DSA211 - Statistical Learning With R

Project Part 2 Report

Section Number: G1

Instructor: Professor Goh Jing Rong

Toh Jing Lin Cheryl	**01430750**	**cheryl.toh.2021**
Foo Chuan Wei	**01394670**	**cwfoo.2021**
Nguyen Hanh Trang	**01422031**	**htnguyen.2021**
Seah Li Ping Megan	**01395272**	**megan.seah.2019**
Sharafinaz Binte Shawal	**01443439**	**sharafinazs.2021**

1.0 Introduction

The aim of this report is to find the best predictive model to predict Balance against the given variables Income, Limit, Rating, Cards, Age, Education, Gender, Married and Ethnicity in the Bank2023P.csv dataset. This will be done by directly estimating the test mean squared error (MSE) as it is more accurate than using indirect estimators. The best model obtained will be the model that has the lowest test MSE out of all methods used in this report.

2.0 Justification of Validation Method

There are many validation methods available such as validation set, leave-one-out cross validation (LOOCV), k-fold cross validation (CV) and bootstrap method. Our team determined that the k-fold cross validation to be the best method. The reasons are as follows.

Firstly, the validation set method was rejected as it is not comprehensive since only a subset of the data is used to fit the model and its validation set error may overestimate the test error. In addition, when the model is parsed through multiple different validations, the MSE tends to fluctuate and can be highly variable.

Next, we have the bootstrap method, which is a resampling method. The use of bootstrap data for cross validation as the training set whilst the original as the test set is not considered as about ⅔ of the original data will appear in each bootstrap sample leading to underestimation of the true prediction error.

Lastly, we have our cross validation methodology. The LOOCV when k=n was not considered as it has a high variance and low bias stemming from the fact that each the test mse from each fold are highly correlated with one another. Considering the bias-variance trade-off, our team decided on a k-fold cross validation where k=10.

3.0 Justification of Predictive Methods

To build our initial model, we will involve the quadratic term of $Income^2$ due to our preliminary analysis of our pairplot (Appendix, Output 2.0). The Best Subset Selection was used as it is the most comprehensive as compared to Forward/Backward Stepwise Selection. The 10-fold CV was used in tandem so as to obtain the minimum test MSE for us to decide what are the best predictors to use in the model.

Thereafter, regularisation methods such as Lasso and Ridge Regression with 10-fold Cross validation were used to optimise the model further with the aim of obtaining a lower MSE than the initial models.

Additionally, we explored the use of tree-based methods, specifically bagging and random forest. As compared to a single decision tree which would underestimate the complexity of the problem, the bagging and random forest procedures utilise multiple trees, which would result in a significantly higher prediction accuracy and hence a lower test MSE.

4.0 Best Subset Selection

Firstly, we conducted the Best Subset Selection that considers all exhaustive possible models before selecting the best subset of variables, with all the main factors inclusive of the quadratic term of $Income^2$ by using the k-folds Cross Validation to select the optimal model. We set k to be 10, which would be the default number of folds.

```
  (Intercept)        Rating         Cards  GenderFemale   I(Income^2)
 9.815291e+02  6.335510e-01 -6.172657e+01 -4.813148e+02  3.727802e-04
```

Figure 1: Best Subset Selection

From the results, we were able to identify the factors Rating, Cards, Gender and $Income^2$ to be the most important factors. We will reconsider the Income variable again in our best linear model as we are not certain that when Income = 0 , the global extremum occurs at that particular point, hence we cannot omit Income = 0. In addition when comparing a linear model including Income with the above factors, it is observed that it has a lower test MSE value of (369589.5) as compared to (369872.2) when it has no Income as shown in Figures 2 and 3. Hence, when including income, we call this model **M0** and it has a test MSE value of 369589.5. The best subset selection model helps in identifying a subset of the 10 predictors that are related to the response, in which we fit all ${}_{10}C_k$ models which contain exactly k (where k = 1, 2, … 10) predictors and pick the best model. We then used 10-fold cross-validation onto the models in finding the model with lowest test MSE value.

```
Warning: non-uniform 'Rounding' sampler used
Call:
glm(formula = Balance ~ Rating + Cards + Gender + I(Income^2),
    data = bank)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1968.02   -398.66     15.78    426.85   1681.70

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   9.815e+02  6.002e+01  16.352  < 2e-16 ***
Rating        6.336e-01  1.494e-01   4.241 2.44e-05 ***
Cards        -6.173e+01  1.547e+01  -3.990 7.09e-05 ***
GenderFemale -4.813e+02  3.839e+01 -12.538  < 2e-16 ***
I(Income^2)   3.728e-04  2.091e-06 178.288  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 367593.9)

    Null deviance: 1.2288e+10  on 999  degrees of freedom
Residual deviance: 3.6576e+08  on 995  degrees of freedom
AIC: 15660

Number of Fisher Scoring iterations: 2

[1] 369872.2
```

Test MSE: 369872.2

Figure 2: Linear Model **M0** (Including Income)

```
Call:
glm(formula = Balance ~ Rating + Cards + Gender + I(Income^2) +
    Income, data = bank)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1913.30   -404.00     15.02    425.57   1699.91

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   8.752e+02  8.260e+01  10.595  < 2e-16 ***
Rating        6.376e-01  1.492e-01   4.273 2.12e-05 ***
Cards        -6.228e+01  1.545e+01  -4.030 6.01e-05 ***
GenderFemale -4.813e+02  3.834e+01 -12.552  < 2e-16 ***
I(Income^2)   3.572e-04  8.595e-06  41.557  < 2e-16 ***
Income        9.532e-02  5.092e-02   1.872   0.0615 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 366671.3)

    Null deviance: 1.2288e+10  on 999  degrees of freedom
Residual deviance: 3.6447e+08  on 994  degrees of freedom
AIC: 15658

Number of Fisher Scoring iterations: 2

[1] 369589.5
```

Test MSE: 369589.5

Figure 3: Linear Model (Not including Income)

Considering our best linear model, we conducted another best subset selection by using 10-fold cross validation to find the best model with interaction terms between the identified factors up to 2 degrees. We did not consider the interactions between Income and $Income^2$ as the interaction between them would suggest a cubic relationship while on the pairplot, we can clearly see the relationship is quadratic. With that, the best subset is looking at 14 models with up to 14 variables. Our team noted that this method of selecting the best model with interaction terms may omit the main effects of the interaction terms, hence violating the hierarchy principle. However, despite this, our team still proceeded with choosing the model with the lowest test MSE value as it is a model provided by the best subset that produces the lowest test MSE value. With that, we have the following results as shown on Figure 4.

```
   (Intercept)                   Cards            GenderFemale            I(Income^2)      Cards:GenderFemale
  7.817934e+02           -9.265952e+01            1.860636e+02           4.024314e-04            7.693033e+01
 Rating:Income GenderFemale:I(Income^2)
  2.034295e-04           -8.633219e-05
```

Figure 4: Best Subset Selection with 2nd Degree Interaction Terms **(M1)**

We call this model **M1,** which went through another round of 10-fold cross validation to obtain a test MSE value of 206463. Our team then decided to investigate the best model obtainable for interaction terms up to third degree to see whether there are any presence of 3rd degree interactions. Similarly, as per the 2nd degree, we omitted the interactions between Income and $Income^2$. The results we obtained from the best subset selection of 3rd degree interactions was the same model M1 as that of the best subset selection on 2nd degree interaction. Hence, this seems to suggest that there is no presence of a 3rd degree interaction. With that, we conclude that the best model obtained thus far was M1 with the lowest test MSE value of 206463 as opposed to the linear model M0's 369589.5.

5.0 Ridge Regression

Ridge regression is a regularisation method used when collinearity is present in the data. Ridge regression reduces the coefficient values of the model by introducing a penalty term in order to obtain the minimum test MSE value in each case.

