Credit Card Project Writeup

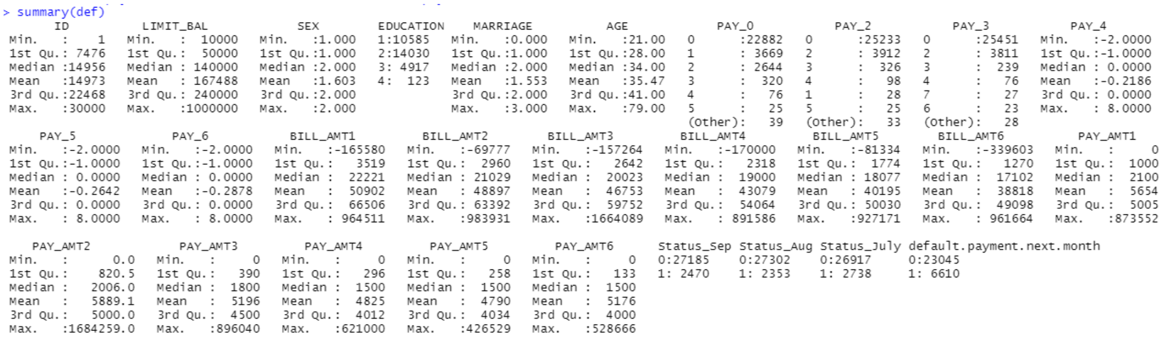
I. Summary Statistics and Data Cleaning

At the first glance of the data, the variables sex, marriage status, and age seemed irrelevant to whether the next payment in credit card will default or not. Also, the process of collecting those data could have embedded some biases, so I decided to exclude them from the analysis. Below were a few assumptions I made for the remaining data cleaning process:

1. For the variable "education", as data description only specified what values 1 to 4 mean, I assumed any number out of that range was an error and assigned it as "NA". In total, there were 345 NAs, and I removed the rows with missing values, given that 345 was a small portion of the data set and would not necessarily change the data structure.
2. In real life, if card holders missed payments for three months, their status and the next payment would be automatically considered as default. That means the credit history in the most recent three months could have the strongest correlation with default payment. Therefore, I picked the payment history in the last three months (X6 to X8) to be my predictors. Similarly, If I were using the bill statement and payment amount, I would only consider those of September, August, and July.
3. The variables "payment history" (X6-X11) have some values other than those specified in the data description. More than 10 thousand data has a value of 0, and we are not sure about what it means. As the amount of the data with value of 0 is more than 1/3 of the data set, I suspected it might be caused by a clerical error where researchers might have put a wrong Excel formula and the data has been assigned to a value higher than the original value by 1. In other words, the data with the value of 0 might indicate they paid the balance on time. In data cleaning section, I assigned the value of 0 to all of the payment paid as R sometimes cannot process negative numbers for categorical variables.
4. I assumed payment often occurs one month later than the bill statement, so I decided not to use the bill statement data in September. More specifically, if I want to know if the payment in August is paid on time, I will probably use the bill statement in August and payment amount in September.
5. Normally speaking, if card holders are able to pay at least 2% of their billing statement every month, the payments will not be considered as default.

When I looked further into the data, I noticed that there might be some correlation between the amount of bill statement and the amount of previous payment. If there is a higher balance in the bill statement, the card holders probably need to pay more to satisfy the minimum amount of the payment (roughly 2%). I wondered if I could create a new variable by using bill statement and previous payment to further explain their relationship. In the Excel sheet, three columns are added and labeled as "Status_Sep", "Status_Aug", and "Status_July", which represent the payment status in September, August, and July (referring to the second assumption). I set the outcome to be binary with value 0 and 1. If the payment amount is equal to or greater than 2% of the bill statement, it returns 0, meaning the payment does not default; otherwise, it returns 1, suggesting the payment defaults for that month.

