Project Group:
Pradyothan Govrineni
Sohaib Syed
Uttam Kotadiya
Harsh Gordhan Dungrani

Group Leader:
Pradyothan Govrineni

A formal description of the project with a stated research goal.

Billions of dollars of loss are caused every year due to fraudulent credit card transactions. The aim of this project is to build a classifier system that can detect credit card fraudulent transactions. We will use R programming language and Card Transactions dataset that will be able to discern fraudulent from non-fraudulent ones. By the end of this project, We will learn how to implement various algorithms to perform classification.

A specific question or set of questions that the project seeks to address.

1. How to navigate class imbalance issues when training models to detect outliers?
2. Across multiple datasets is there a feature that is shared among fraudulent transactions?
3. What makes a relevant feature when it comes to credit card fraud?
4. Because outlier detection problems face class imbalance issues, is it safer to coerce model decisions to positive class for credit card fraud if the decision is near 50%?

A proposed methodology/approach to the analysis that will be performed.
Due to data availability, the time frame of this analysis is restricted to 2010-2020.
Data Preparation:
The data will be imported using the R import library.
In order to get the dataset ready for the following stage, try to reduce its complexity.
Both missing data and stacked fields are present in our data. In order to reunite variables, we must first separate them.

Data Discovery:
Discover our dataset to better and more effectively handle the next procedures.
In order to decide on data cleaning, and modeling, we must go thoroughly into our dataset, and represent each feature.
Some information about each feature in our dataset is the result of this stage.

Data Modeling:

Attempt to complete this project, and respond to the key questions we have.

To identify the best model to suit the data and determine the likelihood of each credit card theft, we may clean, reorganize, and split the data.

Root mean square error is a commonly used statistic for determining the disparities between values predicted by a model and the values actually observed.

A metric or set of metrics which will measure analysis results.

We will use the following performance metrics to evaluate the classifier models:

1. Confusion matrix
2. Accuracy Score
3. Precision Score
4. Recall Score
5. F1 Score
6. Area Under Curve (AUC) Score
7. ROC Curve
8. Log loss/ binary cross entropy
9. Matthew's correlation coefficient (MCC)

Literature review and related work

D. Prajapati et al. [1] performed credit card fraud detection using Random Forest, XGBoost, ANN (Artificial Neural Network) and used sampling techniques to counter the imbalance dataset. A. S. Rathore et al. [2] compared the performance of Decision Tree, Random Forest, K-nearest neighbors, and logistic regression on highly imbalanced data. Ileberi, E. et al. [3] proposed a machine learning (ML) based credit card fraud detection engine using the genetic algorithm (GA) for feature selection. After the optimized features are chosen, the proposed detection engine uses the following ML classifiers: Decision Tree (DT), Random Forest (RF), Logistic Regression (LR), Artificial Neural Network (ANN), and Naive Bayes (NB) on imbalanced dataset generated from European cardholders.

V. Nisha Jenipher et al. [4] used two library files such as PyCaret and Synthetic Minority Oversampling Technique (SMOTE) for data balancing. Random Forest, Logistic Regression, Isolation Forest, Naïve Bayes and XGBoost Classifier models were compared and evaluated. V Ghai and Sandeep Singh [5] presented numerous advantages in utilizing machine learning for fraud detection, particularly in cases of credit card, because machines are faster and more accurate than manual reviews.

Lei Zhang et al. [6] created a Tabnet based Card Fraud detection algorithm using feature engineering and compared it with traditional models like Naive Bayes and XGBoost classifier. Dennis et al. [7] dealt with binary classification problems in imbalanced financial data using SMOTE algorithm variants and performed Principal Component Analysis (PCA) for dimensionality reduction and built a logistic regression model with cross validation for deciding the best hyperparameters. Hema Gonaboina and et al. [8] used the European credit card fraud dataset and used Logistic Regression, Random Forest and CatBoost models to train and test the dataset with Random Forest and CatBoost performing the best.

