Group Members:

Rannuo Li   
Yuanliang Song  
Chenghao Wu

IST707 Final Project Report

Group 2

## 1.Introduction

This dataset 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. There are 25 variables:

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, PAY\_3, PAY\_4,PAY\_5 and PAY\_6 (scale the same as above). BILL\_AMT1: Amount of bill statement in September 2005 (NT dollar)BILL\_AMT2,BILL\_AMT3,BILL\_AMT4,BILL\_AMT5 and BILL\_AMT6. PAY\_AMT1: Amount of previous payment in September 2005 (NT dollar). PAY\_AMT2,PAY\_AMT3,PAY\_AMT4, PAY\_AMT5 and PAY\_AMT6 default.payment.next.month: Default payment (1=yes, 0=no)

Given the background of this dataset, we are going to find important features and predict who is likely to default. We perform the following operations:

1.Processing data

2.Descriptive Analysis

3.Feature Engineering

4.Training model using SVM, Random Forest and NeuralNet algorithms

## 2.Data Processing

1.Firstly, we imported the CSV file as the data frame.

2.We deleted the first column with customer ID which would be useless when predicting.

3.Besides, we replaced Education levels 5 and 6 with 0 since Education 5 and 6 mean nothing. Also, we notice that marriage has level 0, which is unknown too. Then we remove those rows.

4.We put cleaned data into a new data frame.

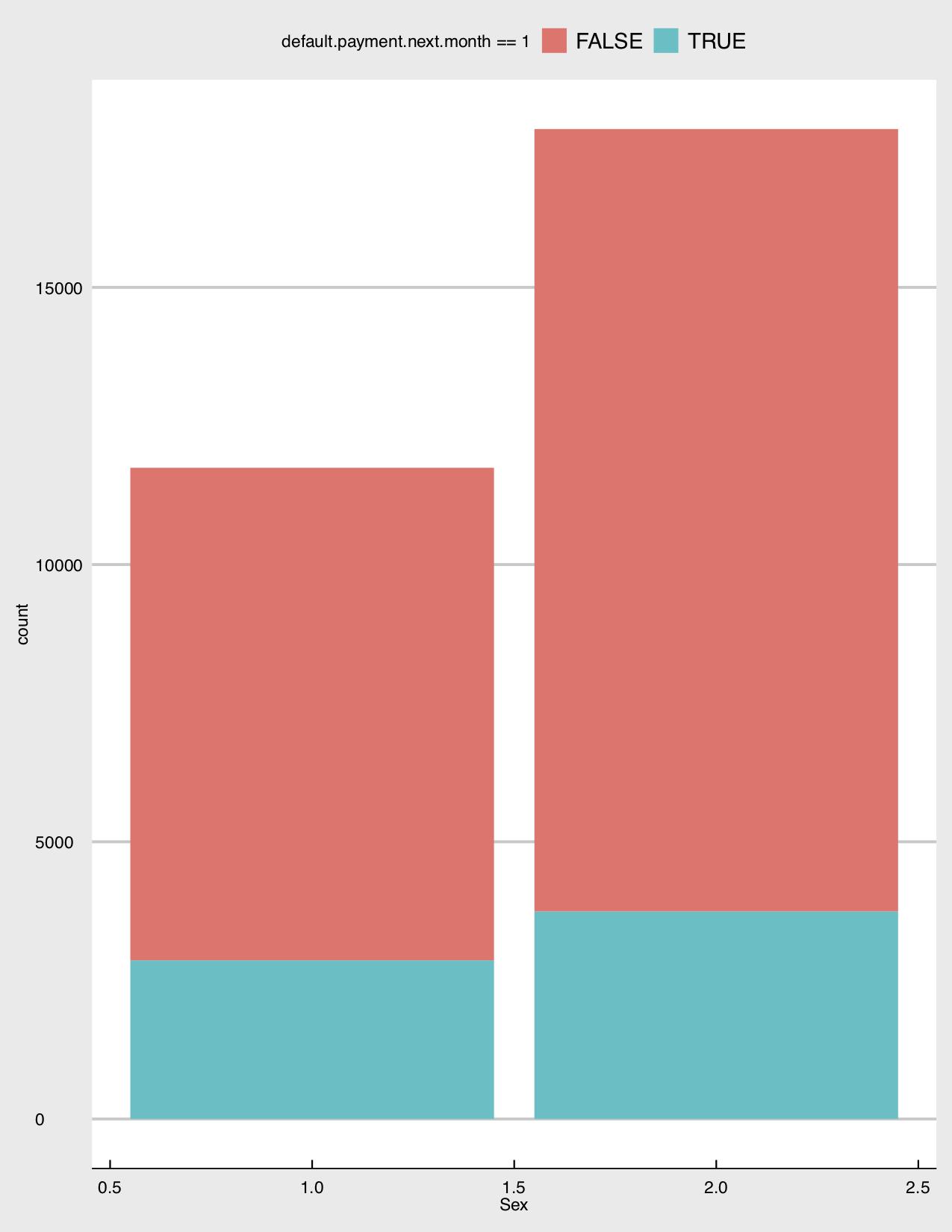
5. Lastly, we converted quantitative variables to new factorial variables.