I also decided to drop “bill statement” and “previous payment” for my predictors because it seems like it is simply exploring if a higher or lower amount of bill statement or payment would impact the default payment, but not the relationship between the two variables. I applied log function to the only numeric variable “LIMIT_BAL” and converted the categorical variables. The statistics is summarized below:

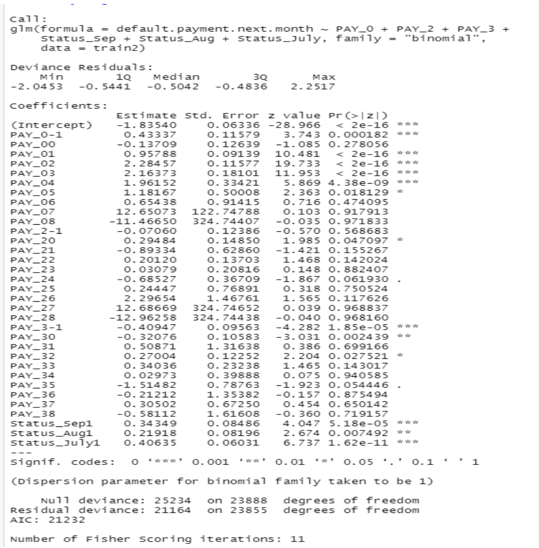


```
> summary(def)
       ID          LIMIT_BAL            SEX        EDUCATION    MARRIAGE          AGE            PAY_0           PAY_2           PAY_3           PAY_4        
 Min.   :    1   Min.   :  10000   Min.   :1.000   1:10585   Min.   :0.000   Min.   :21.00   0      :22882   0      :25233   0      :25451   Min.   :-2.0000  
 1st Qu.: 7476   1st Qu.:  50000   1st Qu.:1.000   2:14030   1st Qu.:1.000   1st Qu.:28.00   1      : 3669   2      : 3912   2      : 3811   1st Qu.:-1.0000  
 Median :14956   Median : 140000   Median :2.000   3: 4917   Median :2.000   Median :34.00   2      : 2644   3      :  326   3      :  239   Median : 0.0000  
 Mean   :14973   Mean   : 167488   Mean   :1.603   4:  123   Mean   :1.553   Mean   :35.47   3      :  320   4      :   98   4      :   76   Mean   :-0.2186  
 3rd Qu.:22468   3rd Qu.: 240000   3rd Qu.:2.000             3rd Qu.:2.000   3rd Qu.:41.00   4      :   76   1      :   28   7      :   27   3rd Qu.: 0.0000  
 Max.   :30000   Max.   :1000000   Max.   :2.000             Max.   :3.000   Max.   :79.00   5      :   25   5      :   25   6      :   23   Max.   : 8.0000  
                                                                                             (Other):   39   (Other):   33   (Other):   28                    
     PAY_5             PAY_6            BILL_AMT1         BILL_AMT2        BILL_AMT3         BILL_AMT4         BILL_AMT5        BILL_AMT6          PAY_AMT1     
 Min.   :-2.0000   Min.   :-2.0000   Min.   :-165580   Min.   :-69777   Min.   :-157264   Min.   :-170000   Min.   :-81334   Min.   :-339603   Min.   :     0  
 1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:   3519   1st Qu.:  2960   1st Qu.:   2642   1st Qu.:   2318   1st Qu.:  1774   1st Qu.:   1270   1st Qu.:  1000  
 Median : 0.0000   Median : 0.0000   Median :  22221   Median : 21029   Median :  20023   Median :  19000   Median : 18077   Median :  17102   Median :  2100  
 Mean   :-0.2642   Mean   :-0.2878   Mean   :  50902   Mean   : 48897   Mean   :  46753   Mean   :  43079   Mean   : 40195   Mean   :  38818   Mean   :  5654  
 3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.:  66506   3rd Qu.: 63392   3rd Qu.:  59752   3rd Qu.:  54064   3rd Qu.: 50030   3rd Qu.:  49098   3rd Qu.:  5005  
 Max.   : 8.0000   Max.   : 8.0000   Max.   : 964511   Max.   :983931   Max.   :1664089   Max.   : 891586   Max.   :927171   Max.   : 961664   Max.   :873552  
                                                                                                                                                                  
    PAY_AMT2            PAY_AMT3          PAY_AMT4          PAY_AMT5          PAY_AMT6       Status_Sep Status_Aug Status_July default.payment.next.month
 Min.   :      0.0   Min.   :     0   Min.   :     0   Min.   :     0   Min.   :     0   0:27185    0:27302    0:26917     0:23045                   
 1st Qu.:    820.5   1st Qu.:   390   1st Qu.:   296   1st Qu.:   258   1st Qu.:   133   1: 2470    1: 2353    1: 2738     1: 6610                   
 Median :   2006.0   Median :  1800   Median :  1500   Median :  1500   Median :  1500                                                               
 Mean   :   5889.1   Mean   :  5196   Mean   :  4825   Mean   :  4790   Mean   :  5176                                                               
 3rd Qu.:   5000.0   3rd Qu.:  4500   3rd Qu.:  4012   3rd Qu.:  4034   3rd Qu.:  4000                                                               
 Max.   :1684259.0   Max.   :896040   Max.   :621000   Max.   :426529   Max.   :528666                                                               
```