References:

[1] D. Prajapati, A. Tripathi, J. Mehta, K. Jhaveri and V. Kelkar, "Credit Card Fraud Detection Using Machine Learning," 2021 International Conference on Advances in Computing, Communication, and Control (ICAC3), 2021, pp. 1-6, doi: 10.1109/ICAC353642.2021.9697227.

[2] A. S. Rathore, A. Kumar, D. Tomar, V. Goyal, K. Sarda and D. Vij, "Credit Card Fraud Detection using Machine Learning," 2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART), 2021, pp. 167-171, doi: 10.1109/SMART52563.2021.9676262.

[3] Ileberi, E., Sun, Y. & Wang, Z. A machine learning based credit card fraud detection using the GA algorithm for feature selection. J Big Data 9, 24 (2022). https://doi.org/10.1186/s40537-022-00573-8

[4] V. N. Jenipher, J. Dafni Rose, M. Sabharam and M. Nithin, "Learning Algorithms with Data Balancing in Credit Card Fraud Detection Application," 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2021, pp. 1-6, doi: 10.1109/I-SMAC52330.2021.9640731

[5] V. Ghai and S. S. Kang, "Role of Machine Learning in Credit Card Fraud Detection," 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), 2021, pp. 939-943, doi: 10.1109/ICAC3N53548.2021.9725540.

[6] L. Zhang, K. Ma, F. Yuan and W. Fang, "A Tabnet based Card Fraud detection Algorithm with Feature Engineering," 2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE), 2022, pp. 911-914, doi: 10.1109/ICCECE54139.2022.9712822.

[7] Dennis, I. R. Budianto, R. K. Azaria and A. A. S. Gunawan, "Machine Learning-based Approach on Dealing with Binary Classification Problem in Imbalanced Financial Data," 2021 International Seminar on Machine Learning, Optimization, and Data Science (ISMODE), 2022, pp. 152-156, doi: 10.1109/ISMODE53584.2022.9742834.

[8] Vaishnavi Nath Dornadula, S Geetha, “Credit Card Fraud Detection using Machine Learning Algorithms, Procedia Computer Science”, Volume 165, 2019, Pages 631-641, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.01.057.(https://www.sciencedirect.com/science/article/pii/S187705092030065X).

All data sources and reference data with descriptions

https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud
https://www.kaggle.com/datasets/yashpaloswal/fraud-detection-credit-card
https://www.kaggle.com/datasets/mishra5001/credit-card

Data processing and pipeline

The biggest issue we will face in our data is making sure there is a balance between positive and negative classes in order to take away value from results
Again, as class imbalance will be an issue simply removing observations with empty or missing data can be ill-advised, so deciding to fill in missing values with appropriate values will be necessary
Splitting the train and test data will need to be done intelligently so that the distribution of positive and negative classes avoids over/under fitting.

Data stylized facts

Dataset 1:
https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud
Dataset 2:
https://www.kaggle.com/datasets/mishra5001/credit-card

For Dataset 1:

The first dataset consists of 284807 rows and 31 attributes

```
'Number of rows in dataset: 284807'
'Number of attributes in dataset: 31'
```

The data types of the 31 attributes are:

