[**Business Analytics with R**](https://elearning.utdallas.edu/webapps/blackboard/execute/courseMain?course_id=_342104_1)

### [**Project final report**](https://elearning.utdallas.edu/webapps/assignment/uploadAssignment?content_id=_7844870_1&course_id=_342104_1&group_id=&mode=view)

**Credit Card Fraud Detection**

**Contents**

1. Executive Summary
2. Project motivation
3. Data description
4. Exploratory Data Analysis
5. Data Mining
6. BI model
7. Interpretations
8. Conclusion
9. References

**Executive Summary**

This project represents a comprehensive exploration into the realm of credit card fraud detection, leveraging a diverse array of machine learning algorithms. In response to the staggering annual financial losses incurred through fraudulent credit card activities, this endeavor aims to mitigate such losses through the development and implementation of robust fraud detection models. The utilization of Logistic Regression, Decision Tree, Gradient Boosting, and Artificial Neural Network models on the dataset—initially partitioned into training and test sets via the caTools library—unveiled distinct insights and challenges. These algorithms, while promising, confront significant hurdles stemming from the dynamic nature of data distribution, prevalent class imbalances, and the continuous influx of transactional data. Addressing these challenges becomes imperative in enhancing fraud detection mechanisms, particularly considering the scarcity of publicly available data due to confidentiality concerns. The project aligns with the critical need for credit card companies to discern fraudulent transactions, ensuring customers are safeguarded from unwarranted charges, ultimately bolstering trust and reliability within financial systems.

**Project Motivation**

This project is driven by the imperative to combat the escalating threat of credit card fraud through the application of advanced data analysis techniques. With a surge in credit card fraud incidents, the need for robust security measures is paramount in real-world financial scenarios. Meticulous data preprocessing, which addresses issues like missing values and duplicates, forms the bedrock for in-depth analysis. By employing powerful machine learning algorithms such as Logistic Regression, Decision Tree, Gradient Boosting, and Artificial Neural Networks, this initiative seeks to construct predictive models tailored for precise fraud detection. In a world where the security and integrity of financial transactions are constantly under siege, the successful implementation of these models not only fortifies transactional security but also significantly contributes to elevating the overall customer experience. This project addresses a critical real-world need by enhancing security measures and instilling trust in financial transactions amid the escalating challenge of credit card fraud.

**Data Description**

The dataset utilized in this project encompasses credit card transactions spanning a two-year period, from January 1, 2019, to December 31, 2020. Covering interactions between 1000 customers and 800 diverse merchants, it mirrors the intricacies of real-world financial dealings. Customer data, likely anonymized for privacy, offers insights into demographic details, while transaction records encompass vital information like amounts, timestamps, and merchant identifiers. Crucially, this dataset incorporates both legitimate and fraudulent transactions, enabling the development of machine learning models capable of identifying potential fraud patterns. Featuring a broad array of merchants, it reflects the varied nature of consumer transactions. The dataset attributes include unique identifiers, transaction details, customer demographics, geographic information, and a crucial fraud flag denoting the target class for model development. This rich dataset forms the foundation for training precise fraud detection models using the R programming language.

The attributes in the dataset are as follows:

