BT2103 Optimization Methods in Business Analytics

Final Project

Chen Zuo Hui A0233620U

Ng Chek Khau A0234465A

Ryu Kairin Anaqi Suzuki A0233269B

Tan Chong Ren A0234511U

BT2103 Project

2022-11-08

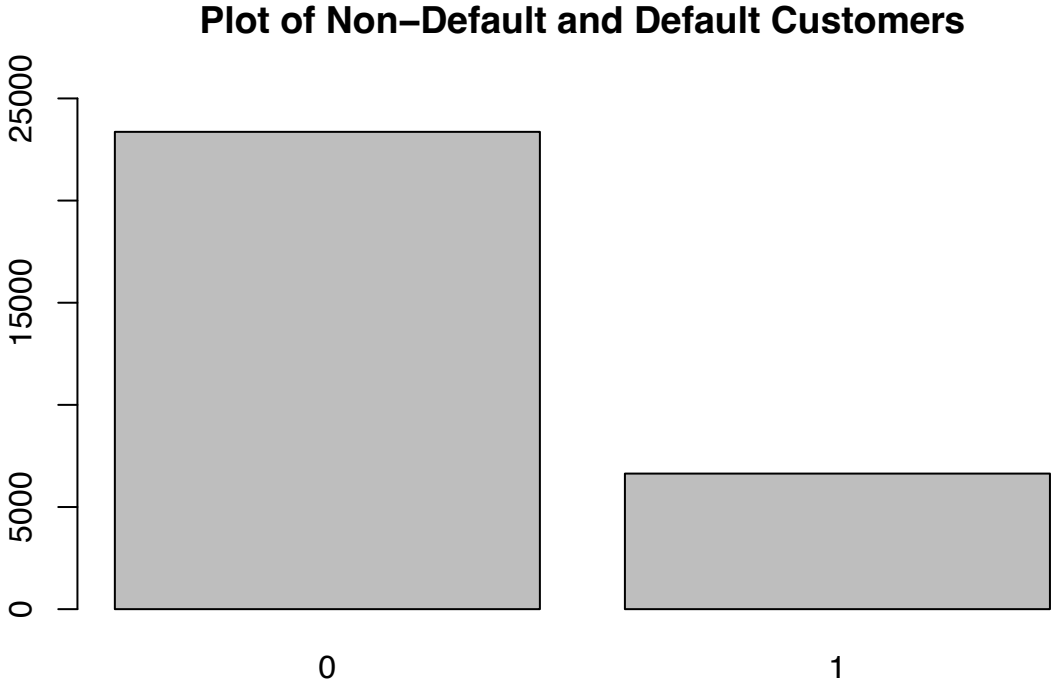
Brief Introduction of Data Set and Data Modeling Problem

```
##   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
## 6  6     50000   1         1        2  37     0     0     0     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
## 6     64400     57069     57608     19394     19619     20024     2500     1815
##   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
## 6      657     1000     1000      800                          0
```

```
##        ID          LIMIT_BAL           SEX          EDUCATION
##  Min.   :    1   Min.   :  10000   Min.   :1.000   Min.   :0.000
##  1st Qu.: 7501   1st Qu.:  50000   1st Qu.:1.000   1st Qu.:1.000
##  Median :15000   Median : 140000   Median :2.000   Median :2.000
##  Mean   :15000   Mean   : 167484   Mean   :1.604   Mean   :1.853
##  3rd Qu.:22500   3rd Qu.: 240000   3rd Qu.:2.000   3rd Qu.:2.000
##  Max.   :30000   Max.   :1000000   Max.   :2.000   Max.   :6.000
##     MARRIAGE          AGE            PAY_0             PAY_2
##  Min.   :0.000   Min.   :21.00   Min.   :-2.0000   Min.   :-2.0000
##  1st Qu.:1.000   1st Qu.:28.00   1st Qu.:-1.0000   1st Qu.:-1.0000
##  Median :2.000   Median :34.00   Median : 0.0000   Median : 0.0000
##  Mean   :1.552   Mean   :35.49   Mean   :-0.0167   Mean   :-0.1338
##  3rd Qu.:2.000   3rd Qu.:41.00   3rd Qu.: 0.0000   3rd Qu.: 0.0000
##  Max.   :3.000   Max.   :79.00   Max.   : 8.0000   Max.   : 8.0000
##      PAY_3             PAY_4             PAY_5             PAY_6
##  Min.   :-2.0000   Min.   :-2.0000   Min.   :-2.0000   Min.   :-2.0000
##  1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:-1.0000
##  Median : 0.0000   Median : 0.0000   Median : 0.0000   Median : 0.0000
##  Mean   :-0.1662   Mean   :-0.2207   Mean   :-0.2662   Mean   :-0.2911
##  3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000
##  Max.   : 8.0000   Max.   : 8.0000   Max.   : 8.0000   Max.   : 8.0000
##    BILL_AMT1         BILL_AMT2        BILL_AMT3         BILL_AMT4
```

```
##  Min.   :-165580   Min.   :-69777   Min.   :-157264   Min.   :-170000
##  1st Qu.:   3559   1st Qu.:  2985   1st Qu.:   2666   1st Qu.:   2327
##  Median :  22382   Median : 21200   Median :  20088   Median :  19052
##  Mean   :  51223   Mean   : 49179   Mean   :  47013   Mean   :  43263
##  3rd Qu.:  67091   3rd Qu.: 64006   3rd Qu.:  60165   3rd Qu.:  54506
##  Max.   : 964511   Max.   :983931   Max.   :1664089   Max.   : 891586
##    BILL_AMT5        BILL_AMT6          PAY_AMT1         PAY_AMT2
##  Min.   :-81334   Min.   :-339603   Min.   :     0   Min.   :      0
##  1st Qu.:  1763   1st Qu.:   1256   1st Qu.:  1000   1st Qu.:    833
##  Median : 18104   Median :  17071   Median :  2100   Median :   2009
##  Mean   : 40311   Mean   :  38872   Mean   :  5664   Mean   :   5921
##  3rd Qu.: 50190   3rd Qu.:  49198   3rd Qu.:  5006   3rd Qu.:   5000
##  Max.   :927171   Max.   : 961664   Max.   :873552   Max.   :1684259
##     PAY_AMT3         PAY_AMT4         PAY_AMT5           PAY_AMT6
##  Min.   :     0   Min.   :     0   Min.   :     0.0   Min.   :     0.0
##  1st Qu.:   390   1st Qu.:   296   1st Qu.:   252.5   1st Qu.:   117.8
##  Median :  1800   Median :  1500   Median :  1500.0   Median :  1500.0
##  Mean   :  5226   Mean   :  4826   Mean   :  4799.4   Mean   :  5215.5
##  3rd Qu.:  4505   3rd Qu.:  4013   3rd Qu.:  4031.5   3rd Qu.:  4000.0
##  Max.   :896040   Max.   :621000   Max.   :426529.0   Max.   :528666.0
##  default.payment.next.month
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.2212
##  3rd Qu.:0.0000
##  Max.   :1.0000
```