## 3.Dataset Exploration

By using ggplot2, we explore the relationship between default and sex, education, age, marriage variable.

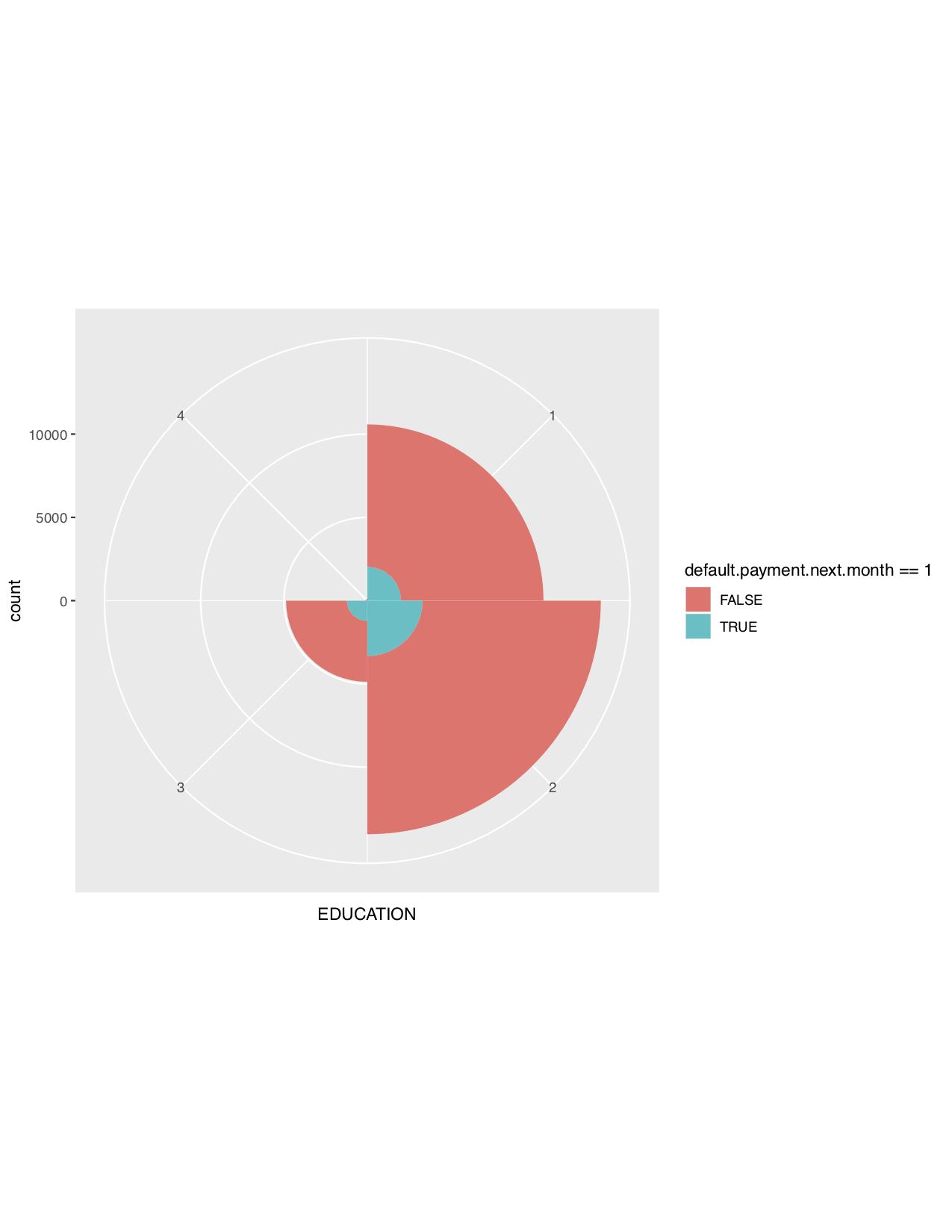
1.Sex

we plot a bar plot to demonstrate the relationship between sex and their default records. From the demographic below, we are able to know that more than half of our customers are female, and the male has a relatively higher tendency to default since the plot shows a higher percentage of default for male customers.