1. **index** - Unique Identifier for each row

2. **trans\_date\_trans\_time** - Transaction Date Time

3. **cc\_num** - Credit Card Number of Customer

4. **merchant** - Merchant Name

5. **category** - Category of Merchant

6. **amt** - Amount of Transaction

7. **first** - First Name of Credit Card Holder

8. **last** - Last Name of Credit Card Holder

9. **gender** - Gender of Credit Card Holder

10. **street** - Street Address of Credit Card Holder

11. **city** - City of Credit Card Holder

12. **state** - State of Credit Card Holder

13. **zip** - Zip of Credit Card Holder

14. **lat** - Latitude Location of Credit Card Holder

15. **long** - Longitude Location of Credit Card Holder

16. **city\_pop** - Credit Card Holder's City Population

17. **job** - Job of Credit Card Holder

18. **dob** - Date of Birth of Credit Card Holder

19. **trans\_num** - Transaction Number

20. **unix\_time** - UNIX Time of transaction

21. **merch\_lat** - Latitude Location of Merchant

22. **merch\_long** - Longitude Location of Merchant

23. **is\_fraud** - Fraud Flag <--- Target Class

**Exploratory Data Analysis**

1. **Data Import and Library Usage**

- Library Usage: Libraries like `ranger`, `caret`, and `data.table` are essential for various functionalities in R. `ranger` is useful for building random forests, `caret` aids in machine learning modeling, and `data.table` offers efficient data manipulation.

- Package Loading: The loading of packages like `lattice` and `ggplot2` provides tools for data visualization, enabling the creation of insightful plots and graphs.

2. **Dataset Import and Overview**

- Dataset Import: Importing the dataset `creditcard.csv` into the R environment using `read.csv` allows for further analysis.

- Initial Data Examination: The `head(df)` function provides a glimpse of the initial rows, showcasing the structure of the dataset and the first few observations, aiding in understanding the data's attributes.

3. **Checking for Missing Values**

- Missing Values Check: The function `sum(is.na(df))` is used to identify the presence of missing values across the dataset. The result of zero missing entries indicates a lack of missing data.

- Visualization of Missing Values: Utilizing the `vis\_miss(df)` function from the `naniar` package visually displays any missing data patterns, assisting in understanding the completeness of the dataset.

4. **Data Attributes and Summary Statistics**

- Dataset Dimensions: The function `dim(df)` displays the number of rows and columns in the dataset, giving an overview of the dataset's size and shape.

- Target Variable Frequency: Using `table(df$Class)` provides a frequency table for the target variable 'Class', showcasing the distribution of classes, especially important for imbalanced datasets.

- Summary Statistics: The `summary(df$Amount)` function offers descriptive statistics like minimum, maximum, and quartiles for the 'Amount' attribute, aiding in understanding its distribution and range.

A graph of a number of numbers

Description automatically generated

A graph of a class

Description automatically generated

5. **Attribute Inspection and Data Transformation**

- Attribute Names Inspection: Checking attribute names via `names(df)` provides a list of all attributes in the dataset, essential for understanding the variables available for analysis.

- Variance and Standard Deviation: Calculating the variance and standard deviation using `var(df$Amount)` and `sd(df$Amount)` respectively gives insights into the spread and dispersion of the 'Amount' variable.

- Data Transformation: Standardizing the 'Amount' variable with `scale(df$Amount)` allows for normalization, facilitating easier comparison and analysis across variables. Displaying the transformed data using `head(df\_1)` demonstrates the effect of the transformation on the dataset.

