BT2103 Group Project

Report on the Default of Credit Card Clients

Group Members

Kenny Low Sheng Wei A0233869N

Ng Wei Han A0234338E

Dylan Wo Qiying A0233634J

Report Contents

Introduction

About the Data Set

The data set we are provided contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005.

The 25 variables (columns) in the data set are as follows:

- `ID`: ID of each client
- `LIMIT_BAL`: Amount of given credit in NT dollars (includes individual and family/supplementary credit
- `SEX`: Gender (1=male, 2=female)
- `EDUCATION`: (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)
- `MARRIAGE`: Marital status (1=married, 2=single, 3=others)
- `AGE`: Age in years
- `PAY_0`: Repayment status in September, 2005 (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months, ... 8=payment delay for eight months, 9=payment delay for nine months and above)
- `PAY_2`: Repayment status in August, 2005 (scale same as above)
- `PAY_3`: Repayment status in July, 2005 (scale same as above)
- `PAY_4`: Repayment status in June, 2005 (scale same as above)
- `PAY_5`: Repayment status in May, 2005 (scale same as above)
- `PAY_6`: Repayment status in April, 2005 (scale same as above)
- `BILL_AMT1`: Amount of bill statement in September, 2005 (NT dollar)
- `BILL_AMT2`: Amount of bill statement in August, 2005 (NT dollar)
- `BILL_AMT3`: Amount of bill statement in July, 2005 (NT dollar)
- `BILL_AMT4`: Amount of bill statement in June, 2005 (NT dollar)
- `BILL_AMT5`: Amount of bill statement in May, 2005 (NT dollar)
- `BILL_AMT6`: Amount of bill statement in April, 2005 (NT dollar)
- `PAY_AMT1`: Amount of previous payment in September, 2005 (NT dollar)
- `PAY_AMT2`: Amount of previous payment in August, 2005 (NT dollar)
- `PAY_AMT3`: Amount of previous payment in July, 2005 (NT dollar)
- `PAY_AMT4`: Amount of previous payment in June, 2005 (NT dollar)
- `PAY_AMT5`: Amount of previous payment in May, 2005 (NT dollar)
- `PAY_AMT6`: Amount of previous payment in April, 2005 (NT dollar)
- `default.payment.next.month`: Default payment (1=yes, 0=no)

Data Modeling Problem

Since credit score cards are based on historical data, we are interested to find out whether it is possible to train a model to be profficient in predicting whether a customer will default based on their past/present attributes.

In this report, we will focus on building and testing a machine learning model that is capable of classifying applicants as 'good' or 'bad', with 'good' applicants being individuals that will not default their credit payments (Negaitve class) and 'bad' applicants being individuals that will default their credit payments (Positive class).

Exploratory Data Analysis

General Data Exploration

When presented with a new data set, we first would like to carry out a generalized check on the data set's characteristics and features.

Firstly, let us peek at the first few entries.

```
head(data, 5)
```

```
##   ID LIMIT_BAL SEX EDUCATION MARRIAGE AGE PAY_0 PAY_2 PAY_3 PAY_4 PAY_5 PAY_6
## 1  1     20000   2         2        1  24     2     2    -1    -1    -2    -2
## 2  2    120000   2         2        2  26    -1     2     0     0     0     2
## 3  3     90000   2         2        2  34     0     0     0     0     0     0
## 4  4     50000   2         2        1  37     0     0     0     0     0     0
## 5  5     50000   1         2        1  57    -1     0    -1     0     0     0
##   BILL_AMT1 BILL_AMT2 BILL_AMT3 BILL_AMT4 BILL_AMT5 BILL_AMT6 PAY_AMT1 PAY_AMT2
## 1      3913      3102       689         0         0         0        0      689
## 2      2682      1725      2682      3272      3455      3261        0     1000
## 3     29239     14027     13559     14331     14948     15549     1518     1500
## 4     46990     48233     49291     28314     28959     29547     2000     2019
## 5      8617      5670     35835     20940     19146     19131     2000    36681
##   PAY_AMT3 PAY_AMT4 PAY_AMT5 PAY_AMT6 default.payment.next.month
## 1        0        0        0        0                          1
## 2     1000     1000        0     2000                          1
## 3     1000     1000     1000     5000                          0
## 4     1200     1100     1069     1000                          0
## 5    10000     9000      689      679                          0
```

By calling the above functions, we can visually validate that the variables in the data set match its description given previously.

From our observations above, we can also make the following minor adjustments:

- remove the ID column as it is unnecessary for our purpose.
- change column name `PAY_0` to `PAY_1`, so that it is consistent with `BILL_AMT` and `PAY_AMT`.

```
data = data %>% rename(c("PAY_0" = "PAY_1"))
data = select(data, -ID)
```

We would like to next check for the completeness of the data. Below, we call 2 functions to check for `NA` values and duplicated data respectively.

```
any(is.na(data))
```

```
## [1] FALSE
```

```
any(duplicated(data))
```

```
## [1] TRUE
```

From these results, we can conclude that the data does not contain missing and/or redundant values

Secondly, let us check our variable types.

```
str(data)
```

From the code chunk above (Output in annex), we can observe that the current categorical variables of `SEX`, `EDUCATION`, `MARRIAGE`,`PAY` and `default.payment.next.month`, are of type `int`.

We shall now convert them to categorical variables (factor variables), and give them more representative factor names instead of numbers.

The corrected categorical variables are as follows:

`SEX` now has factors "Male" or "Female".

- Male = 1
- Female = 2

`EDUCATION` now has factors "Others","Graduate school", "University", "High school".

- Graduate school = 1
- University = 2
- High school = 3
- Others = 0,4,5,6

`MARRIAGE` now has factors "Others","Married", "Single", "Divorce".

- Married = 1
- Single = 2
- Divorced = 3
- Others = 0

Here, `PAY` variables can be further formatted. Since 1 to 8 means payment delay for "x" months, we will call all of them "Payment delay".

6 additional columns, from `PAY0_DELAY` to `PAY6_DELAY`, will be added to keep track of the delay in payment for each month.

`PAY` now has factors "No consumption","Fully paid", "Use of revolving credit", "Payment delay".

- No consumption = -2
- Fully paid = -1
- Use of revolving credit = 0
- Payment delay = 1,2,..,8

`default.payment.next.month` now has factors "Non_Default", "Default".

- Non_Default = 0
- Default = 1

Splitting of data

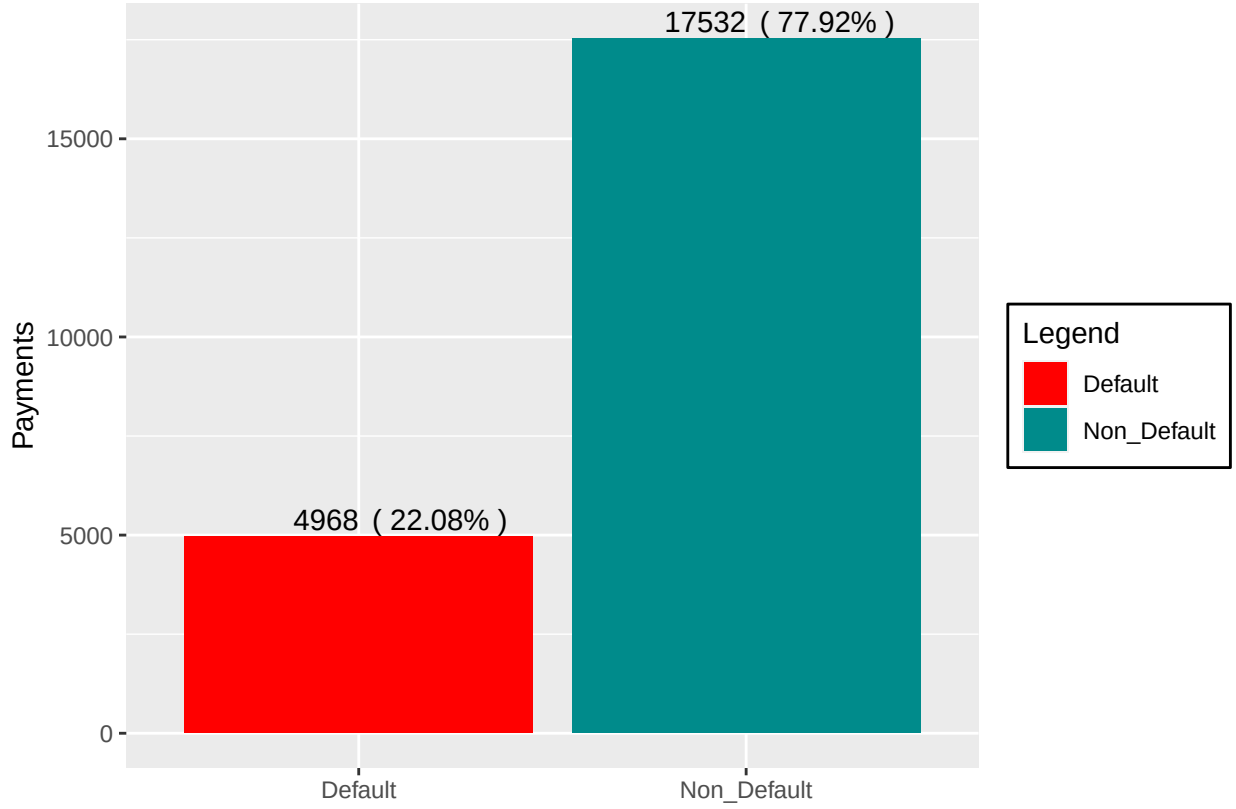
Now, let us split our data into training and test sets.

```
set.seed(1234)
n = length(data$LIMIT_BAL)
index <- 1:nrow(data)
testindex <- sample(index, trunc(n)/4)
test.data <- data[testindex,]
train.data <- data[-testindex,]
```

Data Visualization

Now, let us visualize our data.

