Credit Card Project, HarvardX: PH125.9x

Flavio Alejandro Chavarri Miranda

12/4/2020

1. Excecutive Summary

This is the final project for the HarvardX: PH125.9x Data Science: Capstone course and for the HarvardX Data Science Professional Certificate. All the courses of the professional certificate gave us the knowledge and the tools to make this capstone project. In this opportunity, I will create a model that can analyze and predict when a transaction made with a credit card was a fraud or a legal transaction.

Following the recommendations, the dataset used in this model was downloaded from Kaggle. In Kaggle, the data was uploaded by Machine Learning Group - ULB and it is called "Credit Card Fraud Detection". The name of the excel file in a CSV format is creditcard.csv. To make it easier, I downloaded the file and uploaded to my Google Drive Account. In this project, the data is automatically downloaded from my Google Drive account, therefore there is no need to download the data from another source. The name of the dataset in this model is "creditcard". In case you want to download the data and then upload it to Rmd, you can download it from Kaggle or my Github account. Below are the links to download the data in my Google Drive account, in Kaggle, and my Github account:

https://drive.google.com/uc?id=1QpCApwEac85j2zCmcXWA03zjsZX24f7E&export=download&authuser=0 https://www.kaggle.com/mlg-ulb/creditcardfraud https://github.com/flavio7910/Harvard-Capstone-Project

This dataset has a total of 284807 transactions and a total of 492 frauds. This dataset was collected from the transactions made by European credit cardholders in September 2013. Due to privacy concerns, the variables V2, V2, ... V28 does not have the names of the variables and for instance, we do not know its features. The only two variables that we have the names are the time and amount variables.

This model seeks to recognize when a transaction was legal or when it was a fraud. To evaluate the data, I split the data in the most common way making a 70/30 split, 70 percent for the training and 30 percent for the test set. The goal of this project is to make a model that can reach an accuracy rate above 99.95% so that we can have an almost perfect amount of true positives and true negatives in future transactions. To make this happen, I will use four models: Naive Baseline Model,Logistic Regression Model, Decision Tree Model, and Random Forest Model.

2. Analysis and Methods

The credit card fraud dataset has 31 colums: time, V1 to V28, amount, and class.

```
##   Time         V1          V2        V3         V4          V5          V6
## 1    0 -1.3598071 -0.07278117 2.5363467  1.3781552 -0.33832077  0.46238778
## 2    0  1.1918571  0.26615071 0.1664801  0.4481541  0.06001765 -0.08236081
## 3    1 -1.3583541 -1.34016307 1.7732093  0.3797796 -0.50319813  1.80049938
## 4    1 -0.9662717 -0.18522601 1.7929933 -0.8632913 -0.01030888  1.24720317
```

```
## 5    2 -1.1582331  0.87773676 1.5487178  0.4030339 -0.40719338  0.09592146
## 6    2 -0.4259659  0.96052304 1.1411093 -0.1682521  0.42098688 -0.02972755
##            V7          V8         V9         V10        V11         V12
## 1  0.23959855  0.09869790  0.3637870  0.09079417 -0.5515995 -0.61780086
## 2 -0.07880298  0.08510165 -0.2554251 -0.16697441  1.6127267  1.06523531
## 3  0.79146096  0.24767579 -1.5146543  0.20764287  0.6245015  0.06608369
## 4  0.23760894  0.37743587 -1.3870241 -0.05495192 -0.2264873  0.17822823
## 5  0.59294075 -0.27053268  0.8177393  0.75307443 -0.8228429  0.53819555
## 6  0.47620095  0.26031433 -0.5686714 -0.37140720  1.3412620  0.35989384
##          V13        V14        V15        V16         V17         V18
## 1 -0.9913898 -0.3111694  1.4681770 -0.4704005  0.20797124  0.02579058
## 2  0.4890950 -0.1437723  0.6355581  0.4639170 -0.11480466 -0.18336127
## 3  0.7172927 -0.1659459  2.3458649 -2.8900832  1.10996938 -0.12135931
## 4  0.5077569 -0.2879237 -0.6314181 -1.0596472 -0.68409279  1.96577500
## 5  1.3458516 -1.1196698  0.1751211 -0.4514492 -0.23703324 -0.03819479
## 6 -0.3580907 -0.1371337  0.5176168  0.4017259 -0.05813282  0.06865315
##           V19         V20          V21          V22         V23         V24
## 1  0.40399296  0.25141210 -0.018306778  0.277837576 -0.11047391  0.06692808
## 2 -0.14578304 -0.06908314 -0.225775248 -0.638671953  0.10128802 -0.33984648
## 3 -2.26185709  0.52497973  0.247998153  0.771679402  0.90941226 -0.68928096
## 4 -1.23262197 -0.20803778 -0.108300452  0.005273597 -0.19032052 -1.17557533
## 5  0.80348692  0.40854236 -0.009430697  0.798278495 -0.13745808  0.14126698
## 6 -0.03319379  0.08496767 -0.208253515 -0.559824796 -0.02639767 -0.37142658
##          V25        V26          V27         V28 Amount Class
## 1  0.1285394 -0.1891148  0.133558377 -0.02105305 149.62     0
## 2  0.1671704  0.1258945 -0.008983099  0.01472417   2.69     0
## 3 -0.3276418 -0.1390966 -0.055352794 -0.05975184 378.66     0
## 4  0.6473760 -0.2219288  0.062722849  0.06145763 123.50     0
## 5 -0.2060096  0.5022922  0.219422230  0.21515315  69.99     0
## 6 -0.2327938  0.1059148  0.253844225  0.08108026   3.67     0
```