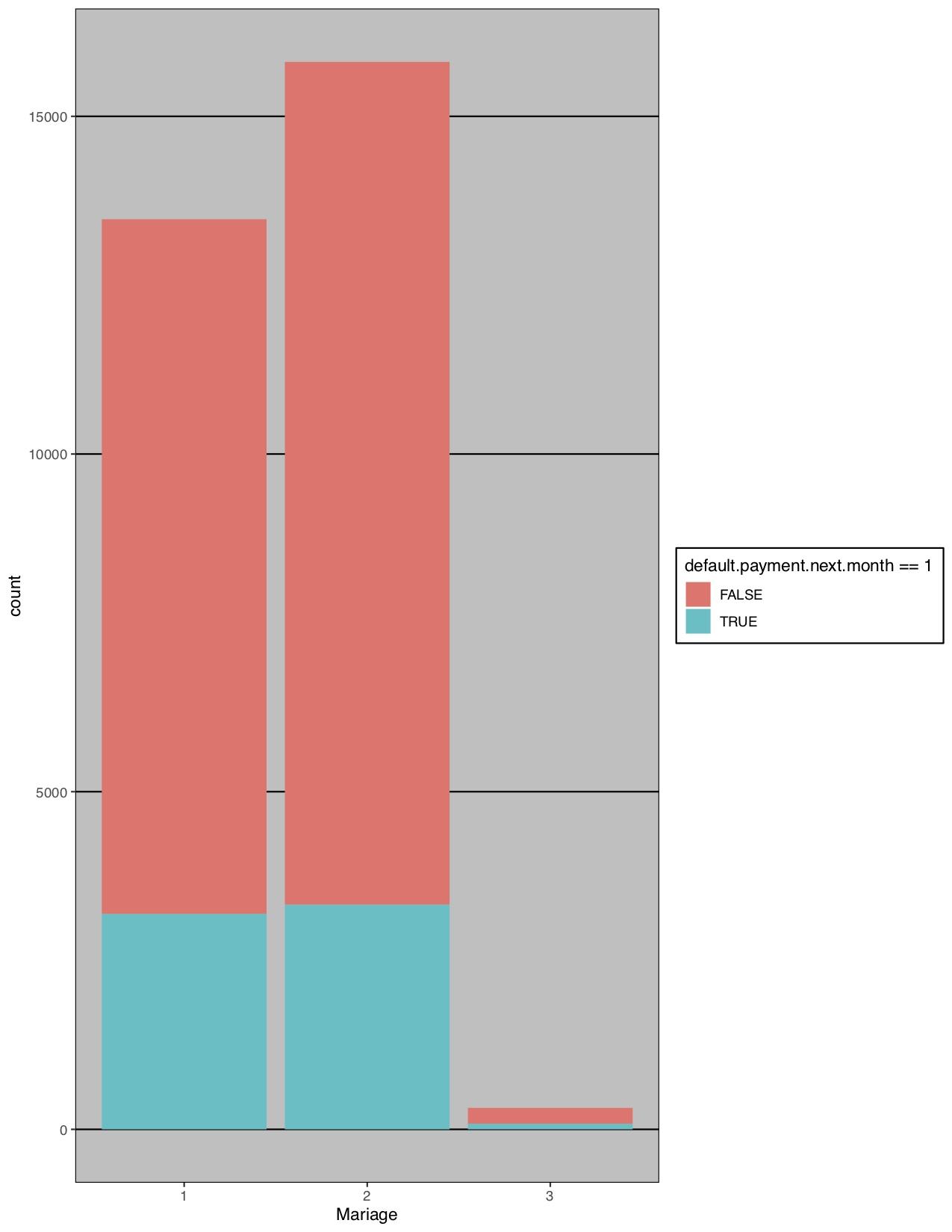
2.Education

According to the radar chart below, we can clearly see that the composition of our customers mostly has a university or graduate school degree. Among them, people have graduate school degree has the largest number of default, but we cannot draw a conclusion that they are more likely to default because all three levels have the relatively same percentage of people who default.