We applied ridge regression on the best subset model M1 (up to 14 variables) to see whether it can help improve the model further. Additionally we also tested the global second degree regression model M2 and the global third degree regression model M3 (which we had used previously in the best subset selection) to see whether a better model can be found. Train-validation split is set to 80:20 as empirical studies have shown that using 70-80% of the data for the training set and 20-30% of the data for the validation set gives the best performance (Gholamy et al., 2018). The threshold is set at 1e-7 since the smaller the threshold is, the better the estimate will be. We have also chosen to implement the function over the default grid.

When applied to the best subset model M1, the best lambda obtained was 348.1693 and the corresponding test MSE value was 430769.5 (Appendix, 5.0 Output). For M2 regression model, M2 = Balance ~ Rating + Gender + Cards + Income + I(Income^2) + Income*Rating + Rating*Gender + Rating*Cards + Rating*I(Income^2) + Income*Cards + Income*Gender + I(Income^2)*Gender + I(Income^2)*Cards + Gender*Cards. The lambda obtained was 348.1693 and the test MSE value was 22795742.

Lastly, we applied ridge regression on M3 regression model, M3 = Balance ~ Rating*Cards*Gender + Rating*Income*Cards + Rating*I(Income^2)*Cards + Rating*Income*Gender + Rating*I(Income^2)*Gender + Income*Cards*Gender + I(Income^2)*Cards*Gender. The lambda obtained was 348.1693 and the test MSE value was 343435. We call this model **M4**.

```
          (Intercept)                Rating                    Cards              GenderFemale
         8.238732e+02         -8.515295e-01            -1.407854e+02             -4.470687e+01
               Income           I(Income^2)             Rating:Cards       Rating:GenderFemale
         6.796392e-01          1.636782e-04            -7.811637e-02              1.981655e-01
   Cards:GenderFemale         Rating:Income             Cards:Income        Rating:I(Income^2)
         6.353085e+01          2.868507e-04             8.183316e-03              1.625202e-07
    Cards:I(Income^2)   GenderFemale:Income   GenderFemale:I(Income^2)   Rating:Cards:GenderFemale
         1.522924e-05         -6.299565e-02             7.882790e-06              2.197922e-01
  Rating:Cards:Income  Rating:Cards:I(Income^2)  Rating:GenderFemale:Income Rating:GenderFemale:I(Income^2)
        -7.463330e-05          5.024893e-09            -3.133168e-04             -3.991427e-08
```

Figure 5: Best Model for Ridge Regression (**M4**)

We observe that the M4 regression model had the lowest MSE amongst the 3 models but much higher than the test MSE of the best subset model M1. Hence, we conclude that ridge regression is not suitable to be used in this case.

6.0 Lasso Regression

Lasso regression is a regularisation method to generate more accurate predictions. Lasso may drop predictors by reducing coefficients to zero. The main objective is to reduce the quantity of predictors. As in ridge regression, selecting the best lambda is important and this was conducted using 10-fold cross validation.

Similar to ridge regression, firstly, we applied lasso regression on M1 using the regression model in Section 4.0 to see whether there are further improvements to our current best model. The lambda obtained was 5.674292 and the test MSE value was 203054.9 which is smaller than the test MSE value from ridge regression and the best subset regression. The final predictors returned were I(Income^2), GenderFemale, Cards, Rating*Income.

Additionally, we tested second degree regression model M2 and third degree regression model M3 to see whether there is a missing presence of interaction terms that was not captured previously in the best subset selection.

For M2 regression model, the lambda obtained was 5.674292 and the test MSE value was 203188.3 which is larger than the test MSE value from M1. The final predictors returned were Rating, GenderFemale, Cards, Income, I(Income^2), Rating*Income,Rating*GenderFemale,Rating*I(Income^2),GenderFemale*I(Income^2). Hence, we do not consider M2 as our final model.

For M3 regression model, the lambda obtained was 4.710897 and the test MSE value was 201781.3 which has the smallest test MSE value. The final predictors returned are Cards, GenderFemale, Income, I(Income^2), Rating*GenderFemale, Cards*GenderFemale,Rating*Income,Rating*I(Income^2),GenderFemale*I(Income^2),Rating*Cards*GenderFemale,Rating*Cards*I(Income^2),Rating*GenderFemale*I(Income^2). Since the test MSE value for this model is the lowest, we consider this lasso regression to be our final model and we call this model **M5.**

```
             (Intercept)                    Cards             GenderFemale                  Income
            7.339071e+02            -8.612623e+01             1.028961e+02            1.201578e-01
             I(Income^2)      Rating:GenderFemale       Cards:GenderFemale           Rating:Income
            3.749728e-04             1.749931e-01             6.591505e+01            1.014376e-05
      Rating:I(Income^2) GenderFemale:I(Income^2) Rating:Cards:GenderFemale Rating:Cards:I(Income^2)
            4.319330e-08            -7.447831e-05             3.387013e-03            1.377175e-09
Rating:GenderFemale:I(Income^2)
           -2.260941e-08
```

Figure 6: Best Model for lasso regression (**M5**)

7.0 Tree-based Methods

For the tree-based methods, we used only the base model, i.e., the full set of first-degree terms (Income, Limit, Rating, Cards, Age, Education, Gender, Married and Ethnicity) and $Income^2$. Interaction terms were not explicitly included because tree-based methods inherently account for interactions between variables.

We explored the use of both bagging and random forest procedures. For bagging, mtry is set to p, where p is the total number of predictors, in this case, 10, hence there is no random selection of predictors. While for random forest, mtry is set to approximately $\sqrt{p}$, in this case, 3, meaning that 3 predictors are randomly selected to build the decision trees at each step of tree splitting. Due to the abovementioned difference between the two procedures, random forest is usually a better approach than the bagging procedure due to the reduction of correlation relationship among the trees, which reduces the variance when averaging the trees.

The test MSE value obtained from the bagging procedure was 249680.6 (Appendix, 7.1 Output) while the test MSE value obtained from the random forest procedure was 272117.8 (Appendix, 7.2 Output).

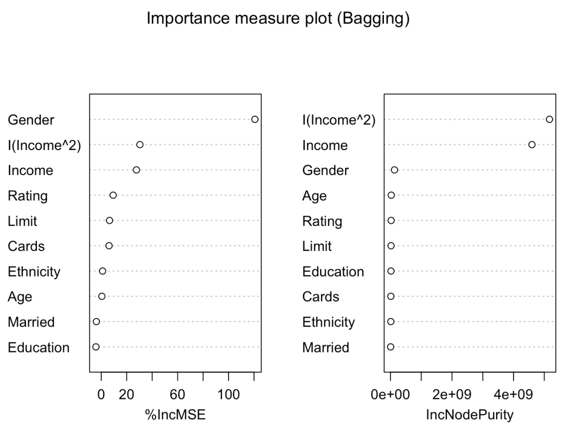


Figure 7: Importance measure plot (Bagging)

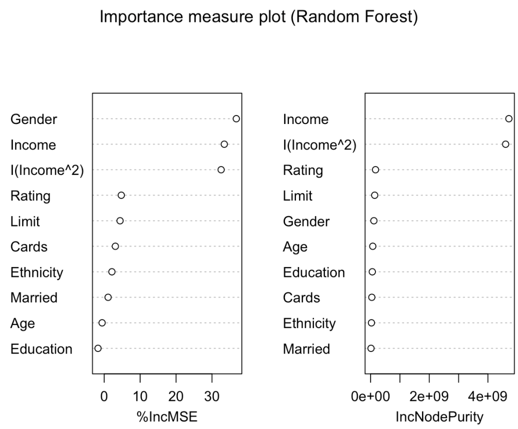


Figure 8: Importance measure plot (Random Forest)

Based on the importance measure plots as shown in Figures 7 and 8 zooming into the %IncMSE plots which indicate the increase in the test MSE due to permutations of each variable, we observed that the ranking of the importance of the variables are similar for both bagging and random forest procedures, with the exception of Married and Age. The variables with highest %IncMSE include Gender, $Income^2$, Income, which is consistent with our results from M5.

However, the test MSE values for both tree-based methods are higher than that of M5, likely because random forest cannot capture some interaction effects, particularly those between variables that do not have sufficiently strong marginal effects (Hornung & Boulesteix, 2022).