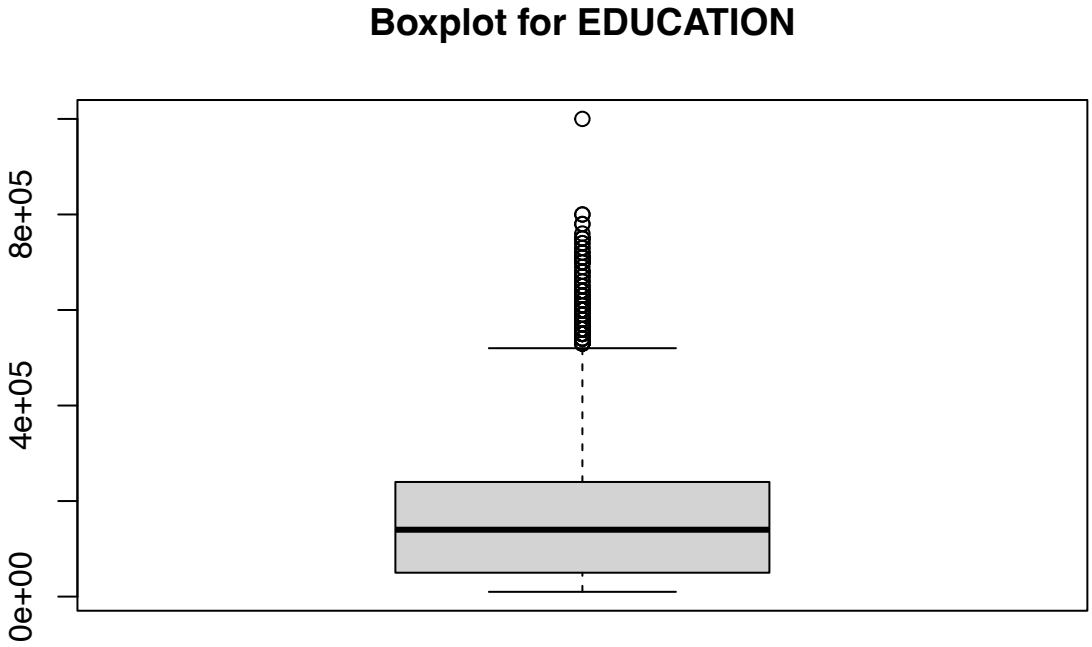
This Data Set contains payment information of 30,000 credit card holders obtained from a bank in Taiwan. Each data sample is described by 23 feature attributes (columns V2 to V24). The target feature (column V25) to be predicted is binary valued 0 (not default) or 1 (default). The data can be fitted using logistic regression, Support Vector Machine, as well as neural network. As there are a large number of variables, we will first perform feature selection in order to reduce the dimensionality of the data. This will help increase the efficiency of the models we run, as well as help prevent the problem of overfitting. We will split the data into training and test sets and proceed to fit the model using a training set, and then use a test set in order to test the accuracy of our models, as well as derive other metrics, such as recall and accuracy.