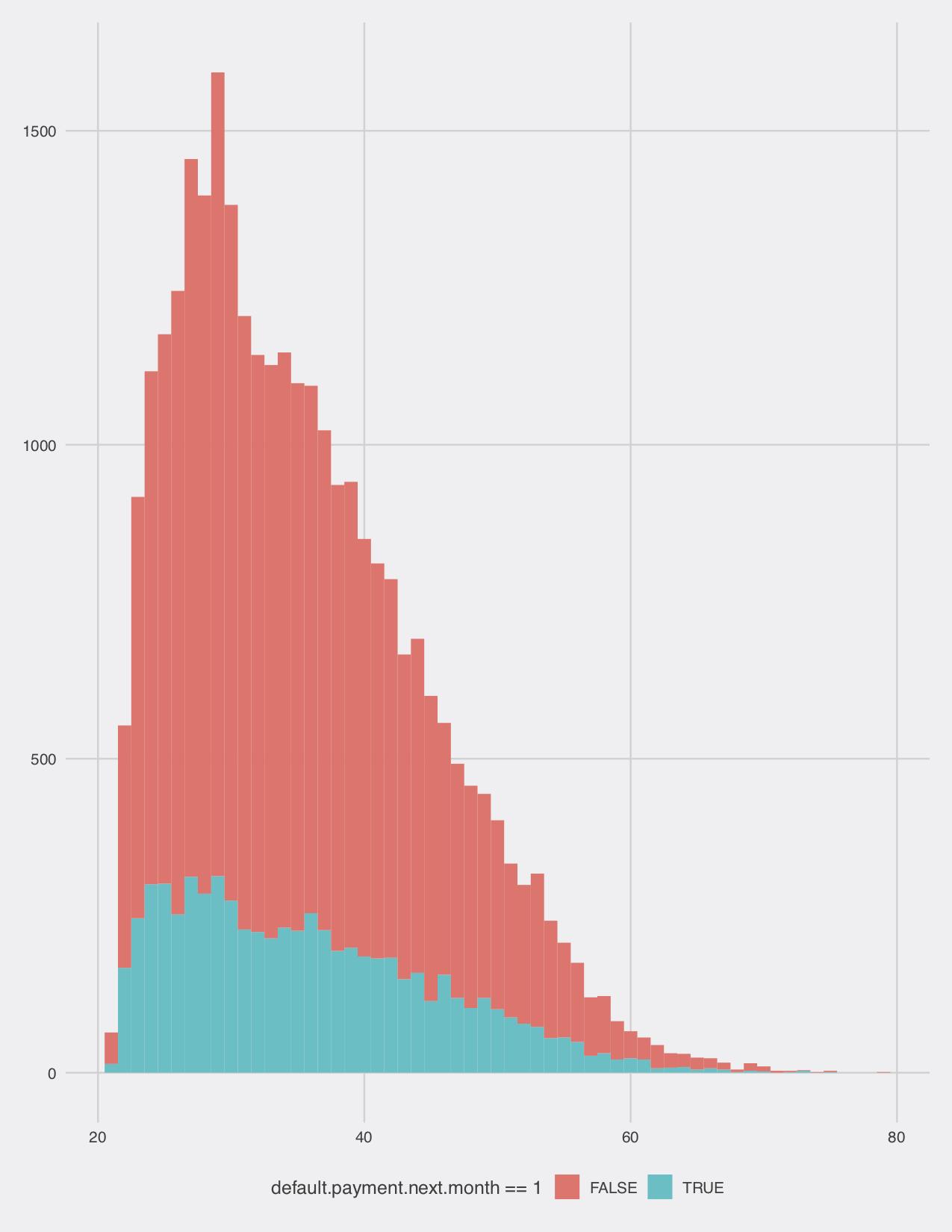
3. Marriage status

The bar chart below illustrates the relationship between the marriage status and default. We can omit the marriage status of others. More than half of our customers are single person and they have a lower tendency to default compared to people who have marriage. They have approximately the equal number of people who default.