A graph showing a number of classes

Description automatically generated

A graph of a graph

Description automatically generated with medium confidence

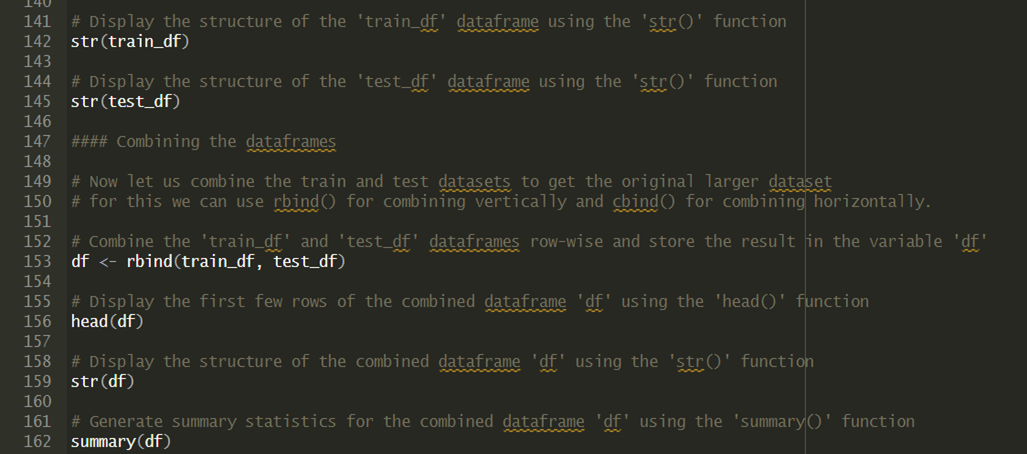
**Data Mining:**

1. **Objective**

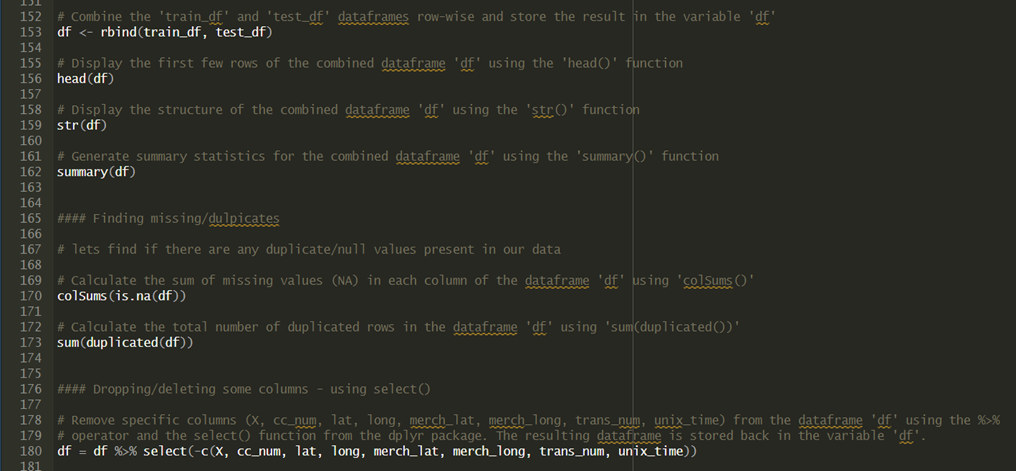
The objective of Mining is to preprocess and prepare the credit card transaction data for analysis and modeling in the context of fraud detection. The code performs various operations such as loading necessary libraries, reading, and combining datasets, handling missing and duplicate values, deriving new columns, converting data types, and reordering columns. These operations are aimed at preparing the data for further analysis, feature engineering, and model building to detect fraudulent credit card transactions.

1. **Data Preparation:**

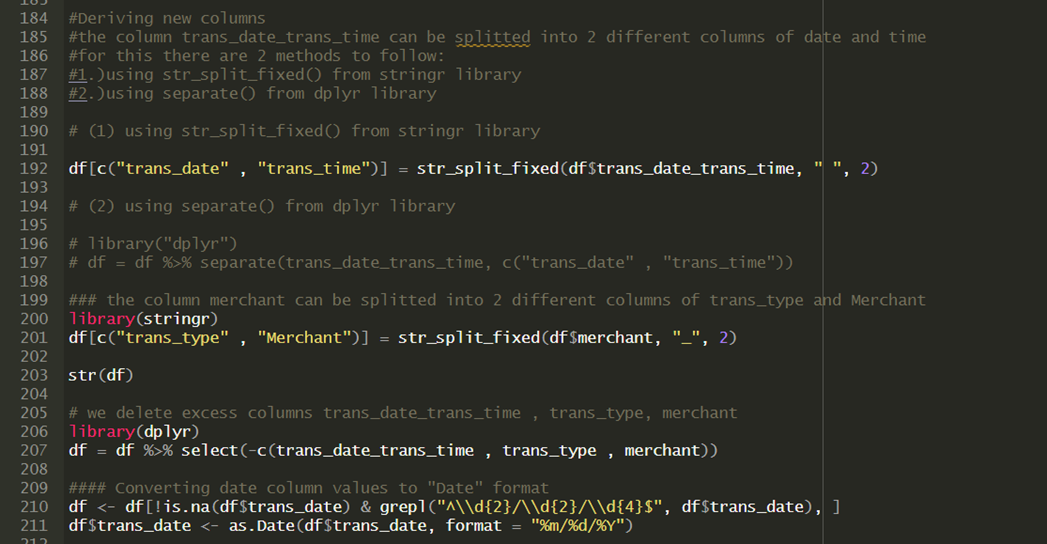
* Structure Exploration: Uses str() function to explore the structure of the datasets.
* Combining Datasets: Combines the training and testing datasets vertically using rbind() and stores the result in the variable df.
* Summary Statistics: Generates summary statistics for the combined dataset using summary() function.