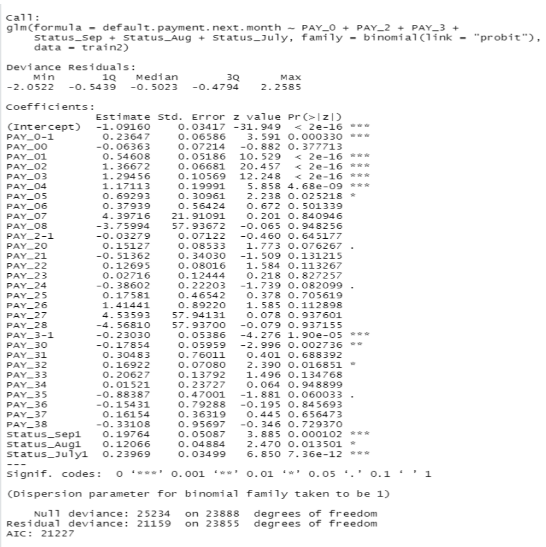
II. Exploratory Analysis

It is likely that education could help predict the default payment for next month because it may affect people’s understanding on the concepts of credits and how to manage credits. Income may also have an indirect relationship with default payment. Moreover, the amount of credit may be a good predictor because credit score may reflect past default status. More specifically, as the numbers of default payment increase, the credit score will probably decrease. In order to test out whether these two variables have statistical significance in building the model, I decided to compare the results with and without them.

For my first attempt, I used six variables mentioned above for my logistic and probit regression models: “Pay_0”, “Pay_2”, “Pay_3”, “Status_Sep”, “Status_Aug”, and “Status_July”. The logistic model returned an AIC of 21232, and probit model had a slightly lower AIC of 21227 as shown in the regression tables below. Both logistic and probit models gave a decently high accuracy score of 81.96% and 81.82% respectively. The result also suggested that the repayment status in the past months is meaning given its low p-value. It can be reasonably inferred that the records of past payment behavior can help predict the future performance. For example, if the past three payments are paid on time, the next payment probably will not default either. After running the data for several times, it was found that probit model always has a slightly lower AIC score, and the accuracy scores of both models are within the range of 80% to 82%.

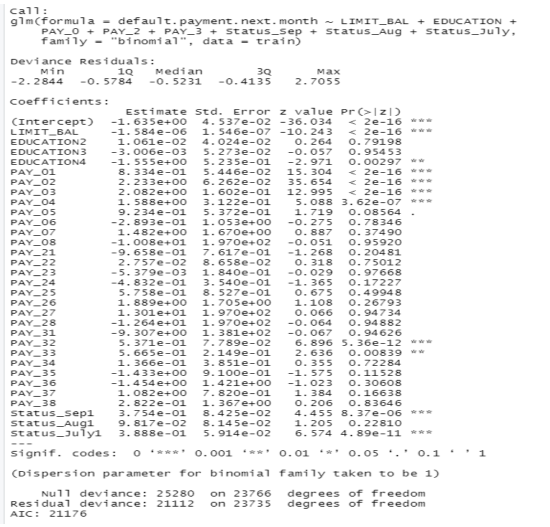


```
Call:
glm(formula = default.payment.next.month ~ PAY_0 + PAY_2 + PAY_3 +
    Status_Sep + Status_Aug + Status_July, family = "binomial",
    data = train2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0453  -0.5441  -0.5042  -0.4836   2.2517

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.83540    0.06336 -28.966  < 2e-16 ***
PAY_0-1       0.43337    0.11579   3.743 0.000182 ***
PAY_00       -0.13709    0.12639  -1.085 0.278056
PAY_01        0.95788    0.09139  10.481  < 2e-16 ***
PAY_02        2.28457    0.11577  19.733  < 2e-16 ***
PAY_03        2.16373    0.18101  11.953  < 2e-16 ***
PAY_04        1.96152    0.33421   5.869 4.38e-09 ***
PAY_05        1.18167    0.50008   2.363 0.018129 *
PAY_06        0.65438    0.91415   0.716 0.474095
PAY_07       12.65073  122.74788   0.103 0.917913
PAY_08      -11.46650  324.74407  -0.035 0.971833
PAY_2-1      -0.07060    0.12386  -0.570 0.568683
PAY_20        0.29484    0.14850   1.985 0.047097 *
PAY_21       -0.89334    0.62860  -1.421 0.155267
PAY_22        0.20120    0.13703   1.468 0.142024
PAY_23        0.03079    0.20816   0.148 0.882407
PAY_24       -0.68527    0.36709  -1.867 0.061930 .
PAY_25        0.24447    0.76891   0.318 0.750524
PAY_26        2.29654    1.46761   1.565 0.117626
PAY_27       12.68669  324.74652   0.039 0.968837
PAY_28      -12.96258  324.74438  -0.040 0.968160
PAY_3-1      -0.40947    0.09563  -4.282 1.85e-05 ***
PAY_30       -0.32076    0.10583  -3.031 0.002439 **
PAY_31        0.50871    1.31638   0.386 0.699166
PAY_32        0.27004    0.12252   2.204 0.027521 *
PAY_33        0.34036    0.23238   1.465 0.143017
PAY_34        0.02973    0.39888   0.075 0.940585
PAY_35       -1.51482    0.78763  -1.923 0.054446 .
PAY_36       -0.21212    1.35382  -0.157 0.875494
PAY_37        0.30502    0.67250   0.454 0.650142
PAY_38       -0.58112    1.61608  -0.360 0.719157
Status_Sep1   0.34349    0.08486   4.047 5.18e-05 ***
Status_Aug1   0.21918    0.08196   2.674 0.007492 **
Status_July1  0.40635    0.06031   6.737 1.62e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 25234  on 23888  degrees of freedom
Residual deviance: 21164  on 23855  degrees of freedom
AIC: 21232

Number of Fisher Scoring iterations: 11
```