8.0 Conclusion

We conclude the best model came from the lasso regression, M5. The respective coefficients can be found in Figure 6 (Section 6.0). The lambda value was 4.710897 and the test MSE value was 201781.3 which is the smallest test MSE value out of all the models we tested.

References

Gholamy, A., Kreinovich, V., & Kosheleva, O. (2018). *Why 70/30 or 80/20 relation between training and testing sets: A pedagogical explanation*. ScholarWorks@UTEP. https://scholarworks.utep.edu/cs_techrep/1209/#:~:text=We%20first%20train%20our%20model,of%20the%20data%20for%20training

Hornung, R., & Boulesteix, A.-L. (2022). Interaction forests: Identifying and exploiting interpretable quantitative and qualitative interaction effects. *Computational Statistics & Data Analysis*, *171*, 107460. https://doi.org/10.1016/j.csda.2022.107460

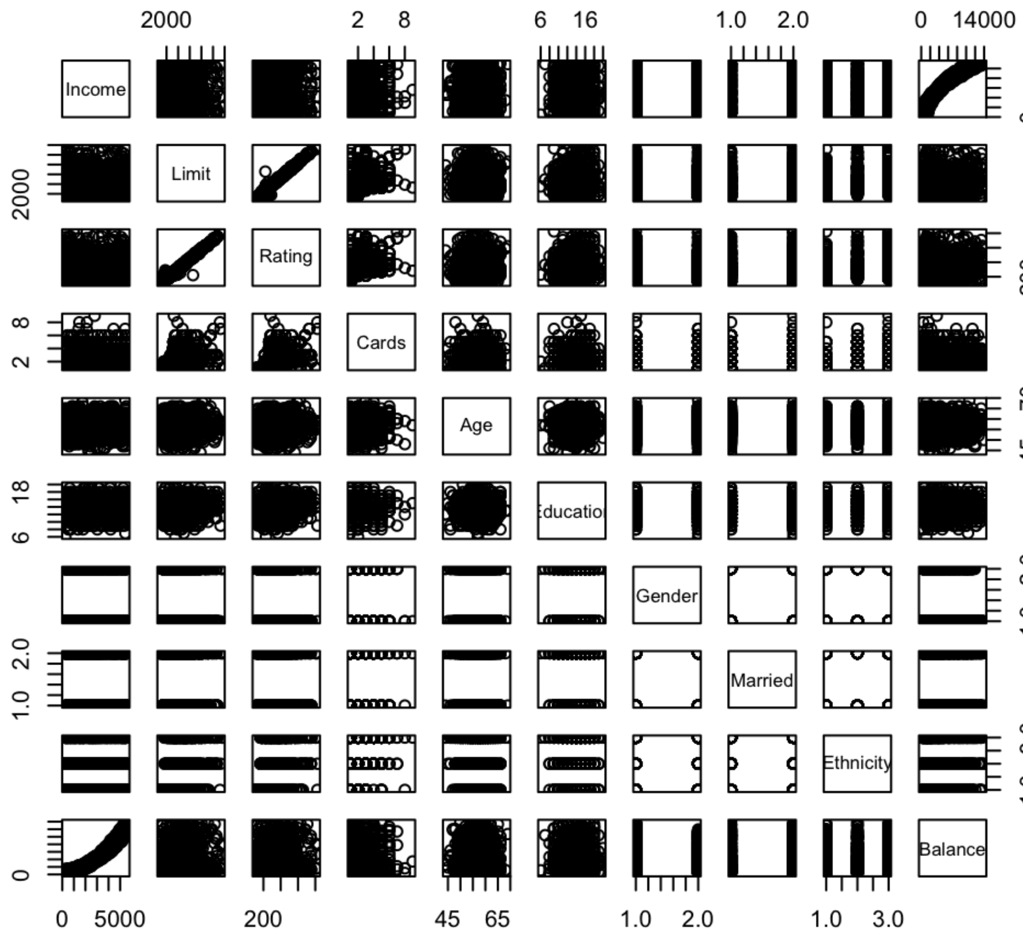
Appendix: R Inputs and Outputs

All figures and methods used in the report may be found in the Appendix below.

3.0 Input

```
bank <- read.csv("Bank2023P.csv", stringsAsFactors = TRUE)
attach(bank)
#spotting linear relationships
pairs(bank)
```

3.0 Output



4.0 Input

```
RNGkind(sample.kind = "Rounding")
set.seed(123)

lm1 <- lm(Balance~. + I(Income^2), data = bank)
summary(lm1)

library(leaps)
set.seed(123)

regfit1.all <- regsubsets(Balance~. + I(Income^2), bank, nvmax = 10)
summary(regfit1.all)
```

```
RNGkind(sample.kind = "Rounding")
set.seed(123)

predict.regsubsets <- function(object, newdata, id){
  form <- as.formula(object$call[[2]])
  mat <- model.matrix(form, newdata)
  coefi <- coef(object, id = id)
  xvars <- names(coefi)
  mat[, xvars]%*%coefi
}

k <- 10
folds <- sample(1:k, nrow(bank), replace = TRUE)

cv.errors <- matrix(NA, k, 10, dimnames = list(NULL, paste(1:10)))
for (j in 1:k) {
  best.fit <- regsubsets(Balance~.+ I(Income^2), data=bank[folds!=j,], nvmax=10)
  for (i in 1:10){
    pred <- predict.regsubsets(best.fit, bank[folds==j,], id=i)
    cv.errors[j,i] <- mean((bank$Balance[folds==j]-pred)^2)
```

```
  }
}

mean.cv <- apply(cv.errors, 2, mean)
mean.cv
bb <- which.min(mean.cv)
bb

coef(regfit1.all, bb)
```

```
# Linear Model
RNGkind(sample.kind = "Rounding")
set.seed(123)
library(boot)

# Linear Model (Not including Income)
L1 <- glm(Balance~Rating+Cards+Gender+I(Income^2), data = bank)
summary(L1)
f]

# Linear Model (Including Income)
L2 <- glm(Balance~Rating+Cards+Gender+I(Income^2) + Income, data = bank)
summary(L2)
cv.error2 <- cv.glm(bank, L2, K = 10)
cv.error2$delta[1]
```

```
#2nd Degree Interaction best subset
library(leaps)
bank <- read.csv("Bank2023P.csv",stringsAsFactors = TRUE)
attach(df)

RNGkind(sample.kind = "Rounding")
set.seed(123)
```

```
regfit1.all <-
regsubsets(Balance~Rating+Gender+Cards+Income+I(Income^2)+Income*Rating+
Rating*Gender+Rating*Cards+Rating*I(Income^2)+Income*Cards+Income*Gender
+I(Income^2)*Gender+I(Income^2)*Cards+Gender*Cards, bank, nvmax=20)  # run
the Best Subset Selection
summary(regfit1.all)
# write a function to do the prediction
set.seed(123)
predict.regsubsets <- function(object, newdata, id){
  form <- as.formula(object$call[[2]])
  mat <- model.matrix(form, newdata)
  coefi <- coef(object, id = id)
  xvars <- names(coefi)
  mat[, xvars]%*%coefi
}

k <- 10
folds <- sample(1:k, nrow(bank), replace = TRUE)

cv.errors <- matrix(NA, k, 14, dimnames = list(NULL, paste(1:14)))
for (j in 1:k) {
  best.fit <-
regsubsets(Balance~Rating+Gender+Cards+Income+I(Income^2)+Income*Rating+
Rating*Gender+Rating*Cards+Rating*I(Income^2)+Income*Cards+Income*Gender
+I(Income^2)*Gender+I(Income^2)*Cards+Gender*Cards, data=bank[folds!=j,],
nvmax=14)
  for (i in 1:14){
    pred <- predict.regsubsets(best.fit, bank[folds==j,], id=i)
    cv.errors[j,i] <- mean((bank$Balance[folds==j]-pred)^2)
  }
}
```

```
mean.cv <- apply(cv.errors, 2, mean)
mean.cv
bb <- which.min(mean.cv)
bb
coef(regfit1.all, bb)
```

```
#Test MSE of M1
library(boot)
RNGkind(sample.kind = "Rounding")
set.seed(123)
CW1 <- glm(Balance~Gender + Cards + I(Income^2) + Rating*Income +
Gender*I(Income^2) + Gender*Cards - Rating - Income, data = bank)
cv.err1 <- cv.glm(bank, CW1, K = 10)
cv.err1$delta[1]
```

```
RNGkind(sample.kind = "Rounding")
set.seed(123)

regfit1.all <- regsubsets(Balance~Rating*Cards*Gender + Rating*Income*Cards +
Rating*I(Income^2)*Cards + Rating*Income*Gender+Rating*I(Income^2)*Gender +
Income*Cards*Gender + I(Income^2)*Cards*Gender, bank, nvmax=21)  # run the
Best Subset Selection
summary(regfit1.all)
# write a function to do the prediction
set.seed(123)
predict.regsubsets <- function(object, newdata, id){
  form <- as.formula(object$call[[2]])
  mat <- model.matrix(form, newdata)
  coefi <- coef(object, id = id)
  xvars <- names(coefi)
  mat[, xvars]%*%coefi
```

```
}

k <- 10
folds <- sample(1:k, nrow(bank), replace = TRUE)

cv.errors <- matrix(NA, k, 21, dimnames = list(NULL, paste(1:21)))
for (j in 1:k) {
  best.fit <- regsubsets(Balance~Rating*Cards*Gender + Rating*Income*Cards + 
Rating*I(Income^2)*Cards + Rating*Income*Gender+Rating*I(Income^2)*Gender + 
Income*Cards*Gender + I(Income^2)*Cards*Gender, data=bank[folds!=j,], 
nvmax=21)
  for (i in 1:21){
    pred <- predict.regsubsets(best.fit, bank[folds==j,], id=i)
    cv.errors[j,i] <- mean((bank$Balance[folds==j]-pred)^2)
  }
}

mean.cv <- apply(cv.errors, 2, mean)
mean.cv
#min  hierarchy principle may be violated but isok

bb <- which.min(mean.cv)

coef(regfit1.all,bb )
```