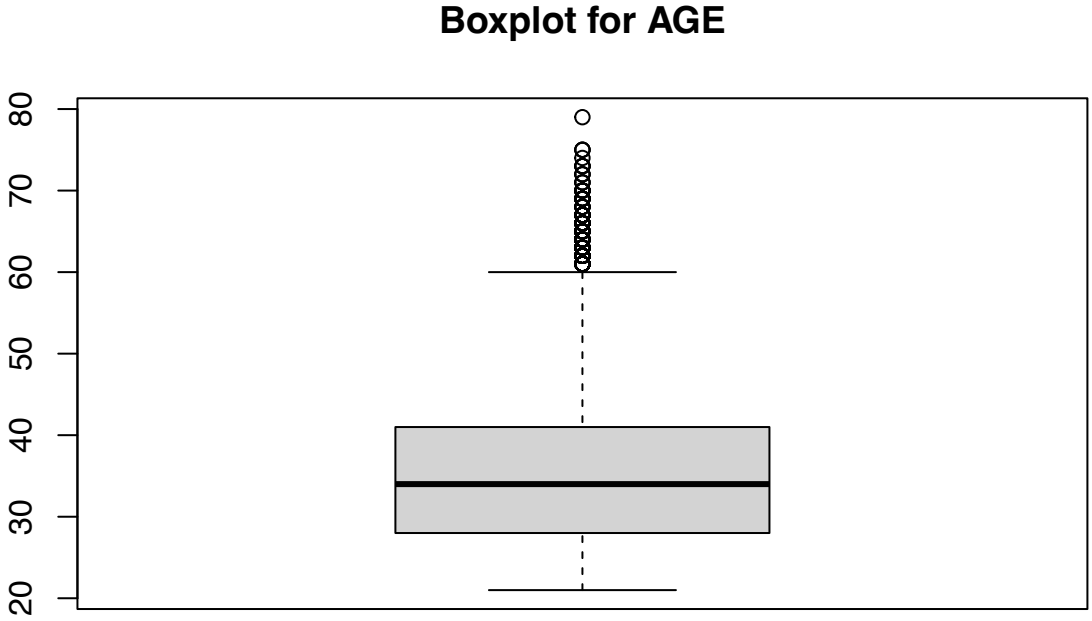


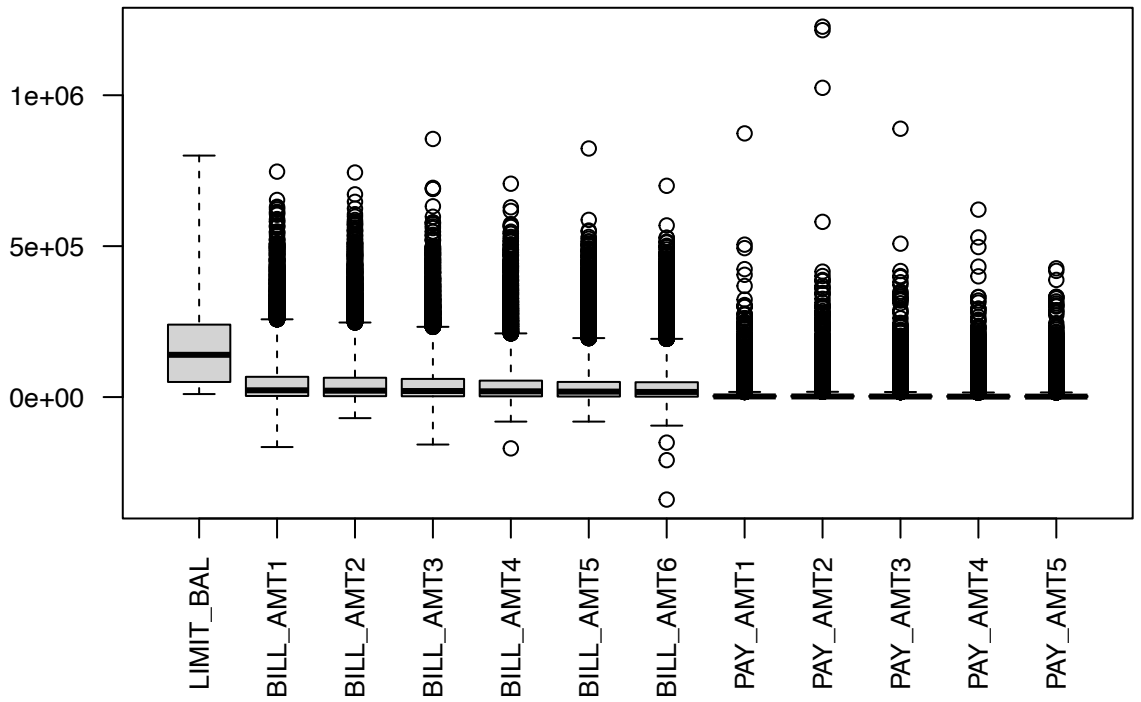
From the plot shown above, we observe that the dataset is not balanced.

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 (-2=no consumption, -1=pay duly, 0=the use of revolving credit, 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 Pre-Processing





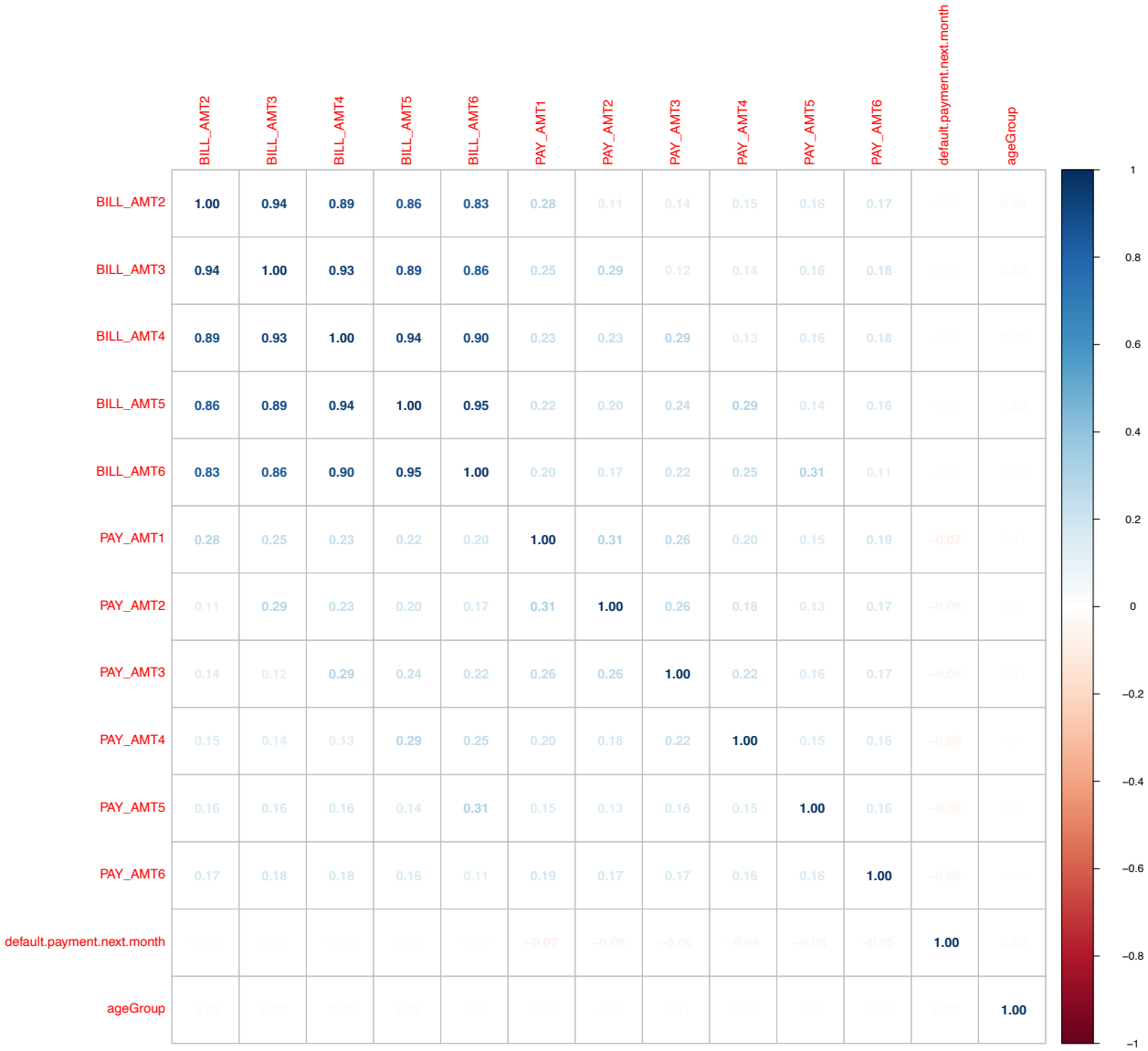


We encoded all values of 5 and 6 under the column EDUCATION to 0, since they are unknown. The dataset consists of people aged between 21 and 79, which are reasonable ages, hence no data cleaning is required to be done on age.