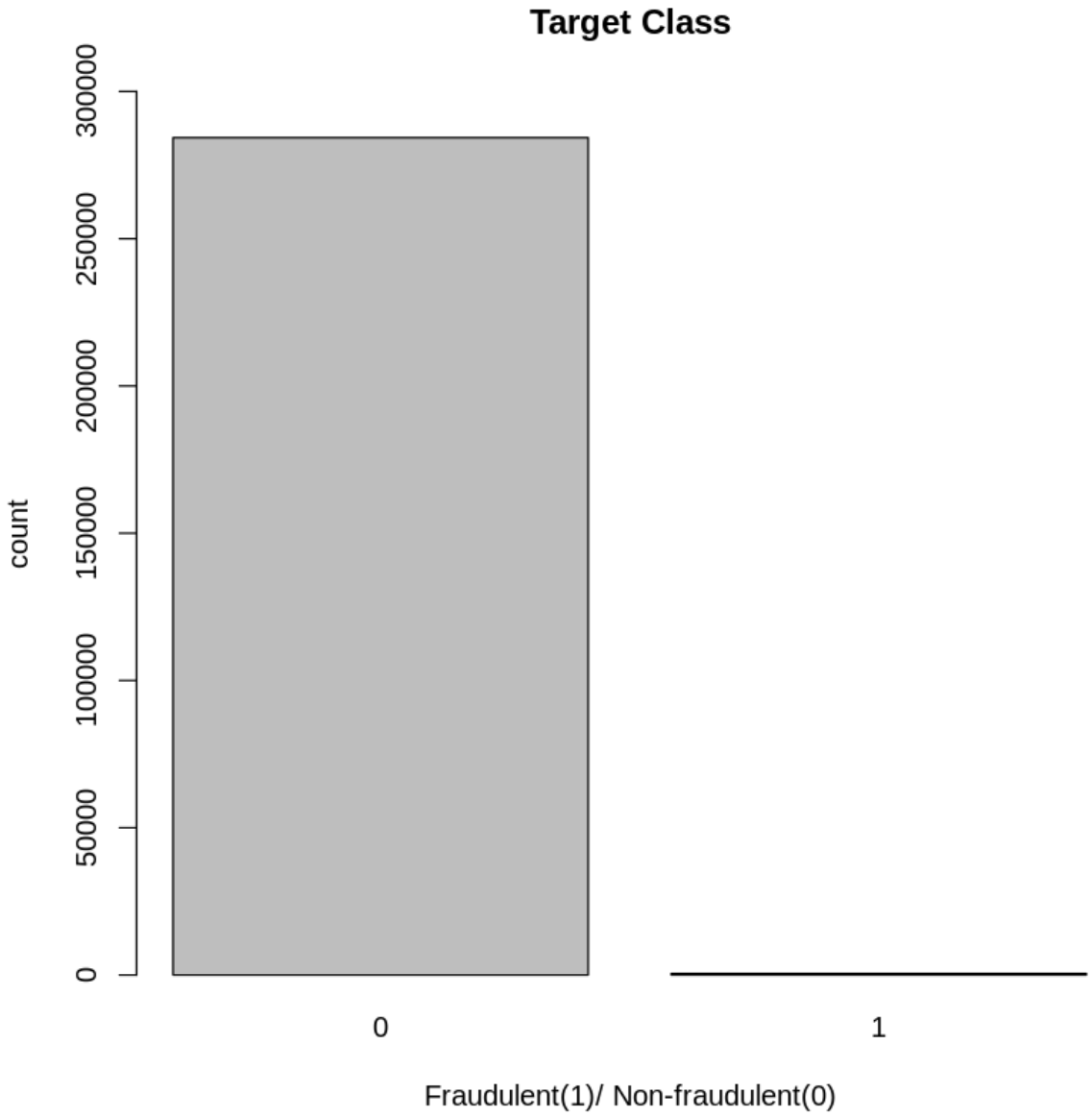
```
integer numeric
      1      30
```

The structure of dataset 1 is:

```
'data.frame':   284807 obs. of  31 variables:
 $ Time  : num  0 0 1 1 2 2 4 7 7 9 ...
 $ V1    : num  -1.36 1.192 -1.358 -0.966 -1.158 ...
 $ V2    : num  -0.0728 0.2662 -1.3402 -0.1852 0.8777 ...
 $ V3    : num  2.536 0.166 1.773 1.793 1.549 ...
 $ V4    : num  1.378 0.448 0.38 -0.863 0.403 ...
 $ V5    : num  -0.3383 0.06 -0.5032 -0.0103 -0.4072 ...
 $ V6    : num  0.4624 -0.0824 1.8005 1.2472 0.0959 ...
 $ V7    : num  0.2396 -0.0788 0.7915 0.2376 0.5929 ...
 $ V8    : num  0.0987 0.0851 0.2477 0.3774 -0.2705 ...
 $ V9    : num  0.364 -0.255 -1.515 -1.387 0.818 ...
 $ V10   : num  0.0908 -0.167 0.2076 -0.055 0.7531 ...
 $ V11   : num  -0.552 1.613 0.625 -0.226 -0.823 ...
 $ V12   : num  -0.6178 1.0652 0.0661 0.1782 0.5382 ...
 $ V13   : num  -0.991 0.489 0.717 0.508 1.346 ...
 $ V14   : num  -0.311 -0.144 -0.166 -0.288 -1.12 ...
 $ V15   : num  1.468 0.636 2.346 -0.631 0.175 ...
 $ V16   : num  -0.47 0.464 -2.89 -1.06 -0.451 ...