4.0 Output

```
Call:
lm(formula = Balance ~ . + I(Income^2), data = bank)

Residuals:
    Min      1Q  Median      3Q     Max 
-1922.4  -400.6     4.1   412.4  1693.9 

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)    
(Intercept)         1.315e+03  2.950e+02   4.458 9.21e-06 ***
Income              9.932e-02  5.099e-02   1.948 0.051693 .  
Limit               3.166e-03  5.973e-02   0.053 0.957743    
Rating              6.314e-01  9.108e-01   0.693 0.488309    
Cards              -6.083e+01  1.617e+01  -3.762 0.000178 ***
Age                -3.616e+00  4.611e+00  -0.784 0.433148    
Education          -1.498e+01  8.654e+00  -1.731 0.083784 .  
GenderFemale       -4.806e+02  3.843e+01 -12.507  < 2e-16 ***
MarriedYes         -4.792e+01  3.967e+01  -1.208 0.227370    
EthnicityAsian     -3.532e+01  5.726e+01  -0.617 0.537560    
EthnicityCaucasian -1.281e+01  5.307e+01  -0.241 0.809375    
I(Income^2)         3.564e-04  8.609e-06  41.396  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 605.6 on 988 degrees of freedom
Multiple R-squared:  0.9705,    Adjusted R-squared:  0.9702 
F-statistic:  2956 on 11 and 988 DF,  p-value: < 2.2e-16

Subset selection object
Call: regsubsets.formula(Balance ~ . + I(Income^2), bank, nvmax = 10)
11 Variables  (and intercept)
                   Forced in Forced out
Income                 FALSE      FALSE
Limit                  FALSE      FALSE
Rating                 FALSE      FALSE
Cards                  FALSE      FALSE
Age                    FALSE      FALSE
Education              FALSE      FALSE
GenderFemale           FALSE      FALSE
MarriedYes             FALSE      FALSE
EthnicityAsian         FALSE      FALSE
EthnicityCaucasian     FALSE      FALSE
I(Income^2)            FALSE      FALSE
1 subsets of each size up to 10
Selection Algorithm: exhaustive

          Income Limit Rating Cards Age Education GenderFemale MarriedYes EthnicityAsian
1  ( 1 )  " "    " "   " "    " "   " " " "       " "          " "        " "           
2  ( 1 )  " "    " "   " "    " "   " " " "       "*"          " "        " "           
3  ( 1 )  " "    "*"   " "    " "   " " " "       "*"          " "        " "           
4  ( 1 )  " "    " "   "*"    "*"   " " " "       "*"          " "        " "           
5  ( 1 )  "*"    " "   "*"    "*"   " " " "       "*"          " "        " "           
6  ( 1 )  "*"    " "   "*"    "*"   " " "*"       "*"          " "        " "           
7  ( 1 )  "*"    " "   "*"    "*"   " " "*"       "*"          "*"        " "           
8  ( 1 )  "*"    " "   "*"    "*"   "*" "*"       "*"          "*"        " "           
9  ( 1 )  "*"    " "   "*"    "*"   "*" "*"       "*"          "*"        "*"           
10  ( 1 ) "*"    " "   "*"    "*"   "*" "*"       "*"          "*"        "*"           
          EthnicityCaucasian I(Income^2)
1  ( 1 )  " "                "*"        
2  ( 1 )  " "                "*"        
3  ( 1 )  " "                "*"        
4  ( 1 )  " "                "*"        
5  ( 1 )  " "                "*"        
6  ( 1 )  " "                "*"        
7  ( 1 )  " "                "*"        
8  ( 1 )  " "                "*"        
9  ( 1 )  " "                "*"        
10  ( 1 ) "*"                "*"        
```

```
> mean.cv

       1        2        3        4        5        6        7        8        9
433553.7 377325.3 377057.5 369240.1 370007.4 369290.4 369394.0 369807.1 370582.4
      10
370562.4
```

```
> bb

4

4

> coef(regfit1.all, bb)

  (Intercept)        Rating         Cards  GenderFemale   I(Income^2)
 9.815291e+02  6.335510e-01 -6.172657e+01 -4.813148e+02  3.727802e-04
```

```
Warning: non-uniform 'Rounding' sampler used
Call:
glm(formula = Balance ~ Rating + Cards + Gender + I(Income^2),
    data = bank)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1968.02   -398.66     15.78    426.85   1681.70

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   9.815e+02  6.002e+01  16.352  < 2e-16 ***
Rating        6.336e-01  1.494e-01   4.241 2.44e-05 ***
Cards        -6.173e+01  1.547e+01  -3.990 7.09e-05 ***
GenderFemale -4.813e+02  3.839e+01 -12.538  < 2e-16 ***
I(Income^2)   3.728e-04  2.091e-06 178.288  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 367593.9)

    Null deviance: 1.2288e+10  on 999  degrees of freedom
Residual deviance: 3.6576e+08  on 995  degrees of freedom
AIC: 15660

Number of Fisher Scoring iterations: 2

[1] 369872.2
```

```
Call:
glm(formula = Balance ~ Rating + Cards + Gender + I(Income^2) +
    Income, data = bank)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1913.30   -404.00     15.02    425.57   1699.91

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   8.752e+02  8.260e+01  10.595  < 2e-16 ***
Rating        6.376e-01  1.492e-01   4.273 2.12e-05 ***
Cards        -6.228e+01  1.545e+01  -4.030 6.01e-05 ***
GenderFemale -4.813e+02  3.834e+01 -12.552  < 2e-16 ***
I(Income^2)   3.572e-04  8.595e-06  41.557  < 2e-16 ***
Income        9.532e-02  5.092e-02   1.872   0.0615 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 366671.3)

    Null deviance: 1.2288e+10  on 999  degrees of freedom
Residual deviance: 3.6447e+08  on 994  degrees of freedom
AIC: 15658

Number of Fisher Scoring iterations: 2

[1] 369589.5
```

```
Subset selection object
Subset selection object
Call: regsubsets.formula(Balance ~ Rating + Gender + Cards + Income +
    I(Income^2) + Income * Rating + Rating * Gender + Rating *
    Cards + Rating * I(Income^2) + Income * Cards + Income *
    Gender + I(Income^2) * Gender + I(Income^2) * Cards + Gender *
    Cards, bank, nvmax = 20)
14 Variables  (and intercept)
                         Forced in Forced out
Rating                       FALSE      FALSE
GenderFemale                 FALSE      FALSE
Cards                        FALSE      FALSE
Income                       FALSE      FALSE
I(Income^2)                  FALSE      FALSE
```

```
Rating:Income                    FALSE      FALSE
Rating:GenderFemale              FALSE      FALSE
Rating:Cards                     FALSE      FALSE
Rating:I(Income^2)               FALSE      FALSE
Cards:Income                     FALSE      FALSE
GenderFemale:Income              FALSE      FALSE
GenderFemale:I(Income^2)         FALSE      FALSE
Cards:I(Income^2)                FALSE      FALSE
GenderFemale:Cards               FALSE      FALSE
1 subsets of each size up to 14
Selection Algorithm: exhaustive
          Rating GenderFemale Cards Income I(Income^2) Rating:Income
Rating:GenderFemale Rating:Cards Rating:I(Income^2)
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 ) "*"    "*"          "*"   "*"    "*"         "*"           "*"
" "          "*"
14  ( 1 ) "*"    "*"          "*"   "*"    "*"         "*"           "*"
```

```
"*"            "*"
          Cards:Income GenderFemale:Income GenderFemale:I(Income^2)
Cards:I(Income^2) GenderFemale:Cards
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 ) "*"          "*"                 "*"                        "*"
"*"
14  ( 1 ) "*"          "*"                 "*"                        "*"
"*"
       1        2        3        4        5        6        7        8
9        10        11        12        13        14
433553.7 235378.2 218952.9 220555.3 212529.9 207599.6 208621.8 208681.6
209062.9 208996.0 209290.1 209288.2 209269.6 209225.6
6
6

   (Intercept)                 Cards          GenderFemale           I(Income^2)   Cards:GenderFemale
  7.817934e+02         -9.265952e+01          1.860636e+02          4.024314e-04         7.693033e+01
 Rating:Income GenderFemale:I(Income^2)
  2.034295e-04         -8.633219e-05
```

```
[1] 2064663
```

```
Subset selection object
Call: regsubsets.formula(Balance ~ Rating * Cards * Gender + Rating *
    Income * Cards + Rating * I(Income^2) * Cards + Rating *
    Income * Gender + Rating * I(Income^2) * Gender + Income *
    Cards * Gender + I(Income^2) * Cards * Gender, bank, nvmax = 21)
21 Variables  (and intercept)
                                Forced in Forced out
Rating                              FALSE      FALSE
Cards                               FALSE      FALSE
GenderFemale                        FALSE      FALSE
Income                              FALSE      FALSE
I(Income^2)                         FALSE      FALSE
Rating:Cards                        FALSE      FALSE
Rating:GenderFemale                 FALSE      FALSE
Cards:GenderFemale                  FALSE      FALSE
Rating:Income                       FALSE      FALSE
Cards:Income                        FALSE      FALSE
Rating:I(Income^2)                  FALSE      FALSE
Cards:I(Income^2)                   FALSE      FALSE
GenderFemale:Income                 FALSE      FALSE
GenderFemale:I(Income^2)            FALSE      FALSE
Rating:Cards:GenderFemale           FALSE      FALSE
Rating:Cards:Income                 FALSE      FALSE
Rating:Cards:I(Income^2)            FALSE      FALSE
Rating:GenderFemale:Income          FALSE      FALSE
Rating:GenderFemale:I(Income^2)     FALSE      FALSE
Cards:GenderFemale:Income           FALSE      FALSE
Cards:GenderFemale:I(Income^2)      FALSE      FALSE
1 subsets of each size up to 21
Selection Algorithm: exhaustive
          Rating Cards GenderFemale Income I(Income^2) Rating:Cards
Rating:GenderFemale Cards:GenderFemale Rating:Income
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 ) " "    "*"   "*"         " "    "*"         "*"          " "
"*"                 "*"
14  ( 1 ) " "    "*"   "*"         " "    "*"         "*"          " "
"*"                 "*"
15  ( 1 ) "*"    "*"   "*"         " "    "*"         "*"          " "
"*"                 "*"
16  ( 1 ) "*"    "*"   "*"         " "    "*"         "*"          "*"
"*"                 "*"
17  ( 1 ) "*"    "*"   "*"         "*"    "*"         "*"          "*"
"*"                 "*"
18  ( 1 ) "*"    "*"   "*"         "*"    "*"         "*"          "*"
"*"                 "*"
19  ( 1 ) "*"    "*"   "*"         "*"    "*"         "*"          "*"
"*"                 "*"
20  ( 1 ) "*"    "*"   "*"         "*"    "*"         "*"          "*"
"*"                 "*"
21  ( 1 ) "*"    "*"   "*"         "*"    "*"         "*"          "*"
"*"                 "*"
          Cards:Income Rating:I(Income^2) Cards:I(Income^2)
GenderFemale:Income GenderFemale:I(Income^2) Rating:Cards:GenderFemale
```