Upon observation of the combined box plots of column 2 and 13 to 24, we filtered for anomalies for the amount of given credit for the column LIMIT_BAL, removing the data sample that has above 1000000 credit since it is significantly further away from the interquartile range. We also filtered away values of BILL_AMT3 that exceed 1500000.

We discretize continuous values of age into 3 groups of values 1,2 and 3 under the new column ageGroup to better categorize individuals of different age groups. The value of 1 represents customers aged 21-39, the value of 2 represents customers aged 40-60 and value of 3 represents customers above age 60. After which, we removed the initial age column of continuous variables.

We then split the data into training and test sets.




```
##
## Call:
## lm(formula = paste(response, "~", paste(preds, collapse = " + ")),
##     data = l)
##
## Coefficients:
## (Intercept)        PAY_0        PAY_2    BILL_AMT1    MARRIAGE1    MARRIAGE2
##    -0.19239      0.96163      0.20569     -0.38161      0.13414      0.10437
##   MARRIAGE3        PAY_3     PAY_AMT1   EDUCATION1   EDUCATION2   EDUCATION3
##     0.10282      0.12268     -0.57026      0.13497      0.11621      0.11633
##  EDUCATION4    LIMIT_BAL     ageGroup          SEX2     PAY_AMT5        PAY_4
##     0.06197     -0.06743      0.02147     -0.01322     -0.15310      0.08024
##    PAY_AMT2
##    -0.33805
```

```
## Subset selection object
## Call: regsubsets.formula(default.payment.next.month ~ ., data = train.data,
##     method = "backward", nvmax = 13)
```

```
## 29 Variables  (and intercept)
##             Forced in Forced out
## ID              FALSE      FALSE
## LIMIT_BAL       FALSE      FALSE
## SEX2            FALSE      FALSE
## EDUCATION1      FALSE      FALSE
## EDUCATION2      FALSE      FALSE
## EDUCATION3      FALSE      FALSE
## EDUCATION4      FALSE      FALSE
## MARRIAGE1       FALSE      FALSE
## MARRIAGE2       FALSE      FALSE
## MARRIAGE3       FALSE      FALSE
## PAY_0           FALSE      FALSE
## PAY_2           FALSE      FALSE
## PAY_3           FALSE      FALSE
## PAY_4           FALSE      FALSE
## PAY_5           FALSE      FALSE
## PAY_6           FALSE      FALSE
## BILL_AMT1       FALSE      FALSE
## BILL_AMT2       FALSE      FALSE
## BILL_AMT3       FALSE      FALSE
## BILL_AMT4       FALSE      FALSE
## BILL_AMT5       FALSE      FALSE
## BILL_AMT6       FALSE      FALSE
## PAY_AMT1        FALSE      FALSE
## PAY_AMT2        FALSE      FALSE
## PAY_AMT3        FALSE      FALSE
## PAY_AMT4        FALSE      FALSE
## PAY_AMT5        FALSE      FALSE
## PAY_AMT6        FALSE      FALSE
## ageGroup        FALSE      FALSE
## 1 subsets of each size up to 13
## Selection Algorithm: backward
##           ID  LIMIT_BAL SEX2 EDUCATION1 EDUCATION2 EDUCATION3 EDUCATION4
## 1  ( 1 )  " " " "       " "  " "        " "        " "        " "
## 2  ( 1 )  " " " "       " "  " "        " "        " "        " "
## 3  ( 1 )  " " " "       " "  " "        " "        " "        " "
## 4  ( 1 )  " " " "       " "  " "        " "        " "        " "
## 5  ( 1 )  " " " "       " "  " "        " "        " "        " "
## 6  ( 1 )  " " " "       " "  " "        " "        " "        " "
## 7  ( 1 )  " " " "       " "  "*"        " "        " "        " "
## 8  ( 1 )  " " " "       " "  "*"        " "        "*"        " "
## 9  ( 1 )  " " " "       " "  "*"        "*"        "*"        " "
## 10  ( 1 ) " " "*"       " "  "*"        "*"        "*"        " "
## 11  ( 1 ) " " "*"       " "  "*"        "*"        "*"        " "
## 12  ( 1 ) " " "*"       "*"  "*"        "*"        "*"        " "
## 13  ( 1 ) " " "*"       "*"  "*"        "*"        "*"        " "
##           MARRIAGE1 MARRIAGE2 MARRIAGE3 PAY_0 PAY_2 PAY_3 PAY_4 PAY_5 PAY_6
## 1  ( 1 )  " "       " "       " "       "*"   " "   " "   " "   " "   " "
## 2  ( 1 )  " "       " "       " "       "*"   " "   " "   " "   " "   " "
## 3  ( 1 )  " "       " "       " "       "*"   "*"   " "   " "   " "   " "
## 4  ( 1 )  "*"       " "       " "       "*"   "*"   " "   " "   " "   " "
## 5  ( 1 )  "*"       " "       " "       "*"   "*"   "*"   " "   " "   " "
## 6  ( 1 )  "*"       " "       " "       "*"   "*"   "*"   " "   " "   " "
```

```
## 7  ( 1 )  "*"       " "       " "       "*"   "*"   "*"   " "   " "   " "
## 8  ( 1 )  "*"       " "       " "       "*"   "*"   "*"   " "   " "   " "
## 9  ( 1 )  "*"       " "       " "       "*"   "*"   "*"   " "   " "   " "
## 10  ( 1 ) "*"       " "       " "       "*"   "*"   "*"   " "   " "   " "
## 11  ( 1 ) "*"       " "       " "       "*"   "*"   "*"   " "   " "   " "
## 12  ( 1 ) "*"       " "       " "       "*"   "*"   "*"   " "   " "   " "
## 13  ( 1 ) "*"       " "       " "       "*"   "*"   "*"   " "   " "   " "
##           BILL_AMT1 BILL_AMT2 BILL_AMT3 BILL_AMT4 BILL_AMT5 BILL_AMT6 PAY_AMT1
## 1  ( 1 )  " "       " "       " "       " "       " "       " "       " "
## 2  ( 1 )  "*"       " "       " "       " "       " "       " "       " "
## 3  ( 1 )  "*"       " "       " "       " "       " "       " "       " "
## 4  ( 1 )  "*"       " "       " "       " "       " "       " "       " "
## 5  ( 1 )  "*"       " "       " "       " "       " "       " "       " "
## 6  ( 1 )  "*"       " "       " "       " "       " "       " "       "*"
## 7  ( 1 )  "*"       " "       " "       " "       " "       " "       "*"
## 8  ( 1 )  "*"       " "       " "       " "       " "       " "       "*"
## 9  ( 1 )  "*"       " "       " "       " "       " "       " "       "*"
## 10  ( 1 ) "*"       " "       " "       " "       " "       " "       "*"
## 11  ( 1 ) "*"       " "       " "       " "       " "       " "       "*"
## 12  ( 1 ) "*"       " "       " "       " "       " "       " "       "*"
## 13  ( 1 ) "*"       " "       " "       " "       " "       " "       "*"
##           PAY_AMT2 PAY_AMT3 PAY_AMT4 PAY_AMT5 PAY_AMT6 ageGroup
## 1  ( 1 )  " "      " "      " "      " "      " "      " "
## 2  ( 1 )  " "      " "      " "      " "      " "      " "
## 3  ( 1 )  " "      " "      " "      " "      " "      " "
## 4  ( 1 )  " "      " "      " "      " "      " "      " "
## 5  ( 1 )  " "      " "      " "      " "      " "      " "
## 6  ( 1 )  " "      " "      " "      " "      " "      " "
## 7  ( 1 )  " "      " "      " "      " "      " "      " "
## 8  ( 1 )  " "      " "      " "      " "      " "      " "
## 9  ( 1 )  " "      " "      " "      " "      " "      " "
## 10  ( 1 ) " "      " "      " "      " "      " "      " "
## 11  ( 1 ) " "      " "      " "      " "      " "      "*"
## 12  ( 1 ) " "      " "      " "      " "      " "      "*"
## 13  ( 1 ) "*"      " "      " "      " "      " "      "*"
```