 $ V17   : num  0.208 -0.115 1.11 -0.684 -0.237 ...
 $ V18   : num  0.0258 -0.1834 -0.1214 1.9658 -0.0382 ...
 $ V19   : num  0.404 -0.146 -2.262 -1.233 0.803 ...
 $ V20   : num  0.2514 -0.0691 0.525 -0.208 0.4085 ...
 $ V21   : num  -0.01831 -0.22578 0.248 -0.1083 -0.00943 ...
 $ V22   : num  0.27784 -0.63867 0.77168 0.00527 0.79828 ...
 $ V23   : num  -0.11 0.101 0.909 -0.19 -0.137 ...
 $ V24   : num  0.0669 -0.3398 -0.6893 -1.1756 0.1413 ...
 $ V25   : num  0.129 0.167 -0.328 0.647 -0.206 ...
 $ V26   : num  -0.189 0.126 -0.139 -0.222 0.502 ...
 $ V27   : num  0.13356 -0.00898 -0.05535 0.06272 0.21942 ...
 $ V28   : num  -0.0211 0.0147 -0.0598 0.0615 0.2152 ...
 $ Amount: num  149.62 2.69 378.66 123.5 69.99 ...
 $ Class : int  0 0 0 0 0 0 0 0 0 0 ...
```

Here, we can observe that the 30 predictors are numeric while the target ‘Class’ is integer type.

The major issue that we face in our project is class imbalance which can be observed from the plot below.



The ‘Class’ attribute can be seen to have the following distribution:

