Credit Card Data Modelling

Shaik

2022-10-05

# Part 1

Attacks on client credit cards represents a form of fraud that targets the client back accounts that replace the direct attack on banks. The use of credit cards allowed for fraudsters to be able to easier target specific individual bank accounts in ways that were not possible when the bank accounts are remotely targeted as a whole. The fraudsters make payments from the victims credit cards in transactions to other accounts, with the client only realizing the theft later. Through observation of various aspects relating to the client and their transaction history a fraudulent transaction can be identified. In this case, the aim would be the designing of a Decision Tree model for predicting whether a transaction is fraudulent or not. This would allow for the transaction to be flagged and withheld pending confirmation from the client.

# Part 2

The credit card data was used in the modelling for this case and was collected from <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>.

# Part 3

The code for the importation of the data is as presented below:

#Loading Data  
creditcard = read\_csv("creditcard.csv")

## Rows: 284807 Columns: 31  
## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## dbl (31): Time, V1, V2, V3, V4, V5, V6, V7, V8, V9, V10, V11, V12, V13, V14,...  
##   
## i Use `spec()` to retrieve the full column specification for this data.  
## i Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

# Part 4

The coding for getting the overview and summary of the data is as give below. The overview of the data showed that the data had 31 columns and 284,807 rows. Table 1 below shows that none of the variables in the data had any null entries. The summary statistics for mean, standard deviation, minimum and maximum statistics are presented in Table 2 below for the numeric variables. Additionally, the summary statistics for the frequency of each of the two categories under the Class variables is computed giving 0 with 284,315 observations and 1 with 492 observations.

#Data Summary  
#Overall  
nrow(creditcard)

## [1] 284807

ncol(creditcard)

## [1] 31

status.summary = df\_status(creditcard)

## variable q\_zeros p\_zeros q\_na p\_na q\_inf p\_inf type unique  
## 1 Time 2 0.00 0 0 0 0 numeric 124592  
## 2 V1 0 0.00 0 0 0 0 numeric 275663  
## 3 V2 0 0.00 0 0 0 0 numeric 275663  
## 4 V3 0 0.00 0 0 0 0 numeric 275663  
## 5 V4 0 0.00 0 0 0 0 numeric 275663  
## 6 V5 0 0.00 0 0 0 0 numeric 275663  
## 7 V6 0 0.00 0 0 0 0 numeric 275663  
## 8 V7 0 0.00 0 0 0 0 numeric 275663  
## 9 V8 0 0.00 0 0 0 0 numeric 275663  
## 10 V9 0 0.00 0 0 0 0 numeric 275663  
## 11 V10 0 0.00 0 0 0 0 numeric 275663  
## 12 V11 0 0.00 0 0 0 0 numeric 275663  
## 13 V12 0 0.00 0 0 0 0 numeric 275663  
## 14 V13 0 0.00 0 0 0 0 numeric 275663  
## 15 V14 0 0.00 0 0 0 0 numeric 275663  
## 16 V15 0 0.00 0 0 0 0 numeric 275663  
## 17 V16 0 0.00 0 0 0 0 numeric 275663  
## 18 V17 0 0.00 0 0 0 0 numeric 275663  
## 19 V18 0 0.00 0 0 0 0 numeric 275663  
## 20 V19 0 0.00 0 0 0 0 numeric 275663  
## 21 V20 0 0.00 0 0 0 0 numeric 275663  
## 22 V21 0 0.00 0 0 0 0 numeric 275663  
## 23 V22 0 0.00 0 0 0 0 numeric 275663  
## 24 V23 0 0.00 0 0 0 0 numeric 275663  
## 25 V24 0 0.00 0 0 0 0 numeric 275663  
## 26 V25 0 0.00 0 0 0 0 numeric 275663  
## 27 V26 0 0.00 0 0 0 0 numeric 275663  
## 28 V27 0 0.00 0 0 0 0 numeric 275663  
## 29 V28 0 0.00 0 0 0 0 numeric 275663  
## 30 Amount 1825 0.64 0 0 0 0 numeric 32767  
## 31 Class 284315 99.83 0 0 0 0 numeric 2

kable(status.summary, caption = "Table 1")