```
## Subset selection object
## Call: regsubsets.formula(default.payment.next.month ~ ., data = train.data,
##     method = "forward", nvmax = 13)
## 29 Variables  (and intercept)
##             Forced in Forced out
## ID              FALSE      FALSE
## LIMIT_BAL       FALSE      FALSE
## SEX2            FALSE      FALSE
## EDUCATION1      FALSE      FALSE
## EDUCATION2      FALSE      FALSE
## EDUCATION3      FALSE      FALSE
## EDUCATION4      FALSE      FALSE
## MARRIAGE1       FALSE      FALSE
## MARRIAGE2       FALSE      FALSE
## MARRIAGE3       FALSE      FALSE
## PAY_0           FALSE      FALSE
## PAY_2           FALSE      FALSE
## PAY_3           FALSE      FALSE
```

```
## PAY_4          FALSE      FALSE
## PAY_5          FALSE      FALSE
## PAY_6          FALSE      FALSE
## BILL_AMT1      FALSE      FALSE
## BILL_AMT2      FALSE      FALSE
## BILL_AMT3      FALSE      FALSE
## BILL_AMT4      FALSE      FALSE
## BILL_AMT5      FALSE      FALSE
## BILL_AMT6      FALSE      FALSE
## PAY_AMT1       FALSE      FALSE
## PAY_AMT2       FALSE      FALSE
## PAY_AMT3       FALSE      FALSE
## PAY_AMT4       FALSE      FALSE
## PAY_AMT5       FALSE      FALSE
## PAY_AMT6       FALSE      FALSE
## ageGroup       FALSE      FALSE
## 1 subsets of each size up to 13
## Selection Algorithm: forward
##           ID  LIMIT_BAL SEX2 EDUCATION1 EDUCATION2 EDUCATION3 EDUCATION4
## 1  ( 1 )  " " " "       " "  " "        " "        " "        " "
## 2  ( 1 )  " " " "       " "  " "        " "        " "        " "
## 3  ( 1 )  " " " "       " "  " "        " "        " "        " "
## 4  ( 1 )  " " " "       " "  " "        " "        " "        " "
## 5  ( 1 )  " " " "       " "  " "        " "        " "        " "
## 6  ( 1 )  " " " "       " "  " "        " "        " "        " "
## 7  ( 1 )  " " " "       " "  " "        " "        " "        " "
## 8  ( 1 )  " " "*"       " "  " "        " "        " "        " "
## 9  ( 1 )  " " "*"       " "  "*"        " "        " "        " "
## 10  ( 1 ) " " "*"       "*"  "*"        " "        " "        " "
## 11  ( 1 ) " " "*"       "*"  "*"        " "        " "        " "
## 12  ( 1 ) " " "*"       "*"  "*"        " "        " "        " "
## 13  ( 1 ) " " "*"       "*"  "*"        " "        " "        " "
##           MARRIAGE1 MARRIAGE2 MARRIAGE3 PAY_0 PAY_2 PAY_3 PAY_4 PAY_5 PAY_6
## 1  ( 1 )  " "       " "       " "       "*"   " "   " "   " "   " "   " "
## 2  ( 1 )  " "       " "       " "       "*"   " "   " "   " "   " "   " "
## 3  ( 1 )  " "       " "       " "       "*"   "*"   " "   " "   " "   " "
## 4  ( 1 )  "*"       " "       " "       "*"   "*"   " "   " "   " "   " "
## 5  ( 1 )  "*"       " "       " "       "*"   "*"   "*"   " "   " "   " "
## 6  ( 1 )  "*"       " "       " "       "*"   "*"   "*"   " "   " "   " "
## 7  ( 1 )  "*"       " "       " "       "*"   "*"   "*"   " "   " "   " "
## 8  ( 1 )  "*"       " "       " "       "*"   "*"   "*"   " "   " "   " "
## 9  ( 1 )  "*"       " "       " "       "*"   "*"   "*"   " "   " "   " "
## 10  ( 1 ) "*"       " "       " "       "*"   "*"   "*"   " "   " "   " "
## 11  ( 1 ) "*"       " "       " "       "*"   "*"   "*"   " "   " "   " "
## 12  ( 1 ) "*"       " "       " "       "*"   "*"   "*"   "*"   " "   " "
## 13  ( 1 ) "*"       " "       " "       "*"   "*"   "*"   "*"   " "   " "
##           BILL_AMT1 BILL_AMT2 BILL_AMT3 BILL_AMT4 BILL_AMT5 BILL_AMT6 PAY_AMT1
## 1  ( 1 )  " "       " "       " "       " "       " "       " "       " "
## 2  ( 1 )  "*"       " "       " "       " "       " "       " "       " "
## 3  ( 1 )  "*"       " "       " "       " "       " "       " "       " "
## 4  ( 1 )  "*"       " "       " "       " "       " "       " "       " "
## 5  ( 1 )  "*"       " "       " "       " "       " "       " "       " "
## 6  ( 1 )  "*"       " "       " "       " "       " "       " "       "*"
## 7  ( 1 )  "*"       " "       " "       " "       " "       " "       "*"
```

```
## 8  ( 1 )  "*"       " "       " "       " "       " "       " "       "*"
## 9  ( 1 )  "*"       " "       " "       " "       " "       " "       "*"
## 10  ( 1 ) "*"       " "       " "       " "       " "       " "       "*"
## 11  ( 1 ) "*"       " "       " "       " "       " "       " "       "*"
## 12  ( 1 ) "*"       " "       " "       " "       " "       " "       "*"
## 13  ( 1 ) "*"       " "       " "       " "       " "       " "       "*"
##           PAY_AMT2 PAY_AMT3 PAY_AMT4 PAY_AMT5 PAY_AMT6 ageGroup
## 1  ( 1 )  " "      " "      " "      " "      " "      " "
## 2  ( 1 )  " "      " "      " "      " "      " "      " "
## 3  ( 1 )  " "      " "      " "      " "      " "      " "
## 4  ( 1 )  " "      " "      " "      " "      " "      " "
## 5  ( 1 )  " "      " "      " "      " "      " "      " "
## 6  ( 1 )  " "      " "      " "      " "      " "      " "
## 7  ( 1 )  " "      " "      " "      " "      " "      "*"
## 8  ( 1 )  " "      " "      " "      " "      " "      "*"
## 9  ( 1 )  " "      " "      " "      " "      " "      "*"
## 10  ( 1 ) " "      " "      " "      " "      " "      "*"
## 11  ( 1 ) "*"      " "      " "      " "      " "      "*"
## 12  ( 1 ) "*"      " "      " "      " "      " "      "*"
## 13  ( 1 ) "*"      " "      " "      "*"      " "      "*"
```

