Stat 412

Yolanda Jin

24/11/2021

NA data (Income NA), Minority, Majority Groups

- Added column for NA_indicator
- split groups to minority vs majority group
- Q1: do we want NA to be a separate group? based on EDA, might or might not do so

```
# Read Credit Scoring Data Training Set
#cs_train <- cs_training
cs_train = read.csv("cs-training.csv")

# If we want to split NA to another group
#cs_train_NA <- cs_train[cs_train$MonthlyIncome==NA,]
#head(cs_train_NA)



## 0. NA Monthly Income

# Add col to indicate whether monthly Income is NA or not
cs_train$NA_Indicator <- 0  # Set all NA_Indicator to zero
cs_train$NA_Indicator[is.na(cs_train$MonthlyIncome)] <- 1  # Change NA income indexes to 1
head(cs_train[cs_train$NA_Indicator==1,])  # Check the ones indicated as NA
```

```
##     X SeriousDlqin2yrs RevolvingUtilizationOfUnsecuredLines age
## 7   7                0                            0.30568247  57
## 9   9                0                            0.11695064  27
## 17 17                0                            0.06108612  78
## 33 33                0                            0.08341801  62
## 42 42                0                            0.07289757  81
## 53 53                0                            0.99999990  62
##    NumberOfTime30.59DaysPastDueNotWorse DebtRatio MonthlyIncome
## 7                                     0      5710            NA
## 9                                     0        46            NA
## 17                                    0      2058            NA
## 33                                    0       977            NA
## 42                                    0        75            NA
## 53                                    0         0            NA
##    NumberOfOpenCreditLinesAndLoans NumberOfTimes90DaysLate
## 7                                8                       0
## 9                                2                       0
```

```
## 17                              10                        0
## 33                               6                        0
## 42                               7                        0
## 53                               1                        0
##    NumberRealEstateLoansOrLines NumberOfTime60.89DaysPastDueNotWorse
## 7                             3                                    0
## 9                             0                                    0
## 17                            2                                    0
## 33                            1                                    0
## 42                            0                                    0
## 53                            0                                    0
##    NumberOfDependents NA_Indicator
## 7                   0            1
## 9                  NA            1
## 17                  0            1
## 33                  0            1
## 42                  0            1
## 53                  0            1
```

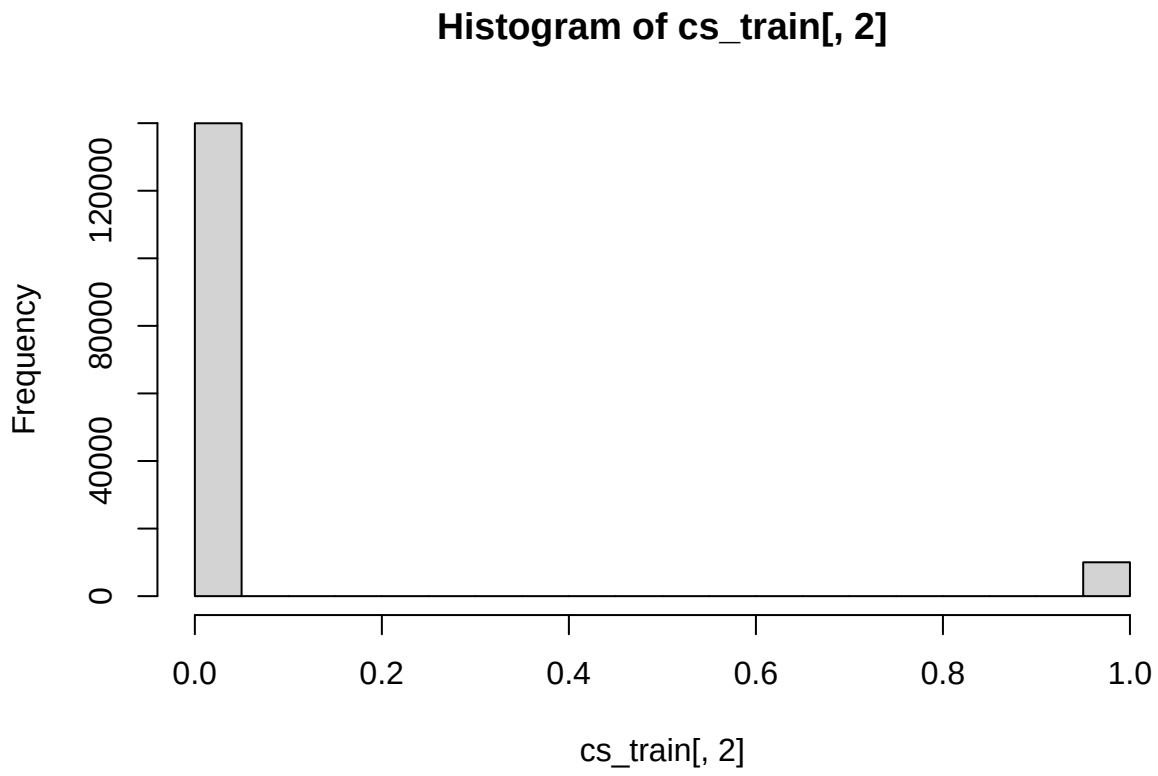
```
## 1. Separate minority data vs majority data vs NA data total 150k
cs_train_min <- cs_train[cs_train$SeriousDlqin2yrs==1,]  # 10,026 obs
cs_train_maj <- cs_train[cs_train$SeriousDlqin2yrs==0,]  # 139,974 obs
```

```
summary(cs_train)
```