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) "*" "*" "*" " "
"*" " "

14 (1) "*" "*" "*" " "
"*" " "

15 (1) "*" "*" "*" " "
"*" " "

16 (1) "*" "*" "*" " "
"*" " "

17 (1) "*" "*" "*" " "
"*" " "

18 (1) "*" "*" "*" " "
"*" " "

19 (1) "*" "*" "*" " "
"*" " "

20 (1) "*" "*" "*" " "
"*" "*"

```
21  ( 1 ) "*"          "*"                 "*"                "*"
"*"                       "*"
          Rating:Cards:Income Rating:Cards:I(Income^2)
Rating:GenderFemale:Income Rating:GenderFemale:I(Income^2)
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 ) "*"                 "*"                        " "
"*"
14  ( 1 ) "*"                 "*"                        " "
"*"
15  ( 1 ) "*"                 "*"                        "*"
" "
16  ( 1 ) "*"                 "*"                        "*"
" "
17  ( 1 ) "*"                 "*"                        "*"
" "
18  ( 1 ) "*"                 "*"                        "*"
" "
```

```
19  ( 1 ) "*"                 "*"                      "*"
"*"
20  ( 1 ) "*"                 "*"                      "*"
"*"
21  ( 1 ) "*"                 "*"                      "*"
"*"
          Cards:GenderFemale:Income Cards:GenderFemale:I(Income^2)
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 ) " "                       " "
14  ( 1 ) " "                       "*"
15  ( 1 ) " "                       "*"
16  ( 1 ) " "                       "*"
17  ( 1 ) " "                       "*"
18  ( 1 ) "*"                       "*"
19  ( 1 ) "*"                       "*"
20  ( 1 ) "*"                       "*"
21  ( 1 ) "*"                       "*"

  (Intercept)                 Cards          GenderFemale           I(Income^2)      Cards:GenderFemale
 7.817934e+02         -9.265952e+01          1.860636e+02          4.024314e-04            7.693033e+01
Rating:Income GenderFemale:I(Income^2)
 2.034295e-04         -8.633219e-05
```

