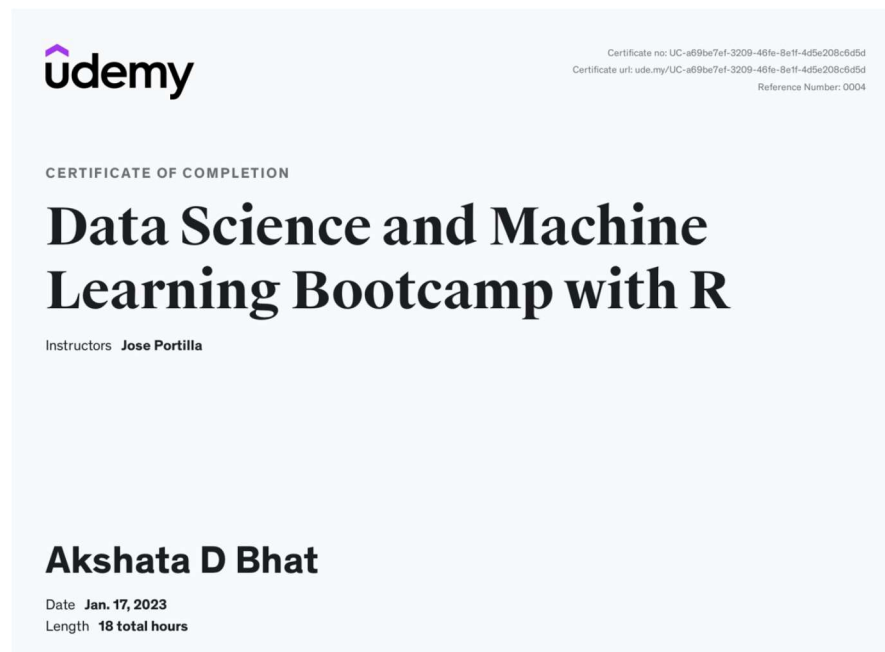
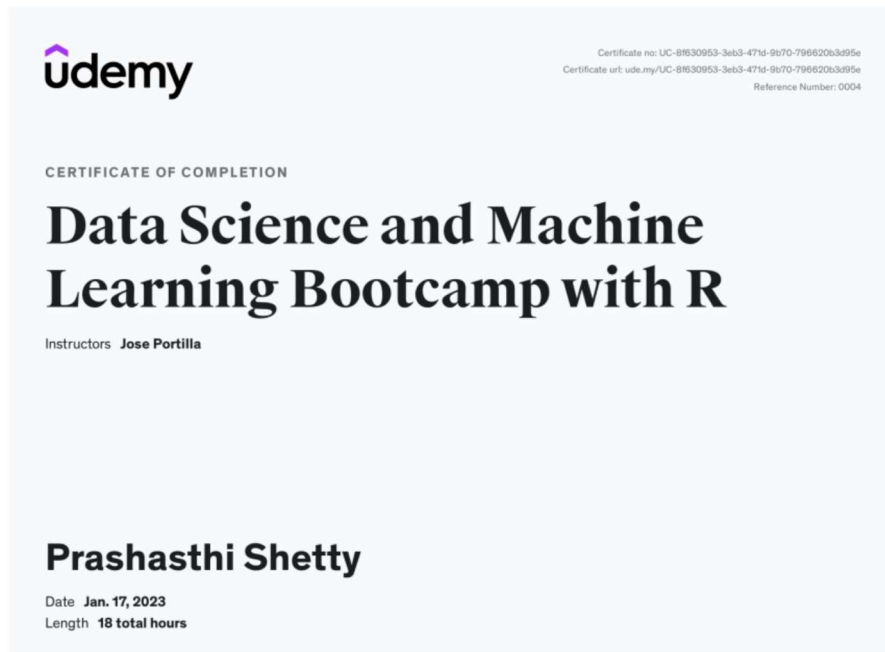
Improvement in prediction or accuracy of the algorithm may be achieved by designing an algorithm which will give lower false positives.

In future, with better advancements in technology, we will try to implement an algorithm or code to create a detection model which will give us nearly 100% accuracy.

## REFERENCES

- [1] <https://github.com/Rpita623/Detecting-Credit-Card-Fraud>
- [2] <https://www.kaggle.com/code/mendozav/credit-card-fraud-detection-project/data>
- [3] <https://www.kaggle.com/code/atharvaingle/credit-card-fraud-detection-with-r-sampling>
- [4] <https://data-flair.training/blogs/data-science-machine-learning-project-credit-card-fraud-detection/>
- [5] [https://github.com/Hima29nshi/Predicting Credit Card Using R/blob/main/Final Project.r](https://github.com/Hima29nshi/Predicting_Credit_Card_Using_R/blob/main/Final_Project.r)
- [6] <https://www.udemy.com/course/data-science-and-machine-learning-bootcamp-with-r/>

## CERTIFICATE





Certificate no: UC-43301483-a2bb-4fc8-8112-5f5e4007e35e  
Certificate url: ude.my/UC-43301483-a2bb-4fc8-8112-5f5e4007e35e  
Reference Number: 0004

CERTIFICATE OF COMPLETION

# Data Science and Machine Learning Bootcamp with R

Instructors **Jose Portilla**

**Akshay Prabhu K**

Date **Jan. 17, 2023**

Length **18 total hours**