We begin with visualizing the distribution of default-ees and non-default-ees.



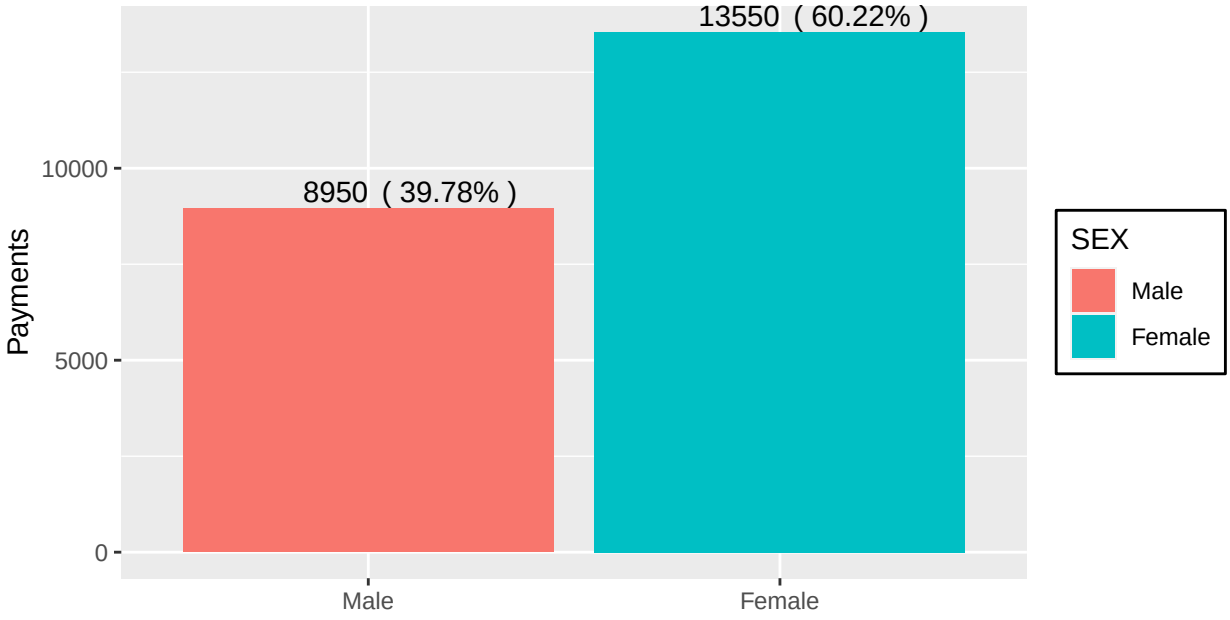
From the plot above, we can see that there of the large number of clients, 77.92% did not default, as compared to 22.08% who defaulted the next month.

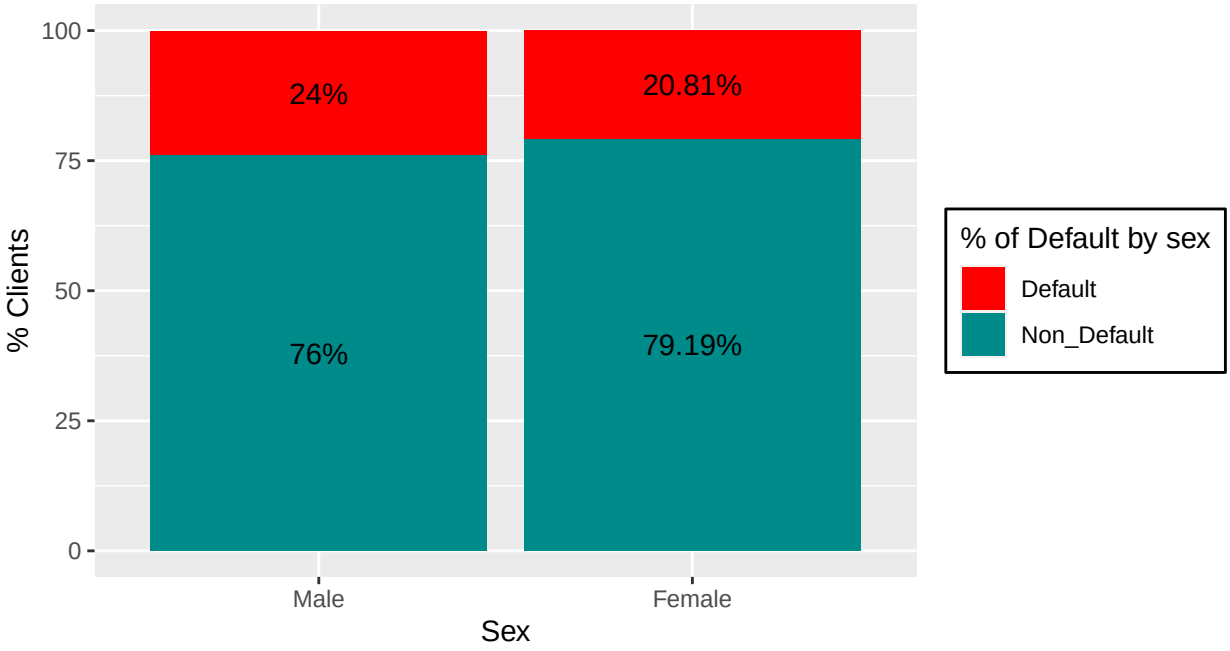
This is a potential issue, as splitting skewed data might result in incorrect model estimates and wrongly computed model accuracy scores.

Also, there is almost thrice as much defaulters as compared to non defaulters. This would pose a problem when splitting the data, as the training set might be populated with mostly the majority class, while the testing set might be populated with the minority class, affecting the accuracy calculated.

Default against Sex

Here, we investigate and compare the default rates between genders.