5.0 Input

```
#Ridge regression on M1
bank = read.csv("Bank2023P.csv", stringsAsFactors = TRUE)
attach(bank)
```

```
library(leaps)
library(glmnet)
RNGkind(sample.kind = "Rounding")
set.seed(123)
train <- sample(1:nrow(bank), 800)
test <- (-train)
bank.train = bank[train,]
bank.test = bank[test,]

train.x = model.matrix(Balance~Gender + Cards + I(Income^2) + Rating*Income +
Gender*I(Income^2) + Gender*Cards - Rating - Income, data = bank.train)
train.y = bank.train$Balance
test.x = model.matrix(Balance~Gender + Cards + I(Income^2) + Rating*Income +
Gender*I(Income^2) + Gender*Cards - Rating - Income,data = bank.test)
test.y = bank.test$Balance

ridge.mod <- glmnet(train.x, train.y, alpha=0)
cvrr.out <- cv.glmnet(train.x, train.y, alpha=0)
r_bestlam <- cvrr.out$lambda.min
r_bestlam
ridge_pred <-predict(ridge.mod, s=r_bestlam, newx=test.x)
mean((ridge_pred - test.y)^2)

x = model.matrix(Balance~Gender + Cards + I(Income^2) + Rating*Income +
Gender*I(Income^2) + Gender*Cards - Rating - Income, bank)
y = bank$Balance
out.rr <- glmnet(x, y,alpha=0)
ridge.coef <- predict(out.rr, type="coefficients", s=r_bestlam)[1:6,]
ridge.coef[ridge.coef!=0]

#Ridge regression on M2
RNGkind(sample.kind = "Rounding")
set.seed(123)
```

```
train <- sample(1:nrow(bank), 800)
test <- (-train)
bank.train = bank[train,]
bank.test = bank[test,]

train.x =
model.matrix(Balance~Rating+Gender+Cards+Income+I(Income^2)+Income*Ratin
g+Rating*Gender+Rating*Cards+Rating*I(Income^2)+Income*Cards+Income*Gen
der+I(Income^2)*Gender+I(Income^2)*Cards+Gender*Cards, data = bank.train)
train.y = bank.train$Balance

test.x =
model.matrix(Balance~Rating+Gender+Cards+Income+I(Income^2)+Income*Ratin
g+Rating*Gender+Rating*Cards+Rating*I(Income^2)+Income*Cards+Income*Gen
der+I(Income^2)*Gender+I(Income^2)*Cards+Gender*Cards, data = bank.train)
test.y = bank.test$Balance

ridge.mod1 <- glmnet(train.x, train.y, alpha=0)
cvrr.out1 <- cv.glmnet(train.x, train.y, alpha=0)
r_bestlam1 <- cvrr.out1$lambda.min
r_bestlam1
ridge_pred1 <-predict(ridge.mod1, s=r_bestlam1, newx=test.x)
mean((ridge_pred1 - test.y)^2)

x =
model.matrix(Balance~Rating+Gender+Cards+Income+I(Income^2)+Income*Ratin
g+Rating*Gender+Rating*Cards+Rating*I(Income^2)+Income*Cards+Income*Gen
der+I(Income^2)*Gender+I(Income^2)*Cards+Gender*Cards, bank)
y = bank$Balance
out.rr1 <- glmnet(x, y,alpha=0)
ridge.coef <- predict(out.rr1, type="coefficients", s=r_bestlam1)[1:14,]
ridge.coef[ridge.coef!=0]
```

```
#Ridge regression on M3
RNGkind(sample.kind = "Rounding")
set.seed(123)
train <- sample(1:nrow(bank), 800)
test <- (-train)
bank.train = bank[train,]
bank.test = bank[test,]

train.x = model.matrix(Balance~Rating*Cards*Gender + Rating*Income*Cards +
Rating*I(Income^2)*Cards + Rating*Income*Gender+Rating*I(Income^2)*Gender
+ Income*Cards*Gender + I(Income^2)*Cards*Gender, data = bank.train)
train.y = bank.train$Balance

test.x = model.matrix(Balance~Rating*Cards*Gender + Rating*Income*Cards +
Rating*I(Income^2)*Cards + Rating*Income*Gender+Rating*I(Income^2)*Gender
+ Income*Cards*Gender + I(Income^2)*Cards*Gender, bank.test)
test.y = bank.test$Balance

ridge.mod2 <- glmnet(train.x, train.y, alpha=0)
cvrr.out2 <- cv.glmnet(train.x, train.y, alpha=0)
r_bestlam2 <- cvrr.out2$lambda.min
r_bestlam2
ridge_pred2 <-predict(ridge.mod2, s=r_bestlam2, newx=test.x)
mean((ridge_pred2 - test.y)^2)

x = model.matrix(Balance~Rating*Cards*Gender + Rating*Income*Cards +
Rating*I(Income^2)*Cards + Rating*Income*Gender+Rating*I(Income^2)*Gender
+ Income*Cards*Gender + I(Income^2)*Cards*Gender, bank)
y = bank$Balance
out.rr2 <- glmnet(x, y,alpha=0)
ridge.coef <- predict(out.rr2, type="coefficients", s=r_bestlam2)[1:21,]
ridge.coef[ridge.coef!=0]
```

5.0 Output

#M1

[1] 348.1693

[1] 430769.5

```
  (Intercept)  GenderFemale         Cards   I(Income^2) Rating:Income 
 1.192945e+03 -2.394884e+02 -1.112152e+02  3.027047e-04  8.393244e-04 
```

#M2

[1] 348.1693

[1] 22795742

```
         (Intercept)                Rating          GenderFemale                 Cards                Income 
        8.701551e+02         -8.370526e-01          1.088094e+02         -1.330945e+02          7.075621e-01 
         I(Income^2)         Rating:Income   Rating:GenderFemale          Rating:Cards    Rating:I(Income^2) 
        1.728151e-04          1.604433e-04         -9.626396e-02         -8.818590e-02          1.470174e-07 
        Cards:Income   GenderFemale:Income GenderFemale:I(Income^2) 
       -9.306217e-03         -1.639671e-01         -2.034968e-05 
```

#M3

[1] 348.1693

[1] 343435

```
               (Intercept)                     Rating                      Cards               GenderFemale 
              8.238732e+02              -8.515295e-01              -1.407854e+02              -4.470687e+01 
                    Income                I(Income^2)               Rating:Cards        Rating:GenderFemale 
              6.796392e-01               1.636782e-04              -7.811637e-02               1.981655e-01 
        Cards:GenderFemale              Rating:Income               Cards:Income         Rating:I(Income^2) 
              6.353085e+01               2.868507e-04               8.183316e-03               1.625202e-07 
         Cards:I(Income^2)        GenderFemale:Income   GenderFemale:I(Income^2)   Rating:Cards:GenderFemale 
              1.522924e-05              -6.299565e-02               7.882790e-06               2.197922e-01 
       Rating:Cards:Income    Rating:Cards:I(Income^2) Rating:GenderFemale:Income Rating:GenderFemale:I(Income^2) 
             -7.463330e-05               5.024893e-09              -3.133168e-04              -3.991427e-08 
```