The total length of the data is 284807. This means that in our dataset we have a total of 284807 transactions. All of the variables are numeric with the exceptions of "Class" that is an integer.

Length	Columns
284807	31

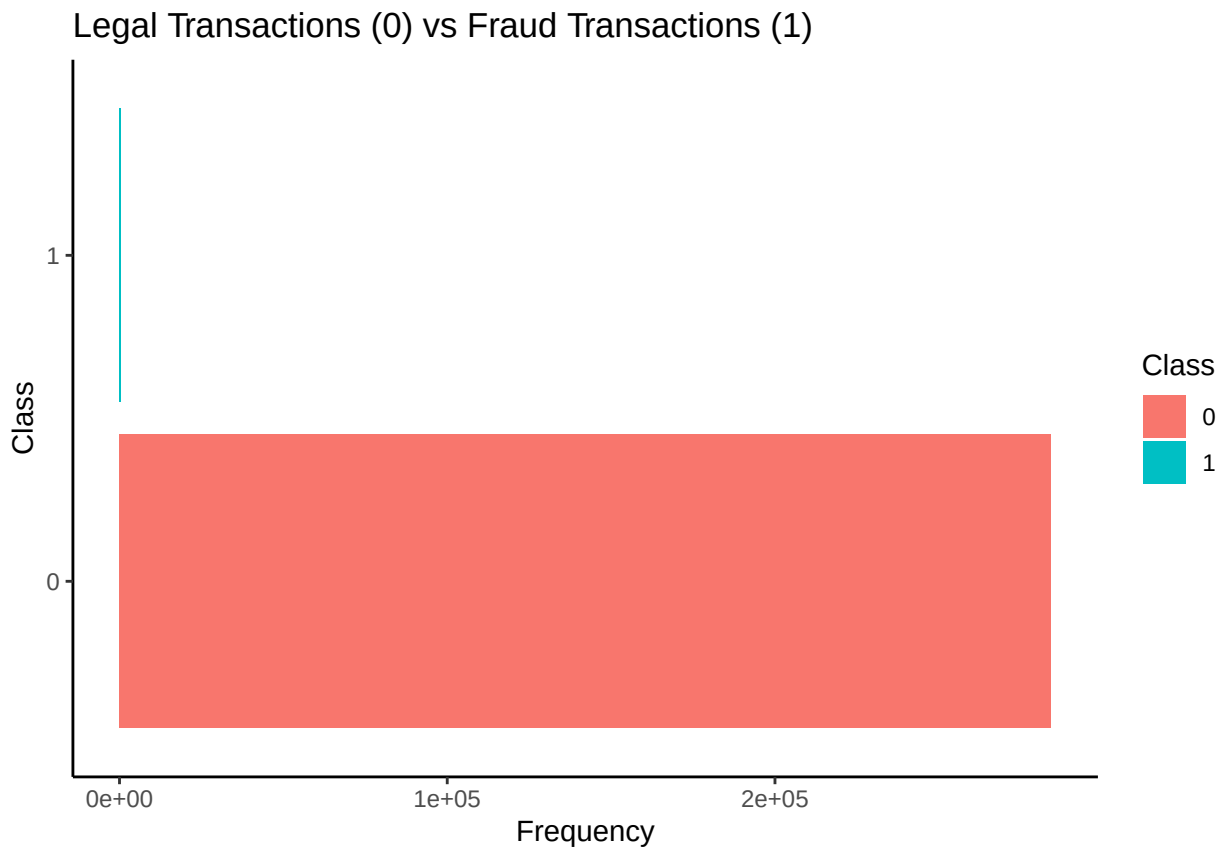
```
##      Time        V1        V2        V3        V4        V5        V6        V7
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##        V8        V9       V10       V11       V12       V13       V14       V15
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##       V16       V17       V18       V19       V20       V21       V22       V23
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##       V24       V25       V26       V27       V28    Amount     Class
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "integer"
```

There are not any missing values in the columns. This will make the analysis easier.

	x
Time	0
V1	0
V2	0
V3	0
V4	0
V5	0
V6	0
V7	0
V8	0
V9	0
V10	0
V11	0
V12	0
V13	0
V14	0
V15	0
V16	0
V17	0
V18	0
V19	0
V20	0
V21	0
V22	0
V23	0
V24	0
V25	0
V26	0
V27	0
V28	0
Amount	0
Class	0

All of the variables have independent numeric values except for the class variable. The class variable is a dummy variable where 0 means that the transaction was legal, and 1 represents that the transaction was a fraud. There is a total of 284315 legal transactions and a total of 492 frauds on the data.