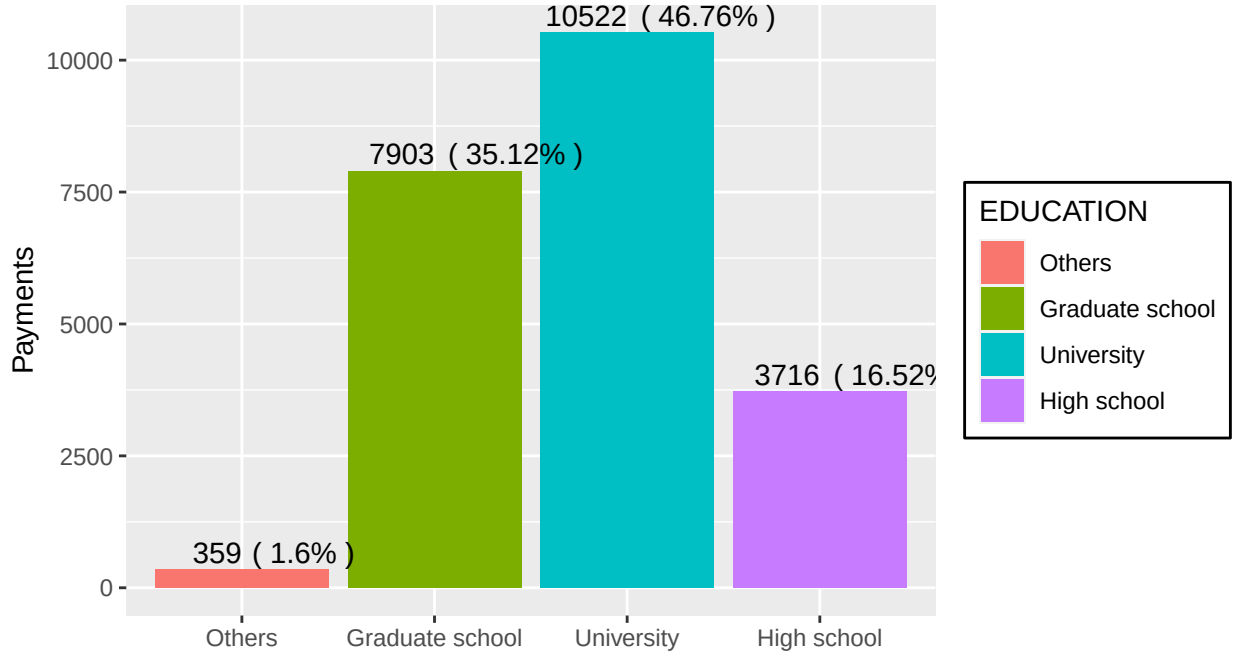


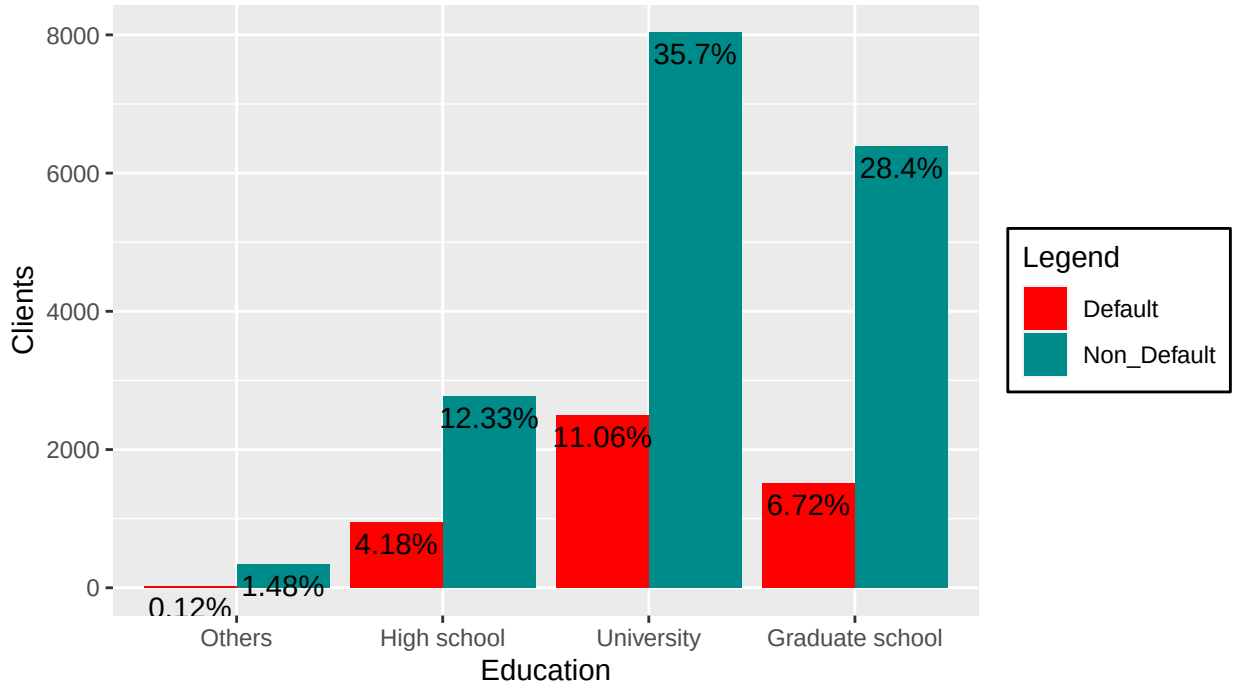
From the first plot, we can see that there is larger number of female clients, 60.37%, as compared to male clients.

However, from the second plot, we observe that despite this, males are in fact more likely to default the next month as compared to females.

Default against Education

As for age, we can do a similar analysis for education level as well.

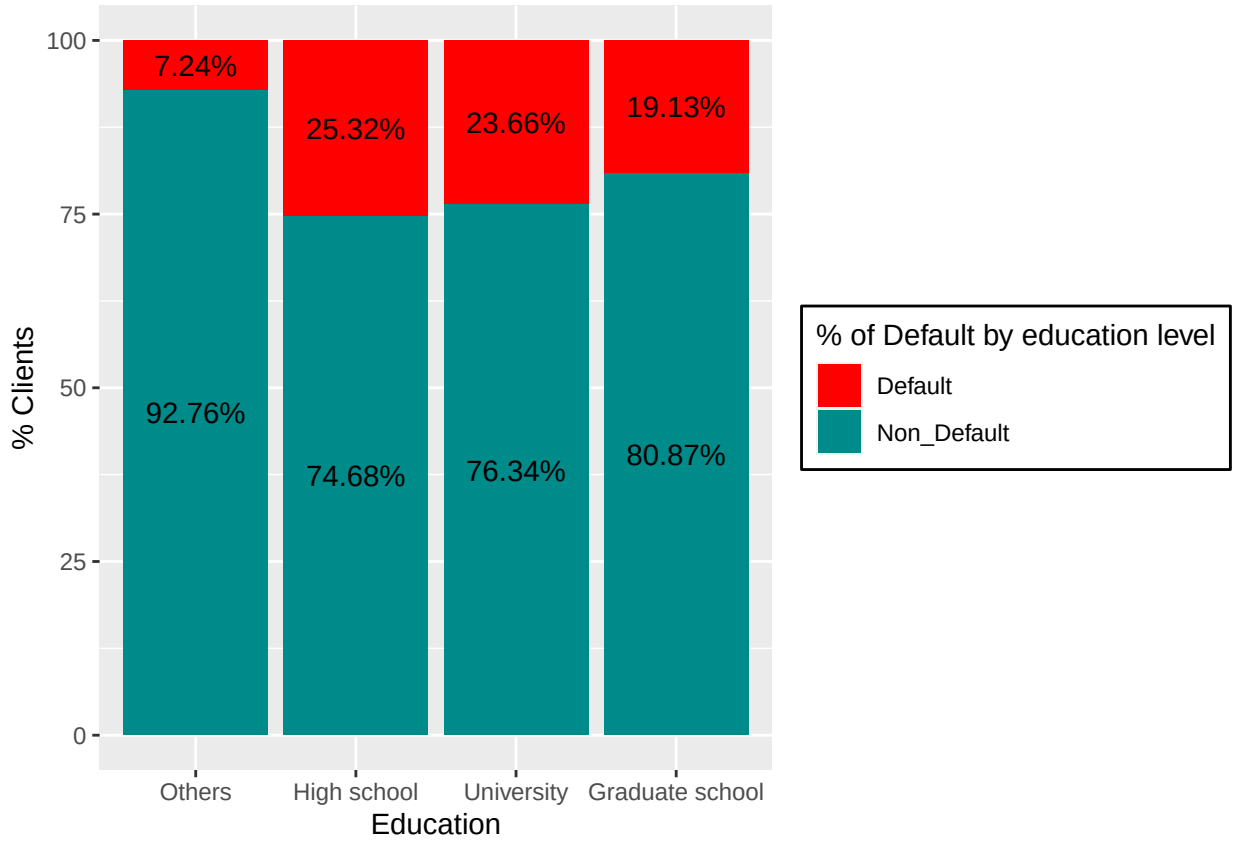




Here, we see that university graduates make up most of the clients. We can also infer that most of the clients, around 80%, are well educated, having graduated from university or graduate school.

Additionally, there seems to be an increase in number of defaults as education level increases. However, as there are different amount of individuals in the groups, it is better to visualize the number of defaults within each education level.

Intuitively, individuals with lower education is more probable to default their credit payments. Let us see whether this is true.



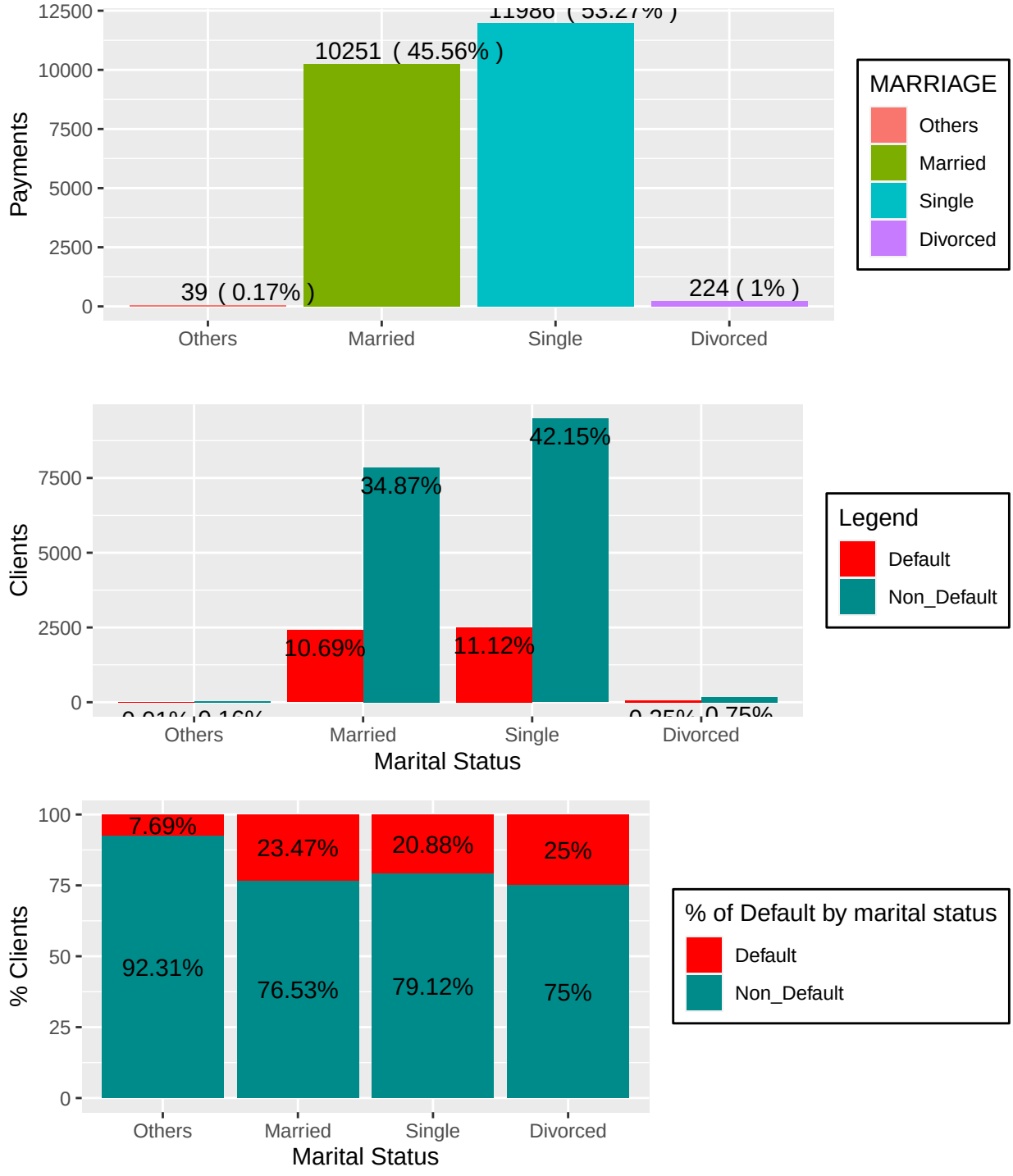
From the figure above, there seems to little difference in number of defaults within each education level.