6.0 Input

 # Lasso regression on M1

```
library(leaps)
library(glmnet)
bank = read.csv("Bank2023P.csv", stringsAsFactors = TRUE)
attach(bank)
RNGkind(sample.kind = "Rounding")
set.seed(123)
```

```
train <- sample(1:nrow(bank), 800)
test <- -train

bank.train = bank[train,]
bank.test = bank[test,]

train.x = model.matrix(Balance~ Gender + Cards + I(Income^2) + Rating*Income +
Gender*I(Income^2) + Gender*Cards - Rating - Income, data = bank.train)
train.y = bank.train$Balance

test.x = model.matrix( Balance~ Gender + Cards + I(Income^2) + Rating*Income +
Gender*I(Income^2) + Gender*Cards - Rating - Income ,bank.test)
test.y = bank.test$Balance

lasso.mod <- glmnet(train.x, train.y, alpha=1)
lassocv.out <- cv.glmnet(train.x, train.y, alpha=1)

lassolam <- lassocv.out$lambda.min
lassolam

lasso.pred <- predict(lasso.mod, s=lassolam, newx=test.x)
mean((lasso.pred-test.y)^2)

x = model.matrix(Balance~ Gender + Cards + I(Income^2) + Rating*Income +
Gender*I(Income^2) + Gender*Cards - Rating - Income, bank)
y = bank$Balance

out.lasso <- glmnet(x,y,alpha=1)
lasso.coef <- predict(out.lasso, type="coefficients", s=lassolam)[1:6,]

lasso.coef[lasso.coef!=0]

# Lasso regression on M2
```

```
RNGkind(sample.kind = "Rounding")
set.seed(123)
train <- sample(1:nrow(bank),800)
test <- -train

bank.train = bank[train,]
bank.test = bank[test,]

train.x =
model.matrix(Balance~Rating+Gender+Cards+Income+I(Income^2)+Income*Ratin
g+Rating*Gender+Rating*Cards+Rating*I(Income^2)+Income*Cards+Income*Gen
der+I(Income^2)*Gender+I(Income^2)*Cards+Gender*Cards, data = bank.train)
train.y = bank.train$Balance

test.x =
model.matrix(Balance~Rating+Gender+Cards+Income+I(Income^2)+Income*Ratin
g+Rating*Gender+Rating*Cards+Rating*I(Income^2)+Income*Cards+Income*Gen
der+I(Income^2)*Gender+I(Income^2)*Cards+Gender*Cards,bank.test)
test.y = bank.test$Balance

lasso.mod <- glmnet(train.x, train.y, alpha=1)
lassocv.out <- cv.glmnet(train.x, train.y, alpha=1)

lassolam <- lassocv.out$lambda.min
lassolam

lasso.pred <- predict(lasso.mod, s=lassolam, newx=test.x)
mean((lasso.pred-test.y)^2)

x =
model.matrix(Balance~Rating+Gender+Cards+Income+I(Income^2)+Income*Ratin
g+Rating*Gender+Rating*Cards+Rating*I(Income^2)+Income*Cards+Income*Gen
der+I(Income^2)*Gender+I(Income^2)*Cards+Gender*Cards, bank)
```

```
y = bank$Balance

out.lasso <- glmnet(x,y,alpha=1)
lasso.coef <- predict(out.lasso, type="coefficients", s=lassolam)[1:14,]

lasso.coef[lasso.coef!=0]

# Lasso regression on M3
RNGkind(sample.kind = "Rounding")
set.seed(123)

train <- sample(1:nrow(bank),800)
test <- -train

bank.train = bank[train,]
bank.test = bank[test,]

train.x = model.matrix(Balance~Rating*Cards*Gender + Rating*Income*Cards +
Rating*I(Income^2)*Cards + Rating*Income*Gender+Rating*I(Income^2)*Gender
+ Income*Cards*Gender + I(Income^2)*Cards*Gender, data = bank.train)
train.y = bank.train$Balance

test.x = model.matrix(Balance~Rating*Cards*Gender + Rating*Income*Cards +
Rating*I(Income^2)*Cards + Rating*Income*Gender+Rating*I(Income^2)*Gender
+ Income*Cards*Gender + I(Income^2)*Cards*Gender,bank.test)
test.y = bank.test$Balance

lasso.mod <- glmnet(train.x, train.y, alpha=1)
lassocv.out <- cv.glmnet(train.x, train.y, alpha=1)

lassolam <- lassocv.out$lambda.min
lassolam
```

```
lasso.pred <- predict(lasso.mod, s=lassolam, newx=test.x)
mean((lasso.pred-test.y)^2)

x = model.matrix(Balance~Rating*Cards*Gender + Rating*Income*Cards +
Rating*I(Income^2)*Cards + Rating*Income*Gender+Rating*I(Income^2)*Gender
+ Income*Cards*Gender + I(Income^2)*Cards*Gender, bank)
y = bank$Balance

out.lasso <- glmnet(x,y,alpha=1)
lasso.coef <- predict(out.lasso, type="coefficients", s=lassolam)[1:21,]

lasso.coef[lasso.coef!=0]
```

6.0 Output

```
#M1
[1] 5.674292
[1] 203054.9

  (Intercept)  GenderFemale         Cards  I(Income^2) Rating:Income
 7.960421e+02  1.527172e+02 -8.290529e+01  3.998380e-04  1.965014e-04

#M2
[1] 5.674292
[1] 203188.3
          (Intercept)                Rating          GenderFemale                 Cards                Income
         7.091430e+02          4.167136e-02          1.446175e+02         -8.035565e+01          1.243958e-01
          I(Income^2)         Rating:Income   Rating:GenderFemale   Rating:I(Income^2) GenderFemale:I(Income^2)
         3.766932e-04          7.111948e-06          1.340007e-02          3.888126e-08         -8.148985e-05

#M3
[1] 4.710897
[1] 201781.3
```

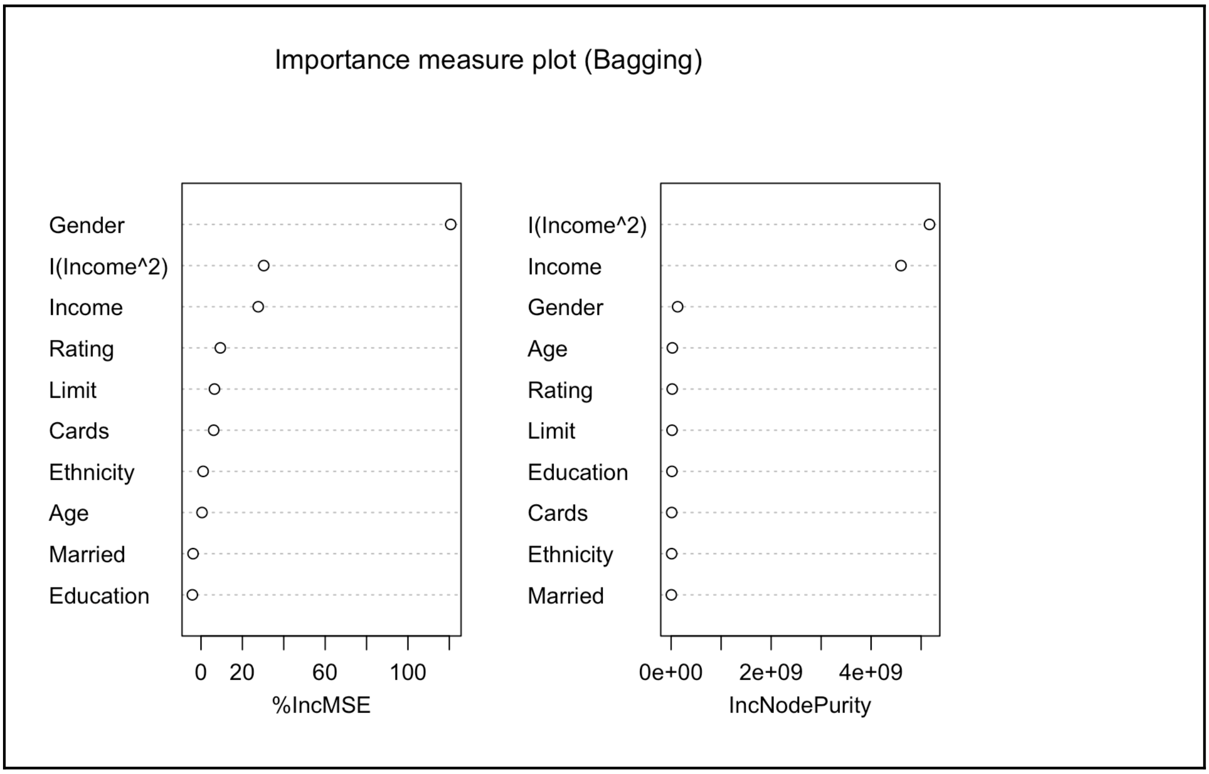
```
                 (Intercept)                        Cards                 GenderFemale                       Income
                7.339071e+02                -8.612623e+01                 1.028961e+02                 1.201578e-01
                 I(Income^2)           Rating:GenderFemale           Cards:GenderFemale                Rating:Income
                3.749728e-04                 1.749931e-01                 6.591505e+01                 1.014376e-05
          Rating:I(Income^2)      GenderFemale:I(Income^2)     Rating:Cards:GenderFemale      Rating:Cards:I(Income^2)
                4.319330e-08                -7.447831e-05                 3.387013e-03                 1.377175e-09
Rating:GenderFemale:I(Income^2)
               -2.260941e-08
```