For feature selection, we used a combination of various methods to derive the most useful features to be fitted to our model. We first plotted a correlation plot of our features with the target variable. Since the target variable is a binary value with values of 0 and 1, the correlation of attributes with the target variable is low and we decided not to use their correlation coefficients with the target variable to determine the selection of attributes.

We first built a regression model from a set of candidate predictor variables by entering and removing predictors based on p values, in a stepwise manner until there is no variable left to enter or remove any more. With this resulting 13 features, we ran a regsubset function on all the attributes to filter out the best 13 features using both the forward and backward method. The result was that both functions yielded the same best 13 features as the initial method by selecting features based on p-values. As such, we will proceed with the model selection using these 13 features.

```
##    LIMIT_BAL SEX EDUCATION MARRIAGE PAY_0 PAY_2 PAY_3 PAY_4 BILL_AMT1
## 1 0.01265823   2         2        1   0.4   0.4   0.1   0.1 0.1857673
## 2 0.13924051   2         2        2   0.1   0.4   0.2   0.2 0.1844181
## 3 0.10126582   2         2        2   0.2   0.2   0.2   0.2 0.2135251
## 4 0.05063291   2         2        1   0.2   0.2   0.2   0.2 0.2329805
## 5 0.05063291   1         2        1   0.1   0.2   0.1   0.2 0.1909230
## 6 0.05063291   1         1        2   0.2   0.2   0.2   0.2 0.2520622
##      PAY_AMT1     PAY_AMT2    PAY_AMT5 train.class ageGroup
## 1 0.000000000 0.0005614947 0.000000000           1        1
## 2 0.000000000 0.0008149415 0.000000000           1        1
## 3 0.001737733 0.0012224122 0.002344506           0        1
## 4 0.002289503 0.0016453668 0.002506277           0        1
## 5 0.002289503 0.0298928678 0.001615365           0        2
## 6 0.002861879 0.0014791188 0.002344506           0        1
```

Model 1: Support Vector Machine

```
##
## Call:
## svm(formula = train.class ~ ., data = train.data, type = "C-classification",
##     kernel = "linear")
##
```

```
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  1
##
## Number of Support Vectors:  11511
```

```
## [1] 7499
```

Table 1: Confusion Matrix for SVM Model

	0	1
0	5663	1255
1	172	409

```
## [1] 0.809708
```

We used a Support Vector Machine model, specifically c-classification, for binary classification to predict whether customers would default in the next month. Each data item is plotted in a 13-dimensional space to find a hyperplane (decision boundary) which distinguishes the two classes of customers. Here we have C as a hyperparameter for penalizing incorrect classifications having a range of values from 0 to infinity. Since our data is most likely inseparable, we map the non-linearly separable data into a higher dimensional space where we can create a hyperplane that can actually separate the classes using linear kernels, which are the same as support vector classifiers which transform our nonlinear data to pass a linear hyperplane to classify data. We then used the SVM model and the obtained hyperplane to predict results based on the attribute columns of data from the test set. Using this model, we obtained an accuracy of 80.97%, a recall of 0.246 and a precision of 0.704.

```
##
## Call:
## svm(formula = train.class ~ ., data = new.data, cross = 10, type = "C-classification",
##     kernel = "linear", cost = 1)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  1
##
## Number of Support Vectors:  15462
```