Class	Count
0	284315
1	492



Taking this fact into consideration, it is much more clear to analyze the basic summary statistics. Based on the chart, the average transaction is worth 88.35. Since the amounts are small, it is better to analyze the fraud transactions alone.

```
##       Time              V1                  V2                  V3
##  Min.   :     0   Min.   :-56.40751   Min.   :-72.71573   Min.   :-48.3256
##  1st Qu.: 54202   1st Qu.: -0.92037   1st Qu.: -0.59855   1st Qu.: -0.8904
##  Median : 84692   Median :  0.01811   Median :  0.06549   Median :  0.1799
##  Mean   : 94814   Mean   :  0.00000   Mean   :  0.00000   Mean   :  0.0000
##  3rd Qu.:139321   3rd Qu.:  1.31564   3rd Qu.:  0.80372   3rd Qu.:  1.0272
##  Max.   :172792   Max.   :  2.45493   Max.   : 22.05773   Max.   :  9.3826
##        V4                  V5                  V6                  V7
##  Min.   :-5.68317   Min.   :-113.74331   Min.   :-26.1605   Min.   :-43.5572
##  1st Qu.:-0.84864   1st Qu.:  -0.69160   1st Qu.: -0.7683   1st Qu.: -0.5541
##  Median :-0.01985   Median :  -0.05434   Median : -0.2742   Median :  0.0401
##  Mean   : 0.00000   Mean   :   0.00000   Mean   :  0.0000   Mean   :  0.0000
##  3rd Qu.: 0.74334   3rd Qu.:   0.61193   3rd Qu.:  0.3986   3rd Qu.:  0.5704
##  Max.   :16.87534   Max.   :  34.80167   Max.   : 73.3016   Max.   :120.5895
##        V8                  V9                 V10                 V11
##  Min.   :-73.21672   Min.   :-13.43407   Min.   :-24.58826   Min.   :-4.79747
##  1st Qu.: -0.20863   1st Qu.: -0.64310   1st Qu.: -0.53543   1st Qu.:-0.76249
##  Median :  0.02236   Median : -0.05143   Median : -0.09292   Median :-0.03276
##  Mean   :  0.00000   Mean   :  0.00000   Mean   :  0.00000   Mean   : 0.00000
##  3rd Qu.:  0.32735   3rd Qu.:  0.59714   3rd Qu.:  0.45392   3rd Qu.: 0.73959
##  Max.   : 20.00721   Max.   : 15.59500   Max.   : 23.74514   Max.   :12.01891
##       V12                 V13                 V14                 V15
##  Min.   :-18.6837   Min.   :-5.79188   Min.   :-19.2143   Min.   :-4.49894
```

