Credit Card Default Research

by Sumaira Afzal, Viraja Ketkar, Murlidhar Loka, Vadim Spirkov

Abstract Credit card default might very well be a life altering event. It happens when a client have become severely delinquent on his/her credit card payment. It's a serious credit card status that not only affects person's standing with that credit card issuer, but also individual's credit standing in general and his/her ability to get approved for credit cards, loans, and other credit-based services. This research will make yet another attempt to predict if a client goint to default on the next payment. Employing verious machine learning technique we also will make an attemt to estime the amount a client would be able to pay when the bill comes. The authors of this study will try to discover who is more likely to default on the payment.

Background

Overdepandance on credit card debt has been an ongoing theme in many countries around the word. For example US consumers started 2018 owing more than $1 trillion in credit card debt (Ref: Comoreanu) It is projected that by the end of 2019 US consumers will increase their collective debt by another 60 billion dollars. Unfortunately many consumers overestimate their ability to pay the debt on time, or the unforeseen circumstances and luck of savings make people default on their payments. This is the least desirable outcome for all parties. Unpaid debt leads, in most cases, to default on the whole outstanding balance causing financial loss for the credit institutions. Majority of the clients go through tremendous emotional and financial stress, risking their credibility. The financial institution make significant efforts to evaluate the prospective client ability to sustain the debt and pay in time to avoid the credit default.

Objective

This study pursues a few goals. First of all employing the client personal characteristics and the last six month payment history we would like to predict ax accurate as possible if the client makes the next month payment or defaults. We will employ a few supervised learning models to attack the problem.

Another objective is to understand which features of the data set have the most impact on the next payment success/ failure.

We are also motivated to unearth, if possible, any trend that might shed light on what make people to default on the payment. And lastly the authors of this study will try to estimate how much a client could pay when the next bill comes

Data Analysis

This research employs the data set sourced from UCI Machine Learning Repository. This real-life data comprises 30000 observations of the credit card payment history of Taiwanese consumers.

Data Dictionary

Column Name	Column Description
ID	Customer ID
LIMIT_BAL	Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit
SEX	Gender (1 = male; 2 = female).
EDUCATION	Education (1 = graduate school; 2 = university; 3 = high school; 4 = others)
MARRIAGE	Marital status (1 = married; 2 = single; 3 = divorced; 0 - other)
AGE	Age (year)

Column Name	Column Description
PAY_1	PAY_1 - PAY_6 are payment statuses over a course of the last six months, where -2: Balance paid in full and no transactions this period (we may refer to this credit card account as having been 'inactive' this period). -1: Balance paid in full, but account has a positive balance at end of period due to recent transactions for which payment has not yet come due; 0: Customer paid the minimum due amount, but not the entire balance. I.e., the customer paid enough for their account to remain in good standing, but did revolve a balance. Positive numbers denote payment delay in months. For example 1 = payment delay for one month; 2 = payment delay for two months; . . .; 9 = payment delay for nine months and above. PAY_1 - Payment status in September
PAY_2	Payment status in August
PAY_3	Payment status in July
PAY_4	Payment status in June
PAY_5	Payment status in May
PAY_6	Payment status in April
BILL_AMT1	BILL_AMT1 - BILL_AMT6 are bill amounts (NT dollar) from April till September. BILL_AMT1: September bill
BILL_AMT2	August bill
BILL_AMT3	July bill
BILL_AMT4	June bill
BILL_AMT5	May bill
BILL_AMT6	April bill
PAY_AMT1	Amount of previous payment (NT dollar). PAY_AMT1: paid in September (August bill)
PAY_AMT2	Amount paid in August (July bill)
PAY_AMT3	Amount paid in July (June bill)
PAY_AMT4	Amount paid in June (May bill)
PAY_AMT5	Amount paid in May (April bill)
PAY_AMT6	Amount paid in April (March bill)
DEFAULT	**Target label that denotes whether the client paid the next month bill (0) or did not (1)**

Data Exploration

Feature Analytics

We start our research with the feature exploration and understanding.

Table 2: Credit Card Payment Data Summary

No	Variable	Stats / Values	Freqs (% of Valid)	Missing
1	ID [integer]	Mean (sd) : 15000.5 (8660.4) min < med < max: 1 < 15000.5 < 30000 IQR (CV) : 14999.5 (0.6)	30000 distinct values (Integer sequence)	0 (0%)
2	LIMIT_BAL [integer]	Mean (sd) : 167484.3 (129747.7) min < med < max: 10000 < 140000 < 1e+06 IQR (CV) : 190000 (0.8)	81 distinct values	0 (0%)
3	SEX [integer]	Min : 1 Mean : 1.6 Max : 2	1 : 11888 (39.6%) 2 : 18112 (60.4%)	0 (0%)
4	EDUCATION [integer]	Mean (sd) : 1.9 (0.8) min < med < max: 0 < 2 < 6 IQR (CV) : 1 (0.4)	7 distinct values	0 (0%)
5	MARRIAGE [integer]	Mean (sd) : 1.6 (0.5) min < med < max: 0 < 2 < 3 IQR (CV) : 1 (0.3)	4 distinct values	0 (0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Missing
6	AGE [integer]	Mean (sd) : 35.5 (9.2) min < med < max: 21 < 34 < 79 IQR (CV) : 13 (0.3)	56 distinct values	0 (0%)
7	PAY_1 [integer]	Mean (sd) : 0 (1.1) min < med < max: -2 < 0 < 8 IQR (CV) : 1 (-67.3)	11 distinct values	0 (0%)
8	PAY_2 [integer]	Mean (sd) : -0.1 (1.2) min < med < max: -2 < 0 < 8 IQR (CV) : 1 (-8.9)	11 distinct values	0 (0%)
9	PAY_3 [integer]	Mean (sd) : -0.2 (1.2) min < med < max: -2 < 0 < 8 IQR (CV) : 1 (-7.2)	11 distinct values	0 (0%)
10	PAY_4 [integer]	Mean (sd) : -0.2 (1.2) min < med < max: -2 < 0 < 8 IQR (CV) : 1 (-5.3)	11 distinct values	0 (0%)
11	PAY_5 [integer]	Mean (sd) : -0.3 (1.1) min < med < max: -2 < 0 < 8 IQR (CV) : 1 (-4.3)	10 distinct values	0 (0%)
12	PAY_6 [integer]	Mean (sd) : -0.3 (1.1) min < med < max: -2 < 0 < 8 IQR (CV) : 1 (-4)	10 distinct values	0 (0%)
13	BILL_AMT1 [integer]	Mean (sd) : 51223.3 (73635.9) min < med < max: -165580 < 22381.5 < 964511 IQR (CV) : 63532.2 (1.4)	22723 distinct values	0 (0%)
14	BILL_AMT2 [integer]	Mean (sd) : 49179.1 (71173.8) min < med < max: -69777 < 21200 < 983931 IQR (CV) : 61021.5 (1.4)	22346 distinct values	0 (0%)
15	BILL_AMT3 [integer]	Mean (sd) : 47013.2 (69349.4) min < med < max: -157264 < 20088.5 < 1664089 IQR (CV) : 57498.5 (1.5)	22026 distinct values	0 (0%)
16	BILL_AMT4 [integer]	Mean (sd) : 43262.9 (64332.9) min < med < max: -170000 < 19052 < 891586 IQR (CV) : 52179.2 (1.5)	21548 distinct values	0 (0%)
17	BILL_AMT5 [integer]	Mean (sd) : 40311.4 (60797.2) min < med < max: -81334 < 18104.5 < 927171 IQR (CV) : 48427.5 (1.5)	21010 distinct values	0 (0%)
18	BILL_AMT6 [integer]	Mean (sd) : 38871.8 (59554.1) min < med < max: -339603 < 17071 < 961664 IQR (CV) : 47942.2 (1.5)	20604 distinct values	0 (0%)
19	PAY_AMT1 [integer]	Mean (sd) : 5663.6 (16563.3) min < med < max: 0 < 2100 < 873552 IQR (CV) : 4006 (2.9)	7943 distinct values	0 (0%)
20	PAY_AMT2 [integer]	Mean (sd) : 5921.2 (23040.9) min < med < max: 0 < 2009 < 1684259 IQR (CV) : 4167 (3.9)	7899 distinct values	0 (0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Missing
21	PAY_AMT3 [integer]	Mean (sd) : 5225.7 (17607) min < med < max: 0 < 1800 < 896040 IQR (CV) : 4115 (3.4)	7518 distinct values	0 (0%)
22	PAY_AMT4 [integer]	Mean (sd) : 4826.1 (15666.2) min < med < max: 0 < 1500 < 621000 IQR (CV) : 3717.2 (3.2)	6937 distinct values	0 (0%)
23	PAY_AMT5 [integer]	Mean (sd) : 4799.4 (15278.3) min < med < max: 0 < 1500 < 426529 IQR (CV) : 3779 (3.2)	6897 distinct values	0 (0%)
24	PAY_AMT6 [integer]	Mean (sd) : 5215.5 (17777.5) min < med < max: 0 < 1500 < 528666 IQR (CV) : 3882.2 (3.4)	6939 distinct values	0 (0%)
25	DEFAULT [integer]	Min : 0 Mean : 0.2 Max : 1	0 : 23364 (77.9%) 1 : 6636 (22.1%)	0 (0%)

Table 2 describes main statistical parameters of each column. It also outputs the values of the binary features. The first thing that jumps at us is that the data set has no missing data! We shall note that our target feature is not balanced. Almost **80%** of the clients do pay on time. Secondly female customers make **60%** of the data set. **Customer ID** column, as usual, will be dropped since it presents no analytical value. Here is a look at the data sample.

ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_1	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	BILL_AMT1	BILL_AMT2	BILL_AMT3
1	20000	2	2	1	24	2	2	-1	-1	-2	-2	3913	3102	689
2	120000	2	2	2	26	-1	2	0	0	0	2	2682	1725	2682
3	90000	2	2	2	34	0	0	0	0	0	0	29239	14027	13559
4	50000	2	2	1	37	0	0	0	0	0	0	46990	48233	49291
5	50000	1	2	1	57	-1	0	-1	0	0	0	8617	5670	35835
6	50000	1	1	2	37	0	0	0	0	0	0	64400	57069	57608
7	500000	1	1	2	29	0	0	0	0	0	0	367965	412023	445007
8	100000	2	2	2	23	0	-1	-1	0	0	-1	11876	380	601
9	140000	2	3	1	28	0	0	2	0	0	0	11285	14096	12108
10	20000	1	3	2	35	-2	-2	-2	-2	-1	-1	0	0	0
11	200000	2	3	2	34	0	0	2	0	0	-1	11073	9787	5535
12	260000	2	1	2	51	-1	-1	-1	-1	-1	2	12261	21670	9966

BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5
3913	3102	689	0	0	0	0	689	0	0	0
2682	1725	2682	3272	3455	3261	0	1000	1000	1000	0
29239	14027	13559	14331	14948	15549	1518	1500	1000	1000	1000
46990	48233	49291	28314	28959	29547	2000	2019	1200	1100	1069
8617	5670	35835	20940	19146	19131	2000	36681	10000	9000	689
64400	57069	57608	19394	19619	20024	2500	1815	657	1000	1000
367965	412023	445007	542653	483003	473944	55000	40000	38000	20239	13750
11876	380	601	221	-159	567	380	601	0	581	1687
11285	14096	12108	12211	11793	3719	3329	0	432	1000	1000
0	0	0	0	13007	13912	0	0	0	13007	1122
11073	9787	5535	2513	1828	3731	2306	12	50	300	3738
12261	21670	9966	8517	22287	13668	21818	9966	8583	22301	0

Table 3: Credit Card Payment Data Sample

Let's review demographic characteristics of the customer base, namely: *EDUCATION*, *MARITAL STATUS* and *AGE*. We immediately can observe some deficiencies in the data quality (Figure: 1). As we see the majority of the credit card holders have a university degree. There are three groups which are not supposed to be in the data set: **Unknown** - code **0**, **Unknown5** - code **5** and **Unkown6** - code **6**. We will assign these customers to the **Other** group, since the description for the aforementioned codes is not provided.

Number of single people is slightly higher than the number of the married ones.

Majority of the credit card holders are people between age of 25 and 50, which does not come as a surprise (Figure: 1)... Let's see if the *AGE* feature has outliers.

```
print(original %>% filter(AGE < 18 || AGE > 100) %>% summarise(COUNT = n()))
```

```
  COUNT
1     0
```

The *AGE* feature maintains perfect data.

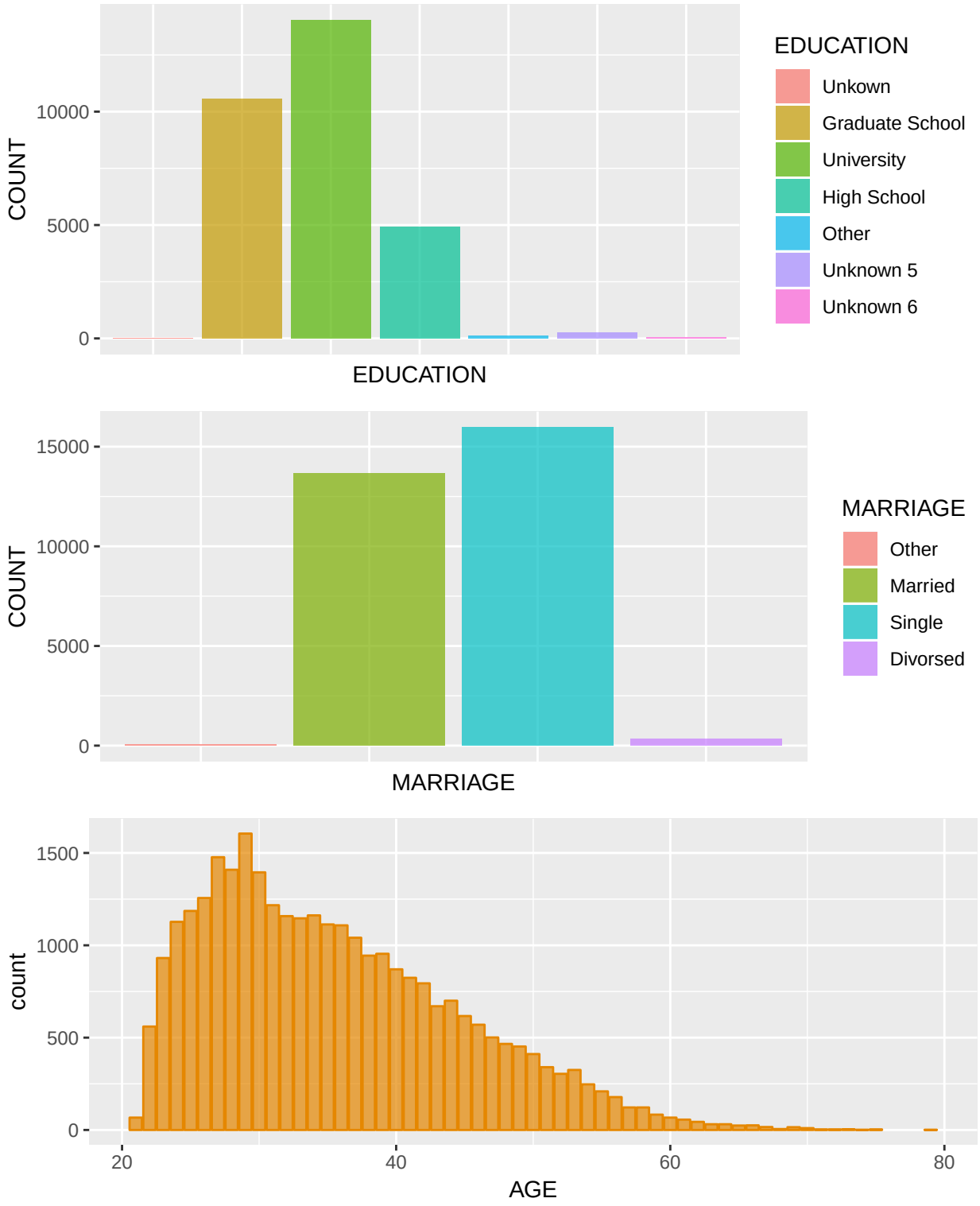


Figure 1: Customer Demographics

The next group of features we are going to explore is payment statuses. As per the data dictionary the payment statuses are supposed to have the status codes in the following range: **-2 : 9**. Let's verify the data integrity of the features.

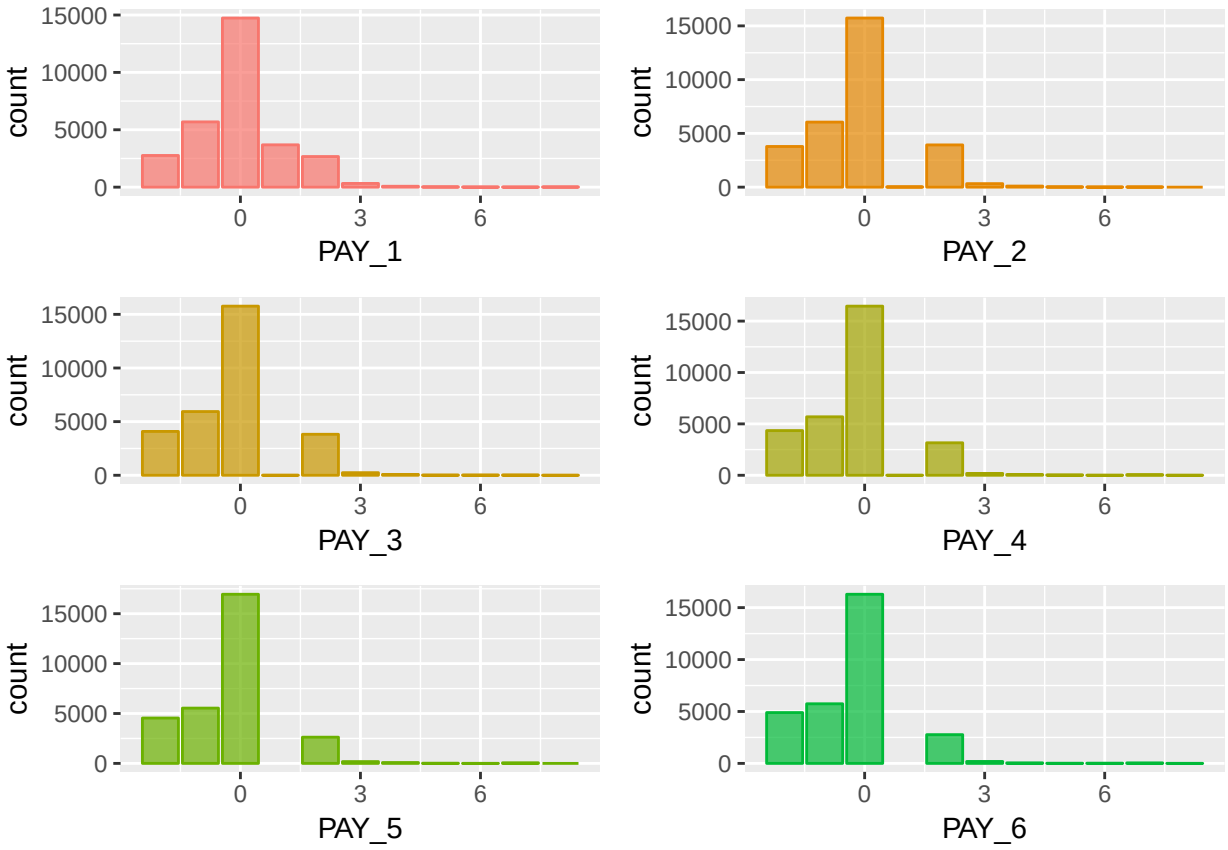


Figure 2: Customer Payment Statuses for the Last Six Months

As we have already noticed many of the credit card holders pay duly, codes: *-2 and -1* (see Figure: 2). Majority of the customers do maintain good standing. Noticeably though they **paid the required minimum or more but not the full balance** (code *0*). There is a rather significant group that falls behind with the payment by one or two months. The next group of features are the bill amounts for the last six months.