However, we can see that the higher the education level of the client, the lower the probability of defaulting the next month. As such, clients with a high school education level have the highest probability of defaulting.

Clients with "Others" education level are the least likely to default, but we are unable to know what it represents and its small size will not have much significance in the final result of our models.

Default against Marital Status

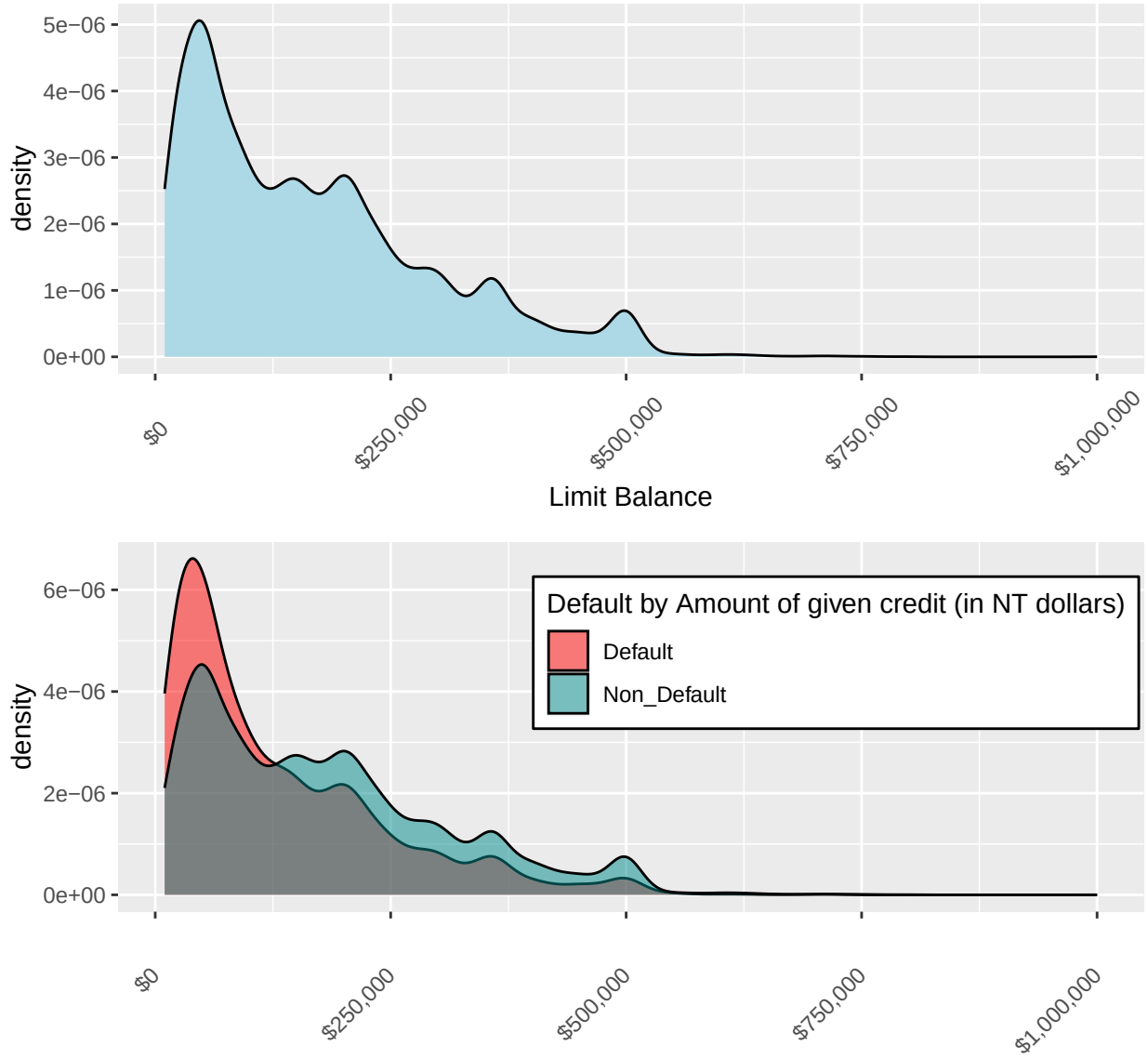
We now observe the relationship between default rate and marital status



Following a similar procedure as before, we recognize that singles make up the highest percentage of clients compared to other marital statuses. There is little difference in percentage of defaults within each marital status, however we can see that divorced clients have the highest default rate, while "Others" have the lowest. Still, we do not know what "Others" represent but its small size will not have much significance in the final result of our models.

Default against Limit Balance

We now observe the relationship between default rate and limit balance.



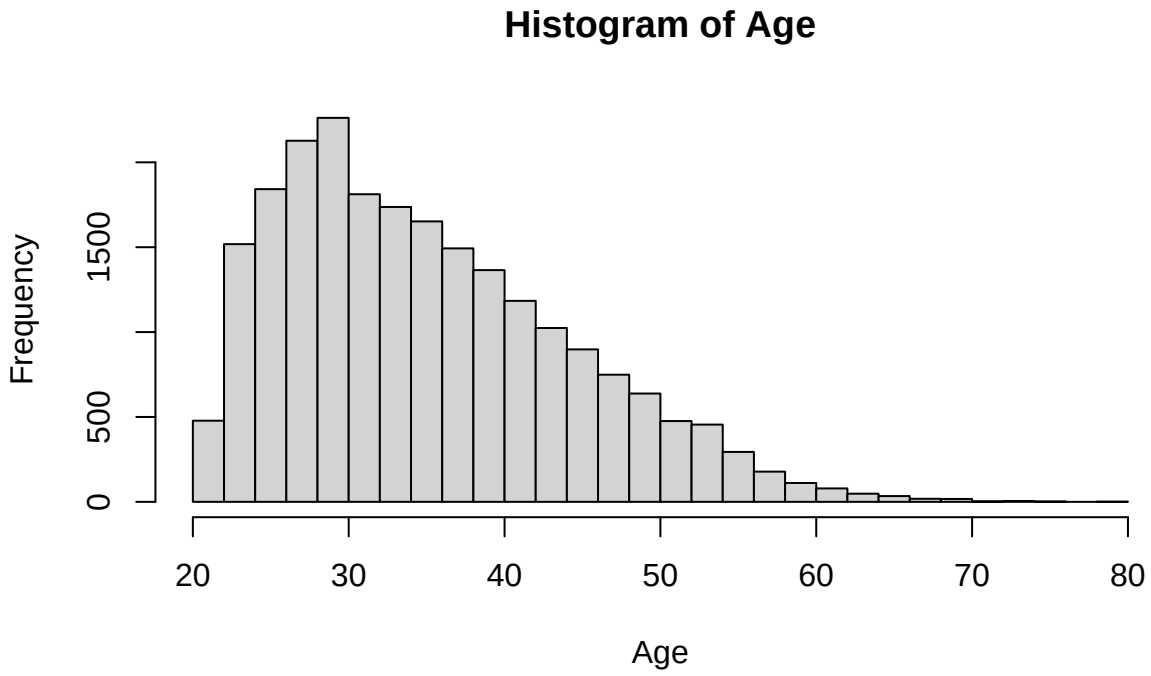
From the first density plot, we can see that most clients are given credit between \$0 - \$500000 and the distribution is skewed to the right.