Table 2: Confusion Matrix for K-fold Cross Validation

	0	1
0	22679	5031
1	683	1605

```
## [1] 0.8095206
```

In order to improve our results, we performed a 10-fold cross validation to maximize the use of the available data for training and then testing. Data is divided into 10 disjoint subsets of almost equal size and for each iteration, 1 subset will be the test set while the other 9 will be the training set. We then compute the average accuracy of the 10 iterations, which is 80.95%. This model has a recall of 0.242 and a precision of 0.701.

```
##
## Call:
## svm(formula = default.payment.next.month ~ ., data = new.train.data,
##     cross = 10, type = "C-classification", kernel = "linear", cost = 1,
##     class.weights = c(NonDefault = 0.3, Default = 0.7))
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  1
##
## Number of Support Vectors:  14432
```

Table 3: Confusion Matrix for Weighted SVM

	NonDefault	Default
NonDefault	5320	939
Default	515	725

```
## [1] 0.8061075
```

Since the data is not balanced, we also performed a weighted SVM as the standard SVM is known to not perform well on imbalanced datasets. Since the proportion of customers who default in our dataset is only 22%, and we know that more emphasis wants to be placed on detecting customers who are likely to default payment in the next month, we want to place more emphasis on correctly classifying customers who default in the next month and hence we set larger weights for Default. This places a larger penalty for wrongly classifying customers who default as customers who do not default payment. In our model, we set a weight of 0.3 for Non Default and 0.7 for Default. For our model, the accuracy is 80.6%, recall is 0.436 and precision is 0.585. These values show us that we can train a model with a similar accuracy while placing a higher emphasis on correctly classifying customers who default.

Model 2: Logistic Regression

```
##
## Call:
## glm(formula = train.class ~ LIMIT_BAL + SEX + EDUCATION + MARRIAGE +
##     PAY_0 + PAY_2 + PAY_3 + PAY_4 + BILL_AMT1 + PAY_AMT1 + PAY_AMT2 +
##     PAY_AMT5 + ageGroup, family = "binomial", data = train.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1631  -0.6972  -0.5474  -0.2866   3.5935
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -5.01539    0.64516  -7.774 7.61e-15 ***
## LIMIT_BAL    -0.60131    0.13984  -4.300 1.71e-05 ***
## SEX2         -0.09777    0.03535  -2.766 0.005680 **
## EDUCATION1    1.29451    0.25590   5.059 4.22e-07 ***
## EDUCATION2    1.17513    0.25505   4.607 4.08e-06 ***
## EDUCATION3    1.16798    0.25723   4.541 5.61e-06 ***
## EDUCATION4    0.47968    0.47477   1.010 0.312337
## MARRIAGE1     1.36059    0.58614   2.321 0.020272 *
```

```
## MARRIAGE2     1.17451    0.58633   2.003 0.045162 *
## MARRIAGE3     1.16151    0.60723   1.913 0.055774 .
## PAY_0         5.87871    0.20428  28.778  < 2e-16 ***
## PAY_2         0.82424    0.23293   3.539 0.000402 ***
## PAY_3         0.80355    0.26246   3.062 0.002201 **
## PAY_4         0.64135    0.22918   2.798 0.005134 **
## BILL_AMT1    -1.34868    0.27718  -4.866 1.14e-06 ***
## PAY_AMT1    -11.46580    2.20500  -5.200 1.99e-07 ***
## PAY_AMT2    -12.48322    2.82407  -4.420 9.86e-06 ***
## PAY_AMT5     -1.53425    0.74434  -2.061 0.039281 *
## ageGroup      0.12305    0.03925   3.135 0.001720 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 23766  on 22498  degrees of freedom
## Residual deviance: 20850  on 22480  degrees of freedom
## AIC: 20888
##
## Number of Fisher Scoring iterations: 6
```

Table 4: Confusion Matrix for Logistic Regression

	0	1
0	5664	1243
1	171	421

We used a logistic regression model to predict the default status of customers in the next month. Firstly, we trained the model using the train dataset, which contained the information of the 13 features that we chose.

After we trained the model, we then passed in the test dataset to the model, with the parameter `type` set to "Response". This allowed us to get a value output between the range of 0 to 1. We then set the threshold to 0.5. Customers who had a model output value of 0.5 and above were predicted to have defaulted, while those with an output value of less than 0.5 were predicted to not have defaulted. Our results are summarized in the table below. We achieved an accuracy of 81.1%, precision of 0.711, and a recall of 0.253. However, the data is not balanced as 77.8% of the dataset contains data from those who have not defaulted. As such, a naive model that predicts everyone as not defaulting would not have fared much worse than this model. Despite this, from the corrplot shown above, there is little correlation between the target value, and the other parameters. As such, a logistic regression may not have been the best choice for a model in this case.