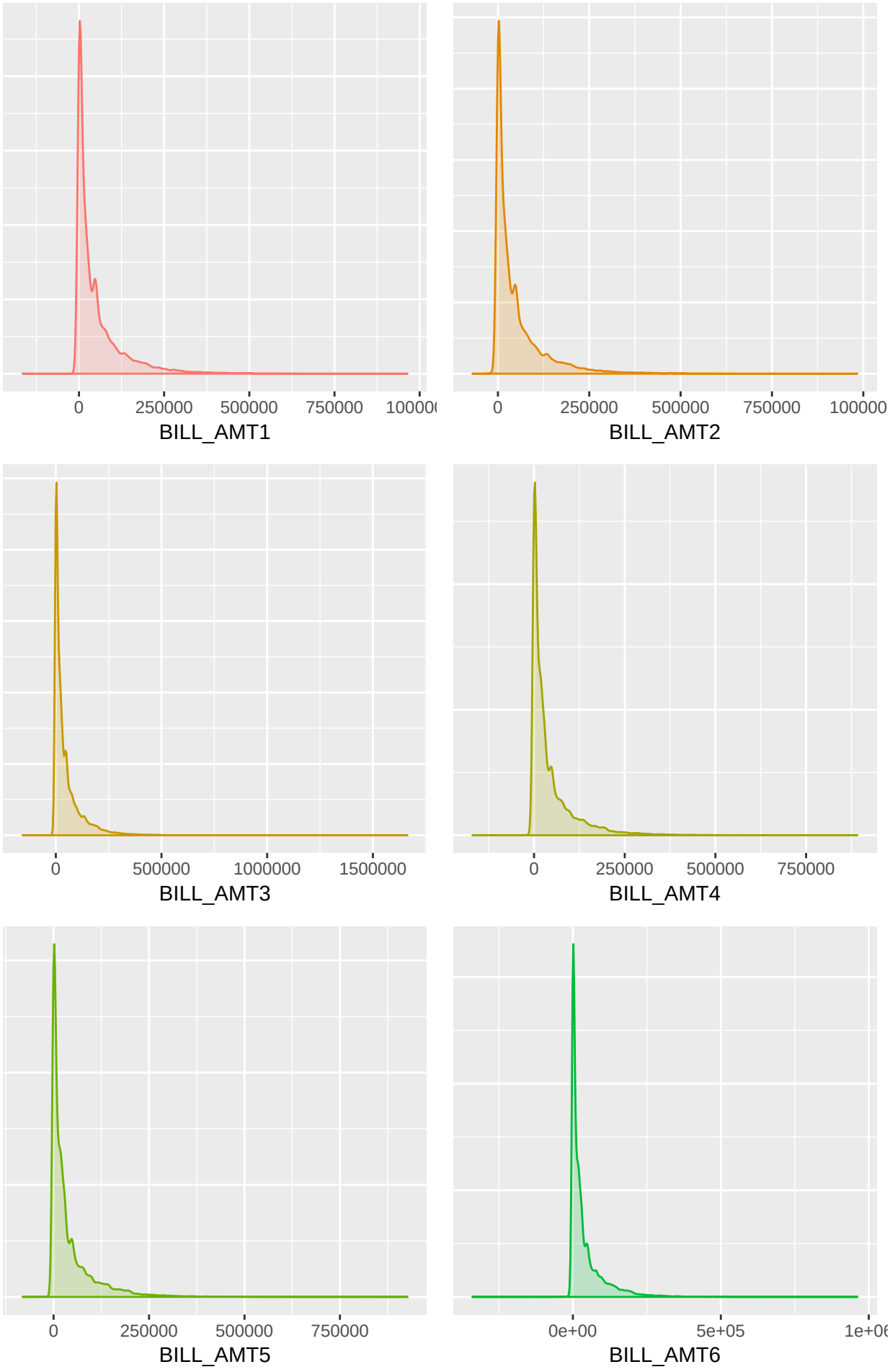


Figure 3: Customer Bill Amount Distribution for the Last Six Months

The bill payment amounts have negative values, significant amounts that reach at times **thens of thousands** of NT dollars! The negative amount on the credit card bill statements happen when a card holder overpaid his/her bill or were issued a credit after he/she already paid the bill.

Noticeably the bill amounts have very long tails. They average in tens of thousands of TN dollars, hovering around 50,000 dollars or so on average (see Table: 2). Thus it makes the negative bill amounts we previously observed more plausible.

The last group of the features is the client monthly payments. The data maintained in those columns appears to be integral (see Table: 2). Let's see how the customer payments are distributed. We employ normal and log-scaled visualization for better presentation.

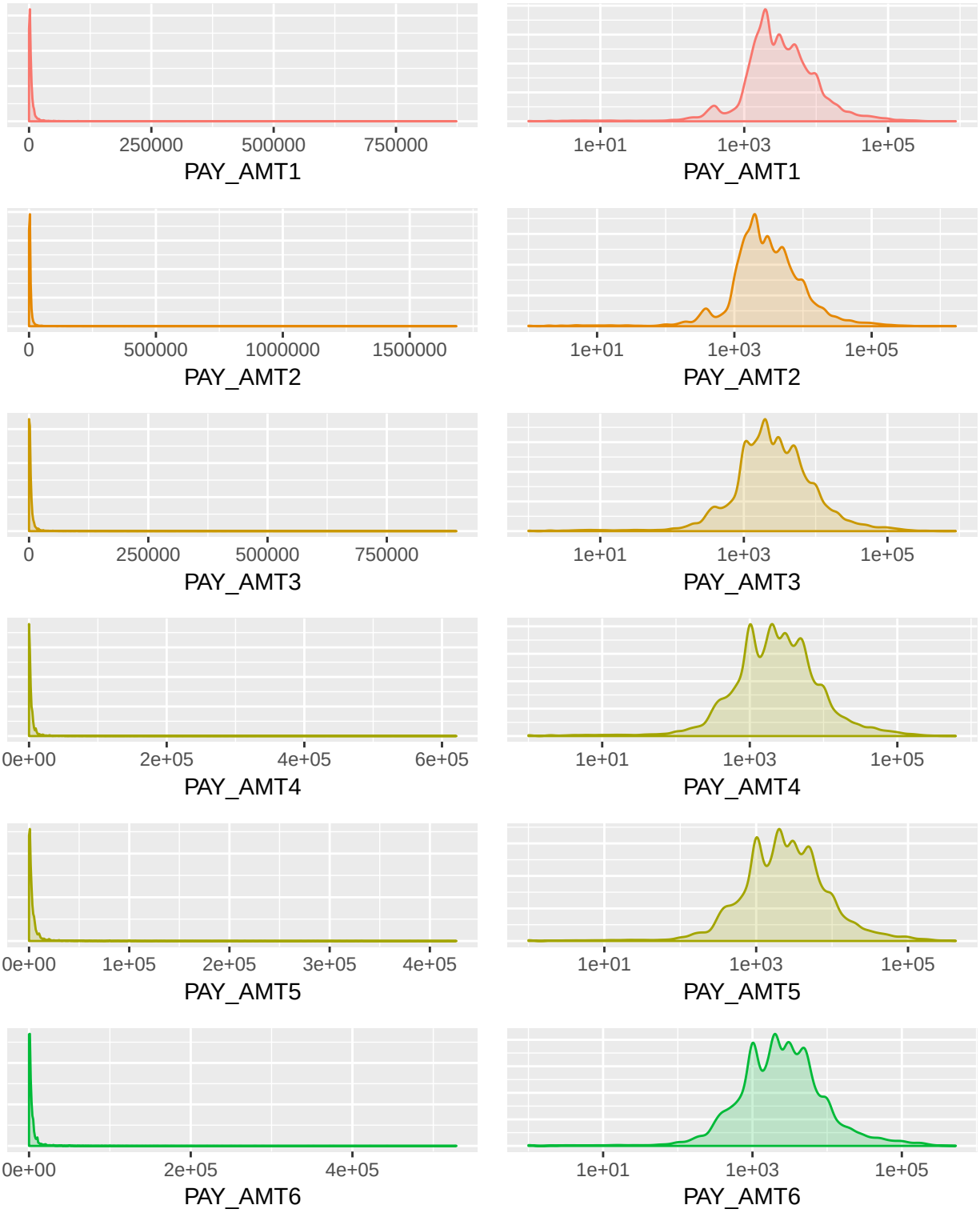


Figure 4: Customer Payment Amount Distribution for the Last Six Months

The pay amounts mirror in the distribution the bill amounts, which is expected. The charts have very long tails which imply that the amounts the card holders pay, very greatly. Most likely the payments that are way outside of the normal distribution curve are lump sum payments. More often than not the clients pay between 1000 and 10000 dollars monthly, which is still way below the average bill amount. This finding and the payment status statistics (see Figure: 2) make us believe that the majority of the credit card holders do have quite significant debt, despite the good standing. This hypotheses also explains the distribution of the payments. To keep the debt growth in check the customers pay lump sums whenever they accumulate some saving. Let's plot the delta between the