4.Age

Lastly, we analyzed the relationship between the age and default by using the histogram below. It’s easy to see that it’s a right-skew graph and most customers are between 30-40 years old. And the number of customers and default customers is decreasing as their age increased.



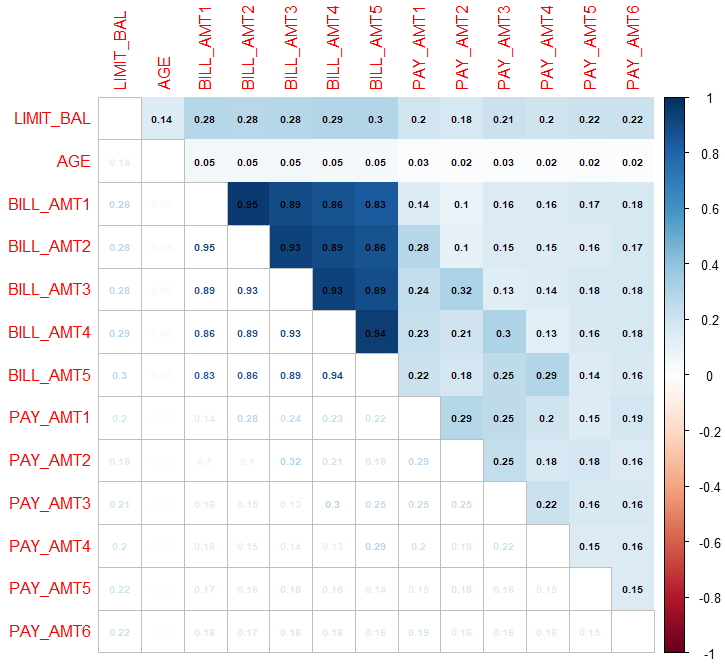
## 4.Feature Engineering

### 4.1 Data Cleaning

We firstly normalizing numerical variables since for some models，predictions are better if data is scaled and centered. Creating DummyVars for categorical variables is also done since some of our fitting models can not take text values or factors(NN), we convert them to dummy variables. Constant and almost constant predictors across samples (called zero and near-zero variance predictors) happen quite often. One reason is that we usually break a categorical variable with many categories into several dummy variables. Hence, when one of the categories has zero observations, it becomes a dummy variable full of zero. These are uninformative predictors. So removing Near Zero Variance variables for better prediction modeling is important.

### 4.2 Correlation and feature combination

We check the numeric features’ relationship using the correlation matrix as below. The goal of the correlation matrix is to see how relevant each feature is, what is its meaning, if it can be used to create new features, and, as usual, play a bit with other basic techniques. When we reflect on the correlations between limit balances, bill amounts, and payments amounts; it presents us that there’s a low correlation between the limit balances and payments and bill amounts. However, it can be seen that bill amounts have a high correlation between each other as expected since the bills reflecting the cumulative amounts. We try to process these high correlation features using feature combination and PCA, we create new columns for total bill amount and total repayment for all clients and choose the principle components which could explain 90% variable. The feature combination seems to perform better on later model work. Maybe the relationship between variables could not only be linearly explained.