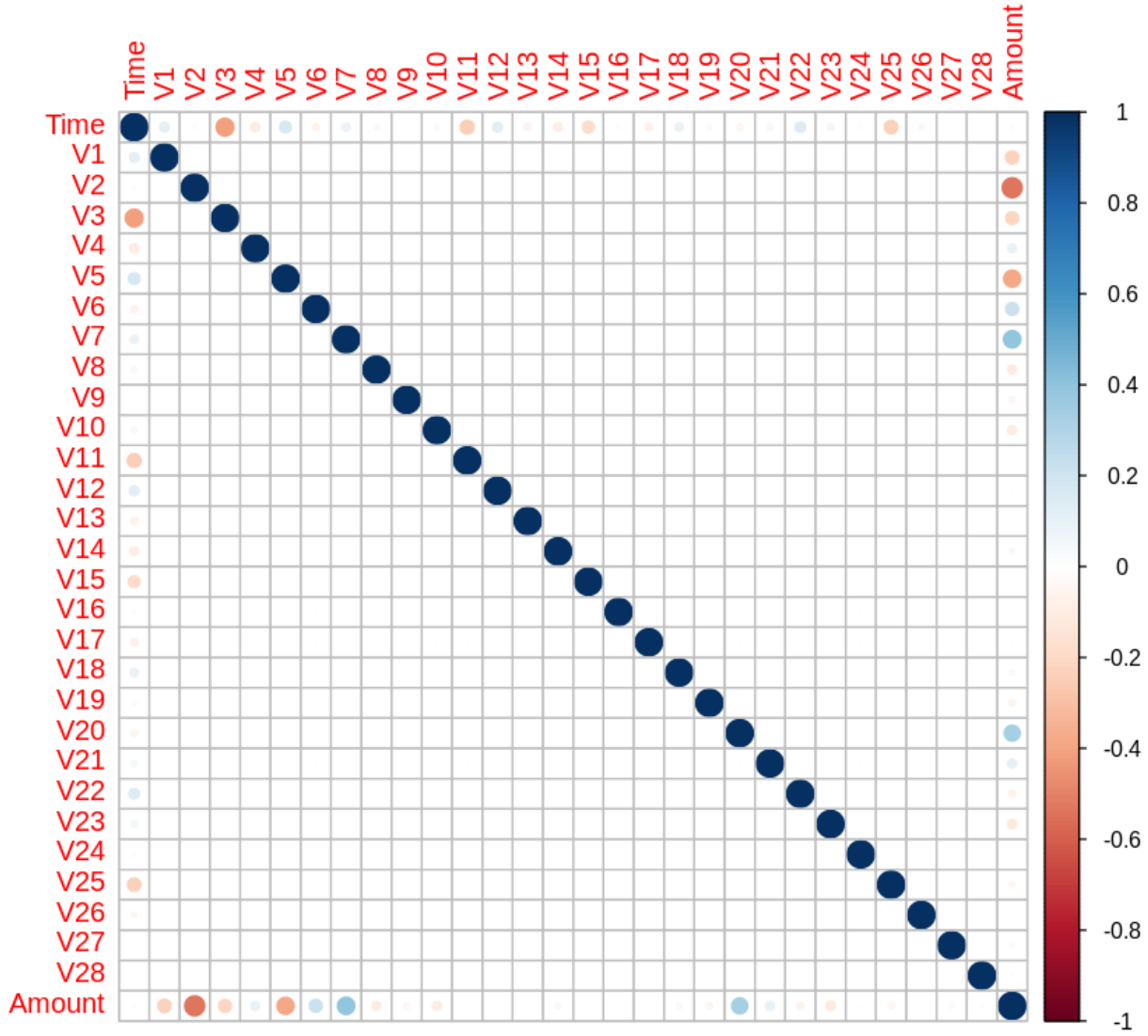
```
table(cd1$Class)
```

```
     0      1 
284315    492 
```

```
table(cd1$Class) / nrow(cd1)
```

```
          0           1 
0.998272514 0.001727486 
```

It is highly imbalanced with 284315 non-fraudulent and 492 fraudulent cases in the dataset.



From the above correlation plot among the 30 predictors we can see that the attributes V3 and Time and attributes V2 and Amount have noticeable negative correlation between each other.

For Dataset 2:

The second dataset consists of 307511 rows and 122 attributes

```
'Number of rows in dataset: 307511'
'Number of attributes in dataset: 122'
```

The data types of the 122 attributes are:

character	integer	numeric
16	41	65

The dataset consists of three different data types namely character, integer and numeric so we would have to transform the character into an appropriate data type for our project.