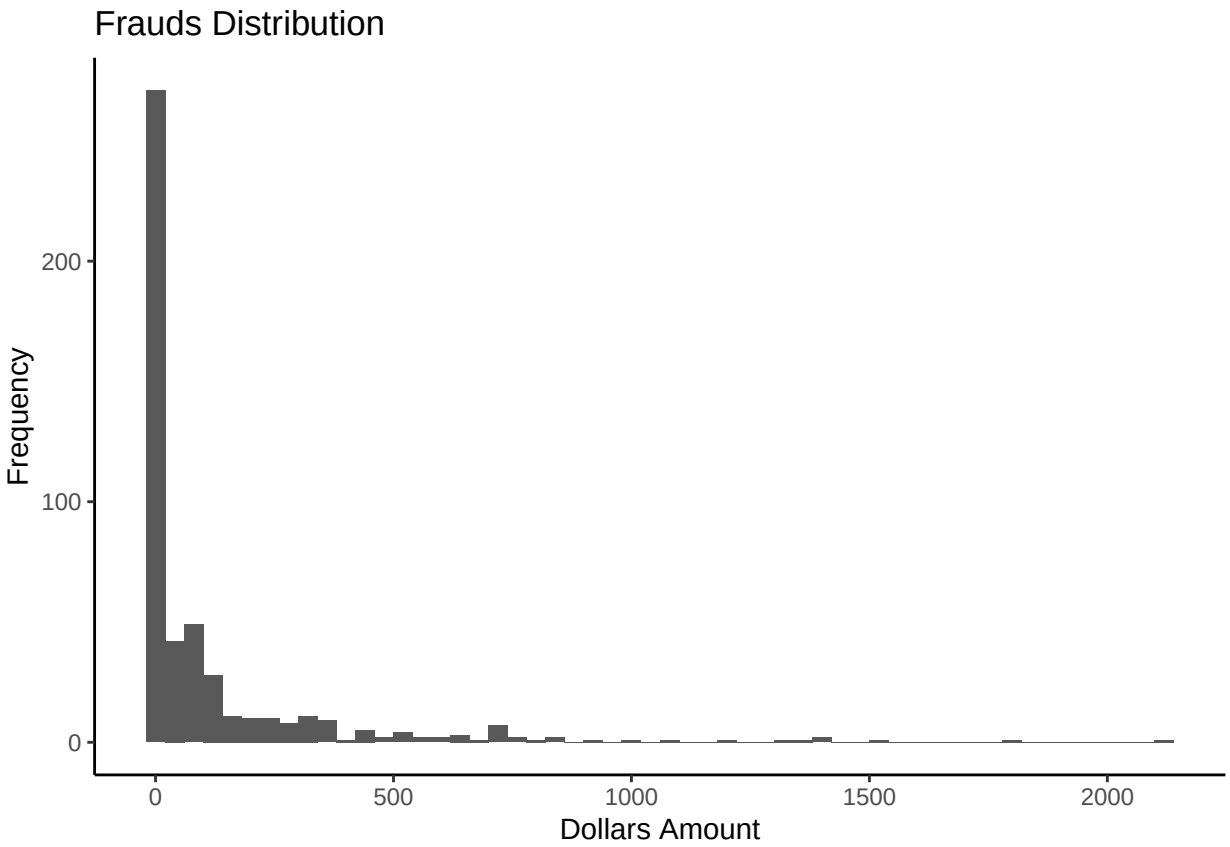
```
##  1st Qu.: -0.4056   1st Qu.:-0.64854   1st Qu.: -0.4256   1st Qu.:-0.58288
##  Median :  0.1400   Median :-0.01357   Median :  0.0506   Median : 0.04807
##  Mean   :  0.0000   Mean   : 0.00000   Mean   :  0.0000   Mean   : 0.00000
##  3rd Qu.:  0.6182   3rd Qu.: 0.66251   3rd Qu.:  0.4931   3rd Qu.: 0.64882
##  Max.   :  7.8484   Max.   : 7.12688   Max.   : 10.5268   Max.   : 8.87774
##       V16                V17                 V18
##  Min.   :-14.12985   Min.   :-25.16280   Min.   :-9.498746
##  1st Qu.: -0.46804   1st Qu.: -0.48375   1st Qu.:-0.498850
##  Median :  0.06641   Median : -0.06568   Median :-0.003636
##  Mean   :  0.00000   Mean   :  0.00000   Mean   : 0.000000
##  3rd Qu.:  0.52330   3rd Qu.:  0.39968   3rd Qu.: 0.500807
##  Max.   : 17.31511   Max.   :  9.25353   Max.   : 5.041069
##       V19                V20                 V21
##  Min.   :-7.213527   Min.   :-54.49772   Min.   :-34.83038
##  1st Qu.:-0.456299   1st Qu.: -0.21172   1st Qu.: -0.22839
##  Median : 0.003735   Median : -0.06248   Median : -0.02945
##  Mean   : 0.000000   Mean   :  0.00000   Mean   :  0.00000
##  3rd Qu.: 0.458949   3rd Qu.:  0.13304   3rd Qu.:  0.18638
##  Max.   : 5.591971   Max.   : 39.42090   Max.   : 27.20284
##       V22                 V23                 V24
##  Min.   :-10.933144   Min.   :-44.80774   Min.   :-2.83663
##  1st Qu.: -0.542350   1st Qu.: -0.16185   1st Qu.:-0.35459
##  Median :  0.006782   Median : -0.01119   Median : 0.04098
##  Mean   :  0.000000   Mean   :  0.00000   Mean   : 0.00000
##  3rd Qu.:  0.528554   3rd Qu.:  0.14764   3rd Qu.: 0.43953
##  Max.   : 10.503090   Max.   : 22.52841   Max.   : 4.58455
##       V25                V26                V27
##  Min.   :-10.29540   Min.   :-2.60455   Min.   :-22.565679
##  1st Qu.: -0.31715   1st Qu.:-0.32698   1st Qu.: -0.070840
##  Median :  0.01659   Median :-0.05214   Median :  0.001342
##  Mean   :  0.00000   Mean   : 0.00000   Mean   :  0.000000
##  3rd Qu.:  0.35072   3rd Qu.: 0.24095   3rd Qu.:  0.091045
##  Max.   :  7.51959   Max.   : 3.51735   Max.   : 31.612198
##       V28                Amount              Class
##  Min.   :-15.43008   Min.   :    0.00   Min.   :0.000000
##  1st Qu.: -0.05296   1st Qu.:    5.60   1st Qu.:0.000000
##  Median :  0.01124   Median :   22.00   Median :0.000000
##  Mean   :  0.00000   Mean   :   88.35   Mean   :0.001728
##  3rd Qu.:  0.07828   3rd Qu.:   77.17   3rd Qu.:0.000000
##  Max.   : 33.84781   Max.   :25691.16   Max.   :1.000000
```