### 4.3 Data unbalance

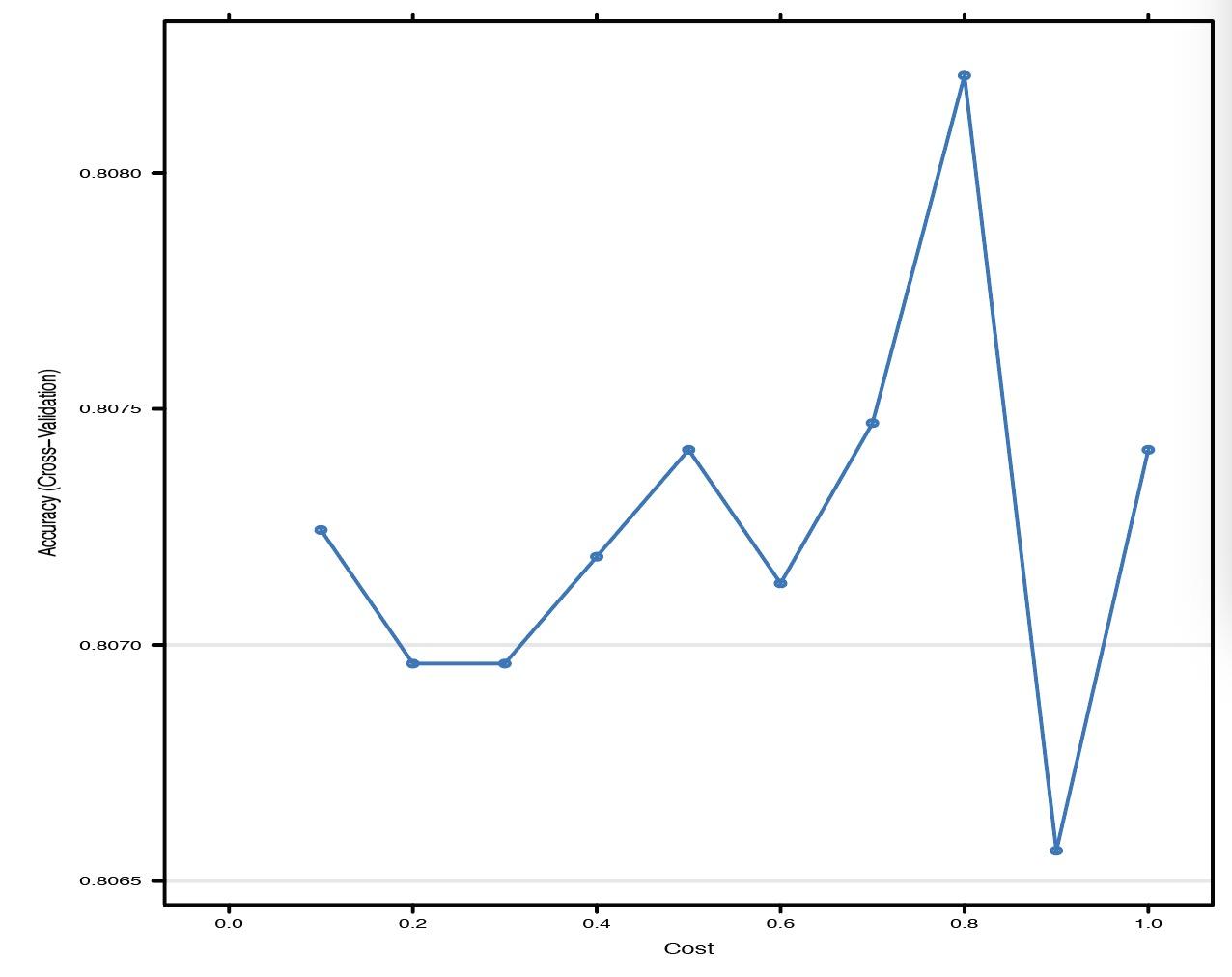
We check the data balance. Models tend to follow the majority of the data. if data is unbalanced then the model will have a preference for data that is in the majority. Accuracy will also be effective because of unbalanced data. For this dataset, the proportion of 0 of default.payment.next.month is approximately 0.8. We can see the above result that our data is unbalanced as we have more of '0' then '1'. Luckily, the caret package makes it very easy to incorporate over- and under-sampling techniques with cross-validation resampling. We can simply add the sampling option to our trainControl and choose down for under- (also called down-) sampling. For over- (also called up-) sampling we simply specify sampling = "up".The rest stays the same as with our original model.

## 5.Model Fitting and evaluation

### 5.1 SVM

An SVM performs classification tasks by constructing hyperplanes in a multidimensional space that separates cases of different class labels. For our model, we have used the *svm()* function in the *e1071* package that implements the SVM supervised learning algorithm.

We use an SVM “Hyperplane Classifier” as we have two possible classes into which we group our customers. This class is calculated based on if they are default customers or not. An optimal hyperplane is the one with the largest margin between classes. For our SVM, we divide the data into training and validation and test datasets. 0.6 of the data is used to train the model, and 0.3 of them is used to validate the model, the remaining for testing.

For training the best tune model with the best average accuracy, we created a grid to adjust parameters using the cross-validation method and set C between 0-1.we generate a plot as below. We found when C=0.8 it has highest accuracy.