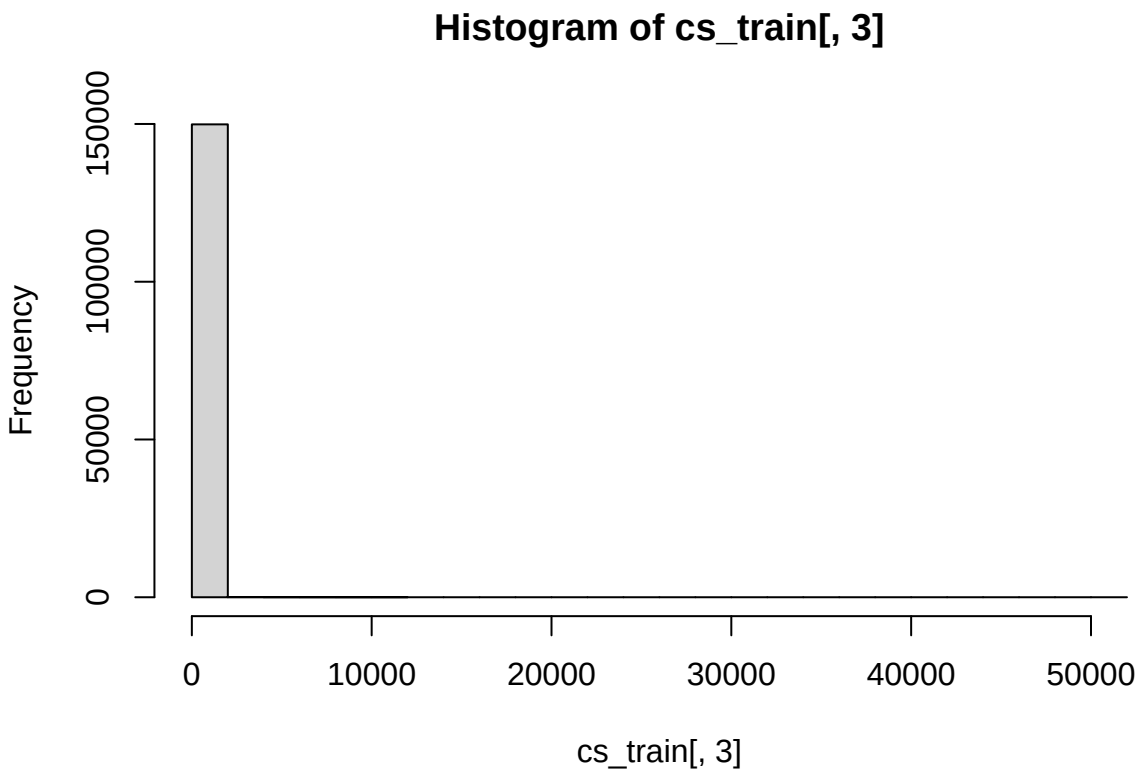
```
##        X          SeriousDlqin2yrs  RevolvingUtilizationOfUnsecuredLines
##  Min.   :     1   Min.   :0.00000   Min.   :    0.00
##  1st Qu.: 37501   1st Qu.:0.00000   1st Qu.:    0.03
##  Median : 75001   Median :0.00000   Median :    0.15
##  Mean   : 75001   Mean   :0.06684   Mean   :    6.05
##  3rd Qu.:112500   3rd Qu.:0.00000   3rd Qu.:    0.56
##  Max.   :150000   Max.   :1.00000   Max.   :50708.00
##
##       age        NumberOfTime30.59DaysPastDueNotWorse   DebtRatio
##  Min.   :  0.0   Min.   : 0.000                       Min.   :     0.0
##  1st Qu.: 41.0   1st Qu.: 0.000                       1st Qu.:     0.2
##  Median : 52.0   Median : 0.000                       Median :     0.4
##  Mean   : 52.3   Mean   : 0.421                       Mean   :   353.0
##  3rd Qu.: 63.0   3rd Qu.: 0.000                       3rd Qu.:     0.9
##  Max.   :109.0   Max.   :98.000                       Max.   :329664.0
##
##  MonthlyIncome     NumberOfOpenCreditLinesAndLoans NumberOfTimes90DaysLate
##  Min.   :      0   Min.   : 0.000                  Min.   : 0.000
##  1st Qu.:   3400   1st Qu.: 5.000                  1st Qu.: 0.000
##  Median :   5400   Median : 8.000                  Median : 0.000
##  Mean   :   6670   Mean   : 8.453                  Mean   : 0.266
##  3rd Qu.:   8249   3rd Qu.:11.000                  3rd Qu.: 0.000
##  Max.   :3008750   Max.   :58.000                  Max.   :98.000
##  NA's   :29731
##  NumberRealEstateLoansOrLines NumberOfTime60.89DaysPastDueNotWorse
##  Min.   : 0.000               Min.   : 0.0000
##  1st Qu.: 0.000               1st Qu.: 0.0000
##  Median : 1.000               Median : 0.0000
```

```
##  Mean   : 1.018              Mean   : 0.2404
##  3rd Qu.: 2.000              3rd Qu.: 0.0000
##  Max.   :54.000              Max.   :98.0000
##
##  NumberOfDependents  NA_Indicator
##  Min.   : 0.000     Min.   :0.0000
##  1st Qu.: 0.000     1st Qu.:0.0000
##  Median : 0.000     Median :0.0000
##  Mean   : 0.757     Mean   :0.1982
##  3rd Qu.: 1.000     3rd Qu.:0.0000
##  Max.   :20.000     Max.   :1.0000
##  NA's   :3924
```