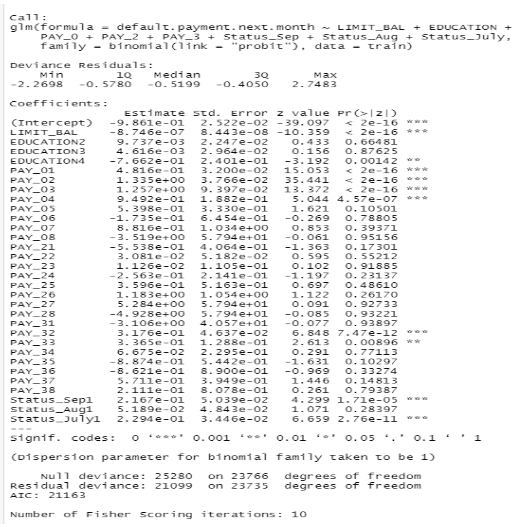


```
Call:
glm(formula = default.payment.next.month ~ PAY_0 + PAY_2 + PAY_3 +
    Status_Sep + Status_Aug + Status_July, family = binomial(link = "probit"),
    data = train2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0522  -0.5439  -0.5023  -0.4794   2.2585

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.09160    0.03417 -31.949  < 2e-16 ***
PAY_0-1      0.23647    0.06586   3.591 0.000330 ***
PAY_00      -0.06363    0.07214  -0.882 0.377713
PAY_01       0.54608    0.05186  10.529  < 2e-16 ***
PAY_02       1.36672    0.06681  20.457  < 2e-16 ***
PAY_03       1.29456    0.10569  12.248  < 2e-16 ***
PAY_04       1.17113    0.19991   5.858 4.68e-09 ***
PAY_05       0.69293    0.30961   2.238 0.025218 *
PAY_06       0.37939    0.56424   0.672 0.501339
PAY_07       4.39716   21.91091   0.201 0.840946
PAY_08      -3.75994   57.93672  -0.065 0.948256
PAY_2-1     -0.03279    0.07122  -0.460 0.645177
PAY_20       0.15127    0.08533   1.773 0.076267 .
PAY_21      -0.51362    0.34030  -1.509 0.131215
PAY_22       0.12695    0.08016   1.584 0.113267
PAY_23       0.02716    0.12444   0.218 0.827257
PAY_24      -0.38602    0.22203  -1.739 0.082099 .
PAY_25       0.17581    0.46542   0.378 0.705619
PAY_26       1.41441    0.89220   1.585 0.112898
PAY_27       4.53593   57.94131   0.078 0.937601
PAY_28      -4.56810   57.93700  -0.079 0.937155
PAY_3-1     -0.23030    0.05386  -4.276 1.90e-05 ***
PAY_30      -0.17854    0.05959  -2.996 0.002736 **
PAY_31       0.30483    0.76011   0.401 0.688392
PAY_32       0.16922    0.07080   2.390 0.016851 *
PAY_33       0.20627    0.13792   1.496 0.134768
PAY_34       0.01521    0.23727   0.064 0.948899
PAY_35      -0.88387    0.47001  -1.881 0.060033 .
PAY_36      -0.15431    0.79288  -0.195 0.845693
PAY_37       0.16154    0.36319   0.445 0.656473
PAY_38      -0.33108    0.95697  -0.346 0.729370
Status_Sep1  0.19764    0.05087   3.885 0.000102 ***
Status_Aug1  0.12066    0.04884   2.470 0.013501 *
Status_July1 0.23969    0.03499   6.850 7.36e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 25234  on 23888  degrees of freedom
Residual deviance: 21159  on 23855  degrees of freedom
AIC: 21227
```