The structure of dataset 2 is:

```
'data.frame':   307511 obs. of  122 variables:
 $ SK_ID_CURR                : int  100002 100003 100004 100006 100007 100008 100009 100010 100011 100012 ...
 $ TARGET                    : int  1 0 0 0 0 0 0 0 0 0 ...
 $ NAME_CONTRACT_TYPE        : chr  "Cash loans" "Cash loans" "Revolving loans" "Cash loans" ...
 $ CODE_GENDER               : chr  "M" "F" "M" "F" ...
 $ FLAG_OWN_CAR              : chr  "N" "N" "Y" "N" ...
 $ FLAG_OWN_REALTY           : chr  "Y" "N" "Y" "Y" ...
 $ CNT_CHILDREN              : int  0 0 0 0 0 0 1 0 0 0 ...
 $ AMT_INCOME_TOTAL          : num  202500 270000 67500 135000 121500 ...
 $ AMT_CREDIT                : num  406598 1293502 135000 312682 513000 ...
 $ AMT_ANNUITY               : num  24700 35698 6750 29686 21866 ...
 $ AMT_GOODS_PRICE           : num  351000 1129500 135000 297000 513000 ...
 $ NAME_TYPE_SUITE           : chr  "Unaccompanied" "Family" "Unaccompanied" "Unaccompanied" ...
 $ NAME_INCOME_TYPE          : chr  "Working" "State servant" "Working" "Working" ...
 $ NAME_EDUCATION_TYPE       : chr  "Secondary / secondary special" "Higher education" "Secondary / secondary special" "Secondary / secondary special" ...
 $ NAME_FAMILY_STATUS        : chr  "Single / not married" "Married" "Single / not married" "Civil marriage" ...
 $ NAME_HOUSING_TYPE         : chr  "House / apartment" "House / apartment" "House / apartment" "House / apartment" ...
 $ REGION_POPULATION_RELATIVE: num  0.0188 0.00354 0.01003 0.00802 0.02866 ...
 $ DAYS_BIRTH                : int  -9461 -16765 -19046 -19005 -19932 -16941 -13778 -18850 -20099 -14469 ...
 $ DAYS_EMPLOYED             : int  -637 -1188 -225 -3039 -3038 -1588 -3130 -449 365243 -2019 ...
 $ DAYS_REGISTRATION         : num  -3648 -1186 -4260 -9833 -4311 ...
 $ DAYS_ID_PUBLISH           : int  -2120 -291 -2531 -2437 -3458 -477 -619 -2379 -3514 -3992 ...
 $ OWN_CAR_AGE               : num  NA NA 26 NA NA NA 17 8 NA NA ...
 $ FLAG_MOBIL                : int  1 1 1 1 1 1 1 1 1 1 ...
 $ FLAG_EMP_PHONE            : int  1 1 1 1 1 1 1 1 0 1 ...
 $ FLAG_WORK_PHONE           : int  0 0 1 0 0 1 0 1 0 0 ...
 $ FLAG_CONT_MOBILE          : int  1 1 1 1 1 1 1 1 1 1 ...
 $ FLAG_PHONE                : int  1 1 1 0 0 1 1 0 0 0 ...
 $ FLAG_EMAIL                : int  0 0 0 0 0 0 0 0 0 0 ...
 $ OCCUPATION_TYPE           : chr  "Laborers" "Core staff" "Laborers" "Laborers" ...
 $ CNT_FAM_MEMBERS           : num  1 2 1 2 1 2 3 2 2 1 ...
 $ REGION_RATING_CLIENT      : int  2 1 2 2 2 2 2 3 2 2 ...
 $ REGION_RATING_CLIENT_W_CITY : int  2 1 2 2 2 2 2 3 2 2 ...