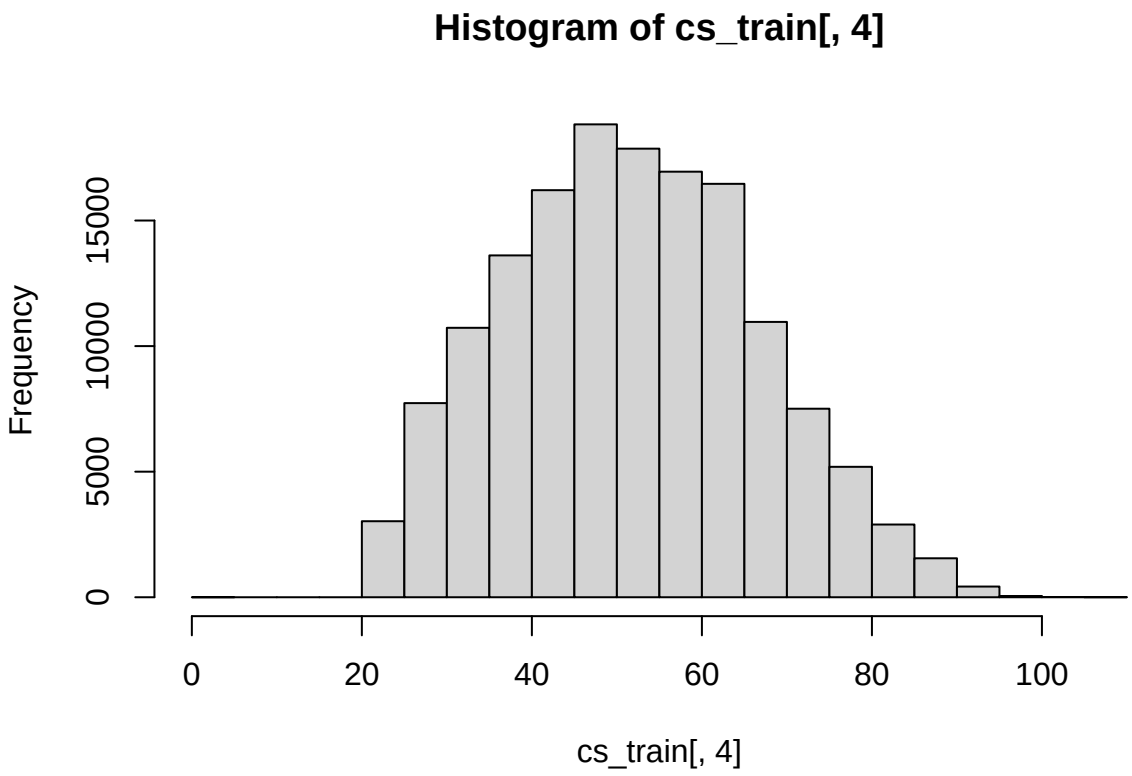
```
hist(cs_train[,2]) #Response Variable
```



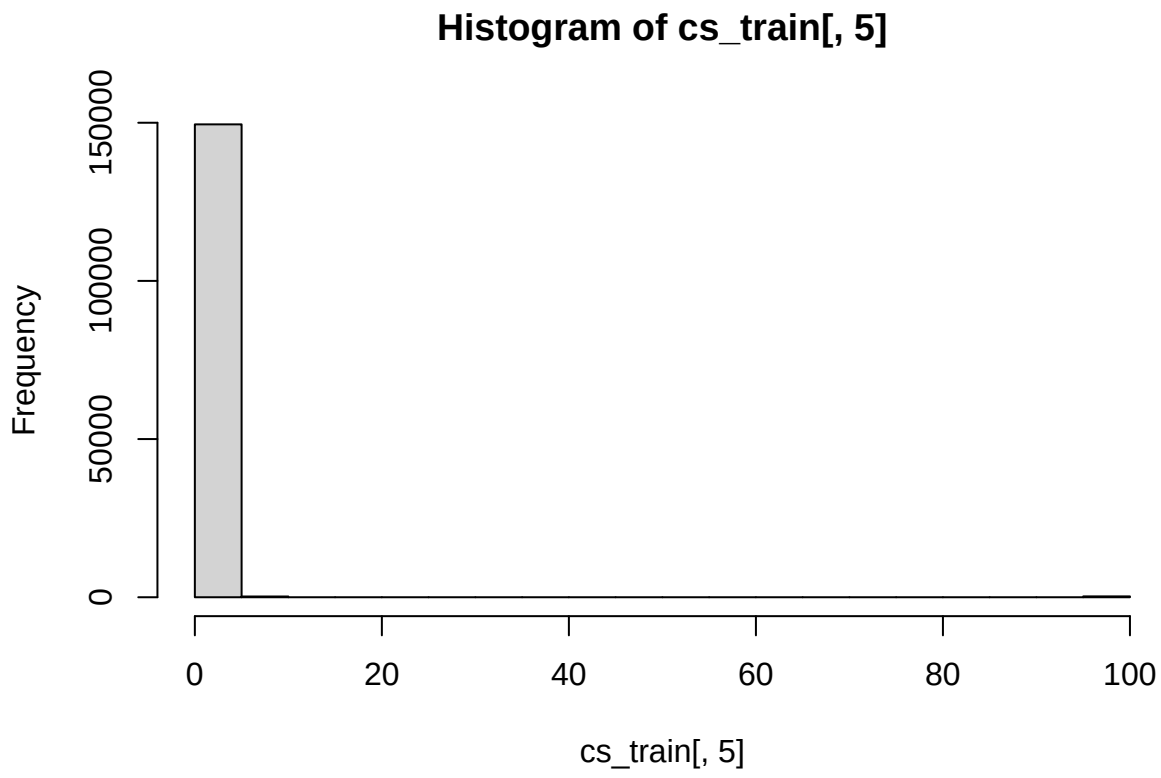

```
hist(cs_train[,3]) #RevolvingUtilizationOfUnsecuredLines
```