bill amounts and the payment amounts to support our theory. Figure 5

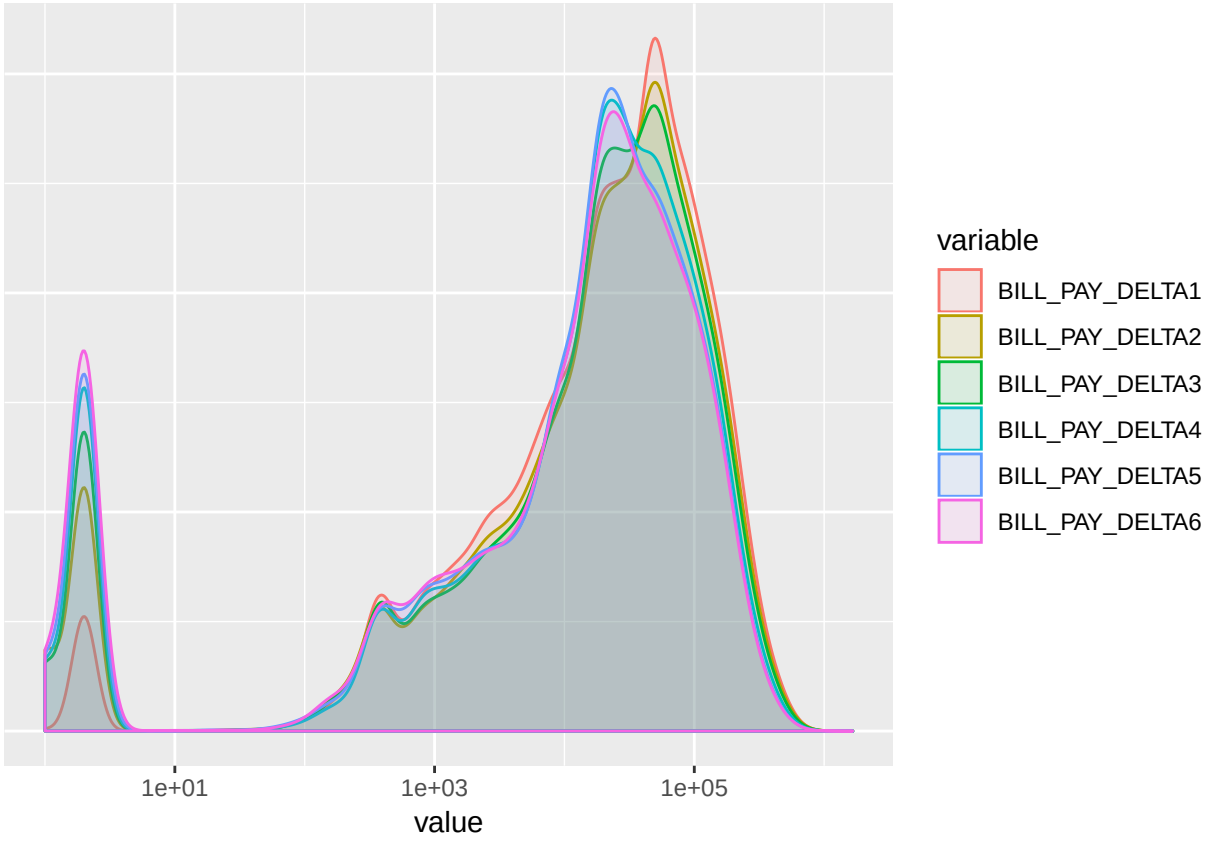


Figure 5: Customer Bill/Payment Amount Delta for Six Months

Data Transformation

Before we proceed further we are going to clean the data set as described in the previous paragraph, namely:

- We will remove *Customer ID* column
- We assign code **4 - Other** to the *EDUCATION* column values that fall out of the declared code range (1:4)

```
original$EDUCATION = with(original, ifelse(EDUCATION == 0 | EDUCATION == 5 | EDUCATION == 6 , 4, EDUCATION))
data = dplyr::select(original, -ID)
data %>% filter(EDUCATION == 0 | EDUCATION >4) %>% summarise(COUNT=n())
```

```
  COUNT
1     0
```

Data Correlation and Principal Component Analysis

In this section of our study we continue exploring the relations between various features of the data set. We put stress on finding the correlated features, the correlation between the features and the target label. We also are going to apply principal component analysis (PCA) to understand which attributes of the data set explain most variance of the original data. If our findings are fruitful we may design a model that requires smaller number of the input parameters without sacrificing the predictive power of the model.

Let's plot the correlation matrix first.

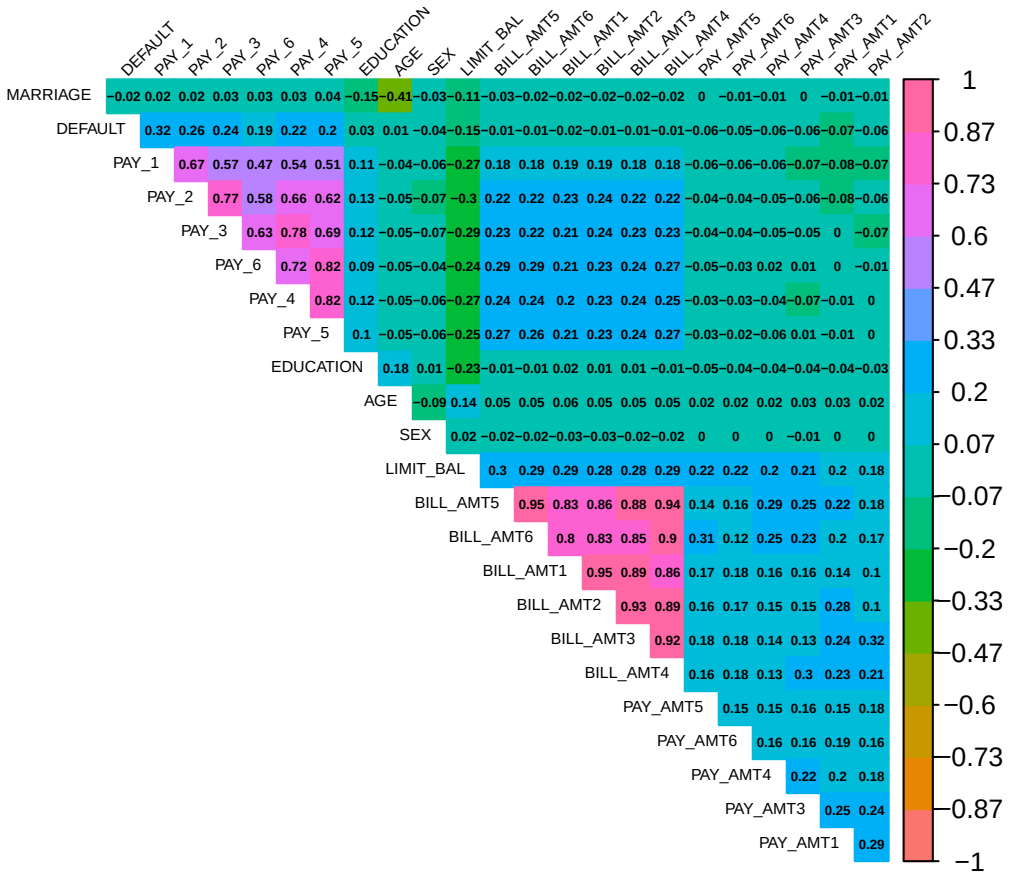


Figure 6: Data Correlation

The correlation matrix does not yield any surprises (see Figure: 6)). Bill payment amounts exhibit higher correlation as well as the payment status group. This does not give us much. The target label has no correlation with any other feature. We proceed with the PCA analysis now. We scale and center the data to achieve meaningful result. We also remove the target feature from the PCA computation. We are going to retain the 15 top components.

```
       eigenvalue variance.percent cumulative.variance.percent
Dim.1  6.54287678       28.4472904                    28.44729
Dim.2  4.10332133       17.8405275                    46.28782
Dim.3  1.55513542        6.7614583                    53.04928
Dim.4  1.47549730        6.4152057                    59.46448
Dim.5  1.02455764        4.4545984                    63.91908
Dim.6  0.95545872        4.1541683                    68.07325
Dim.7  0.90407409        3.9307569                    72.00401
Dim.8  0.88793791        3.8605996                    75.86461
Dim.9  0.87030881        3.7839514                    79.64856
Dim.10 0.78268998        3.4029999                    83.05156
Dim.11 0.73278811        3.1860353                    86.23759
Dim.12 0.68195482        2.9650210                    89.20261
Dim.13 0.57091319        2.4822313                    91.68484
Dim.14 0.51977719        2.2599008                    93.94474
Dim.15 0.40359547        1.7547629                    95.69951
Dim.16 0.25988392        1.1299301                    96.82944
Dim.17 0.24930179        1.0839208                    97.91336
Dim.18 0.18869146        0.8203976                    98.73376
Dim.19 0.13178740        0.5729887                    99.30674
Dim.20 0.07015088        0.3050038                    99.61175
Dim.21 0.04078476        0.1773250                    99.78907
Dim.22 0.02529347        0.1099716                    99.89905
Dim.23 0.02321955        0.1009546                   100.00000
```

Good news! The **top 10 components explain 83%** of the data variance. Let's review what the top

four components are made of.

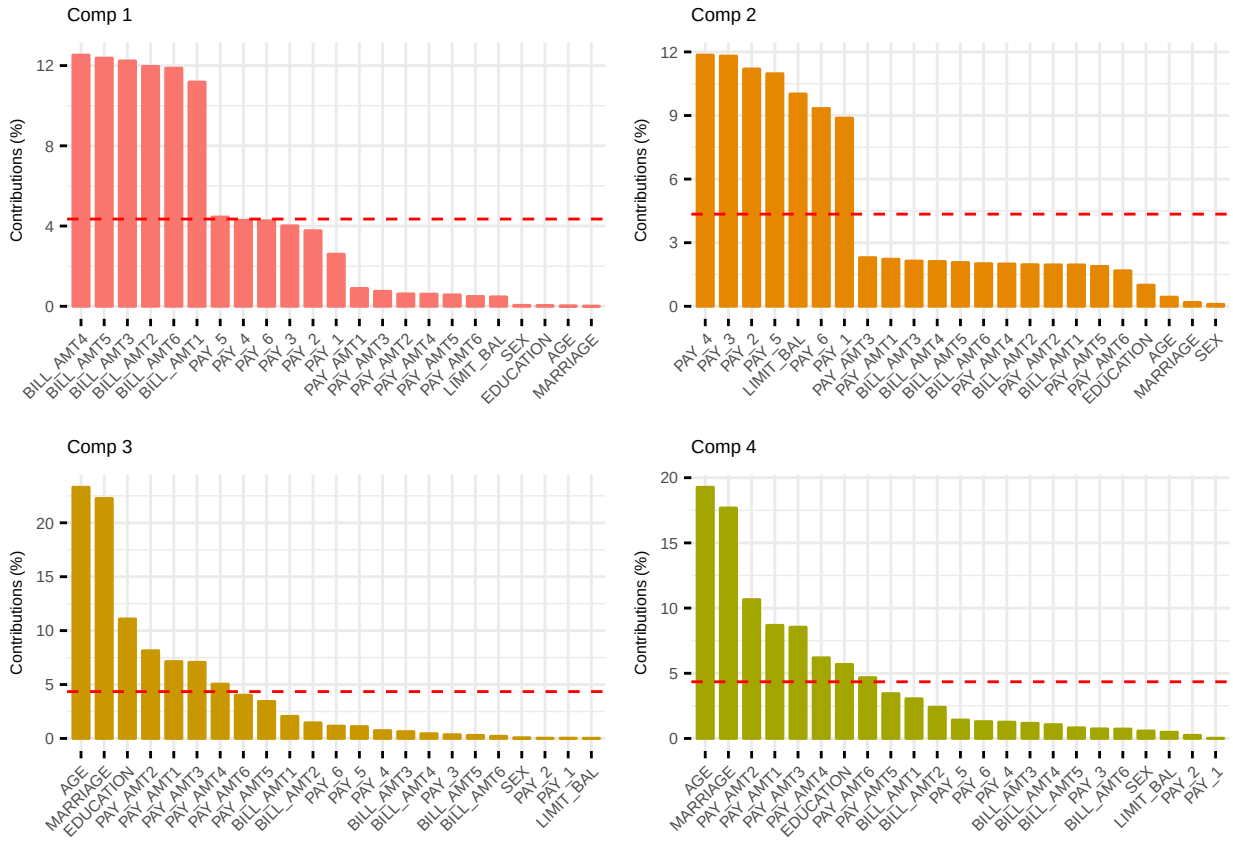


Figure 7: Feature Contribution to the Top Four Components

The red dashed line on the graph (see Figure: 7) indicates the expected average contribution. If the contribution of the variables were uniform, the expected value would about 4.3%. For a given component, a variable with a contribution larger than this cutoff could be considered as important in contributing to the component. The PCA analysis supports the correlation matrix (Figure: 6) in a sense that the bill amounts and the pay statuses are highly correlated and are subject to the dimentionality reduction due to the redundancy. So as we can see the first component is largely being dominated by the bill amounts and pay statuses. The second one mainly includes the payment status features and the credit card limit. The demographic features and the amount paid dominate the third and fourth components.