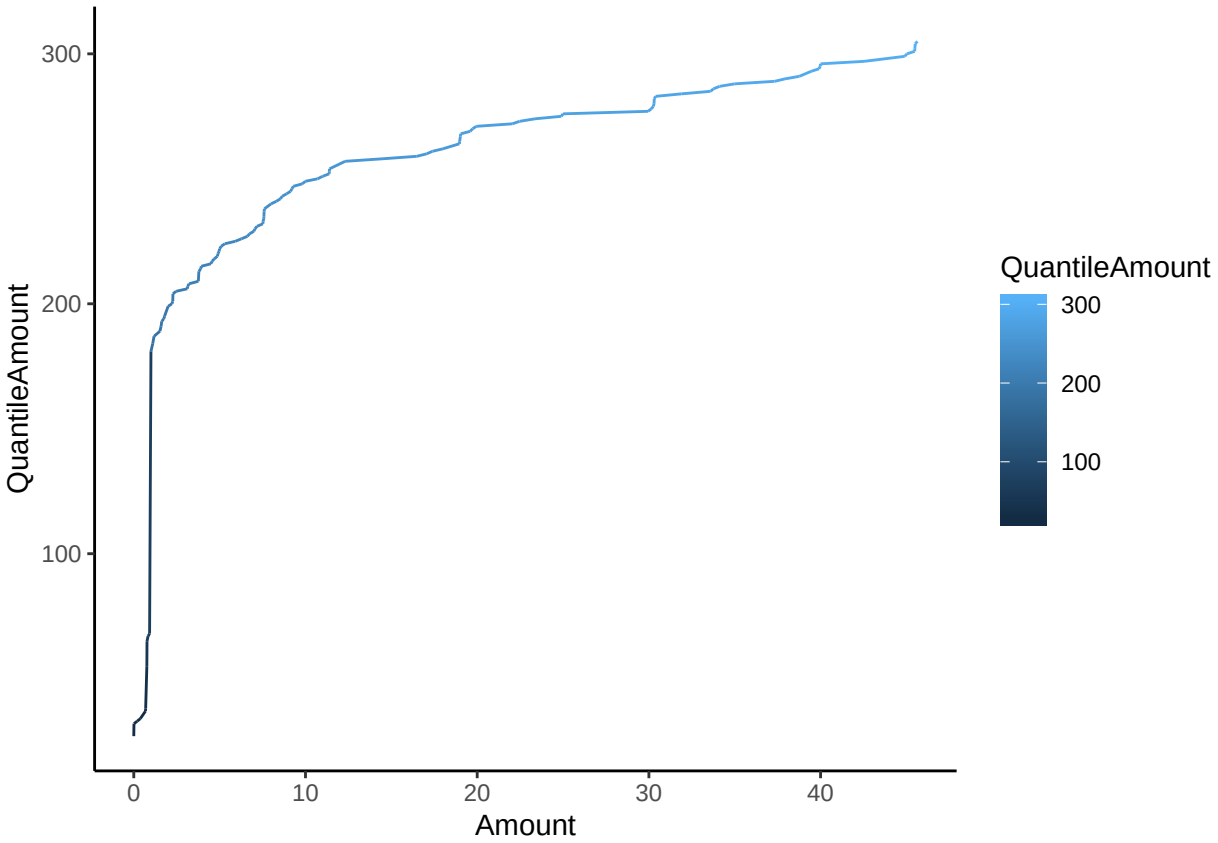
By analyzing only the amount of the fraud transactions, we can observe that the total amount sums 60127 and the mean is 122.2113.

```
## [1] 60127.97
```

```
## [1] 122.2113
```