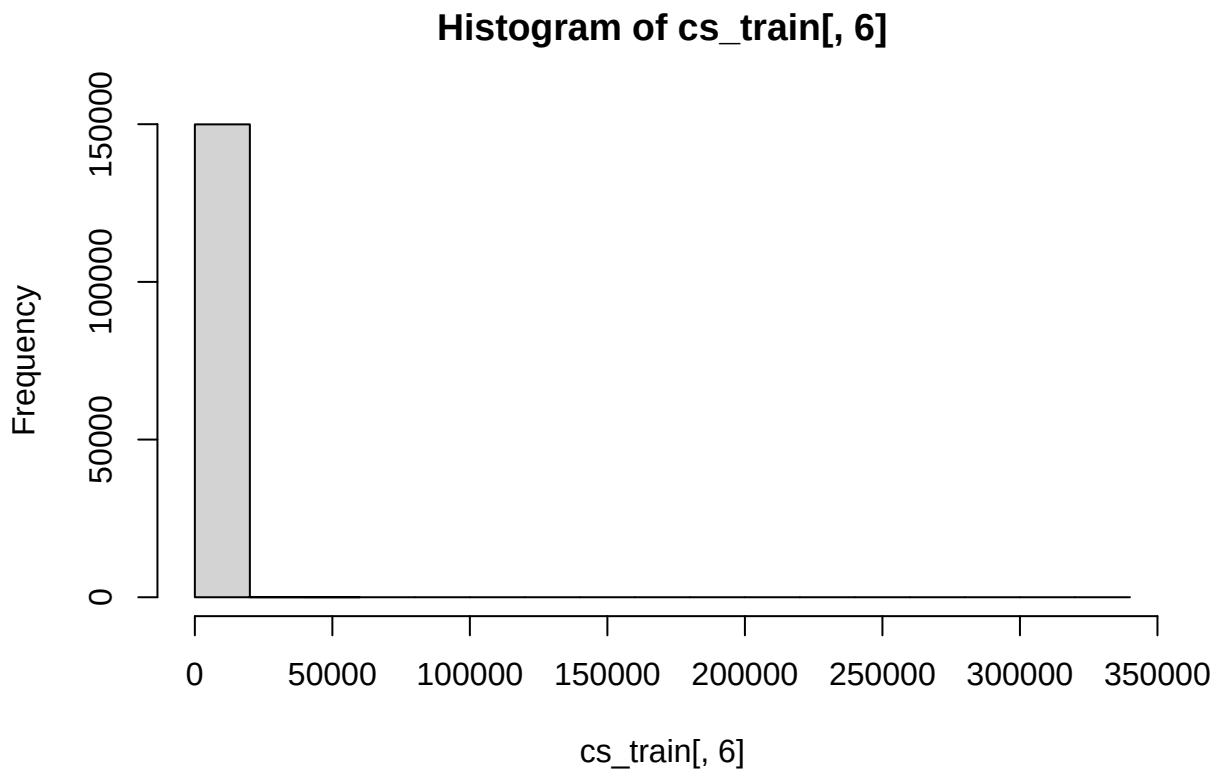

```
hist(cs_train[,4]) #age
```



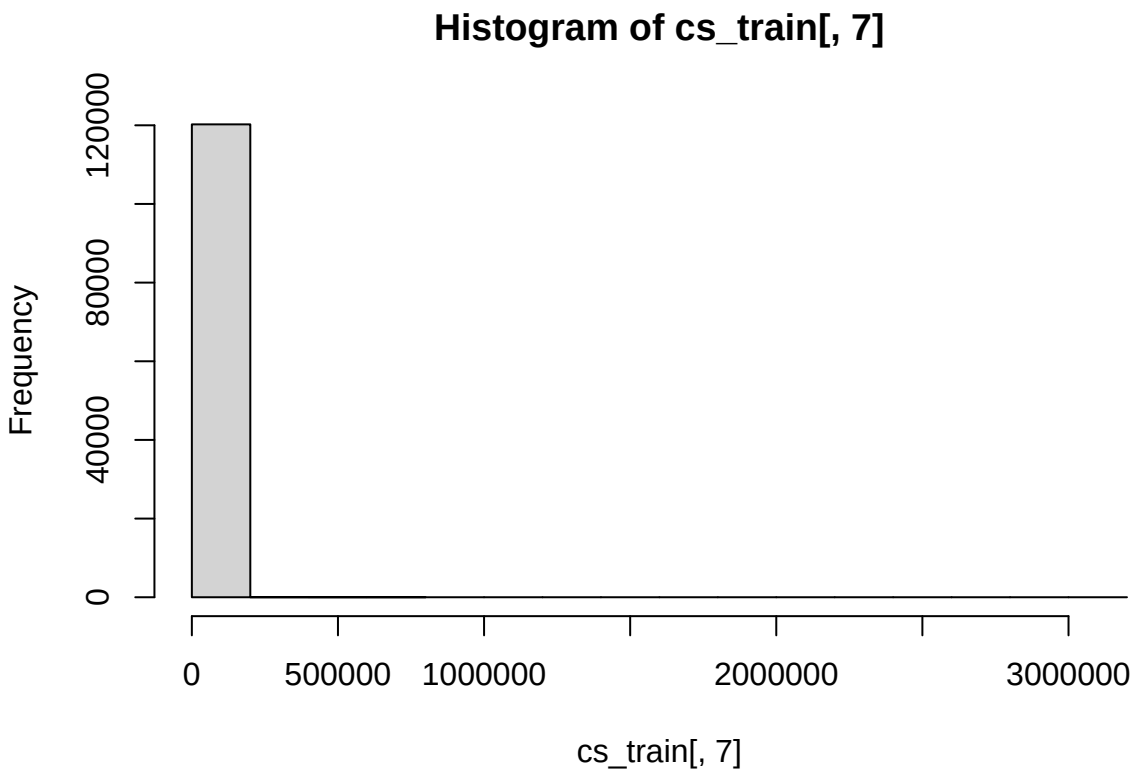

```
hist(cs_train[,5]) #NumberOfTime30.59DaysPastDueNotWorse
```