Handling Missing/Duplicate Values: Identifies and handles missing values and duplicates in the dataset.



* Column Selection: Drops specific columns from the dataset using the select() function from the dplyr package.
* Deriving New Columns: Splits the trans\_date\_trans\_time column into trans\_date and trans\_time using str\_split\_fixed() from the stringr library. Further splits the merchant column into trans\_type and Merchant.



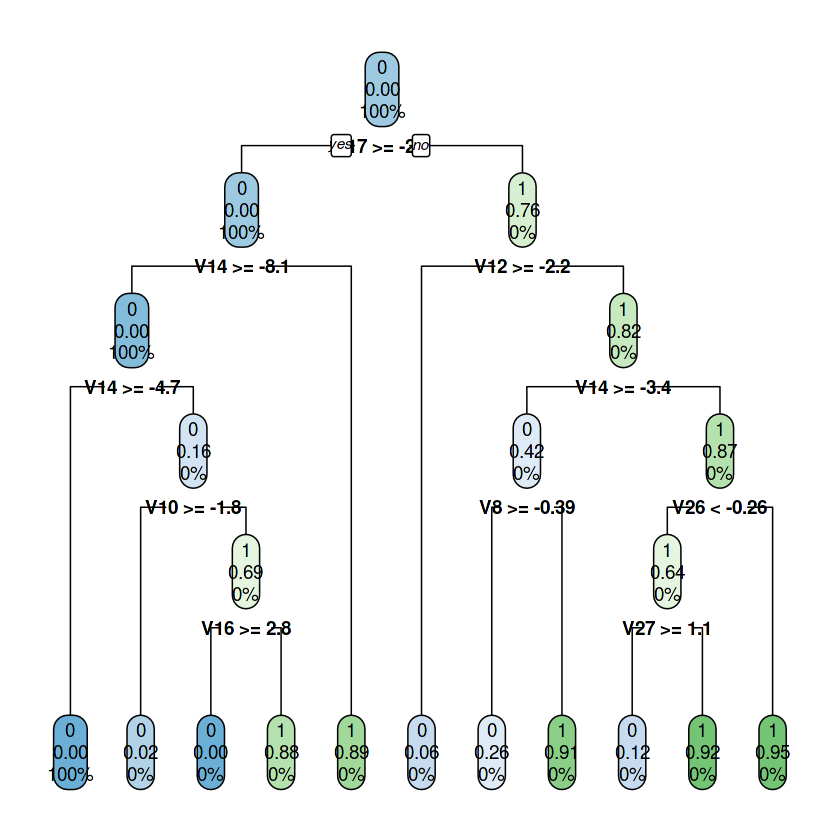
* Date Conversion: Converts the trans\_date column values to "Date" format and calculates the age variable based on the dob column.
* Categorical Variable Conversion: Converts categorical variables to factor type.
* One-Hot Encoding: Creates a new variable Female indicating gender with 0 for Male and 1 for Female using ifelse(), and drops the original gender column.