In order to find out whether education and the amount of credit can be the good predictors on the default rate or not, I tried to incorporate the two variables, “EDUCATION” and “LIMIT_BAL”, with the previous six variables to build a new model. As shown in the table on the left corner, the amount of credit has a p-value smaller than 5%, suggesting it is statistically significant in being a predictor of the probability in default of the next payment. Again, the AIC score of logit regression is slightly higher than that of probit regression. However, the overall accuracy improves to 82.00% and 82.03% respectively. As this group of predictors has a higher accuracy on average compared to the previous attempt, the probit model, which takes the impact of education and credit score into account, seems to be the model that fits the best with the data. Although education has a p-value larger than the significance level, I decided to keep this predictor because the model still maintains a higher accuracy, and the variable itself has a higher practical significance than its statistical significance.



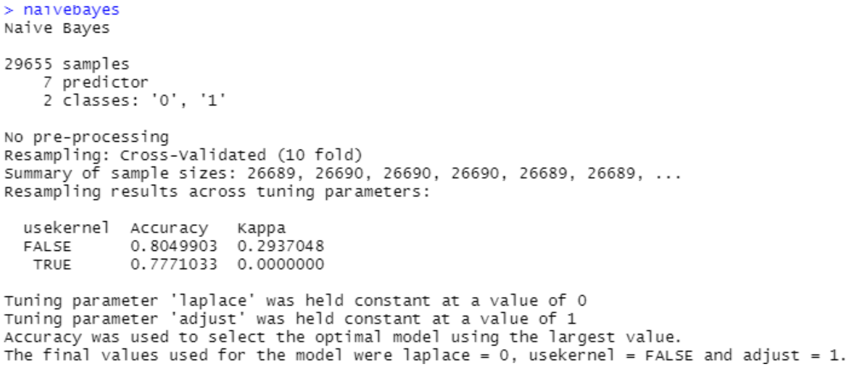
```
Call:
glm(formula = default.payment.next.month ~ LIMIT_BAL + EDUCATION +
    PAY_0 + PAY_2 + PAY_3 + Status_Sep + Status_Aug + Status_July,
    family = "binomial", data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2844  -0.5784  -0.5231  -0.4135   2.7055

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.635e+00  4.537e-02 -36.034  < 2e-16 ***
LIMIT_BAL    -1.584e-06  1.546e-07 -10.243  < 2e-16 ***
EDUCATION2    1.061e-02  4.024e-02   0.264  0.79198
EDUCATION3   -3.006e-03  5.273e-02  -0.057  0.95453
EDUCATION4   -1.555e+00  5.235e-01  -2.971  0.00297 **
PAY_01        8.334e-01  5.446e-02  15.304  < 2e-16 ***
PAY_02        2.233e+00  6.262e-02  35.654  < 2e-16 ***
PAY_03        2.082e+00  1.602e-01  12.995  < 2e-16 ***
PAY_04        1.588e+00  3.122e-01   5.088 3.62e-07 ***
PAY_05        9.234e-01  5.372e-01   1.719  0.08564 .
PAY_06       -2.893e-01  1.053e+00  -0.275  0.78346
PAY_07        1.482e+00  1.670e+00   0.887  0.37490
PAY_08       -1.008e+01  1.970e+02  -0.051  0.95920
PAY_21       -9.658e-01  7.617e-01  -1.268  0.20481
PAY_22        2.757e-02  8.658e-02   0.318  0.75012
PAY_23       -5.379e-03  1.840e-01  -0.029  0.97668
PAY_24       -4.832e-01  3.540e-01  -1.365  0.17227
PAY_25        5.758e-01  8.527e-01   0.675  0.49948
PAY_26        1.889e+00  1.705e+00   1.108  0.26793
PAY_27        1.301e+01  1.970e+02   0.066  0.94734
PAY_28       -1.264e+01  1.970e+02  -0.064  0.94882
PAY_31       -9.307e+00  1.381e+02  -0.067  0.94626
PAY_32        5.371e-01  7.789e-02   6.896 5.36e-12 ***
PAY_33        5.665e-01  2.149e-01   2.636  0.00839 **
PAY_34        1.366e-01  3.851e-01   0.355  0.72284