Table 1

| variable | q\_zeros | p\_zeros | q\_na | p\_na | q\_inf | p\_inf | type | unique |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Time | 2 | 0.00 | 0 | 0 | 0 | 0 | numeric | 124592 |
| V1 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| V2 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| V3 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| V4 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| V5 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| V6 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| V7 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| V8 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| V9 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| V10 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| V11 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| V12 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| V13 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| V14 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| V15 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| V16 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| V17 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| V18 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| V19 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| V20 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| V21 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| V22 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| V23 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| V24 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| V25 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| V26 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| V27 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| V28 | 0 | 0.00 | 0 | 0 | 0 | 0 | numeric | 275663 |
| Amount | 1825 | 0.64 | 0 | 0 | 0 | 0 | numeric | 32767 |
| Class | 284315 | 99.83 | 0 | 0 | 0 | 0 | numeric | 2 |

#Summary Statistics Numeric  
summary.data = rbind(creditcard %>% select(-Class) %>% summarise\_all(mean),   
 creditcard %>% select(-Class) %>% summarise\_all(sd),  
 creditcard %>% select(-Class) %>% summarise\_all(max),   
 creditcard %>% select(-Class) %>% summarise\_all(min))  
t.summary.data = t(summary.data)  
t.summary.data = data.frame(t.summary.data)  
colnames(t.summary.data) = c("mean", "sd", "max", "min")  
kable(t.summary.data, caption = "Table 2")

Table 2

|  | mean | sd | max | min |
| --- | --- | --- | --- | --- |
| Time | 94813.85958 | 4.748815e+04 | 1.727920e+05 | 0.000000 |
| V1 | 0.00000 | 1.958696e+00 | 2.454930e+00 | -56.407510 |
| V2 | 0.00000 | 1.651309e+00 | 2.205773e+01 | -72.715728 |
| V3 | 0.00000 | 1.516255e+00 | 9.382558e+00 | -48.325589 |
| V4 | 0.00000 | 1.415869e+00 | 1.687534e+01 | -5.683171 |
| V5 | 0.00000 | 1.380247e+00 | 3.480167e+01 | -113.743307 |
| V6 | 0.00000 | 1.332271e+00 | 7.330163e+01 | -26.160506 |
| V7 | 0.00000 | 1.237094e+00 | 1.205895e+02 | -43.557242 |
| V8 | 0.00000 | 1.194353e+00 | 2.000721e+01 | -73.216718 |
| V9 | 0.00000 | 1.098632e+00 | 1.559499e+01 | -13.434066 |
| V10 | 0.00000 | 1.088850e+00 | 2.374514e+01 | -24.588262 |
| V11 | 0.00000 | 1.020713e+00 | 1.201891e+01 | -4.797473 |
| V12 | 0.00000 | 9.992014e-01 | 7.848392e+00 | -18.683715 |
| V13 | 0.00000 | 9.952742e-01 | 7.126883e+00 | -5.791881 |
| V14 | 0.00000 | 9.585956e-01 | 1.052677e+01 | -19.214326 |
| V15 | 0.00000 | 9.153160e-01 | 8.877742e+00 | -4.498945 |
| V16 | 0.00000 | 8.762529e-01 | 1.731511e+01 | -14.129855 |
| V17 | 0.00000 | 8.493371e-01 | 9.253526e+00 | -25.162799 |
| V18 | 0.00000 | 8.381762e-01 | 5.041069e+00 | -9.498746 |
| V19 | 0.00000 | 8.140405e-01 | 5.591971e+00 | -7.213527 |
| V20 | 0.00000 | 7.709250e-01 | 3.942090e+01 | -54.497720 |
| V21 | 0.00000 | 7.345240e-01 | 2.720284e+01 | -34.830382 |
| V22 | 0.00000 | 7.257016e-01 | 1.050309e+01 | -10.933144 |
| V23 | 0.00000 | 6.244603e-01 | 2.252841e+01 | -44.807735 |
| V24 | 0.00000 | 6.056471e-01 | 4.584549e+00 | -2.836627 |
| V25 | 0.00000 | 5.212781e-01 | 7.519589e+00 | -10.295397 |
| V26 | 0.00000 | 4.822270e-01 | 3.517346e+00 | -2.604551 |
| V27 | 0.00000 | 4.036325e-01 | 3.161220e+01 | -22.565679 |
| V28 | 0.00000 | 3.300833e-01 | 3.384781e+01 | -15.430084 |
| Amount | 88.34962 | 2.501201e+02 | 2.569116e+04 | 0.000000 |