**BI Model**

Throughout this project, a range of machine learning models was meticulously employed, each bringing its own insights and intricacies to the realm of credit card fraud detection:

1. **Logistic Regression:** This model, while providing initial insights, encountered warnings regarding fitted probabilities. This issue raised concerns about convergence, potentially affecting the model's stability in predicting fraudulent activities.
2. **Decision Tree:** Leveraging the capabilities of the rpart and rpart.plot libraries, the Decision Tree model was visualized. Its graphical representation allowed for a comprehensive understanding of the decision-making process, enhancing interpretability and laying the groundwork for transparent fraud detection.

|  |
| --- |
| library(rpart) library(rpart.plot) decisionTree\_model <- rpart(Class ~ . , df, method = 'class') predicted\_val <- predict(decisionTree\_model, df, type = 'class') probability <- predict(decisionTree\_model, df, type = 'prob')  rpart.plot(decisionTree\_model) |



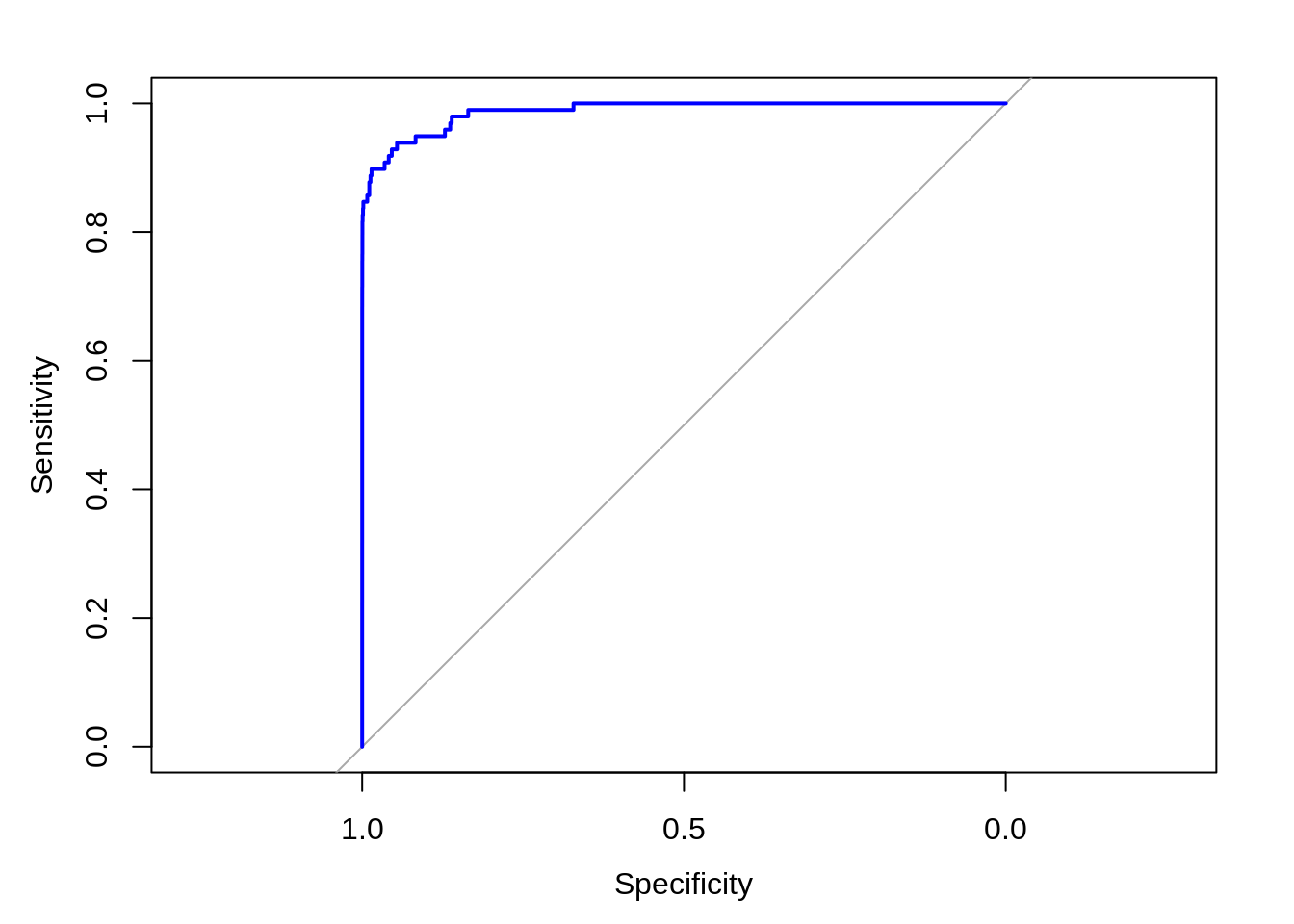
1. **Gradient Boosting:** Among the models, Gradient Boosting emerged as a standout performer. Remarkably, it showcased its prowess by achieving a remarkable AUC of 0.9541. This evaluation was conducted meticulously using the pROC library, affirming its efficacy as a robust tool for identifying fraudulent transactions with high precision.