PAY_35       -1.433e+00  9.100e-01  -1.575  0.11528
PAY_36       -1.454e+00  1.421e+00  -1.023  0.30608
PAY_37        1.082e+00  7.820e-01   1.384  0.16638
PAY_38        2.822e-01  1.367e+00   0.206  0.83646
Status_Sep1   3.754e-01  8.425e-02   4.455 8.37e-06 ***
Status_Aug1   9.817e-02  8.145e-02   1.205  0.22810
Status_July1  3.888e-01  5.914e-02   6.574 4.89e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 25280  on 23766  degrees of freedom
Residual deviance: 21112  on 23735  degrees of freedom
AIC: 21176
```



```
Call:
glm(formula = default.payment.next.month ~ LIMIT_BAL + EDUCATION +
    PAY_0 + PAY_2 + PAY_3 + Status_Sep + Status_Aug + Status_July,
    family = binomial(link = "probit"), data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2698  -0.5780  -0.5199  -0.4050   2.7483

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -9.861e-01  2.522e-02 -39.097  < 2e-16 ***
LIMIT_BAL    -8.746e-07  8.443e-08 -10.359  < 2e-16 ***
EDUCATION2    9.737e-03  2.247e-02   0.433  0.66481
EDUCATION3    4.616e-03  2.964e-02   0.156  0.87625
EDUCATION4   -7.662e-01  2.401e-01  -3.192  0.00142 **
PAY_01        4.816e-01  3.200e-02  15.053  < 2e-16 ***
PAY_02        1.335e+00  3.766e-02  35.441  < 2e-16 ***
PAY_03        1.257e+00  9.397e-02  13.372  < 2e-16 ***
PAY_04        9.492e-01  1.882e-01   5.044 4.57e-07 ***
PAY_05        5.398e-01  3.330e-01   1.621  0.10501
PAY_06       -1.735e-01  6.454e-01  -0.269  0.78805
PAY_07        8.816e-01  1.034e+00   0.853  0.39371
PAY_08       -3.519e+00  5.794e+01  -0.061  0.95156
PAY_21       -5.538e-01  4.064e-01  -1.363  0.17301
PAY_22        3.081e-02  5.182e-02   0.595  0.55212
PAY_23        1.126e-02  1.105e-01   0.102  0.91885
PAY_24       -2.563e-01  2.141e-01  -1.197  0.23137
PAY_25        3.596e-01  5.163e-01   0.697  0.48610
PAY_26        1.183e+00  1.054e+00   1.122  0.26170
PAY_27        5.284e+00  5.794e+01   0.091  0.92733
PAY_28       -4.928e+00  5.794e+01  -0.085  0.93221
PAY_31       -3.106e+00  4.057e+01  -0.077  0.93897
PAY_32        3.176e-01  4.637e-02   6.848 7.47e-12 ***
PAY_33        3.365e-01  1.288e-01   2.613  0.00896 **
PAY_34        6.675e-02  2.295e-01   0.291  0.77113
PAY_35       -8.874e-01  5.442e-01  -1.631  0.10297
PAY_36       -8.621e-01  8.900e-01  -0.969  0.33274
PAY_37        5.711e-01  3.949e-01   1.446  0.14813
PAY_38        2.111e-01  8.078e-01   0.261  0.79387
Status_Sep1   2.167e-01  5.039e-02   4.299 1.71e-05 ***
Status_Aug1   5.189e-02  4.843e-02   1.071  0.28397
Status_July1  2.294e-01  3.446e-02   6.659 2.76e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 25280  on 23766  degrees of freedom
Residual deviance: 21099  on 23735  degrees of freedom
AIC: 21163

Number of Fisher Scoring iterations: 10
```