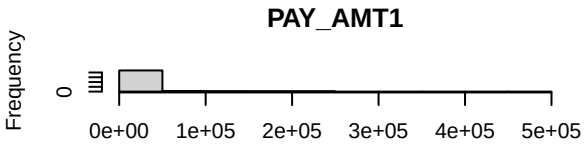
We can also see that the credit can have a very large range, which **can possibly cause problems for some prediction models.**

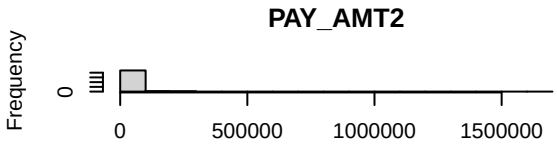
From the second density plot, we can see that at the lower given credit range of \$0 - \$125000, there are more defaulters than non-defaulters.

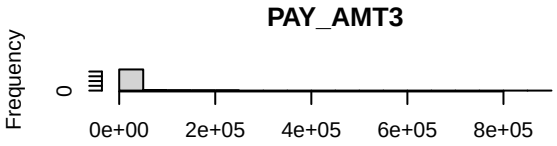
Default against Age, Pay Amount and Bill Amount

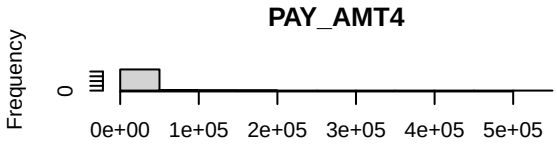
We can also take a look at the distribution of `Age`, `PAY_AMT` and `BILL_AMT`.

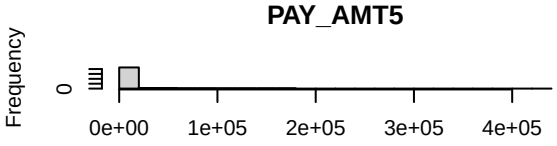


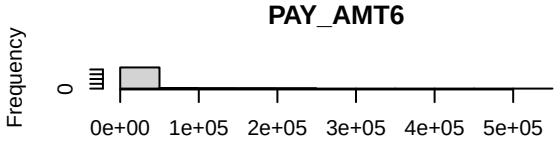


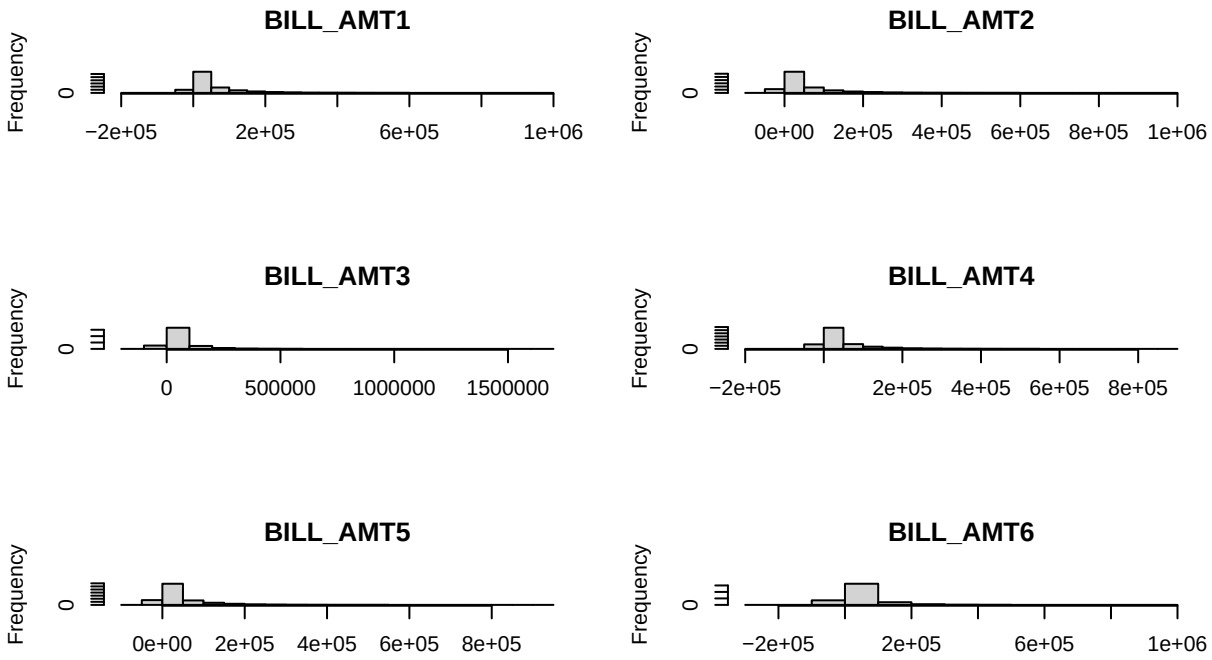












Based on the age distribution, there are more clients between the ages 26 to 30, and the number of clients decreases as age increases past 30.

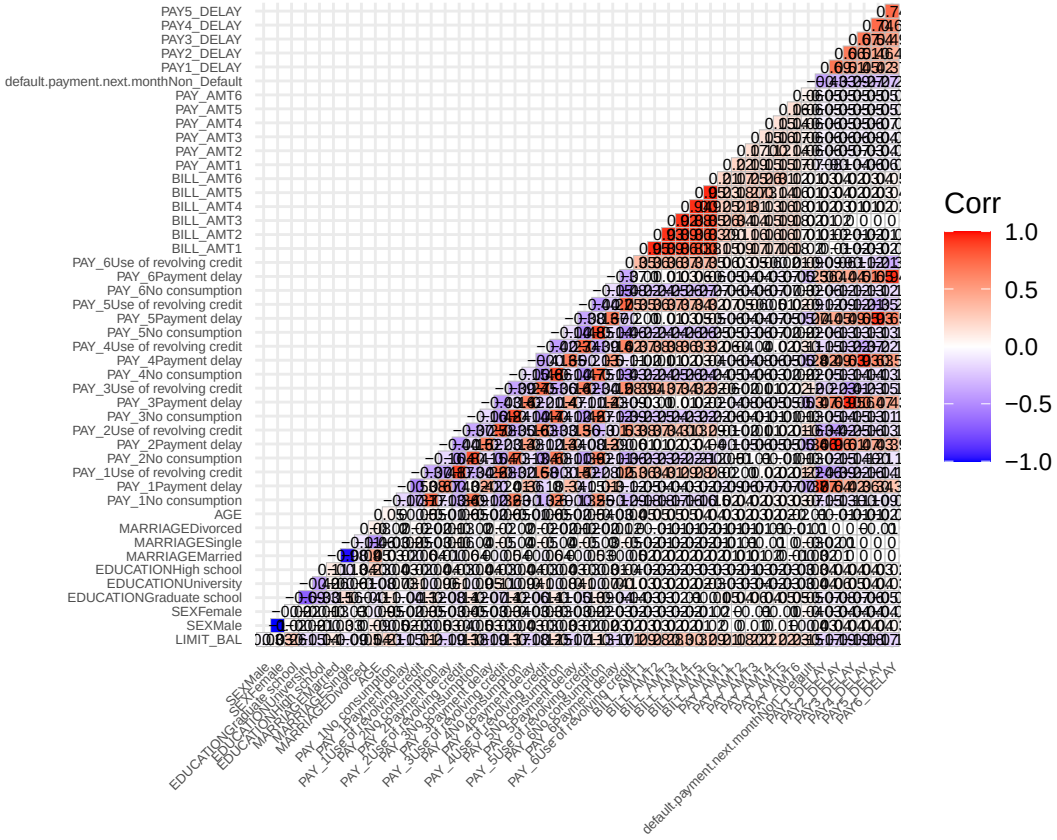
Similar to the `LIMIT_BAL` distribution, the quantities of `PAY_AMT` and `BILL_AMT` have a very large range, so it **might affect our prediction models**.

In all 3 distributions, the data is skewed to the right, so it is important to take that into account.

Correlation Analysis

General Correlation

We will now look at the correlation of our features.

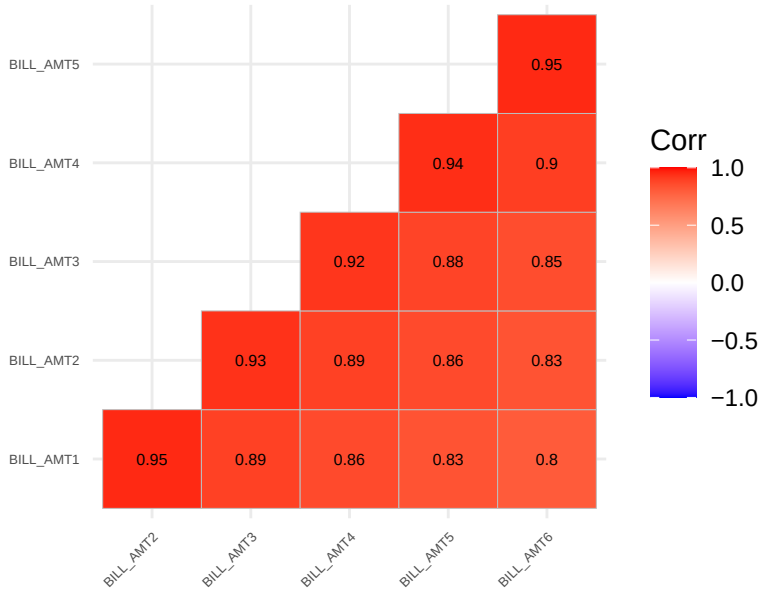


As we can see, there are some features that are highly correlated with each other. They include the `BILL_AMT`, `PAY_AMT` and `PAY`.