|  |
| --- |
| 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)) ) ) |

**OUTPUT:**

|  |
| --- |
| Loaded gbm 2.1.8  user system elapsed  400.057 0.216 400.630 |

|  |
| --- |
| 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") |



1. Artificial Neural Networks (ANN): The ANN models are able to learn the patterns using the historical data and are able to perform classification on the input data. We import the neuralnet package that would allow us to implement our 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. We set a threshold as 0.5, that is, values above 0.5 will correspond to 1 and the rest will be 0. We implement this as follows –

Code:

library(neuralnet)

ANN\_model =neuralnet (Class~.,train\_data,linear.output=FALSE)

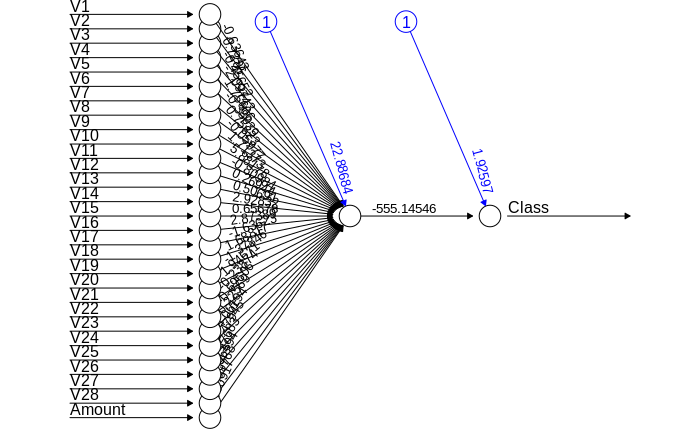
plot(ANN\_model)

predANN=compute(ANN\_model,test\_data)

resultANN=predANN$net.result

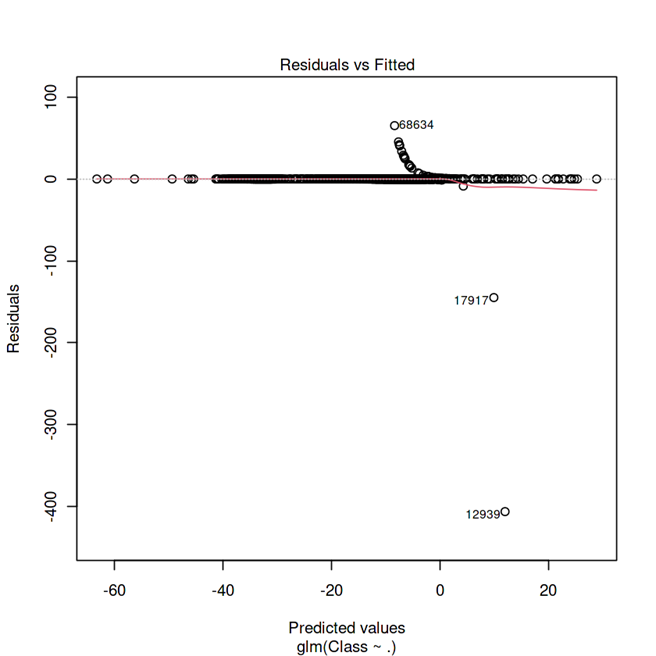
resultANN=ifelse(resultANN>0.5,1,0).