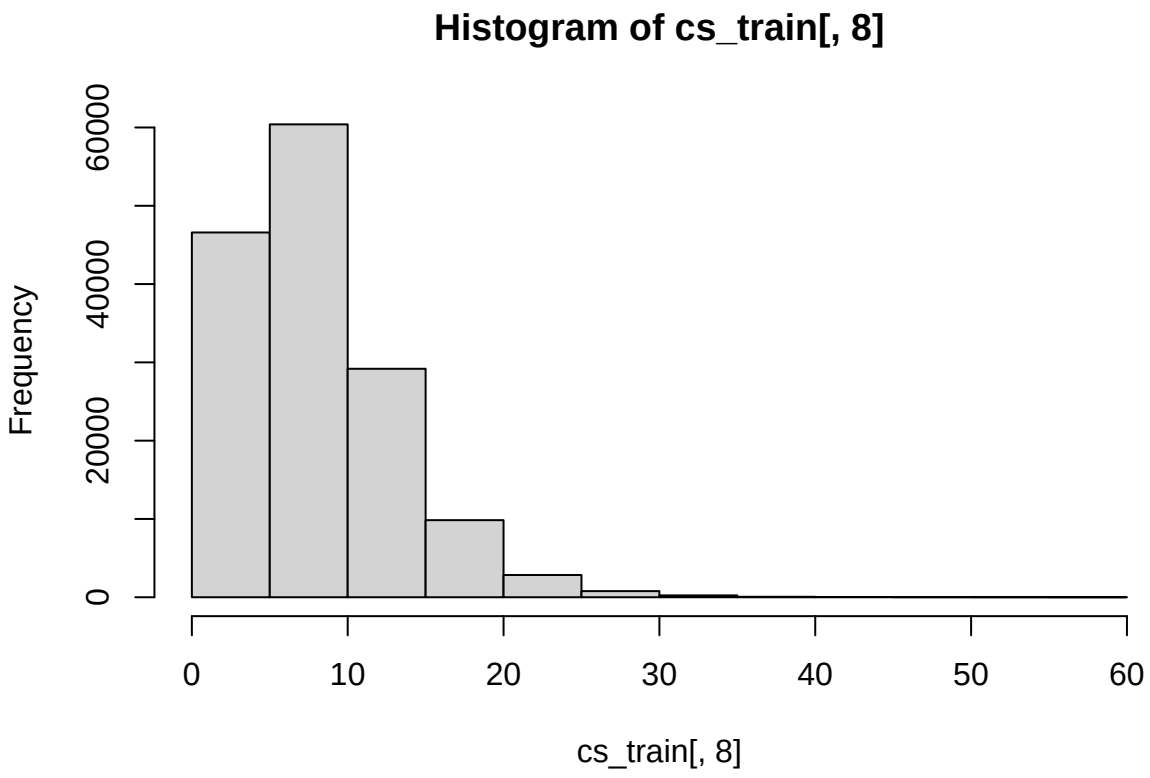

```
hist(cs_train[,6]) #DebtRatio
```



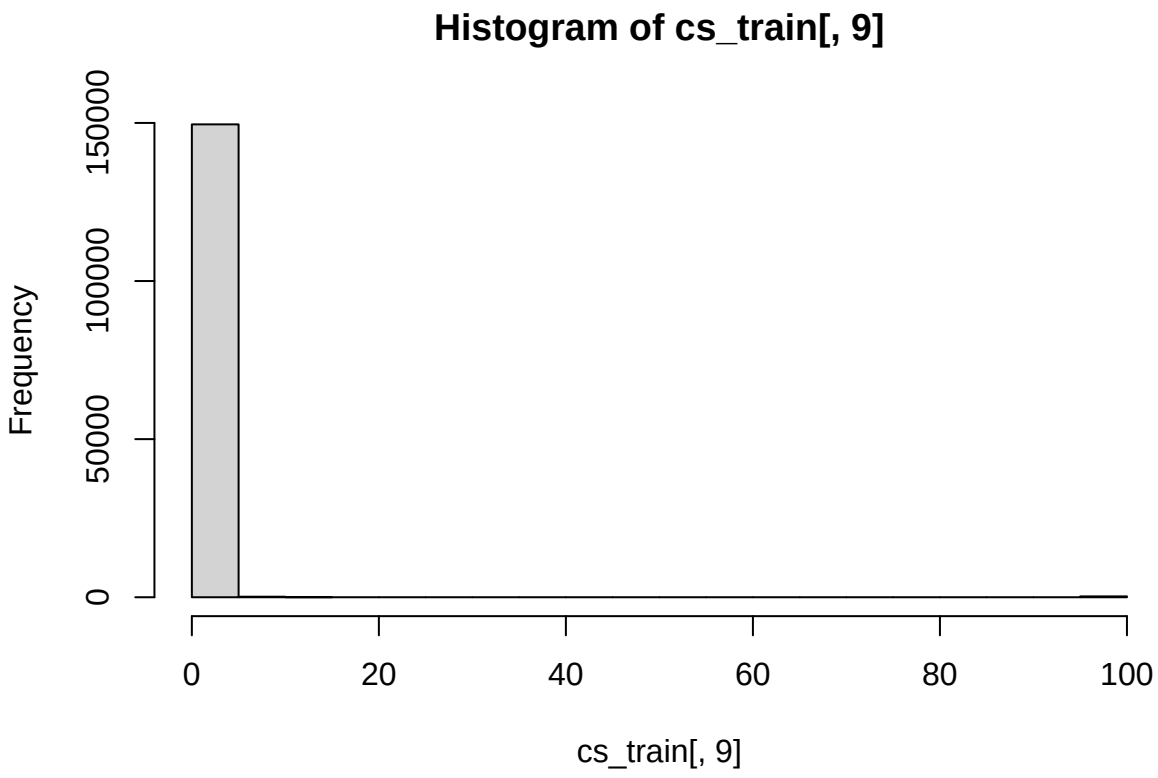

```