It would be very interesting to see how the top two components look like in the context of the target label. Very well, Figure: 9 shows that there is no clear separation between component one and two. As the previous chart highlighted (see Figure: 7) this shall be expected since the top two components largely comprise bill payments and pay statuses. We also observe two linear formations: one is located in the first quadrant; green (no default). The second formation is located in the fourth quadrant; it peppered with the red color that denotes the default green. to understand better what those formations are we will plot the feature vectors in the context of the component one and two.

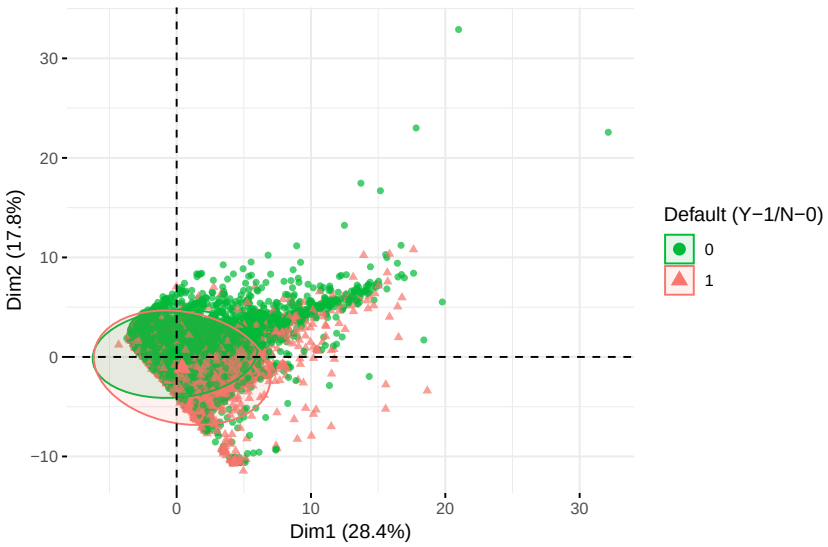


Figure 8: Feature contribution to the Top Two Componenets

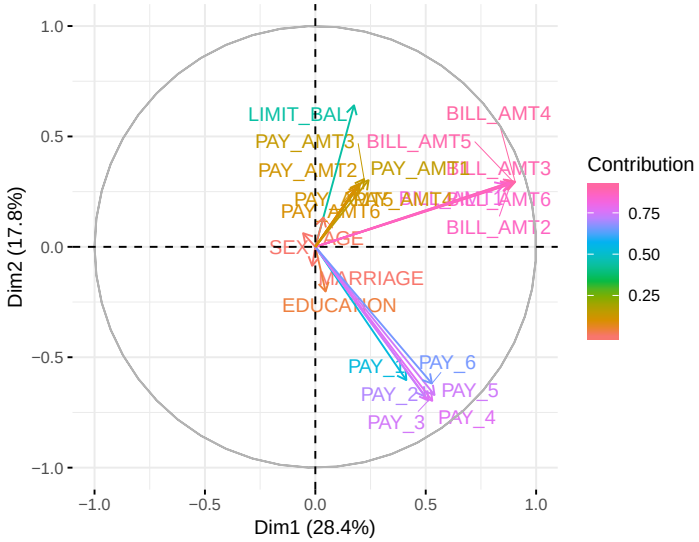


Figure 9: Components and Feature Contribution in the Context of Target Label

The plot submitted above clearly shows that the biggest contributor to the first quadrant is a bill amount group. And the contributors to the fourth quadrant are payments statuses.

Takeaways from Data Exploration Excersize We have concluded the data exploration study. There are a few major points we would like to highlight. They are:

- The majory of the credit card holders maintain good standing (see Figure: 2).
- The clients do maintain debt. On regular basis they pay the amount that meets the minimal required payment but less then the bill amount (see Figure 5).
- The clients tend to reduce the amount of debt paying the outstanding balance in lump sums (Figure: 4).
- Demographically married and single people represented almost equally, women are represented better than men. Majority of the credit card holders have a university degree. The vast majority of the clients are between 25 and 50 year old.
- The data set is not balanced; the number of the customers in a good standing is much higher than the number of people who defaulted on the next payment.

- Principal component analysis showed that we can reduce the number of the features to 10 - 12 sustaining the data variance coverage at about **85%**. The top principal components comprise mainly bill amounts and pay statuses (Figure: 7). The components do not have clear separation and affect equally the target label (Figure: 8).

Modeling and Evalutation

Data Preparation

Prior to fitting the models with the data we would have to usample the trainging set, because our data set is not blanaced; the number of customers who defaults is much smaller than the number of the payning customers.

All models we are going to fit will benefit from the data scaling and centering, thus we are going to upply these transformations.

```
splitIdx = caret::createDataPartition(y=data$DEFAULT, p=0.7, list=FALSE)
trainData = data[splitIdx,]
testData  = data[-splitIdx,]

# sample. REMOVE for final run
trainData = sample_n(trainData, 2000)
testData = sample_n(testData,2000)

preprocParams = c("BoxCox", "center", "scale");

columns = colnames(trainData)
trainData = upSample(x = trainData[, columns[columns != "DEFAULT"] ],
      y = as.factor(trainData$DEFAULT), list = F, yname = "DEFAULT")
print(table(trainData$DEFAULT))
```

```
   0    1
1543 1543
```

Naive Bayes Model

Naïve Bayes classification is a kind of simple probabilistic classification methods based on Bayes' theorem with the assumption of independence between features.

It is simple (both intuitively and computationally), fast, performs well with small amounts of training data, and scales well to large data sets. The greatest weakness of the naïve Bayes classifier is that it relies on an often-faulty assumption of equally important and independent features which results in biased posterior probabilities. Although this assumption is rarely met, in practice, this algorithm works surprisingly well and accurate; however, on average it rarely can compete with the accuracy of advanced tree-based methods (random forests & gradient boosting machines) but is definitely worth having in our toolkit.

```
Naive Bayes

3086 samples
  23 predictor
   2 classes: 'no', 'yes'

Pre-processing: Box-Cox transformation (3), centered (23), scaled (23)
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 2468, 2468, 2470, 2469, 2469
Resampling results across tuning parameters:

  usekernel  ROC        Sens       Spec
  FALSE      0.7567084  0.3513197  0.8457593
   TRUE      0.8077199  0.9138087  0.4569432

Tuning parameter 'fL' was held constant at a value of 0
Tuning
```

```
 parameter 'adjust' was held constant at a value of 1
ROC was used to select the optimal model using the largest value.
The final values used for the model were fL = 0, usekernel = TRUE
 and adjust = 1.

confusionMatrix(data = pred.naiveBayesModel.raw, testDataCopy$DEFAULT)

Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  1449  266
       yes  129  156

               Accuracy : 0.8025
                 95% CI : (0.7844, 0.8197)
    No Information Rate : 0.789
    P-Value [Acc > NIR] : 0.07237

                  Kappa : 0.3268
 Mcnemar's Test P-Value : 7.76e-12

            Sensitivity : 0.9183
            Specificity : 0.3697
         Pos Pred Value : 0.8449
         Neg Pred Value : 0.5474
             Prevalence : 0.7890
         Detection Rate : 0.7245
   Detection Prevalence : 0.8575
      Balanced Accuracy : 0.6440

       'Positive' Class : no
```