 $ WEEKDAY_APPR_PROCESS_START  : chr  "WEDNESDAY" "MONDAY" "MONDAY" "WEDNESDAY" ...
 $ HOUR_APPR_PROCESS_START     : int  10 11 9 17 11 16 16 16 14 8 ...
```

```
$ REG_REGION_NOT_LIVE_REGION  : int  0 0 0 0 0 0 0 0 0 0 ...
$ REG_REGION_NOT_WORK_REGION  : int  0 0 0 0 0 0 0 0 0 0 ...
$ LIVE_REGION_NOT_WORK_REGION : int  0 0 0 0 0 0 0 0 0 0 ...
$ REG_CITY_NOT_LIVE_CITY      : int  0 0 0 0 0 0 0 0 0 0 ...
$ REG_CITY_NOT_WORK_CITY      : int  0 0 0 0 1 0 0 1 0 0 ...
$ LIVE_CITY_NOT_WORK_CITY     : int  0 0 0 0 1 0 0 1 0 0 ...
$ ORGANIZATION_TYPE           : chr  "Business Entity Type 3" "School" "Government" "Business Entity Type 3" ...
$ EXT_SOURCE_1                : num  0.083 0.311 NA NA NA ...
$ EXT_SOURCE_2                : num  0.263 0.622 0.556 0.65 0.323 ...
$ EXT_SOURCE_3                : num  0.139 NA 0.73 NA NA ...
$ APARTMENTS_AVG              : num  0.0247 0.0959 NA NA NA NA NA NA NA NA ...
$ BASEMENTAREA_AVG            : num  0.0369 0.0529 NA NA NA NA NA NA NA NA ...
$ YEARS_BEGINEXPLUATATION_AVG : num  0.972 0.985 NA NA NA ...
$ YEARS_BUILD_AVG             : num  0.619 0.796 NA NA NA ...
$ COMMONAREA_AVG              : num  0.0143 0.0605 NA NA NA NA NA NA NA NA ...
$ ELEVATORS_AVG               : num  0 0.08 NA NA NA NA NA NA NA NA ...
$ ENTRANCES_AVG               : num  0.069 0.0345 NA NA NA NA NA NA NA NA ...
$ FLOORSMAX_AVG               : num  0.0833 0.2917 NA NA NA ...
$ FLOORSMIN_AVG               : num  0.125 0.333 NA NA NA ...
$ LANDAREA_AVG                : num  0.0369 0.013 NA NA NA NA NA NA NA NA ...
$ LIVINGAPARTMENTS_AVG        : num  0.0202 0.0773 NA NA NA NA NA NA NA NA ...
$ LIVINGAREA_AVG              : num  0.019 0.0549 NA NA NA NA NA NA NA NA ...
$ NONLIVINGAPARTMENTS_AVG     : num  0 0.0039 NA NA NA NA NA NA NA NA ...
$ NONLIVINGAREA_AVG           : num  0 0.0098 NA NA NA NA NA NA NA NA ...
$ APARTMENTS_MODE             : num  0.0252 0.0924 NA NA NA NA NA NA NA NA ...
$ BASEMENTAREA_MODE           : num  0.0383 0.0538 NA NA NA NA NA NA NA NA ...
$ YEARS_BEGINEXPLUATATION_MODE: num  0.972 0.985 NA NA NA ...
$ YEARS_BUILD_MODE            : num  0.634 0.804 NA NA NA ...
$ COMMONAREA_MODE             : num  0.0144 0.0497 NA NA NA NA NA NA NA NA ...
$ ELEVATORS_MODE              : num  0 0.0806 NA NA NA NA NA NA NA NA ...
$ ENTRANCES_MODE              : num  0.069 0.0345 NA NA NA NA NA NA NA NA ...
$ FLOORSMAX_MODE              : num  0.0833 0.2917 NA NA NA ...
$ FLOORSMIN_MODE              : num  0.125 0.333 NA NA NA ...
$ LANDAREA_MODE               : num  0.0377 0.0128 NA NA NA NA NA NA NA NA ...
$ LIVINGAPARTMENTS_MODE       : num  0.022 0.079 NA NA NA NA NA NA NA NA ...
```