We shall take a closer look at them.

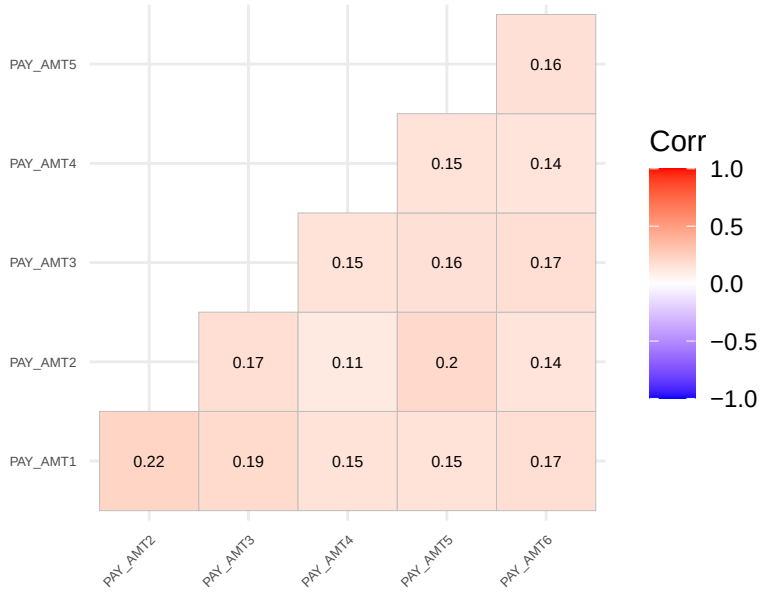
Correlation between Bill Amounts

Zooming in, we have:



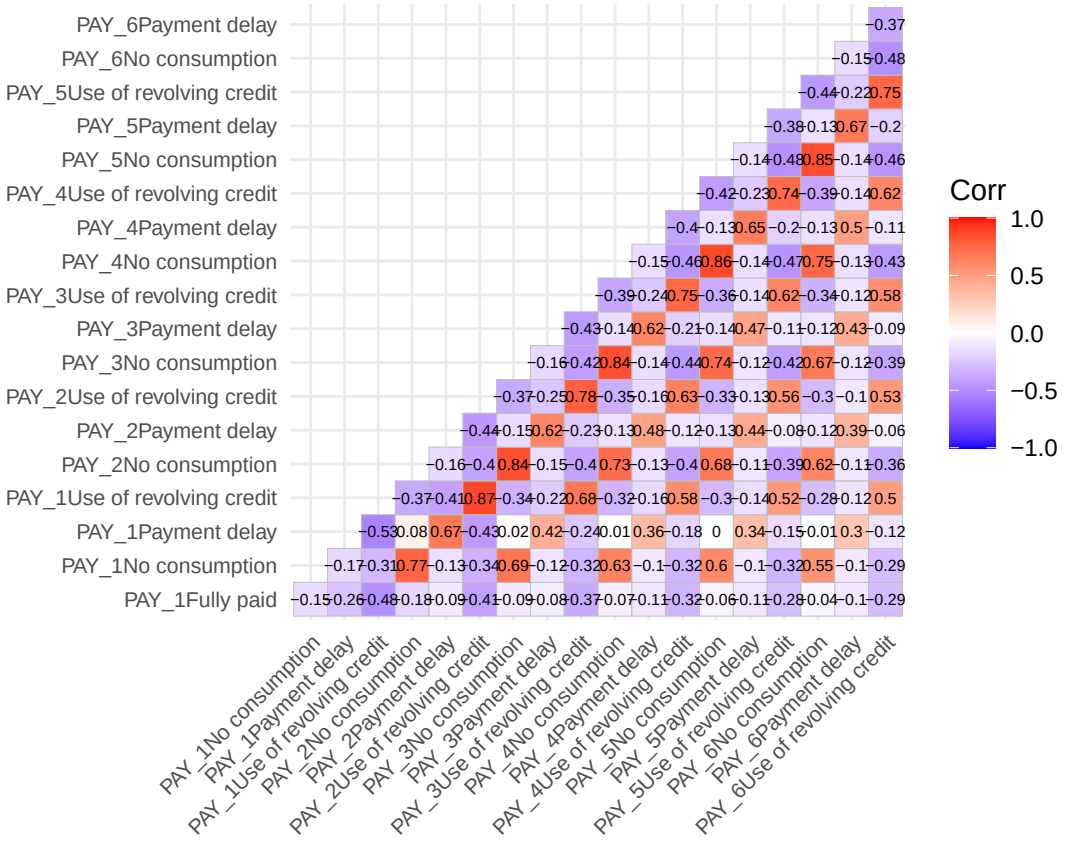
The correlations between `BILL_PAY` are very high. This would mean that if we have an idea about BILL__AMT1, we can know BILL__AMT2 as the correlation is large at 0.95. The correlation between bill statement of a month, `BILL_PAY`, decreases as the difference in months of the other bill statement increases. Let's check the correlation of the `PAY_AMT`.

Correlation between Pay Amounts

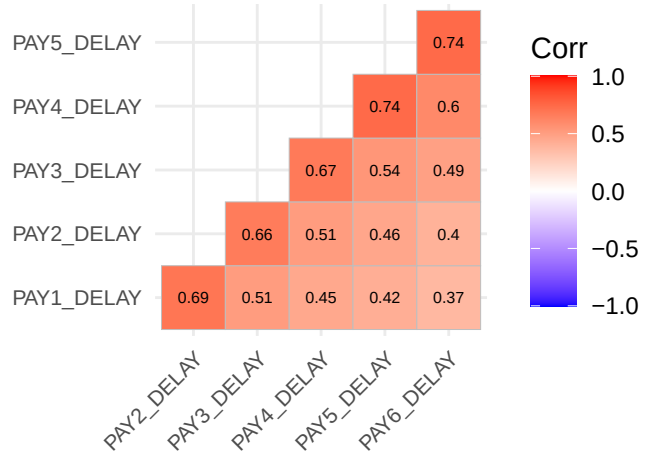


There are no significant correlations between the previous payment amounts for all months. Let's check the correlation between repayment status, `PAY`.

Correlation between Repayment Status



As we can see, there is a strong correlation between same payment statuses (i.e. No consumption, Payment delay, etc), and this correlation decreases as the difference in months of the other bill status increases.



There is also a large positive correlation between duration of payment delays from Sep - Apr. This correlation decreases as the difference in months of the other payment delays increases.

Through thorough analysis of these observations, we can conclude that `BILL_AMT` and `PAY_DELAY` are potentially not useful in our model training, as they have high correlation (positive or negative) with other independent variables.

Data pre-processing

Let us now pre-process our data by standardizing before performing feature selection.

```
standardize.columns <- c("LIMIT_BAL", "AGE", "BILL_AMT1", "BILL_AMT2", "BILL_AMT3", "BILL_AMT4", "BILL_

n <- length(standardize.columns)
for (i in 1:n) {
  column <- standardize.columns[i]
  mean <- mean(train.data[, column])
  sd <- sd(train.data[, column])
  train.data[, column] <- (train.data[, column] - mean) / sd

  mean <- mean(test.data[, column])
  sd <- sd(test.data[, column])
  test.data[, column] <- (test.data[, column] - mean) / sd
}
```

The large range in which the quantities of these numeric variables are allowed to vary might cause issues for our learning models. Additionally, due to the large number of variables in our data, it is essential to ensure that these variables contribute equally in our predictive model.