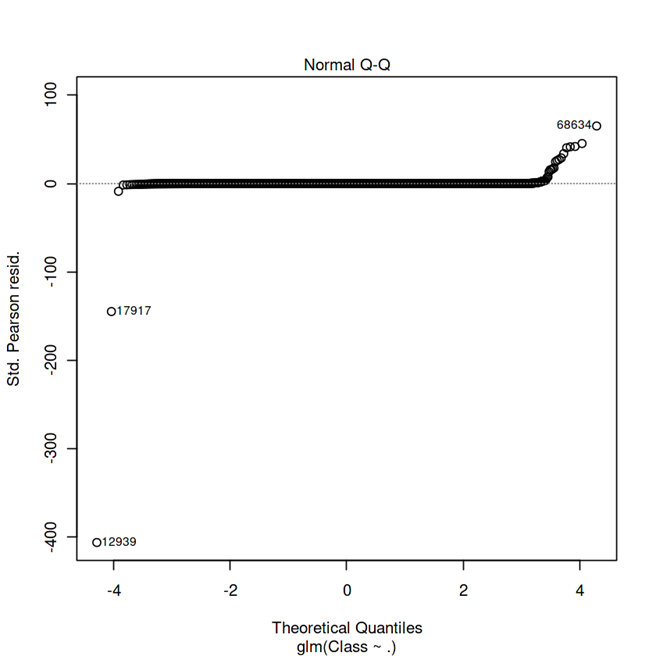
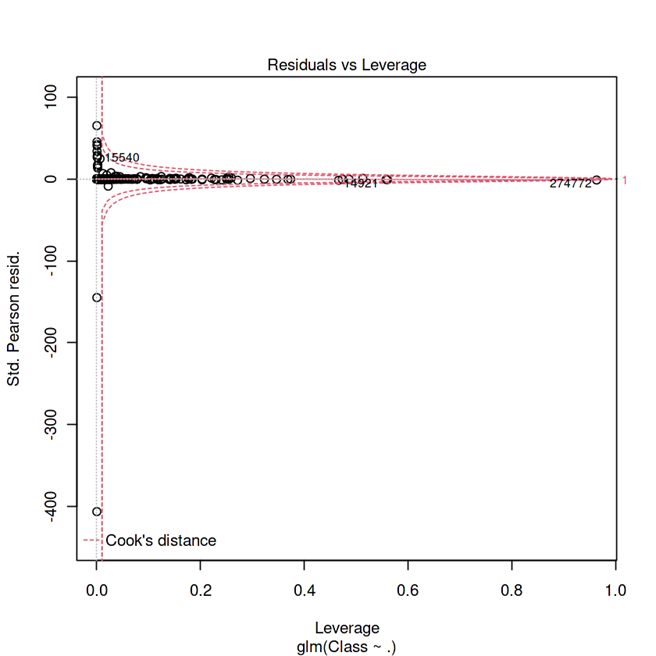
The ANN model stood as a promising prospect for further in-depth analysis and refinement to fortify fraud detection systems. The extensive exploration of these diverse models underlined the critical role of model selection in bolstering security measures. Particularly, the exceptional performance of Gradient Boosting highlighted its potential as an integral component in financial systems, offering an effective shield against fraudulent activities and emphasizing the need for comprehensive analysis in reinforcing security protocols.