Here, the target class is ‘TARGET’ which is an integer type. As there are null values (NA) in our dataset we would have to tackle them to build the optimal model possible.

The ‘TARGET’ attribute can be seen to have the following distribution which again has a majority of non-fraudulent transactions:

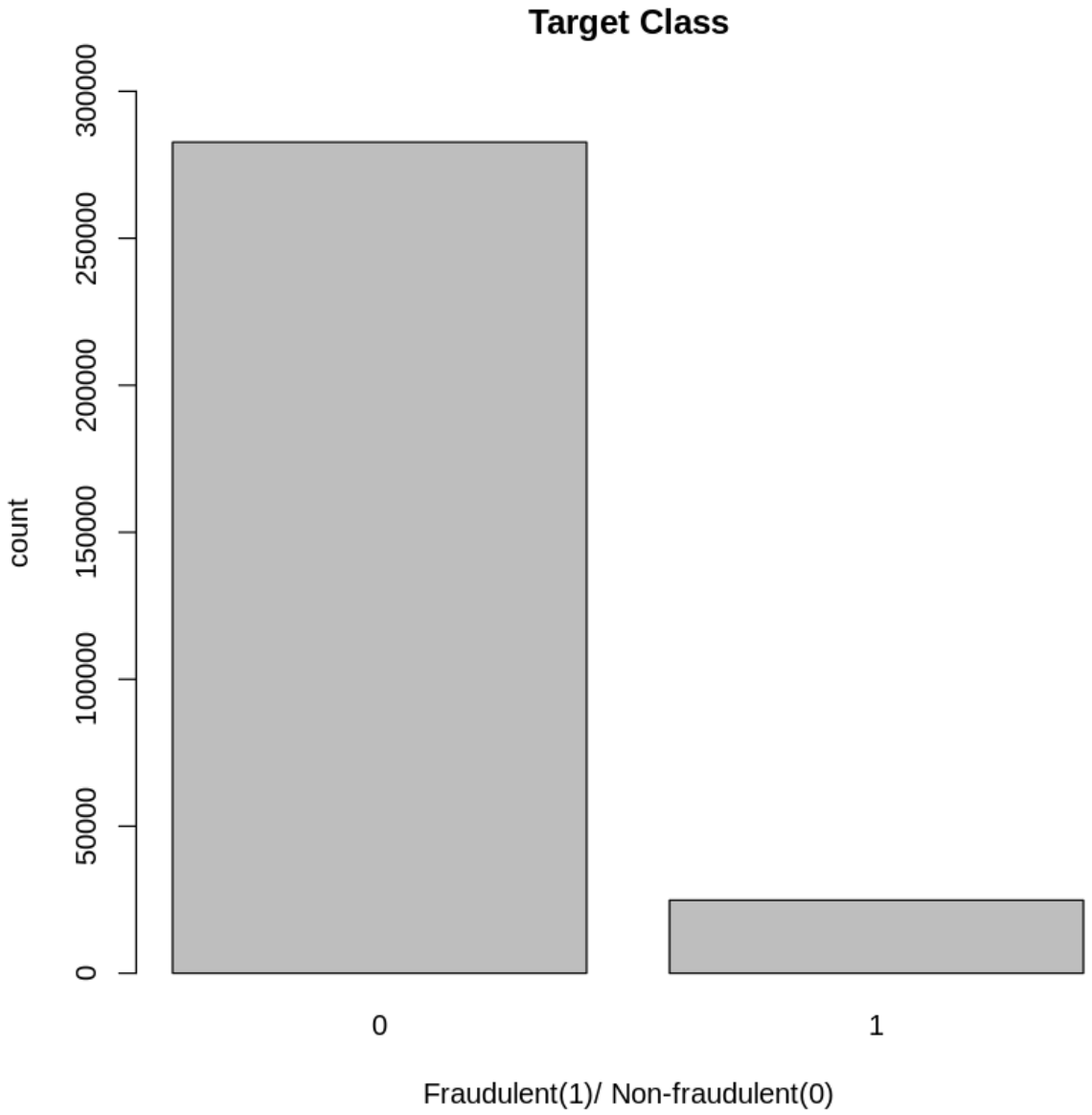
```
table(cd2$TARGET)
```

```
     0      1
282686  24825
```

```
table(cd2$TARGET) / nrow(cd2)
```

```
         0          1
0.91927118 0.08072882
```

The bar plot for ‘TARGET’ attribute:



Model selection

We will use different sampling techniques to tackle class imbalancement issue and build following models for our project:

1)Fitting Logistic Regression Model:

In this section of the credit card fraud detection project, we will fit our first model. We will begin with 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. .

2)Fitting a Decision Tree Model:

The Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too.

3)Artificial Neural Network

Also we use the ROC curve. ROC is also known as Receiver Optimistic Characteristics.

Software packages, applications, libraries, and associated tools

- R language
- GGplot
- NumPy
- SQL