After tuning the model we ran the model when C= 0.8 again and we can use the confusion matrix to describe the performance of this classification model. Down below is the confusion matrix . The accuracy is 0.8217.

|  |  |  |
| --- | --- | --- |
| Prediction\Reference | 0 | 1 |
| 0 | 6697 | 1322 |
| 1 | 272 | 650 |

Accuracy : 0.8217

95% CI : (0.8136, 0.8296)

Sensitivity : 0.9610

Specificity : 0.3296

### 5.2 Random Forest

Random forests attempt to improve the generalization performance by constructing an ensemble of decorrelated decision trees. It uses bagging to select different bootstrap samples of training data for learning decision trees.

Compared with the decision trees in classification, a random forest can produce a more stable and accurate prediction. And more trees in the forest, the more robust would be the prediction and hence more accuracy.

We trained a Random Forest with training data having ntree= 500, which means 500 decision trees in the forest in total. In general, the more the number of decision trees in a random forest, the better accuracy it would be, which means lower bias while higher variance. However, there is a consequence of bias being too low that the model is likely to be overfitted. Having that thought in mind, we choose to use a number of trees of 500 to ensure a high accuracy while being able to avoid overfitting.

In order to train the best tune model with the best average accuracy, we created a grid to adjust parameters using the cross-validation method. After tuning the model, we got the result with mtry=4.89, which means the number of features that would be included in each decision tree in the random forest. Using the above results, we have acquired the model with the highest accuracy of 0.81.

After tuning the model, we run the final Random Forest model with mtry=5( the closed integer to 4.89) and ntree=500.

We can use the confusion matrix to describe the performance of this classification model. And down below is the confusion matrix and statistics for the random forest.

Confusion Matrix and Statistics

|  |  |  |
| --- | --- | --- |
| Prediction\Reference | 0 | 1 |
| 0 | 4623 | 916 |
| 1 | 257 | 500 |