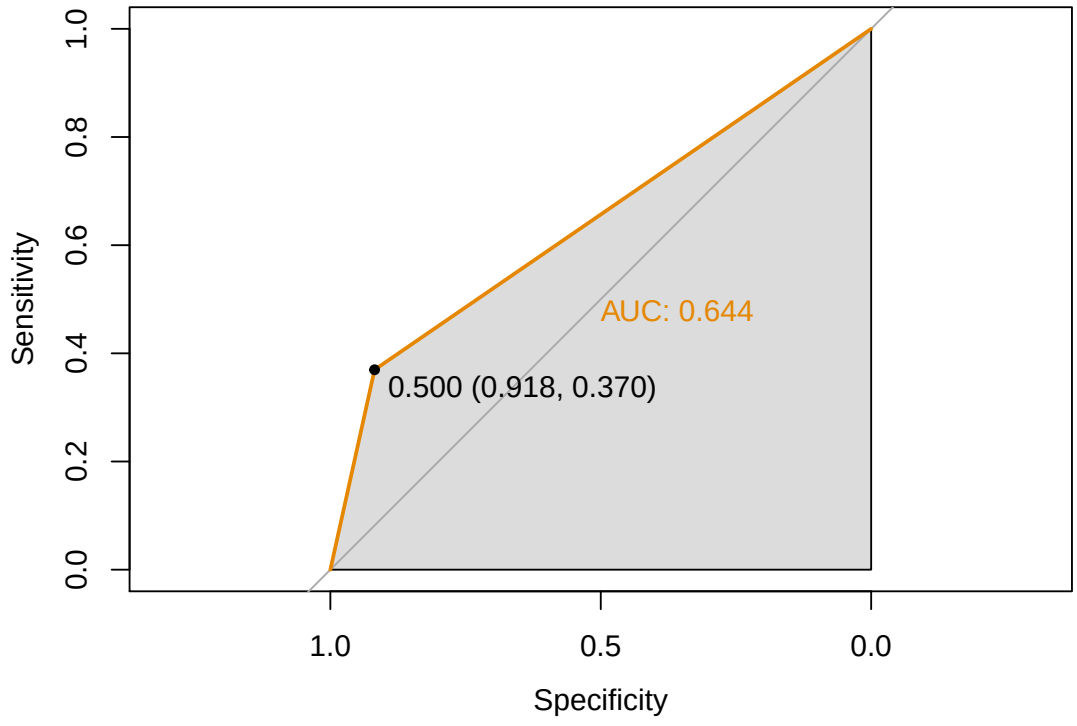


Figure 10: Naive Bayes Model AUC and ROC Curve

Random Forest Model

Random Forest is also considered as a very handy and easy to use algorithm, because it's default hyperparameters often produce a good prediction result. Random Forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model. The main limitation of Random Forest is that a large number of trees can make the algorithm to slow and ineffective for real-time predictions. In general, these algorithms are fast to train, but quite slow to create predictions once they are trained.

```
Random Forest

3086 samples
  23 predictor
   2 classes: 'no', 'yes'

Pre-processing: Box-Cox transformation (3), centered (23), scaled (23)
Resampling: Cross-Validated (3 fold)
Summary of sample sizes: 2057, 2058, 2057
Resampling results across tuning parameters:

  mtry  ROC        Sens       Spec
   2    0.9679013  0.8677824  0.9481433
  12    0.9699482  0.8697254  0.9539786
  23    0.9680196  0.8729704  0.9481483

ROC was used to select the optimal model using the largest value.
The final value used for the model was mtry = 12.

confusionMatrix(data = pred.randomForestModel.raw, testDataCopy$DEFAULT)

Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  1424  228
       yes  154  194

               Accuracy : 0.809
                 95% CI : (0.7911, 0.826)
    No Information Rate : 0.789
    P-Value [Acc > NIR] : 0.0144365

                  Kappa : 0.387
 Mcnemar's Test P-Value : 0.0001877

            Sensitivity : 0.9024
            Specificity : 0.4597
         Pos Pred Value : 0.8620
         Neg Pred Value : 0.5575
             Prevalence : 0.7890
         Detection Rate : 0.7120
   Detection Prevalence : 0.8260
      Balanced Accuracy : 0.6811

       'Positive' Class : no
```

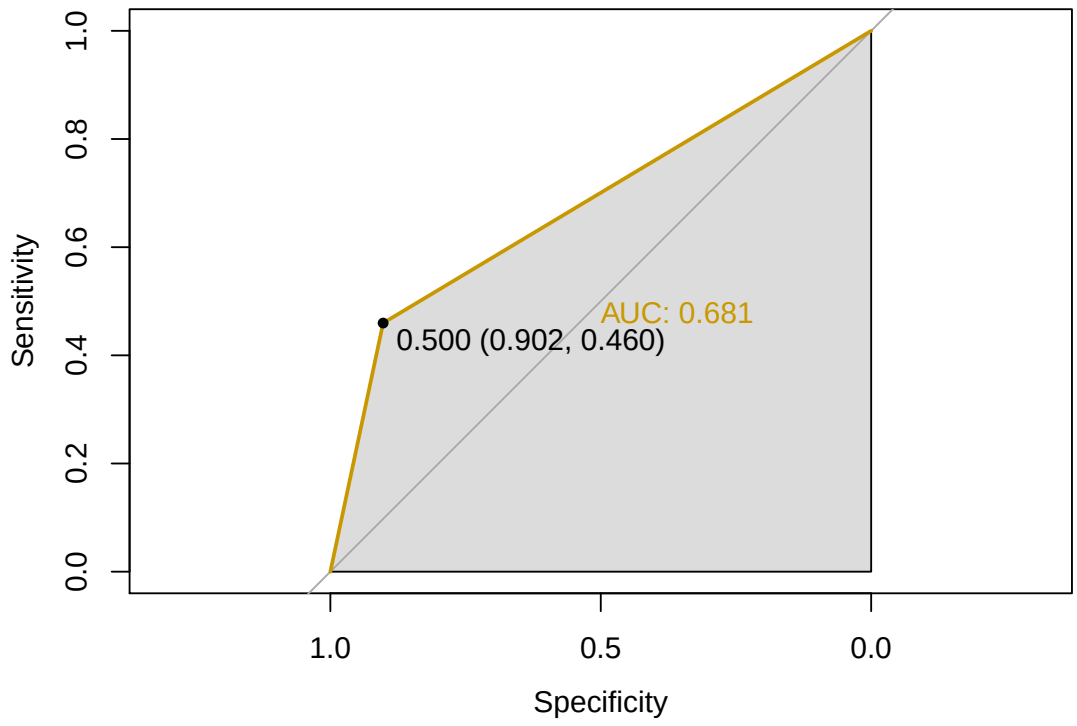


Figure 11: Random Forest Model AUC and ROC Curve

Logistic Regression Model

Logistic regression is an efficient, interpretable and accurate method, which fits quickly with minimal tuning. Logistic regression prediction accuracy will benefit if the data is close to Gaussian distribution. Thus we apply addition transformation to the training data set. We will also be employing 5-fold cross-validation resampling procedure to improve the model.

```
confusionMatrix(data = pred.logRegModel.raw, testDataCopy$DEFAULT)

Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  1102  152
       yes  476  270

               Accuracy : 0.686
                 95% CI : (0.6651, 0.7063)
    No Information Rate : 0.789
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2639
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.6984
            Specificity : 0.6398
         Pos Pred Value : 0.8788
         Neg Pred Value : 0.3619
             Prevalence : 0.7890
         Detection Rate : 0.5510
   Detection Prevalence : 0.6270
      Balanced Accuracy : 0.6691
```

```
'Positive' Class : no
```

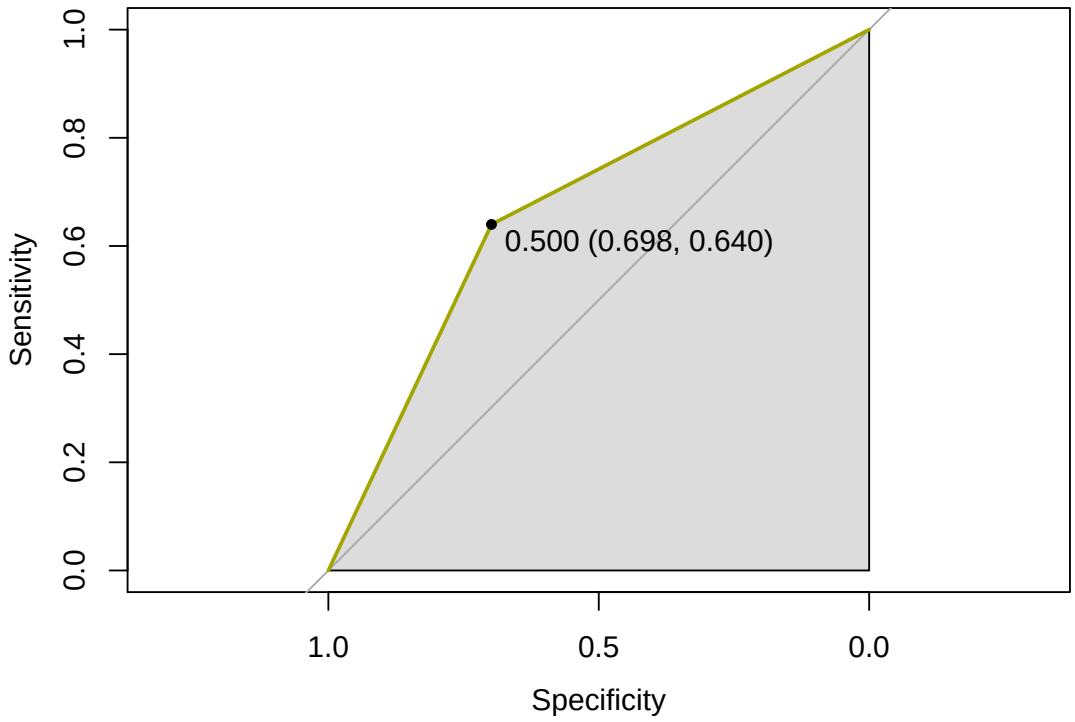


Figure 12: Logistic Regression Model AUC and ROC Curve

Confusion matrix and Figure 12 demonstrate the logistic model performance on the balanced data set. Using the proportion of positive data points that are correctly considered as positive (true positives) and the proportion of negative data points that are mistakenly considered as positive (false negative), we generated a graphic that shows the trade off between the rate at which the model correctly predicts the rain tomorrow with the rate of incorrectly predicting the rain. The value around 0.80 indicates that the model does a good job in discriminating between the two categories.

Let's now compare the performance of the PCA components to the performance of the full data set. Just to remind we selected have identified 10 top componenets that explain *83%* of data variance. We are going to reduce the dimentionality of the training and test data set multiplying them on the principal comonent wiegths matrixd.

```
Confusion Matrix and Statistics

          Reference
Prediction  no yes
       no  804 114
       yes 774 308

               Accuracy : 0.556
                 95% CI : (0.5339, 0.5779)
    No Information Rate : 0.789
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1522
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.5095
            Specificity : 0.7299
         Pos Pred Value : 0.8758
         Neg Pred Value : 0.2847
```

```
            Prevalence : 0.7890
        Detection Rate : 0.4020
  Detection Prevalence : 0.4590
     Balanced Accuracy : 0.6197

      'Positive' Class : no
```

Well the result of the model fitted with the PCA componenets is inferiour to the same very model, whichwas fitted with the whole data set by about 7%. But we shall not forget that the PCA data set had less than a half oof the features of the original data set.

Model Comparison

Now it is time to compare the models side by side and pick a winner.