hist(cs_train[,7]) #MonthlyIncome
```



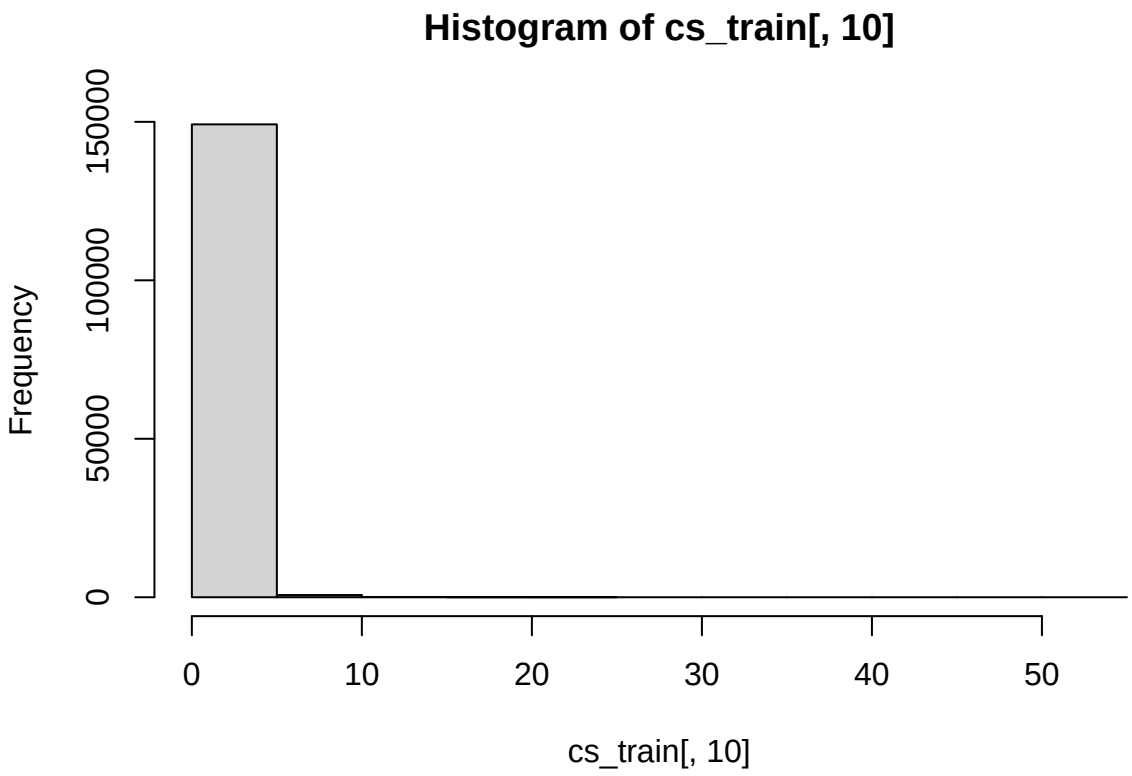

```
hist(cs_train[,8]) #NumberOfOpenCreditLinesAndLoans
```