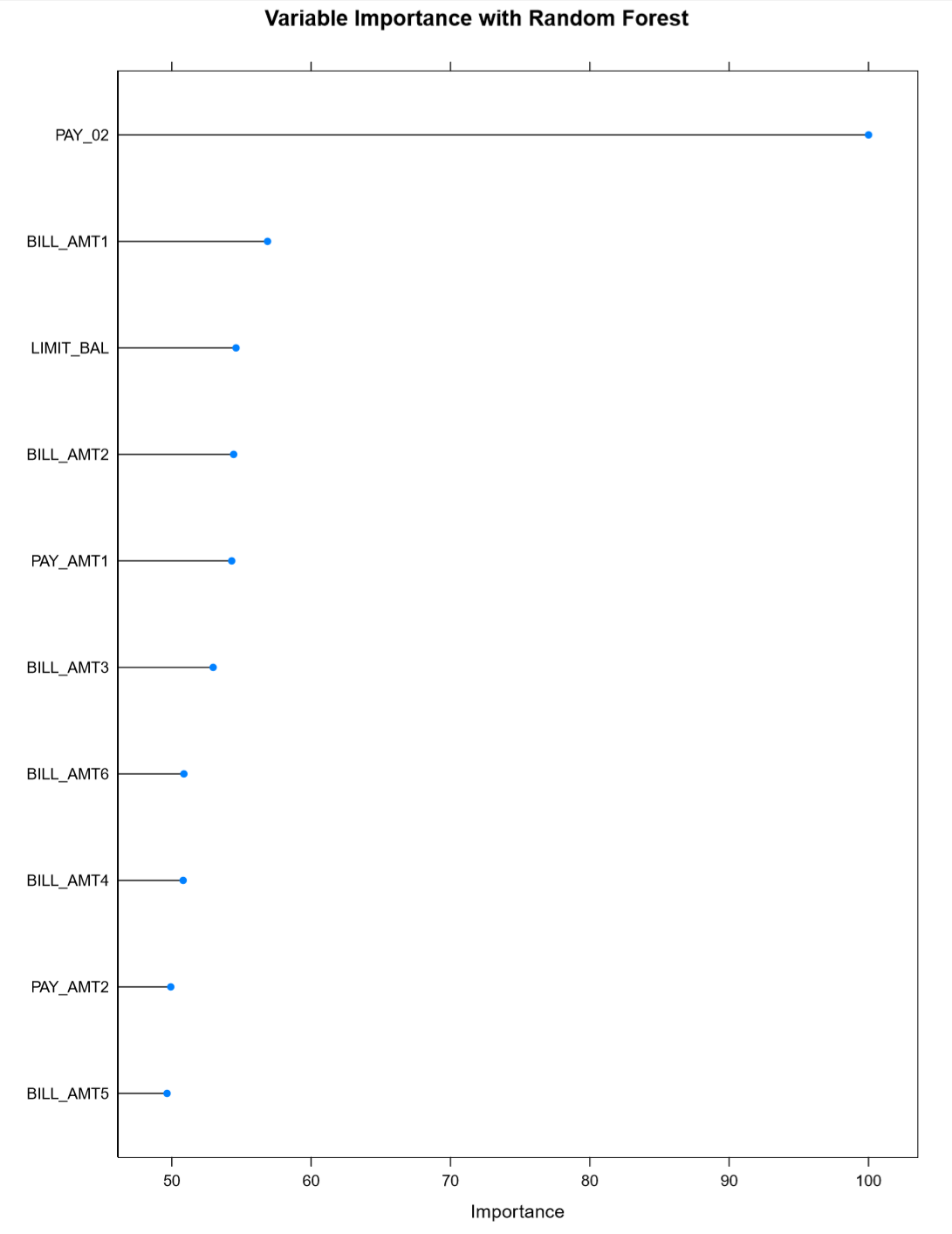
Accuracy : 0.8137

95% CI : (0.8038, 0.8232)

Sensitivity : 0.9473

Specificity : 0.3531

Feature inportance of random forest



### 5.3 Neural Network

A Neural Network is formed with a number of Perceptrons working in conjunction with each other and creating a model used for decision making. The hyperparameters in NN include a number of hidden layers, learning rate, and the activation function. In this case, we use the caret package to train the neural network model and search grid to adjust parameters and find the best tune model with the best average accuracy. For resample method, we use repeated cross-validation which randomly divides the data into 3 blocks of roughly equal size, each of the blocks is left out in turn and the other 2 blocks are used to train the model. And this process will be done twice.

For the hyperparameter in Neural Network, we start at 1 layer neural network, the numbers of nodes start with 3, 5, 7, 10 - then based on results increased and decreased numbers then reached this value. The entropy fitting dacay value(for regularization ) started with - c(5 \* 10^(-5:-1), 0.25 ) then reached at the value used. We finally get a 26-3-1 network with 85 weights, with 26 input variables, 1 hidden layer and 1 output. The hidden layer has 3 neurons. The max iterations are 200, and the initial weight is 0.8, the entropy fitting decay is 0.005.

For this classification model, we use the confusion matrix to describe the performance of it on the test dataset. The outcome is below:

actual

predictions 0 1

0 6623 1315

1 342 704

Accuracy : 0.8156

Precision : 0.67304

Recall: 0.34869

We could see that our model is high accuracy and precision in the validation dataset and low sensitivity. In this model, we try to explore the important features, the top 10 features are below:

importance fields

1 -25.4328674 BILL\_AMT5

2 -15.1779547 BILL\_AMT1

3 -11.8210943 BILL\_AMT3

4 -8.9953835 PAY\_AMT4

5 -8.6787854 PAY\_AMT5

6 -6.6945189 PAY\_AMT2

7 -4.8219024 PAY\_AMT3

8 -3.3787762 PAY\_AMT1

9 -2.4532366 PAY\_AMT6

10 -1.9493336 LIMIT\_BAL

It seems that a little different from other models’ outcomes. We will discuss and compare them in the latter conclusion. Now, we will try to increase the complexity of the model. We increase the nodes of layers to a 26-4-1 network with 113 weights, which increase the model accuracy to 0.8176. It seems to be a little useful on the improvement of accuracy. We could try to increase the layers of the neural network, however, it really takes a lot of time.

## 6.Conclusion

By comparing the three models we trained above, we can see that the neural network has the highest accuracy of 0.8156. However, the random forest has the fastest running time among those three. Thus, our suggestion is if you are going to consider accuracy more other than time efficiency, you should use the neural network to predict the customer credit default. Meanwhile, if you are going to weight time efficiency more, you should use a random forest to do so and be confident as the difference in the accuracy of these two is slight.

Shiny App for our project:

<https://yuanliangsong.shinyapps.io/Project/>