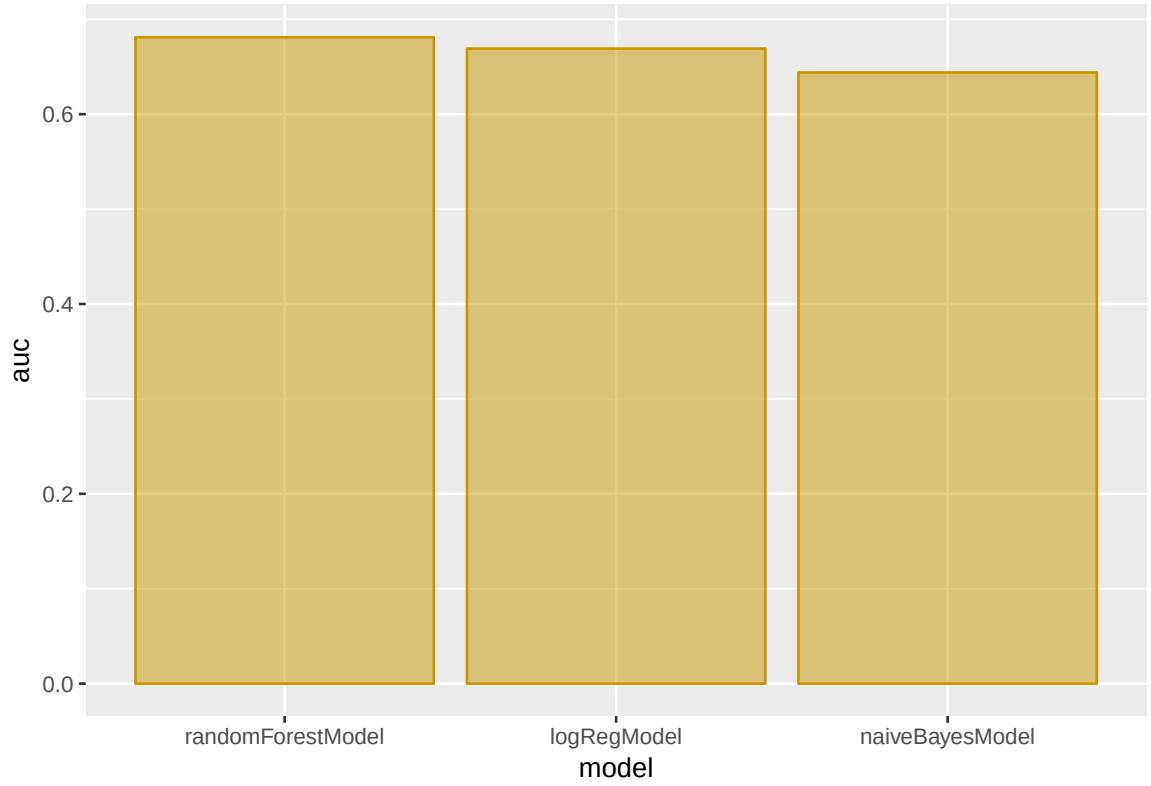


Figure 13: Model AUC Comparison

```
              model       auc
3 randomForestModel 0.6810619
1       logRegModel 0.6690814
2   naiveBayesModel 0.6439596
```

AUC - ROC perfomance

Model interpretibility Logistic Regression and Naive Bayes are all highly interpreatable models. It is easy to explain to the business what impact each input parameter has. The decision tree could be visualized (provided if it is not too large).

Random Forest on the other hand is a black-box model, complex algorithm which is difficult to explain in simple terms.

Data Preparation Random Forest and Naive Bayes can deal with missing data, outliers, numeric and alphanumeric values. Simply speaking they are not very demanding for data quality. It would be

interesting to see how they perform on the original data set without data cleaning. But this is subject of another research...

Logistic regression does require conversion of alphanumeric values to numeric, struggles dealing with the outliers and performs best when fitted with the data that have normal distribution.

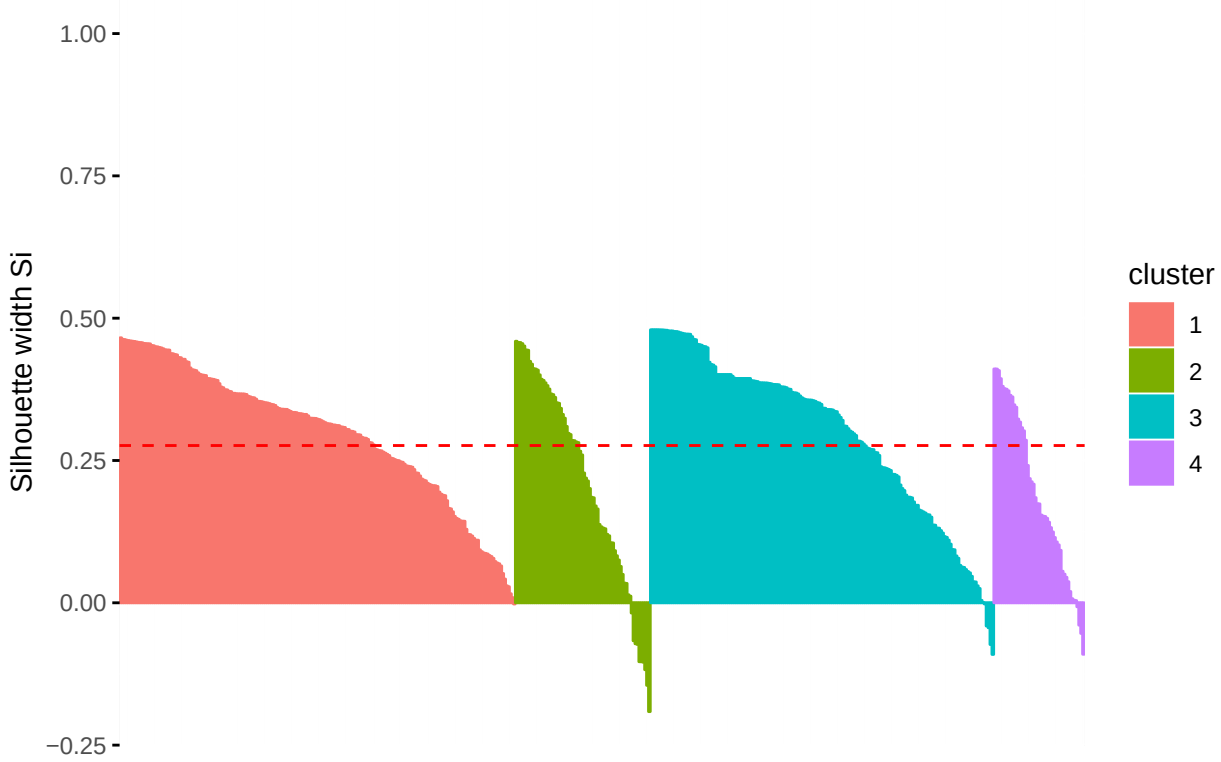
Verdict Despite sensitivity to data quality Random Forest,Logistic Regression outperforms Naive-Bayes model in all other major categories.Both are very close.

Undertanding the Clinet Base Employing Unsupervised Learning

Lastly we believe it would be beneficial to profie the client base. This would add addtional insights into understanding of the credit card holders demographics and spending and borrowing habbits. We will be employing CLARA approach to attack this challange. This study does not make its goal to compare various unsupervised model techniques. CLARA was chozen because it is robust, relatively easy to understand and fast. After many trials and errors we have selected a few features that describe the client financial and demographic profile quite well

```
[1] 0.06032056
```

```
  cluster size ave.sil.width
1       1  205          0.30
2       2   70          0.21
3       3  178          0.30
4       4   47          0.19
```