#Summary Statistics Categorical  
kable(table(creditcard$Class), caption = "Table 3")

Table 3

| Var1 | Freq |
| --- | --- |
| 0 | 284315 |
| 1 | 492 |

# Part 5

The data preparation for the credit card data in this case only involved the changing of the variable type for the categorical variable class. This variable gives information on whether a transaction was noted as fraudulent = 1, or not = 0. However, Table 1 above the variable type for the Class variable is indicated as numeric. Below the code was used in converting the variable type from numeric to factor to enable use in the Decision Tree modelling.

#Data Preparation   
#Setting the Class variable as a factor  
creditcard$Class = factor(creditcard$Class)

# Part 6

The modelling approach for the credit card fraud data applied in this case is the Decision Tree modelling. The model utilizes a tree like structure in the prediction of the possible outcome. The modelling process used the entire data set as the training data for the model. All the variables in the data were used for the modelling the outcome variable, Class. The model was then potted into a tree for the visualization. Finally, the model accuracy was evaluated with the new data set being the original data.

# Part 7 and Part 8

The visual output of the Decision Tree modelling is as presented below in Figure 1. From the model at the top level node 1 the transactions are separated based on the V17 variable into those where V17 is less than or equal to -1.35 and those that are greater than -1.35. On the second level of tree, there are two node for V12 and V14. At this level the transactions that have V17 less than or equal to -1.35 are further divided into 2 groups based on V12 for those with V12 less than or equal to -2.21 and those where v12 is greater than -2.21. On the other hand, at the same level, the transactions that have V17 greater than -1.35 are also further divided into 2 groups based on V14 for those with V14 less than or equal to -2.602 and those where v12 is greater than -2.602. This classification process continues down the tree until the final classification for the Class, as either 0 or 1.

The results below also show the confusion matrix for the predictive performance of the model. The output from the confusion matrix shows that the prediction accuracy of the Decision Tree model 99.96%. This implies that the model correctly predicted the Class level for 99.96% of the transactions is the same data used in the training. Based on the training data, the model has a good predictive performance.

#Modelling   
Decision.Tree = ctree(Class~., data = creditcard)  
plot(Decision.Tree)