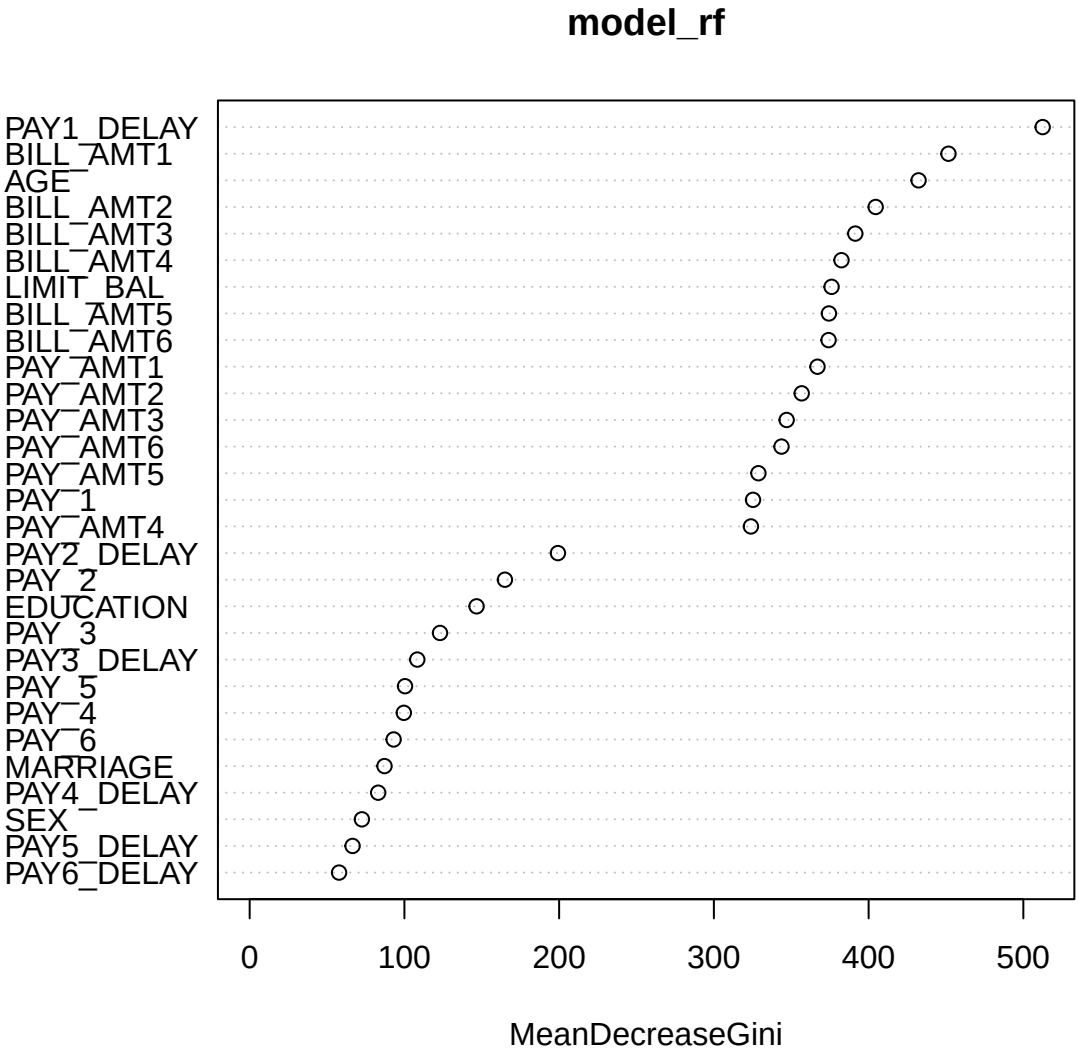
As such. standardization is necessary to put the values on the same scale.

The aforementioned large quantity of variables in our data set hence leads to our next objective of "Feature Selection".

Feature selection

Here, we will the look into feature selection, where irrelevant/redundant features are removed, as the large amount of features can negatively impact the efficiency, accuracy and simplicity of the training models.

We will employ the wrapper method for feature selection, and see whether we can improve the accuracy of our model by using an intelligently selected subset of features, instead of all 29 features.



Instead of creating a large number of possible subsets of features, by forward selection or backward elimination, we will select the top 10 features which have the highest feature importance.

Looking at top 2 features, we notice that `PAY1_DELAY`and `BILL_AMT1` are selected. However, it was already shown **in our data exploration that there is a large correlation within `BILL_AMT` and `PAY_DELAY` across the months**.

Despite these variables begin assumed to be potentially irrelevant in our model training, we can also observe from the above plot that all 6 `BILL_AMT` are in fact included in the top 10 features, while for `PAY_DELAY`, only `PAY1_DELAY` is included. This would mean that our original assumption that `BILL_AMT` is being irrelevant (or not useful) is rejected, while conversely, the assumption for `PAY_DELAY` is not.

Model selection

Choice of Model

Here, we will perform training using two types of models, Generalized Linear Model (Logistic regression) and Support Vector Machine.

For each of the model, we will utilize the top 10 features of highest importance and consequently use them to predict the test data set.

```
fit_glm <- train(default.payment.next.month ~ PAY1_DELAY + BILL_AMT1 + AGE + BILL_AMT2 + BILL_AMT3 + BI
                 data = train.data,
                 method = "glm",
                 trControl = trainControl(method = "cv", number = 10, classProbs = T),
                 preProcess = c("center", "scale", "nzv"))

hat_glm <- predict(fit_glm, newdata = test.data)
```

```
fit_svm <- train(default.payment.next.month ~ PAY1_DELAY + BILL_AMT1 + AGE + BILL_AMT2 + BILL_AMT3 + BI
                 data = train.data,
                 method = "svmLinear2",
                 trControl = trainControl(method = "cv", number = 10, classProbs = T),
                 preProcess = c("center", "scale", "nzv"))

hat_svm <- predict(fit_svm, newdata = test.data)
```

Cost and Benefit Analysis

There are pros and cons of selecting these two models:

Logistic Regression:

- Pros
  - Many ways to regularize the model to tolerate some errors and avoid over-fitting
  - Unlike Support Vector Machines, we can easily take in new data using an online gradient descent method
- Cons
  - It aims to predict based on independent variables, if there are not properly identified, Logistic Regression provides little predictive value
  - Requires observations to be independent of one another

SVM:

- Pros
  - SVM have regularization parameters to tolerate some errors and avoid over-fitting
  - Provides a good out-of-sample generalization, if the parameters C and gamma are appropriate chosen (Might be even more robust even when training sample has some bias)
  - Users can build in expert knowledge about the problem via engineering the kernel
- Cons
  - Bad interpretability as SVMs are black boxes
  - High computational cost: SVMs scale exponentially in training time
  - Users might need to have certain domain knowledge to use kernel function

Additional Analysis

Neural Network Analysis

In this course, another classification model we have learnt would be the Neural Network Model.

The merit(s) of the Neural Network Model are:

- Due to massive amount of data in this data set, traditional machine learning algorithms can possibly reach a level where more data does not improve their performance. On the other hand, deep learning machine learning algorithms such as the Neural Network model can show significant improvement in model performance the more data we input into it, and hence might prove to be a better predictive model.

However, in this report, we did not decide to utilize Neural Network Modelling for the following reasons.

- Black Box:
  - Similar to the Support Vector Machine Model we have built previously, Neural Network Models are also "black-boxes" in nature. This means that on an individual level, we are unsure on how the neurons work together to arrive at the final prediction. Since the SVM model is similar in nature, we concluded that it was unnecessary to train a Neural Network Model.
  - **Additionally, if we contextualize against the data modelling problem statement, it is problematic to use neural networks to predict whether a certain customer is creditworthy, since a reason will be required to reject the customer's credit loan.**
- Computationally Expensive:
  - Due to the large nature of the data set, Neural Network Modelling would be extremely computationally taxing and expensive, and will not be viable (on an efficiency standpoint) with our current lacking computational power.