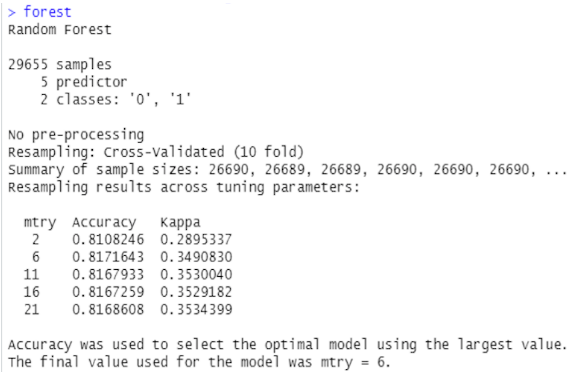
III. Cross-Validations

For Naive Bayes Model, I set the folds to be 10 and I used all of the categorical variables mentioned above. I removed the numeric variable “credit”, as it has some incompatibility with R. It turned out the accuracy was 77.71%, which was around 10% lower than using logistic and probit regression. I removed some variables that seemed less significant and relevant to the model, such as “Pay_2” and “Status_Aug”, while the accuracy level remained the same.

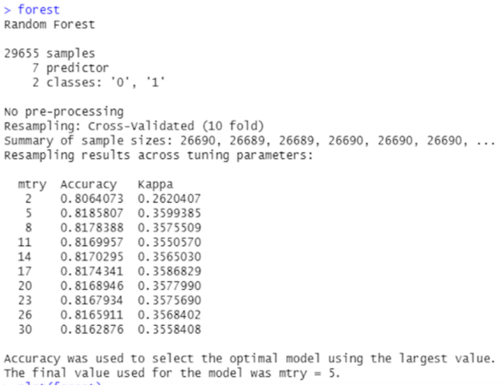


```
> naiveBayes
Naive Bayes

29655 samples
    7 predictor
    2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 26689, 26690, 26690, 26690, 26689, 26689, ...
Resampling results across tuning parameters:

  usekernel  Accuracy   Kappa
  FALSE      0.8049903  0.2937048
   TRUE      0.7771033  0.0000000

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

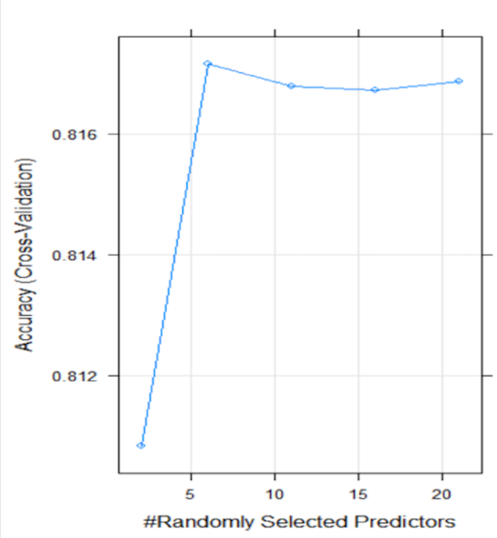
For Random Forest Model, I set the folds to be 10. In my first attempt, I changed the tunelength to three, and used only three variables “Pay_0”, “Status_Sep”, and “Status_July”, which are the ones with the highest statistical significance according to the probit model. The best accuracy score is 81.63%, which is significantly higher than the Naïve Bayes Model. Next, I experimented and increased the tunelength to 5, and it achieved a higher accuracy of 81.72%, as shown in the table on the left. I came up with the hypothesis that a higher tunelength may bring us a higher accuracy, so I decided to further increase the tunelength to 10 while using all of the variables. It turned out this hypothesis was correct and I got an 81.858% of accuracy as shown in the table on the right, which is the best figure among all of the attempts I had for this model.

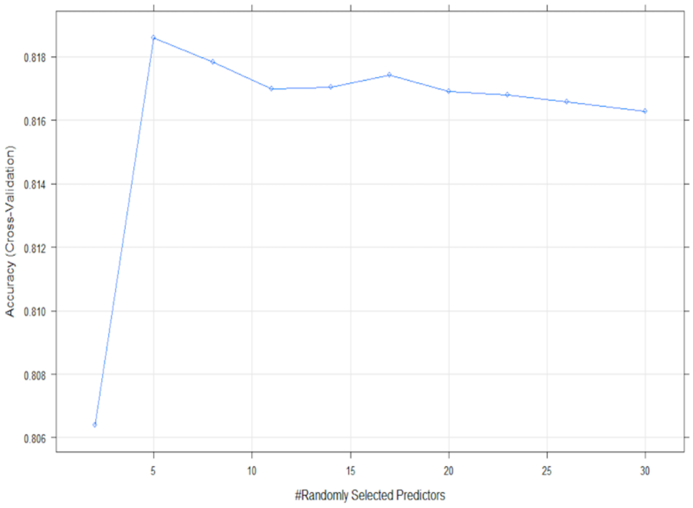