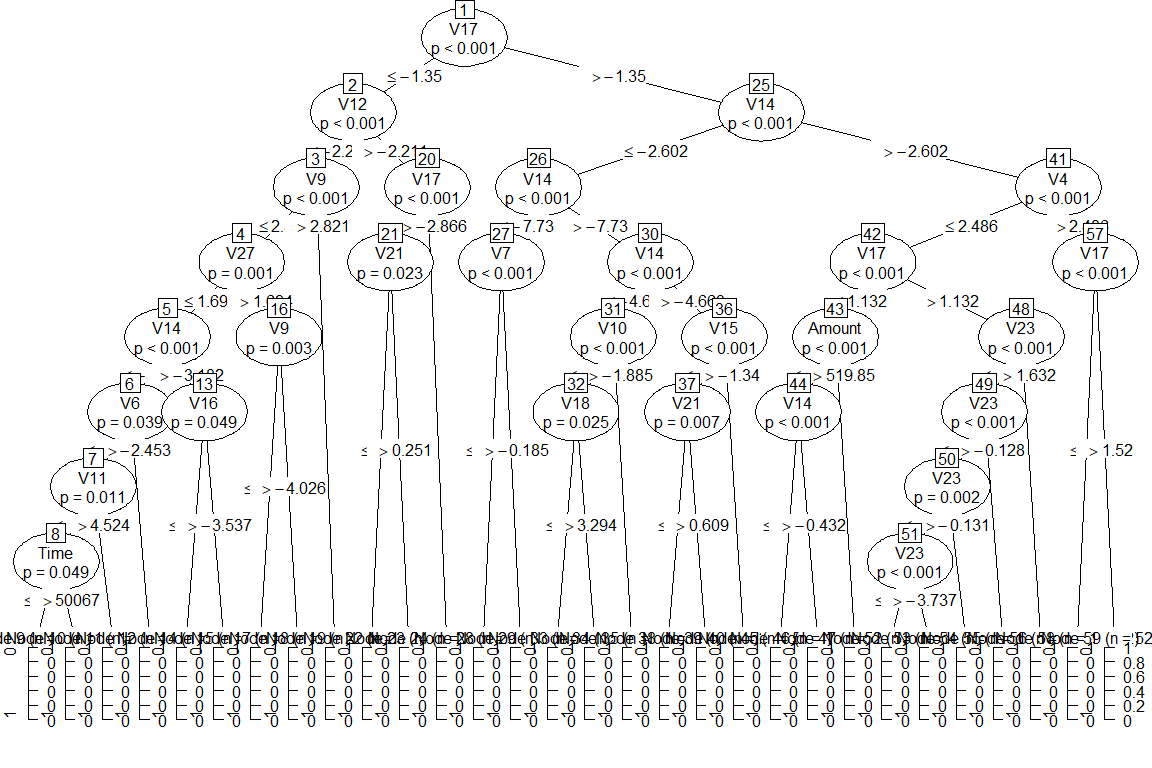


Figure 1

Decision.Tree[1]