Model 3: Neural Network

```
## # weights:  121
## initial  value 13362.610195
## iter  10 value 10553.993434
## iter  20 value 10057.346817
## iter  30 value 9952.067003
## iter  40 value 9835.180474
## iter  50 value 9767.599993
## iter  60 value 9698.399029
## iter  70 value 9665.350832
## iter  80 value 9657.628526
```

```
## iter  90 value 9652.507154
## iter 100 value 9648.588326
## iter 110 value 9646.292015
## iter 120 value 9643.575675
## iter 130 value 9637.597035
## iter 140 value 9633.730615
## iter 150 value 9631.972505
## iter 160 value 9630.651957
## iter 170 value 9630.192312
## iter 180 value 9628.955850
## iter 190 value 9627.508104
## iter 200 value 9626.285196
## iter 210 value 9625.086571
## iter 220 value 9624.225622
## iter 230 value 9623.897013
## iter 240 value 9623.584338
## iter 250 value 9623.485703
## iter 260 value 9623.374440
## iter 270 value 9623.070491
## iter 280 value 9622.321837
## iter 290 value 9621.479596
## iter 300 value 9620.619407
## iter 310 value 9619.253306
## iter 320 value 9617.003792
## iter 330 value 9613.757118
## iter 340 value 9610.774228
## iter 350 value 9606.493928
## iter 360 value 9602.360779
## iter 370 value 9599.860936
## iter 380 value 9598.769649
## iter 390 value 9597.538957
## iter 400 value 9596.101205
## iter 410 value 9594.722251
## iter 420 value 9593.915330
## iter 430 value 9593.145551
## iter 440 value 9592.287854
## iter 450 value 9591.821472
## iter 460 value 9591.287761
## iter 470 value 9590.855733
## iter 480 value 9590.214657
## iter 490 value 9589.737658
## iter 500 value 9589.602657
## iter 510 value 9589.449140
## iter 520 value 9589.028689
## iter 530 value 9588.711019
## iter 540 value 9588.245357
## iter 550 value 9587.737130
## iter 560 value 9587.338802
## iter 570 value 9586.997397
## iter 580 value 9586.593713
## iter 590 value 9585.740136
## iter 600 value 9584.744382
## iter 610 value 9584.310803
## iter 620 value 9584.043252
```

```
## iter 630 value 9583.923617
## iter 640 value 9583.821763
## iter 650 value 9583.625685
## iter 660 value 9583.551747
## iter 670 value 9583.455922
## iter 680 value 9583.383826
## iter 690 value 9583.307469
## iter 700 value 9583.146672
## iter 710 value 9582.962853
## iter 720 value 9582.881397
## iter 730 value 9582.821544
## final  value 9582.818874
## converged
```

Table 5: Confusion Matrix for Neural Network

	0	1
0	5519	316
1	1024	640

```
## [1] 0.8213095
```

Lastly, we used a neural network model to predict the default status of customers in the next month. We passed into the function the formula that comprises the target variable denoted by `train.class` and the 13 features used to train the model.

We then performed grid search hyperparameter tuning on values of decay from 0.01 to 0.10 with a step size of 0.01 in order to penalize large weights and size to vary the number of units in the hidden layer from 3 to 15 with a step size of 1, recording the test accuracy of each iteration in a dataframe. From our dataframe, we found the optimal values of size and decay to be 6 and 0.01 respectively. Hence, we proceed to fit our neural network model with the 13 attributes and these parameter values.

This model ended up with an accuracy of 82.1%, a recall of 38.5% and an average class accuracy of 66.5%.

Model Performance

Table 6: Model Performance

	accuracy	recall	avg.class.accuracy
Neural Network (decay constant = 0.01, number of units in the hidden layer = 6)	82.1	37.9	66.3
Logistic Regression	81.1	25.3	61.2
SVM	81.0	24.6	60.8
SVM (K-fold Cross Validation)	81.0	24.2	60.6
SVM (Weighted)	80.6	43.6	67.3

As we are trying to find out if the customer will default on credit card payment in the next month, we want to heavily penalize false negatives, which are instances where we wrongly classified positive instances as negative. Hence, we include recall as a metric for evaluation as well. If we solely base our evaluation metric on the value of recall, we can consider the fact that if we predict all values to be true, we will get a recall of 100%. However, we also take into account the accuracy of the model so we choose the model that gives us the highest balance between accuracy and recall. We also include the metric of average class accuracy since our dataset has a higher proportion of non-default customers at 78% compared to default customers at

22%. As such, classification accuracy can mask poor performance when data is imbalanced so we include the average class accuracy metric to check for poor performances in either classes.

```
## [1] 0.2209876
```

```
## [1] 0.2218963
```

We also ensured that the proportion of default customers in both the train and test set data are equal so as to ensure that the model will not be biased. Upon calculation, the proportion of default customers in the train and test data are 0.2210 and 0.222 respectively.

Comparing among our 5 models of Neural Network, Logistic Regression and Support Vector Machine models, we have come to the conclusion that the most optimal models to use are the neural network and support vector machine (weighted) model based on our 3 evaluation metrics of Accuracy, recall and average class accuracy.

In terms of prediction accuracy, the neural network model is the most optimal as it has the highest absolute accuracy of 82.1%.

However in this context of defaulting bank customers, a higher recall may be prioritised. This is because the effects of false negatives may be more detrimental for the bank if a customer is predicted as non-default but ends up defaulting. In this case, the weighted support vector machine may be preferred over the neural network model as it has a higher recall of 43.6% compared to that of 38.5% in the neural network model. This is due to the fact that the weights in the SVM model penalise the false negatives more heavily.

Possible Improvements

Our models can be evaluated based on different metrics like the ROC index and Kolmogorov-Smirnov Statistic Index to provide a more complete picture of model performance. For each model, we have come up with some suggestions to possibly improve its performance below.

Support Vector Machine

For our SVM model, we can consider changing the type of kernel used. For classification tasks, another commonly used kernel is the RBF kernel because of its good general performance and the few number of parameters (only two: C and γ). We can also choose to perform gridsearch to obtain optimal parameters like C, gamma and coef0 using the tune.svm() function to obtain the best model. The authors of libsvm suggest trying small and large values for C—like 1 to 1000—first, then to decide which are better for the data by cross validation, and finally to try several γ's for the better C's. gamma is the coefficient of the power term in the radial basis function kernel. The larger gamma is, the richer the feature space is, thus the less error for the training set; however, it may also lead to bad overfitting. Also for the SVM k-fold and weighted model, we can perform gridsearch to obtain the best values of the number of folds and the weights assigned to penalize the wrong classification of each class in order to best fit our objectives.

Logistic Regression

Changing threshold values for the logistic regression. We could plot an ROC curve in order to determine the ideal value for the threshold limit that maximizes the true positive rate and minimizes the false positive rate. Alternatively, we can plot the curve based on false negative rate if we wish to maximize recall.

Neural Network

We can increase the number of hidden layers to the neural network in order to capture the benefits of using a deep-learning learning network. Adding more layers to our model increases its ability to learn the dataset's features more deeply. The argument value of entropy can be varied to specify a different loss function eg. least squares.