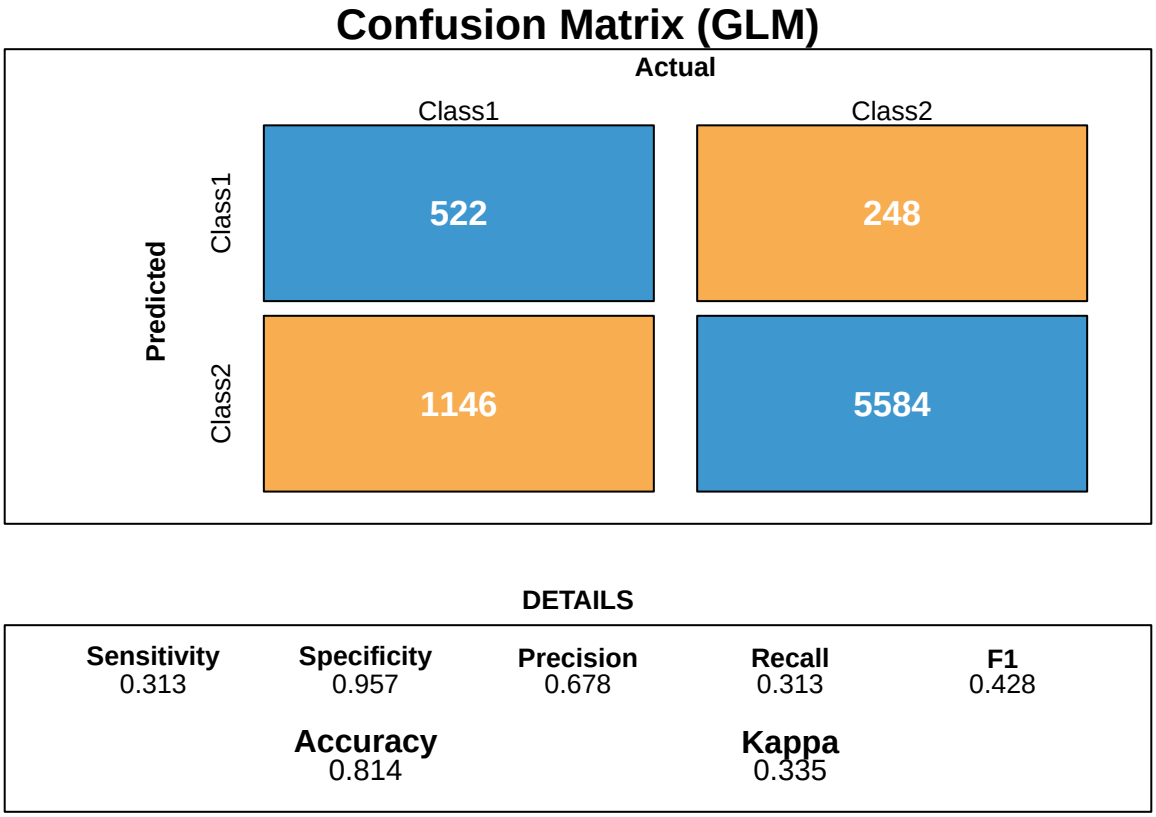
Model evaluation

Model Performance

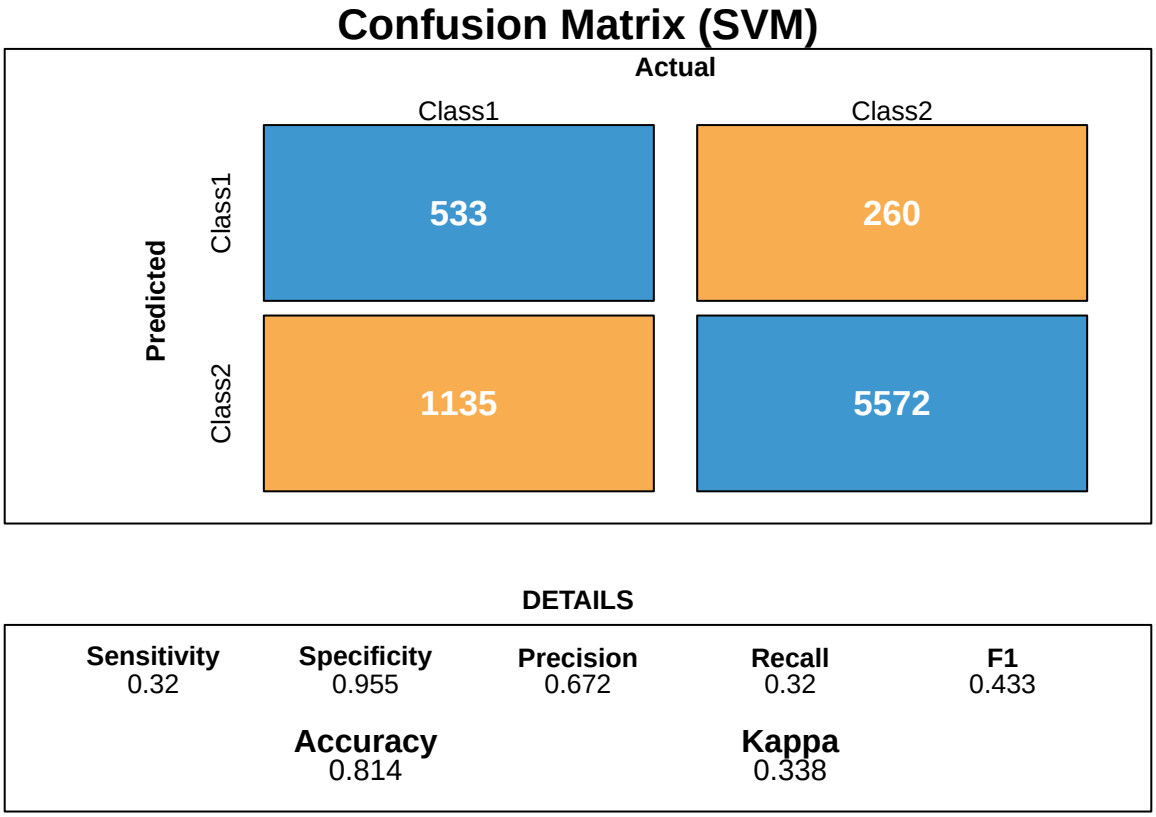
Here, we will evaluate the performance of our models by looking at their confusion matrices and ROC/AUC.

Confusion Matrices

GLM Confusion Matrix:



SVM Confusion Matrix:



The accuracy of both our model are the same at 81.4 percent.

To check if this accuracy is good, we have to check the null accuracy, where we always predict the most frequent class.

```
## [1] 0.7776
```

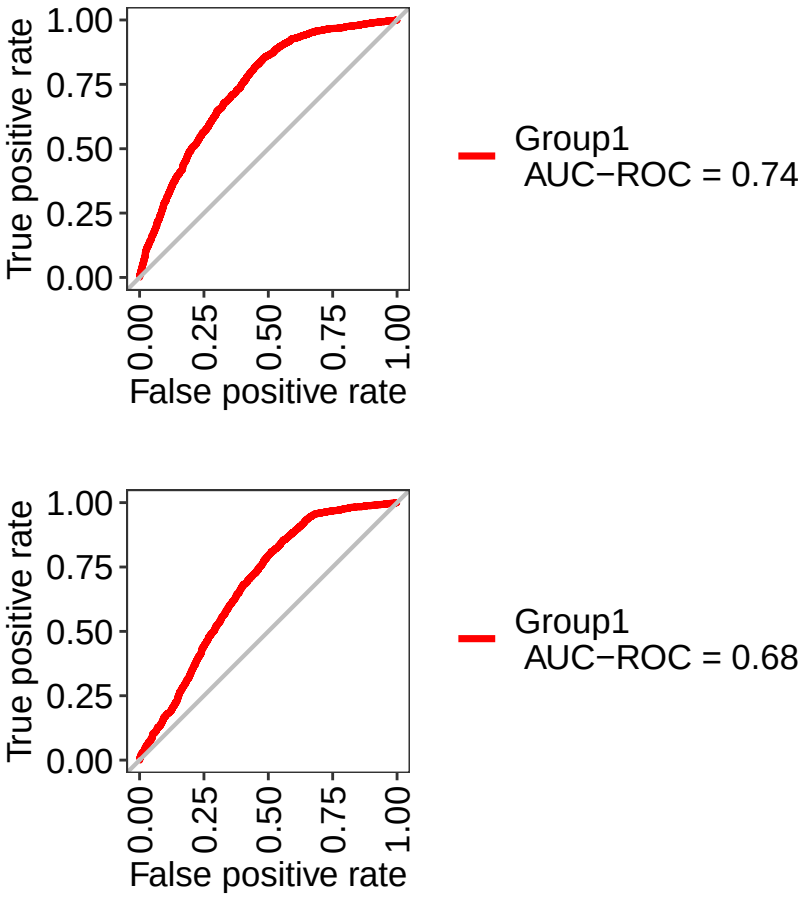
This means that a dumb model that always predicts the most frequent class would be right 77.76% of the time. Our models are definitely much better than this.

As a credit card company, the defaulters are of the biggest concern as they cause the firm to lose money. As such, it is important that we look at the precision and recall of the models, as we are interested in the percentage of defaulters, out of all defaulters, that we predicted correctly (recall), and the percentage of actual defaulters, out of all our predictions of defaulter instances (precision).

Comparing their precision and recall, we notice that the logistic regression model has a recall and precision of 31.24% and 67.75% respectively, as compared to the SVM model's 31.95% and 67.21%. Although the SVM model has a lower precision, its higher recall would mean that more defaulters are accurately identified, albeit a higher possibility of wrongly predicting a non defaulter as a defaulter. Hence, it would be a more beneficial model for the credit card company.

ROCs and AUCs

ROC curves of the respective models are as follows:



Suppose we pick one defaulter and one non defaulter observation. AUC represents the likelihood that the classifier will assign a higher predicted probability to the positive observation, defaulter. Looking at their ROC plots, we can see that AUC for the logistic regression model is higher, at 0.74 as compared to 0.68. This would suggest that the logistic regression is a better classifier.

Conclusion

As a whole, the confusion matrix shows that the SVM model is better than the logistic regression model, while the AUCs shows the opposite. However, computational time is a huge factor when deciding the best model to choose based on the available data, limited resources, cost, and performance. SVM's computational time would grow much faster than the logistic regression, and with the amount of client data increasing over the years, it would be better to choose the logistic regression model, yielding better performance efficiency wise.

Hence, if we were to choose a model to recommend to the company, we would go with the logistic regression model.

Additional Improvements

Possible Model Improvements (GLM tuning)

To further improve our models , we can employ the use of model tuning.

Model tuning achieves the best results by maximizing model performance through the process of scouring to obtain the best hyperparameter values. Note that hyperparameters are the set of values whose values cannot be estimated by the model through the training data.

With respect to our trained logistic regression, we can incorporate model tuning to further improve the model.