##   
## Model formula:  
## Class ~ Time + V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10 +   
## V11 + V12 + V13 + V14 + V15 + V16 + V17 + V18 + V19 + V20 +   
## V21 + V22 + V23 + V24 + V25 + V26 + V27 + V28 + Amount  
##   
## Fitted party:  
## [1] root  
## | [2] V17 <= -1.34996  
## | | [3] V12 <= -2.21069  
## | | | [4] V9 <= 2.82083  
## | | | | [5] V27 <= 1.69361  
## | | | | | [6] V14 <= -3.18217  
## | | | | | | [7] V6 <= -2.45324  
## | | | | | | | [8] V11 <= 4.52432  
## | | | | | | | | [9] Time <= 50067: 0 (n = 23, err = 34.8%)  
## | | | | | | | | [10] Time > 50067: 1 (n = 22, err = 9.1%)  
## | | | | | | | [11] V11 > 4.52432: 1 (n = 63, err = 1.6%)  
## | | | | | | [12] V6 > -2.45324: 1 (n = 240, err = 3.8%)  
## | | | | | [13] V14 > -3.18217  
## | | | | | | [14] V16 <= -3.53706: 1 (n = 12, err = 25.0%)  
## | | | | | | [15] V16 > -3.53706: 0 (n = 31, err = 22.6%)  
## | | | | [16] V27 > 1.69361  
## | | | | | [17] V9 <= -4.02618: 1 (n = 21, err = 0.0%)  
## | | | | | [18] V9 > -4.02618: 0 (n = 30, err = 23.3%)  
## | | | [19] V9 > 2.82083: 0 (n = 17, err = 0.0%)  
## | | [20] V12 > -2.21069  
## | | | [21] V17 <= -2.86575  
## | | | | [22] V21 <= 0.25095: 0 (n = 22, err = 0.0%)  
## | | | | [23] V21 > 0.25095: 0 (n = 7, err = 28.6%)  
## | | | [24] V17 > -2.86575: 0 (n = 2361, err = 0.1%)  
## | [25] V17 > -1.34996  
## | | [26] V14 <= -2.60192  
## | | | [27] V14 <= -7.72975  
## | | | | [28] V7 <= -0.18471: 1 (n = 20, err = 0.0%)  
## | | | | [29] V7 > -0.18471: 0 (n = 9, err = 0.0%)  
## | | | [30] V14 > -7.72975  
## | | | | [31] V14 <= -4.66258  
## | | | | | [32] V10 <= -1.88484  
## | | | | | | [33] V18 <= 3.29369: 1 (n = 28, err = 21.4%)  
## | | | | | | [34] V18 > 3.29369: 0 (n = 9, err = 11.1%)  
## | | | | | [35] V10 > -1.88484: 0 (n = 138, err = 2.2%)  
## | | | | [36] V14 > -4.66258  
## | | | | | [37] V15 <= -1.34018  
## | | | | | | [38] V21 <= 0.60915: 0 (n = 125, err = 2.4%)  
## | | | | | | [39] V21 > 0.60915: 0 (n = 7, err = 14.3%)  
## | | | | | [40] V15 > -1.34018: 0 (n = 2485, err = 0.2%)  
## | | [41] V14 > -2.60192  
## | | | [42] V4 <= 2.48603  
## | | | | [43] V17 <= 1.13215  
## | | | | | [44] Amount <= 519.85  
## | | | | | | [45] V14 <= -0.43169: 0 (n = 54151, err = 0.0%)  
## | | | | | | [46] V14 > -0.43169: 0 (n = 188523, err = 0.0%)  
## | | | | | [47] Amount > 519.85: 0 (n = 7219, err = 0.1%)  
## | | | | [48] V17 > 1.13215  
## | | | | | [49] V23 <= 1.63243  
## | | | | | | [50] V23 <= -0.12843  
## | | | | | | | [51] V23 <= -0.13099  
## | | | | | | | | [52] V23 <= -3.73685: 0 (n = 40, err = 2.5%)  
## | | | | | | | | [53] V23 > -3.73685: 0 (n = 3893, err = 0.2%)  
## | | | | | | | [54] V23 > -0.13099: 0 (n = 40, err = 2.5%)  
## | | | | | | [55] V23 > -0.12843: 0 (n = 11007, err = 0.0%)  
## | | | | | [56] V23 > 1.63243: 0 (n = 218, err = 0.9%)  
## | | | [57] V4 > 2.48603  
## | | | | [58] V17 <= 1.51993: 0 (n = 13522, err = 0.1%)  
## | | | | [59] V17 > 1.51993: 0 (n = 524, err = 2.1%)  
##   
## Number of inner nodes: 29  
## Number of terminal nodes: 30

#Model Accuracy  
Predict.data = predict(Decision.Tree, newdata = creditcard[,-31], type = "prob")  
Predict.data = data.frame(Predict.data)  
Predict.data = Predict.data %>% mutate(Prediction = ifelse(X0 > X1, 0, 1))  
Predict.data$Prediction = factor(Predict.data$Prediction)  
#Checking model accuracy   
confusionMatrix(table(Predict.data$Prediction, creditcard$Class), positive = "1")

## Confusion Matrix and Statistics  
##   
##   
## 0 1  
## 0 284294 107  
## 1 21 385  
##   
## Accuracy : 0.9996   
## 95% CI : (0.9995, 0.9996)  
## No Information Rate : 0.9983   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.8572   
##   
## Mcnemar's Test P-Value : 5.778e-14   
##   
## Sensitivity : 0.782520   
## Specificity : 0.999926   
## Pos Pred Value : 0.948276   
## Neg Pred Value : 0.999624   
## Prevalence : 0.001727   
## Detection Rate : 0.001352   
## Detection Prevalence : 0.001426   
## Balanced Accuracy : 0.891223   
##   
## 'Positive' Class : 1   
##