7.1 Input

```
library(randomForest)
RNGkind(sample.kind = "Rounding")
set.seed(123)

bank <- read.csv("Bank2023P.csv", stringsAsFactors = TRUE)

train <- sample(1:nrow(bank), 0.8*nrow(bank))
test <- -train

# Bagging
bagging <- randomForest(Balance~.+I(Income^2), data=bank, subset=train,
mtry=10, importance=TRUE)
bagging
pred_bagging <- predict(bagging, newdata=bank[test,])
bagging_mse <- mean((pred_bagging-bank[test,]$Balance)^2)
bagging_mse
bagging_all <- randomForest(Balance~.+I(Income^2), data=bank, mtry=10,
importance=TRUE)
bagging_all

importance(bagging)
varImpPlot(bagging, main="Importance measure plot (Bagging)")
```

7.1 Output

```
> library(randomForest)
> RNGkind(sample.kind = "Rounding")
Warning message:
In RNGkind(sample.kind = "Rounding") : non-uniform 'Rounding' sampler used
> set.seed(123)
>
> bank <- read.csv("Bank2023P.csv", stringsAsFactors = TRUE)
>
> train <- sample(1:nrow(bank), 0.8*nrow(bank))
> test <- -train
>
> # Bagging
> bagging <- randomForest(Balance~.+I(Income^2), data=bank, subset=train, mtry=10, importance=TRUE)
> bagging

Call:
 randomForest(formula = Balance ~ . + I(Income^2), data = bank,      mtry = 10, importance = TRUE, subset = train)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 10

          Mean of squared residuals: 264885.8
                    % Var explained: 97.89
> pred_bagging <- predict(bagging, newdata=bank[test,])
> bagging_mse <- mean((pred_bagging-bank[test,]$Balance)^2)
> bagging_mse
[1] 249680.6
> bagging_all <- randomForest(Balance~.+I(Income^2), data=bank, mtry=10, importance=TRUE)
> bagging_all

Call:
 randomForest(formula = Balance ~ . + I(Income^2), data = bank,      mtry = 10, importance = TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 10

          Mean of squared residuals: 259147.7
                    % Var explained: 97.89
>


>
> importance(bagging)
                %IncMSE IncNodePurity
Income       27.6523772    4598172694
Limit         6.5007856      18061007
Rating        9.3436111      21268833
Cards         6.1277652      11657088
Age           0.4908458      24525987
Education    -4.1801827      16036587
Gender      120.6822548     130170638
Married      -3.8375857       4558463
Ethnicity     1.0435212      10766816
I(Income^2)  30.3362314    5167670497
> varImpPlot(bagging, main="Importance measure plot (Bagging)")
```



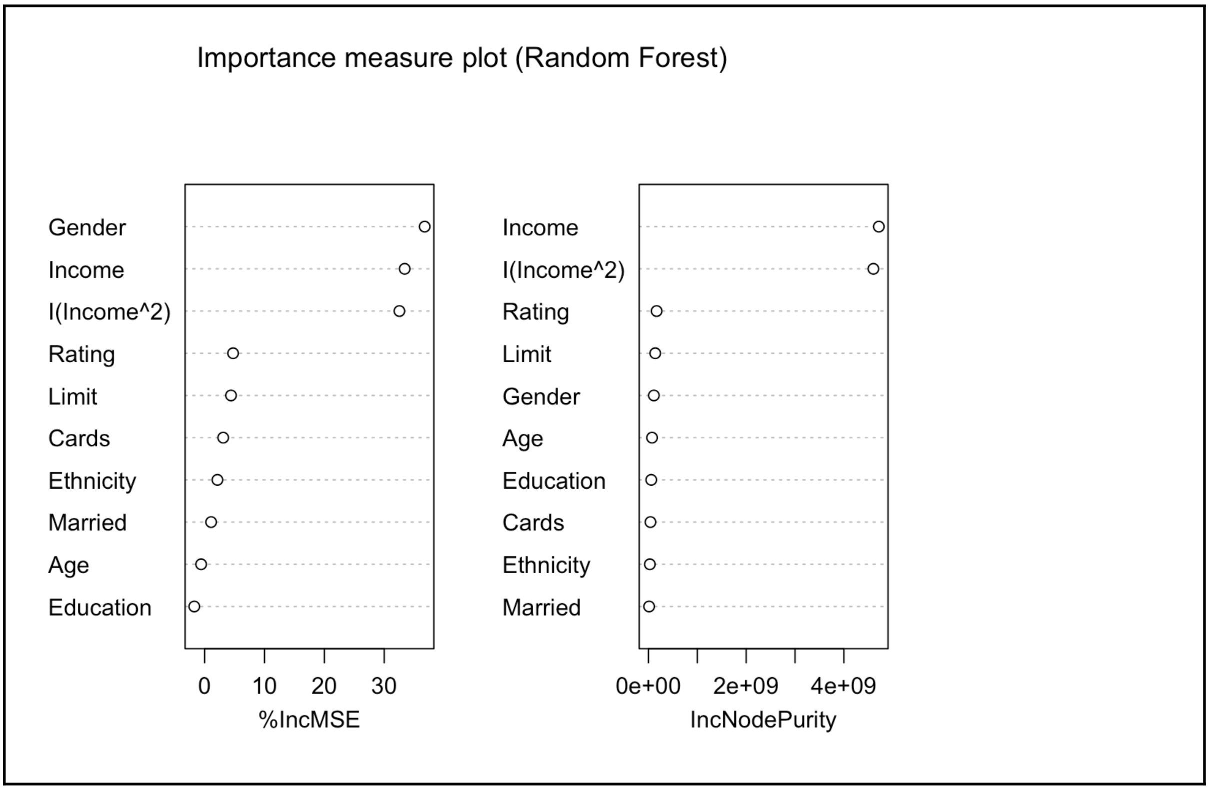
7.2 Input

```
library(randomForest)
RNGkind(sample.kind = "Rounding")
set.seed(123)

bank <- read.csv("Bank2023P.csv", stringsAsFactors = TRUE)

train <- sample(1:nrow(bank), 0.8*nrow(bank))
test <- -train

# Random Forest
rf <- randomForest(Balance~.+I(Income^2), data=bank, subset=train, mtry=3,
importance=TRUE)
rf
pred_rf <- predict(rf, newdata=bank[test,])
rf_mse <- mean((pred_rf-bank[test,]$Balance)^2)
rf_mse
rf_all <- randomForest(Balance~.+I(Income^2), data=bank, mtry=3,
```

```
importance=TRUE)
rf_all

importance(rf)
varImpPlot(rf, main="Importance measure plot (Random Forest)")
```

7.2 Output

```
> # Random Forest
> rf <- randomForest(Balance~.+I(Income^2), data=bank, subset=train, mtry=3, importance=TRUE)
> rf

Call:
 randomForest(formula = Balance ~ . + I(Income^2), data = bank,      mtry = 3, importance = TRUE, subset = train)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 3

          Mean of squared residuals: 286304.5
                    % Var explained: 97.72
> pred_rf <- predict(rf, newdata=bank[test,])
> rf_mse <- mean((pred_rf-bank[test,]$Balance)^2)
> rf_mse
[1] 272117.8
> rf_all <- randomForest(Balance~.+I(Income^2), data=bank, mtry=3, importance=TRUE)
> rf_all

Call:
 randomForest(formula = Balance ~ . + I(Income^2), data = bank,      mtry = 3, importance = TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 3

          Mean of squared residuals: 275506.2
                    % Var explained: 97.76
>
> importance(rf)
               %IncMSE IncNodePurity
Income      33.4225730    4715140420
Limit        4.4066193     139567009
Rating       4.7526310     167836685
Cards        3.1011995      41382316
Age         -0.5873331      73942280
Education   -1.7297789      57749146
Gender      36.7642140     111656398
Married      1.0814323      13738967
Ethnicity    2.1372068      29058033
I(Income^2) 32.5415273    4604042050
> varImpPlot(rf, main="Importance measure plot (Random Forest)")
```


Importance measure plot (Random Forest)
Gender
Income
I(Income^2)
Rating
Limit
Cards
Ethnicity
Married
Age
Education
0
10
20
30
%IncMSE
Income
I(Income^2)
Rating
Limit
Gender
Age
Education
Cards
Ethnicity
Married
0e+00
2e+09
4e+09
IncNodePurity