```
> forest
Random Forest

29655 samples
    5 predictor
    2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 26690, 26689, 26689, 26690, 26690, 26690, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
   2    0.8108246  0.2895337
   6    0.8171643  0.3490830
  11    0.8167933  0.3530040
  16    0.8167259  0.3529182
  21    0.8168608  0.3534399

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 6.
```

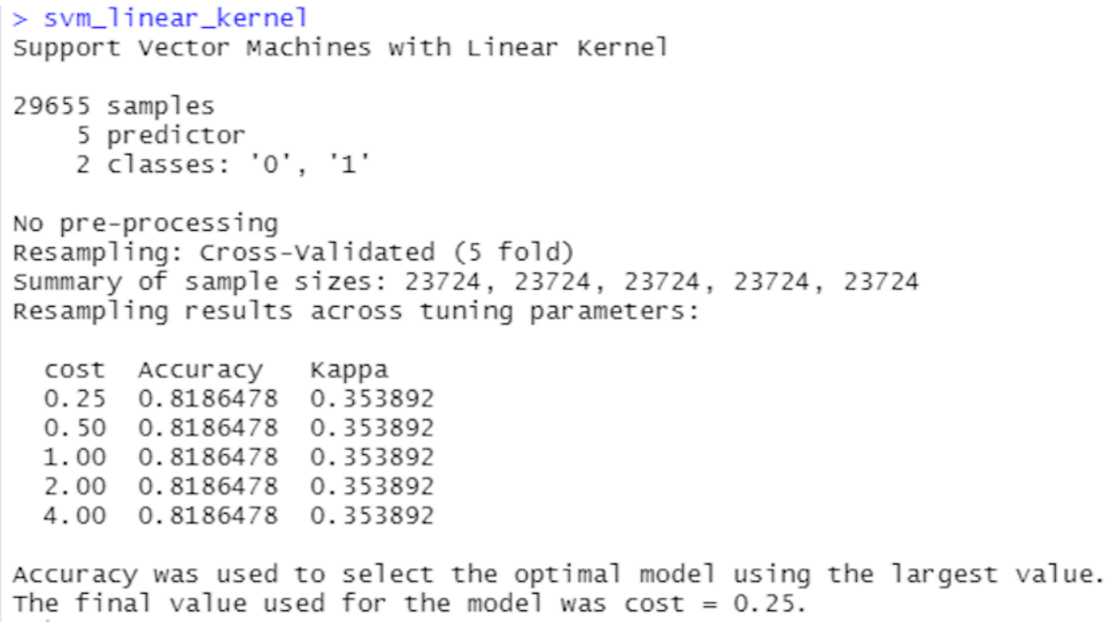


```
> forest
Random Forest

29655 samples
    7 predictor
    2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 26690, 26689, 26689, 26690, 26690, 26690, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
   2    0.8064073  0.2620407
   5    0.8185807  0.3599385
   8    0.8178388  0.3575509
  11    0.8169957  0.3550570
  14    0.8170295  0.3565030
  17    0.8174341  0.3586829
  20    0.8168946  0.3577990
  23    0.8167934  0.3575690
  26    0.8165911  0.3568402
  30    0.8162876  0.3558408

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 5.
```





As running an SVM Linear Model took five hours on my device, I only ran it for once. As the data size of is decently large and may take a while to run, I reduced both the folds and tunelength to five. At the same time, I used only five variables in the model with the highest statistical or practical significance, including “education”, “Pay_0”, “Pay_3”, “Status_Sep”, and “Status_July”. It generated the highest accuracy score among all of the cross-validation models, showing as 81.864%. This figure is slightly better than the previous Random Forest Model with 10 folds, though using less variables and folds. In this sense, it seems like SVM Linear Model has the second-best predictability following the logit and probit model. It is possible that we can get an even better result if we increase the number of folds, tunelength, and predictors.



```
> svm_linear_kernel
Support Vector Machines with Linear Kernel

29655 samples
    5 predictor
    2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 23724, 23724, 23724, 23724, 23724
Resampling results across tuning parameters:

  cost  Accuracy   Kappa
  0.25  0.8186478  0.353892
  0.50  0.8186478  0.353892
  1.00  0.8186478  0.353892
  2.00  0.8186478  0.353892
  4.00  0.8186478  0.353892

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cost = 0.25.
```