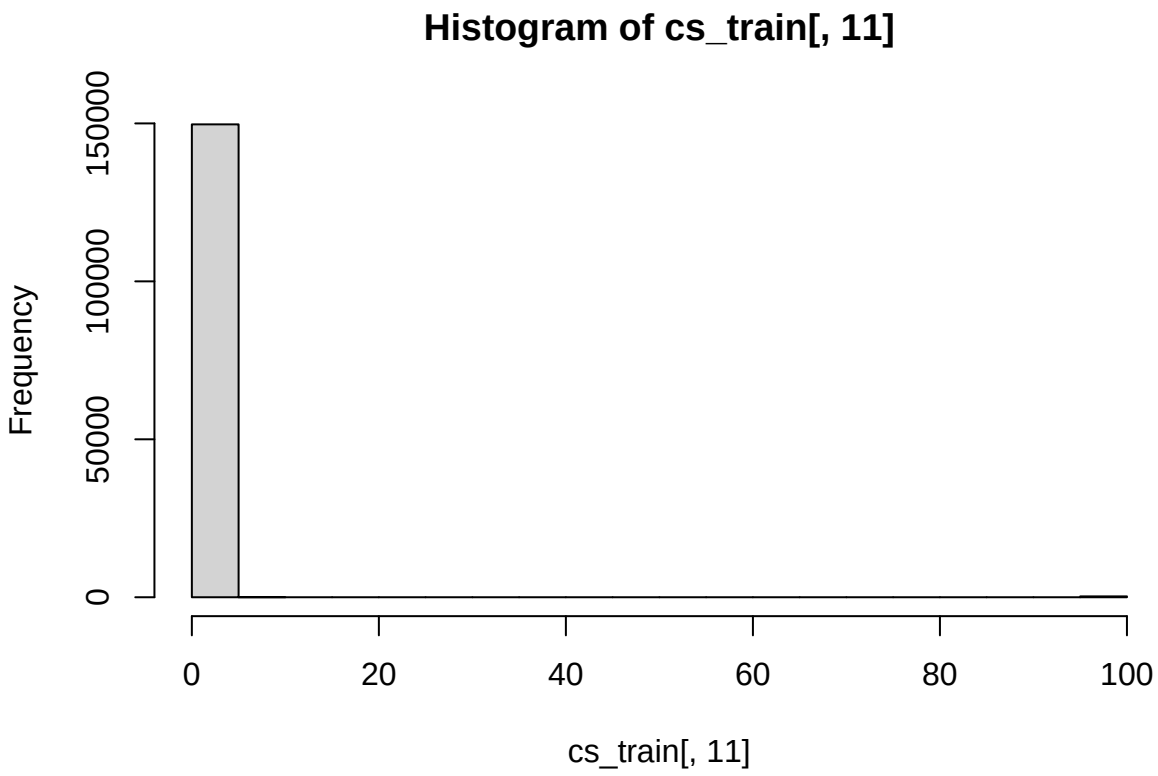

```
hist(cs_train[,9]) #NumberOfTimes90DaysLate
```



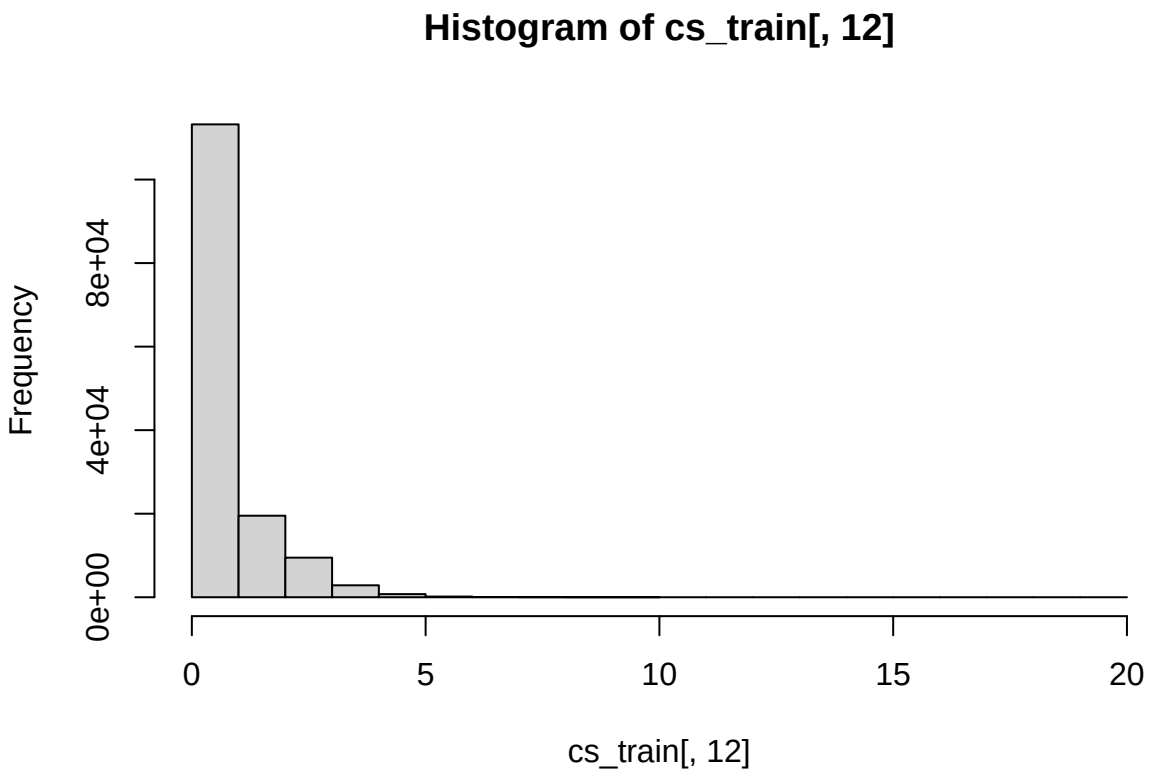

```
hist(cs_train[,10])#NumberRealEstateLoansOrLines
```