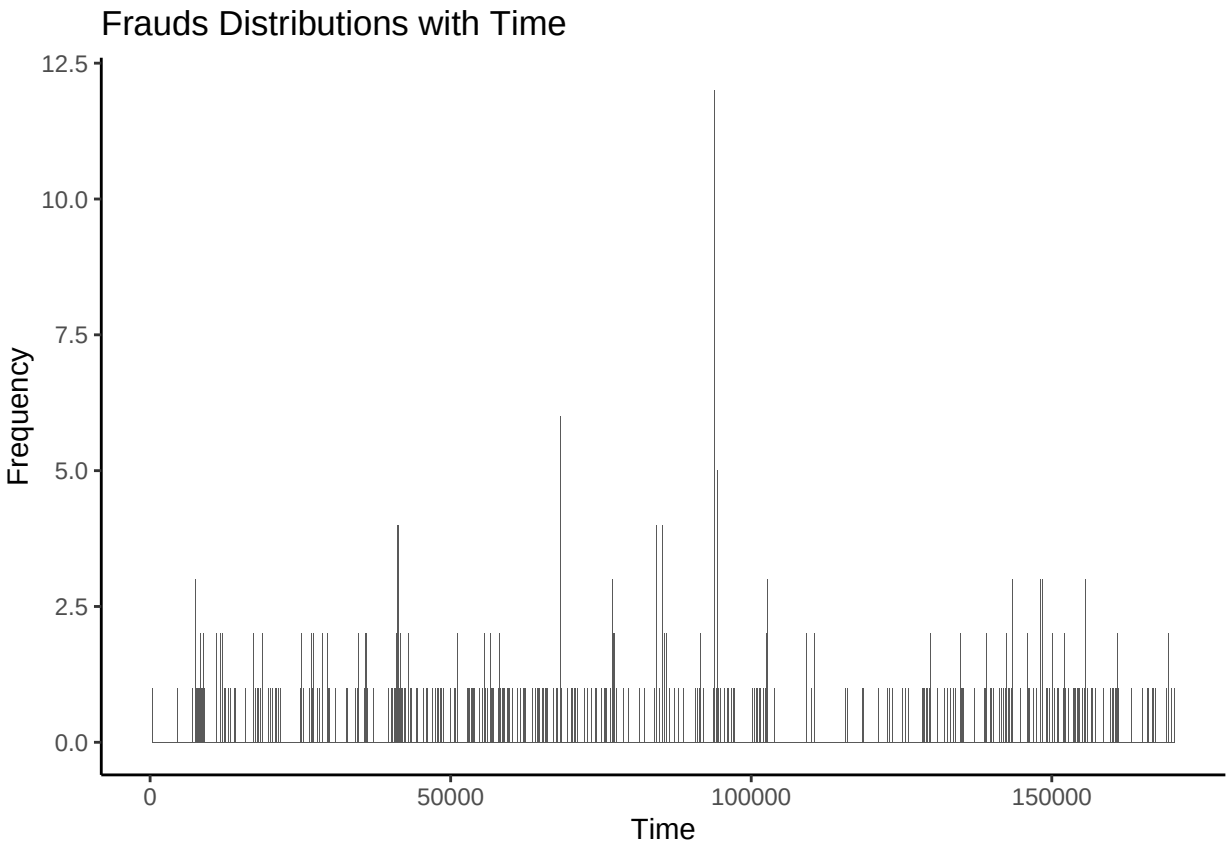
Now, using only the fraud data, we can observe how many times the same amount of money was retired. Making the distribution of the transaction´s amount, we can observe that small amounts of money are more likely to be a fraud. Most of the frauds are in the first quantile as well. Below are the top 10 amounts that were a fraud. One dollar is most likely to be a fraud.


Frauds Distribution
Frequency
200
100
0
0
500
1000
1500
2000
Dollars Amount

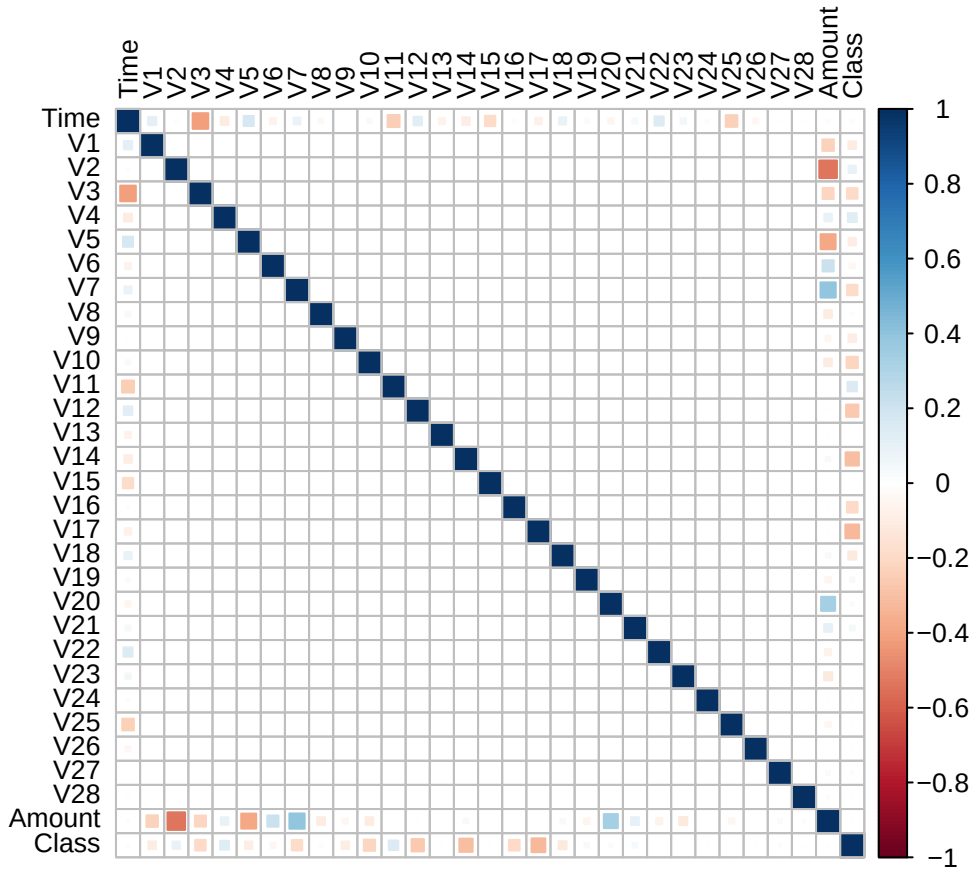


Amount	count
1.00	113
0.00	27
99.99	27
0.76	17
0.77	10
0.01	5
2.00	4
3.79	4
0.68	3
1.10	3