Now plotting our model result

|  |
| --- |
| plot(lm) |





# Applying different models:

# Fitting Logistic Regression Model

|  |  |
| --- | --- |
| lm=glm(Class~.,test\_data,family=binomial()) summary(lm)  Call: glm(formula = Class ~ ., family = binomial(), data = test\_data) | |
| Deviance Residuals:   Min 1Q Median 3Q Max  -4.9019 -0.0254 -0.0156 -0.0078 4.0877  Coefficients:  Estimate Std. Error z value Pr(>|z|)  (Intercept) -12.52800 10.30537 -1.216 0.2241  V1 -0.17299 1.27381 -0.136 0.8920  V2 1.44512 4.23062 0.342 0.7327  V3 0.17897 0.24058 0.744 0.4569  V4 3.13593 7.17768 0.437 0.6622  V5 1.49014 3.80369 0.392 0.6952  V6 -0.12428 0.22202 -0.560 0.5756  V7 1.40903 4.22644 0.333 0.7388  V8 -0.35254 0.17462 -2.019 0.0435 \*  V9 3.02176 8.67262 0.348 0.7275  V10 -2.89571 6.62383 -0.437 0.6620  V11 -0.09769 0.28270 -0.346 0.7297  V12 1.97992 6.56699 0.301 0.7630  V13 -0.71674 1.25649 -0.570 0.5684  V14 0.19316 3.28868 0.059 0.9532  V15 1.03868 2.89256 0.359 0.7195  V16 -2.98194 7.11391 -0.419 0.6751  V17 -1.81809 4.99764 -0.364 0.7160  V18 2.74772 8.13188 0.338 0.7354  V19 -1.63246 4.77228 -0.342 0.7323  V20 -0.69925 1.15114 -0.607 0.5436  V21 -0.45082 1.99182 -0.226 0.8209  V22 -1.40395 5.18980 -0.271 0.7868  V23 0.19026 0.61195 0.311 0.7559  V24 -0.12889 0.44701 -0.288 0.7731  V25 -0.57835 1.94988 -0.297 0.7668  V26 2.65938 9.34957 0.284 0.7761  V27 -0.45396 0.81502 -0.557 0.5775  V28 -0.06639 0.35730 -0.186 0.8526  Amount 0.22576 0.71892 0.314 0.7535  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1    (Dispersion parameter for binomial family taken to be 1)    Null deviance: 1443.40 on 56960 degrees of freedom  Residual deviance: 378.59 on 56931 degrees of freedom  AIC: 438.59    Number of Fisher Scoring iterations: 17 |

**Interpretations**

The Logistic Regression model encountered warnings regarding fitted probabilities, possibly indicating challenges in convergence or data characteristics impacting the model's stability. While Logistic Regression faced challenges related to fitted probabilities, possibly due to convergence issues or data nuances, Decision Tree visualization provided explanatory insights. Yet, the standout was the Gradient Boosting model, showcasing remarkable performance with an AUC of 0.9541, marking it as an effective fraud detection tool.

In real-world applications, this project assumes paramount importance for financial entities and online businesses dealing with transactions. Detecting fraudulent activities in credit card transactions holds immense significance, ensuring financial integrity and user trust. The utilization of machine learning algorithms, exemplified here, serves as a practical approach to unearth suspicious patterns, fortifying security measures. Moreover, this notebook not only demonstrates model implementation but also serves as an instructive resource for mastering R programming within the realm of data analysis and model development.

**Conclusion**

The culmination of this project emphasizes the criticality of employing a spectrum of machine learning algorithms in credit card fraud detection. The project underscored the significance of employing diverse machine learning algorithms for credit card fraud detection. Despite Logistic Regression encountering warnings and Decision Tree enhancing interpretability, Gradient Boosting stood out with its robustness, showcasing a high AUC. This model warrants serious consideration by managers for implementation within financial systems to fortify security measures and curtail fraudulent activities. The insights gleaned from this exhaustive analysis signify Gradient Boosting as a potent fraud detection mechanism, imparting valuable guidance for augmenting security protocols in financial landscapes.

**References**

1. <https://www.sciencedirect.com/science/article/pii/S187705092030065X>
2. <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00573-8>
3. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10280638/>
4. <https://library.ndsu.edu/ir/bitstream/handle/10365/31611/Credit%20Card%20Fraud%20Detection%20Predictive%20Modeling.pdf?sequence=1&isAllowed=y>
5. <https://www.kaggle.com/code/gpreda/credit-card-fraud-detection-predictive-models>