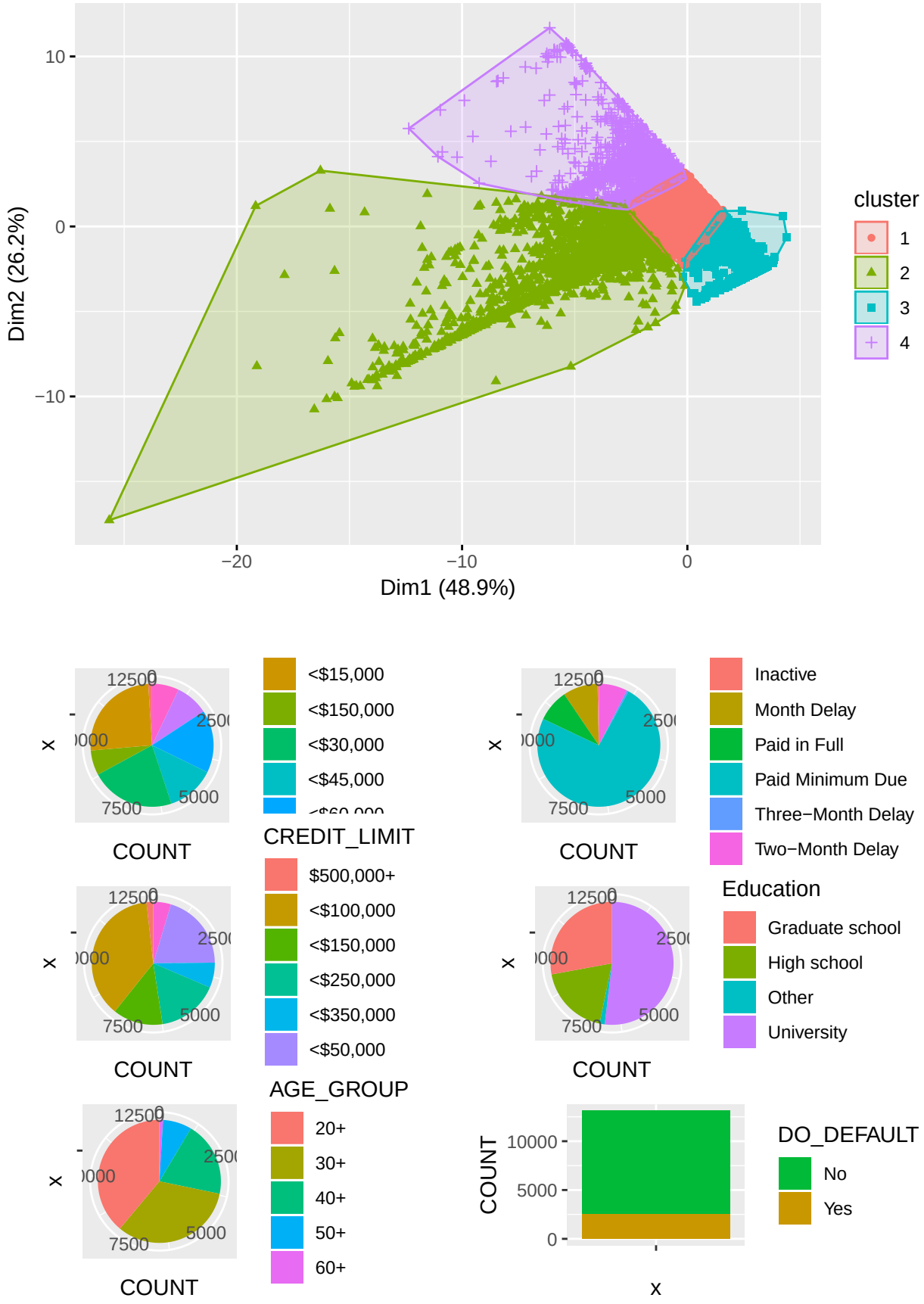


Figure 14: Cluster One

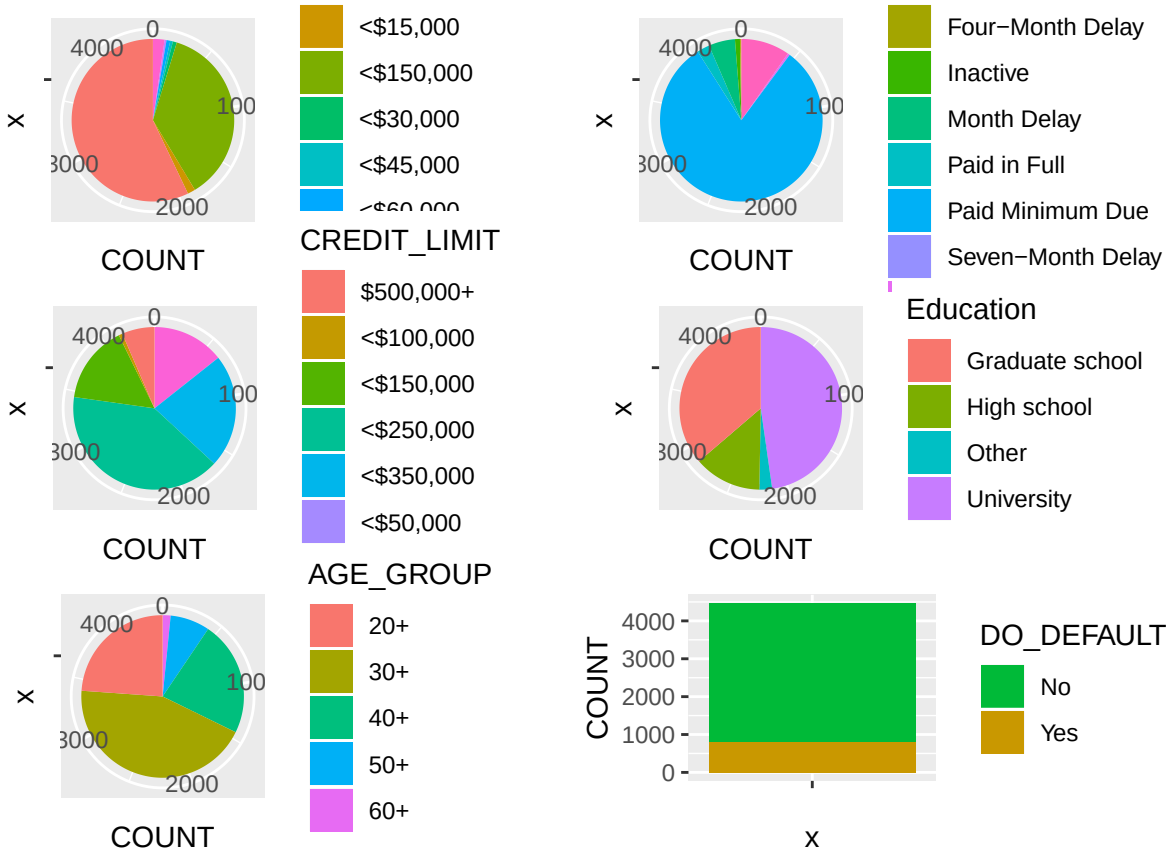


Figure 15: Cluster Two

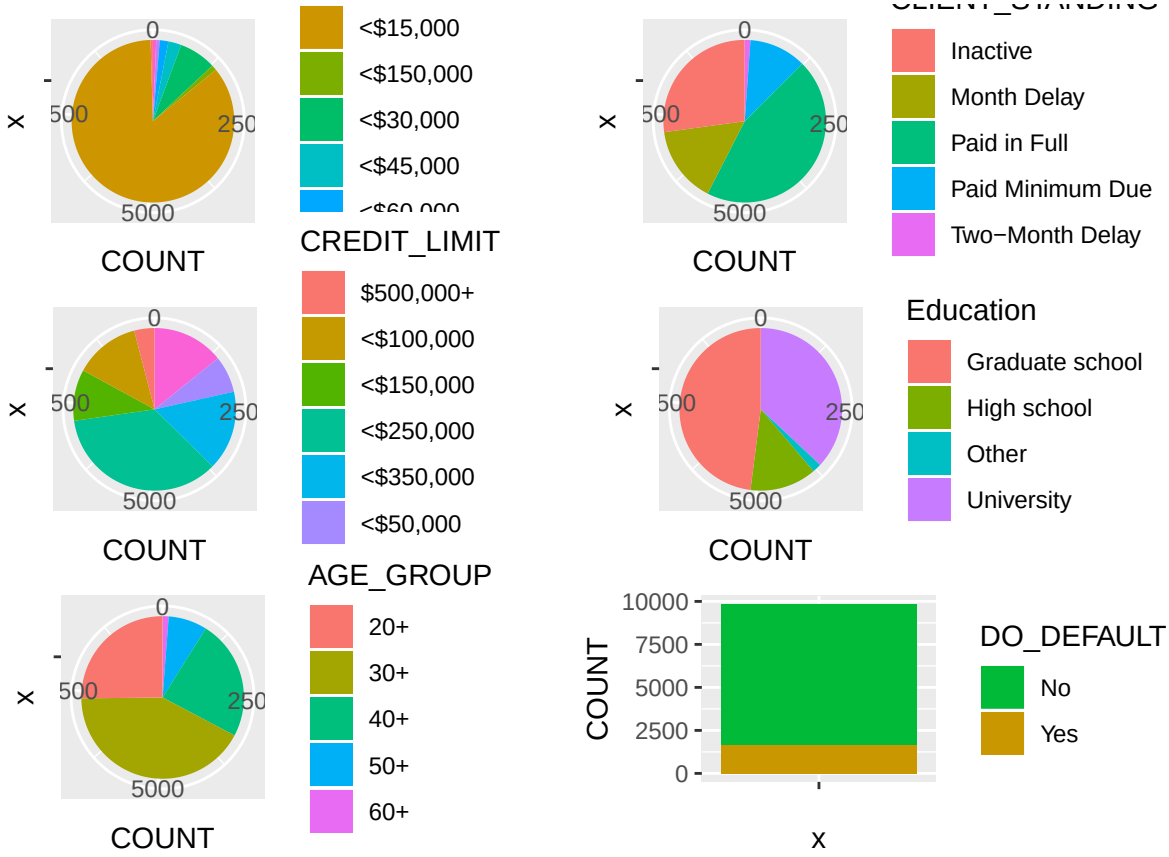


Figure 16: Cluster Three

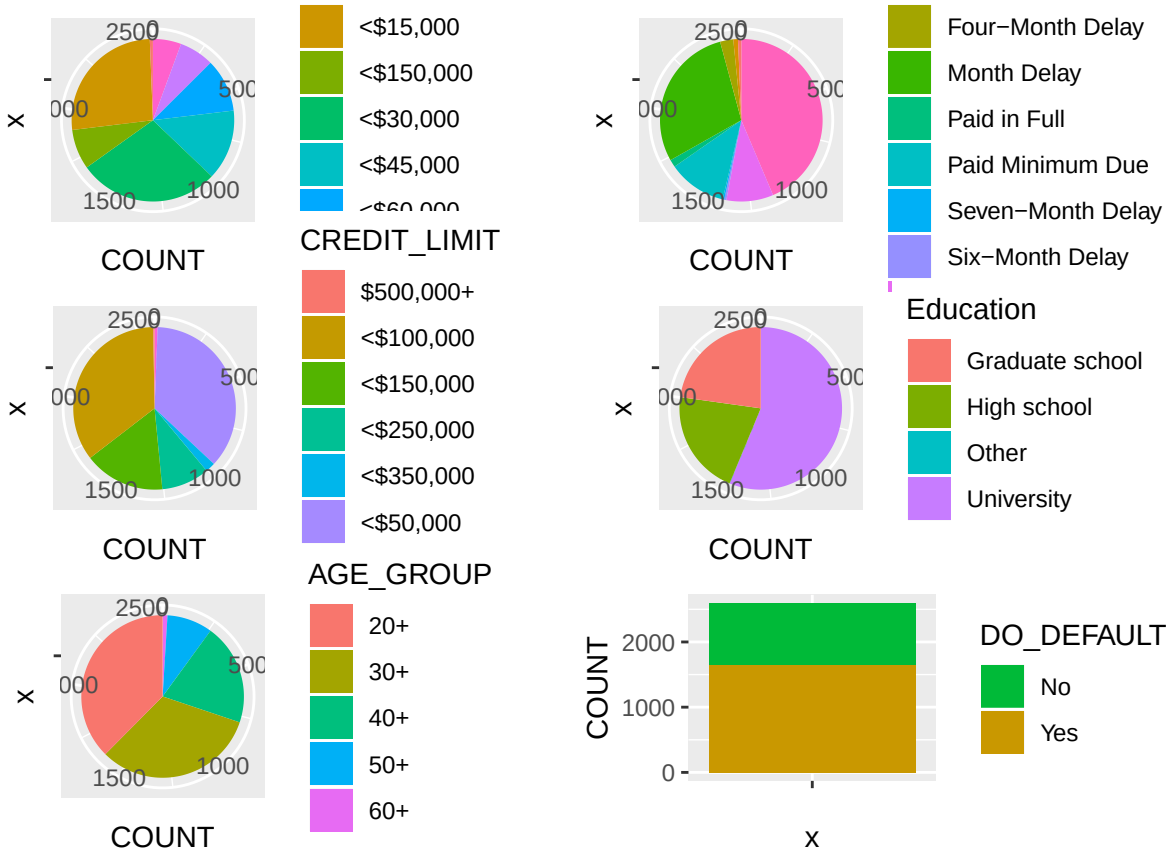


Figure 17: Cluster Four

Model Deployment

Conclusion

Bibliography

A. Comoreanu. Credit card debt study. trends and insights. URL https://wallethub.com/edu/credit-card-debt-study/24400/. [p1]

Note from the Authors

This file was generated using *The R Journal* style article template, additional information on how to prepare articles for submission is here - Instructions for Authors. The article itself is an executable R Markdown file that could be downloaded from Github with all the necessary artifacts.

Sumaira Afzal
York University School of Continuing Studies

Viraja Ketkar
York University School of Continuing Studies

Murlidhar Loka

York University School of Continuing Studies

Vadim Spirkov
York University School of Continuing Studies