Now, analyzing the frauds data only, we will plot a frequency distribution to see if there is a correlation between frauds and time. Watching the correlation matrix below, it is obvious that fraud does not have a correlation with time.



Below is the correlation matrix of all of the variables. The amout variable is more correlated with V7 and V20, but there is not a significant relationship with the variables.



3. Results

Before starting to build models, the class of the credit cards has to be converted to factors in order to get the results. After that, I generated a sequence of random numbers by setting a seed. The data will be split in a train set and in a test set. The proportion of the sets is 70/30. Four models will be used: Naive Baseline Model, Logistic Regression Model, Decision Tree Model, and Random Forest Model.

Naive Baseline Model

The RMSE of the naive baseline model is 256.4233. The Naive model is the most simple model used since it is based on the mean. The RMSE is a prediction based on the average. The score of 256.4233 is huge, therefore it cannot indicate the performance well. For this reason, it does not make sense to analyze further the model. The other models will use accuracy as a measure of performance.

```
##                   model     RMSE
## 1 Naive Baseline Model 256.4223
```

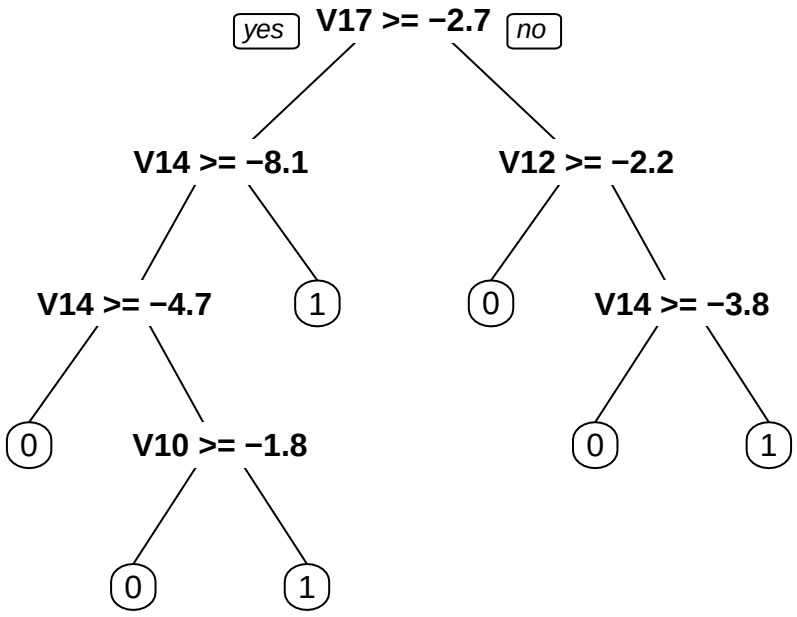
Logistic Regression Model

The logistic regression model estimates the probability of using a cumulative logistic distribution function. This model has an accuracy of 99.900518%. The accuracy is calculated using the formula: (TP+TN)/(TP+TN+FP+FN). The results are shown in the correlation matrix.

```
##
##      FALSE  TRUE
##   0 85279    16
##   1    69    79
```

Decision Tree Model

The decision tree model makest the best split of the credit card using nodes which leads to a lower error rate. This model has an accuracy of 99.925096%. The accuracy of this model is better than the previous one, but the model can still be improved.



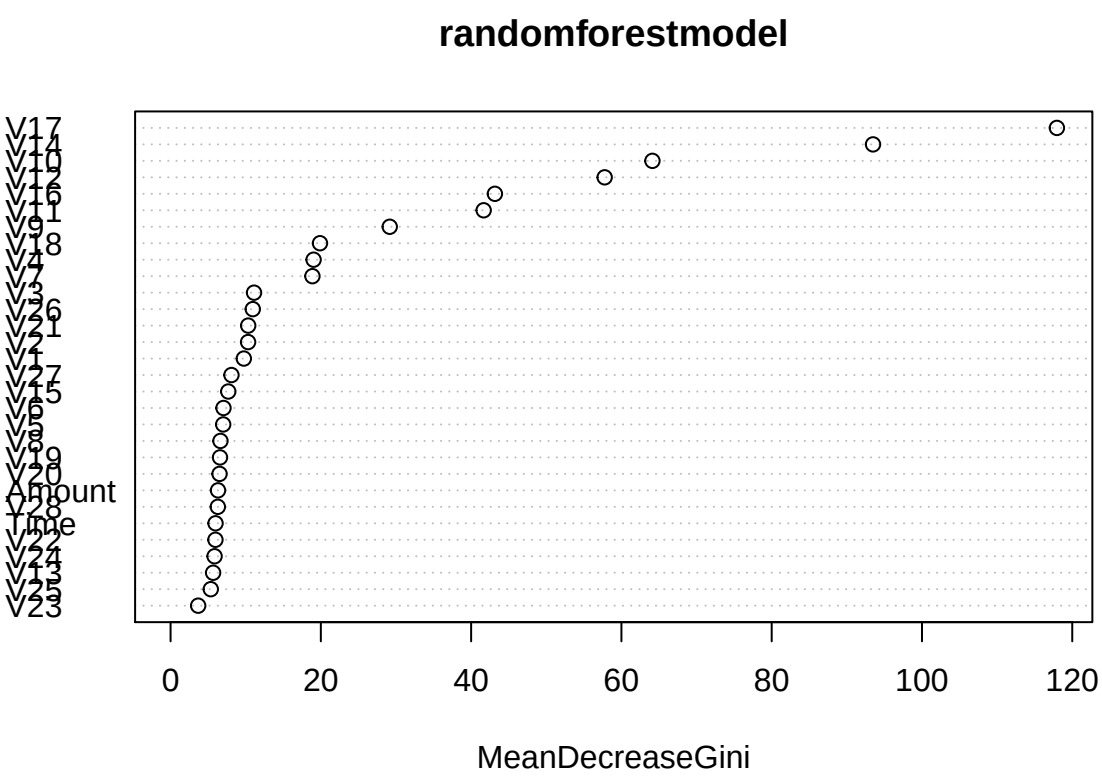

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 85275    20
##          1    44   104
##
##                Accuracy : 0.9993
##                  95% CI : (0.999, 0.9994)
##     No Information Rate : 0.9985
##     P-Value [Acc > NIR] : 2.098e-09
##
##                   Kappa : 0.7643
##
```

```
##  Mcnemar's Test P-Value : 0.00404
##
##             Sensitivity : 0.9995
##             Specificity : 0.8387
##          Pos Pred Value : 0.9998
##          Neg Pred Value : 0.7027
##              Prevalence : 0.9985
##          Detection Rate : 0.9980
##    Detection Prevalence : 0.9983
##       Balanced Accuracy : 0.9191
##
##        'Positive' Class : 0
##
```

Random Forest Model

In order to use the random forest model, it is necessary to build another sequence of random numbers to build multiple decision trees and averages the results. Usually, the random forest uses 500 trees and 3 nodes, but in this case, due to the limitations of my laptop for the amount of the data, 100 trees, and 5 nodes will be used in the model. This model has an accuracy of 99.956692%. For instance, the number of true positives and true negatives are greater than the decision tree model. The model improved just a few compared to the decision tree model, but it was enough to reach the goal of getting an accuracy greater than 99.95%

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 85287     8
##          1    36   112
##
##                Accuracy : 0.9995
##                  95% CI : (0.9993, 0.9996)
##     No Information Rate : 0.9986
##     P-Value [Acc > NIR] : 1.332e-15
##
##                   Kappa : 0.8356
##
##  Mcnemar's Test P-Value : 4.693e-05
##
##             Sensitivity : 0.9996
##             Specificity : 0.9333
##          Pos Pred Value : 0.9999
##          Neg Pred Value : 0.7568
##              Prevalence : 0.9986
##          Detection Rate : 0.9982
##    Detection Prevalence : 0.9983
##       Balanced Accuracy : 0.9665
##
##        'Positive' Class : 0
##
```



4. Conclusion

The goal of this project is to create a model that can examine the transactions made by the users and split them into two categories: fraud and legal transactions. This goal of this model is to achieve an accuracy rate above 99.95% using a machine learning model. After analyzing the four models above, the best model is the random forest model. It has an accuracy rate of 99.956692% which satisfies our goal and achieves the desired performance.

There is a couple of limitations with the data as describes before. Because of the privacy of the users, the variables V1 to V28 are unknown. For instance, I could have made a better analysis of data if I knew what each variable was. At the same time, a huge limitation that I had was my laptop. Unfortunately, I have a core I3 laptop with a low RAM capacity. Every time I had to run the model it took me more than 20 minutes. At the same time, when I run the random forest model with 500 trees and 3 nodes, the computer stop running so I had to decrease it to 100 trees and 3 nodes to analyze the results. It could be possible that by increasing the number of trees and nodes, the model could reach an accuracy level of 100%. Good future work would be to make the same model, but this time more decision trees and nodes and aim for an accuracy level of 100%. By having an accuracy level of 100%, the system would run perfectly and it is a great model to impress local banks since it is based in real transactions.