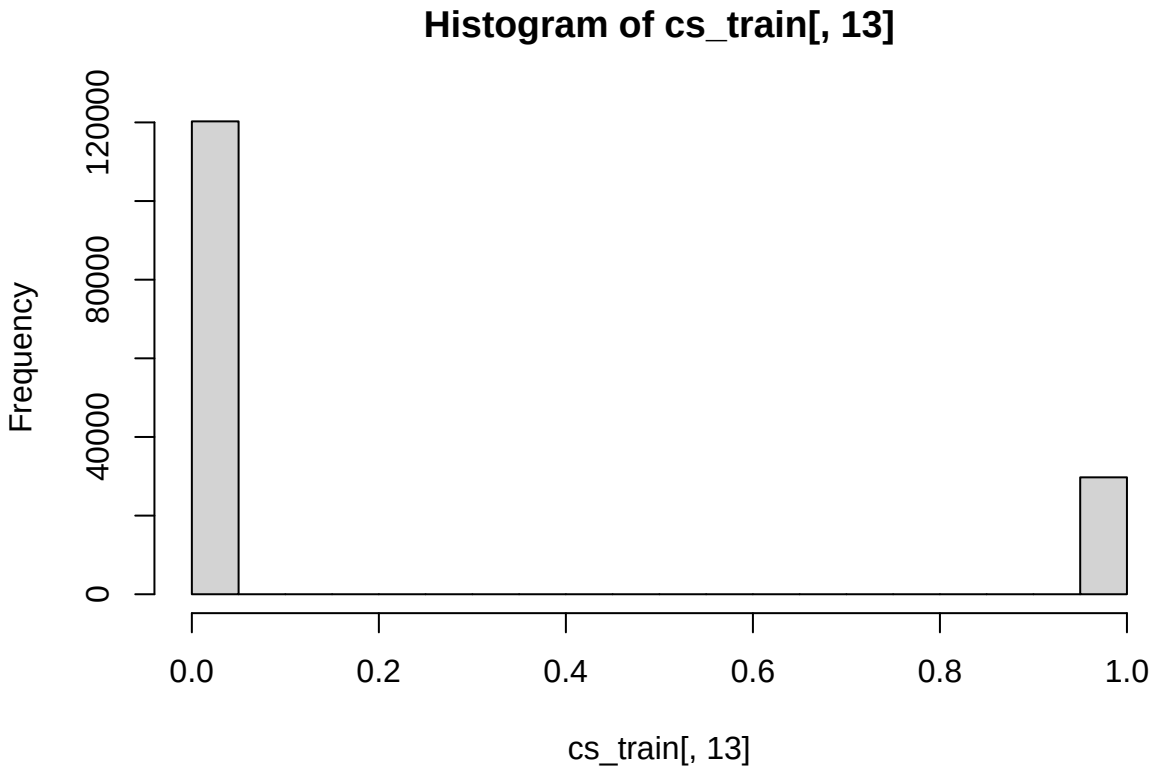

```
hist(cs_train[,11])#NumberOfTime60.89DaysPastDueNotWorse
```




```
hist(cs_train[,12])#NumberOfDependents
```




```
hist(cs_train[,13])#NA_Indicator
```




```
cs_train$HighRevolving <- 0
cs_train$HighRevolving[quantile(cs_train[,3],0.95)] <- 1

cs_train$Many_LessThan2MonthsLate <- 0
cs_train$Many_LessThan2MonthsLate[quantile(cs_train[,5],0.95)] <- 1

cs_train$HighDebtRatio <- 0
cs_train$HighDebtRatio[quantile(cs_train[,6],0.95)] <- 1

cs_train$Many_CurrentLoans <- 0
cs_train$Many_CurrentLoans[quantile(cs_train[,8],0.95)] <- 1

cs_train$Many_AtLeastThreeMonthsLate <- 0
cs_train$Many_AtLeastThreeMonthsLate[quantile(cs_train[,9],0.95)] <- 1

cs_train$Many_HouseLoans <- 0
cs_train$Many_HouseLoans[quantile(cs_train[,10],0.95)] <- 1

cs_train$Many_TwoToThreeMonthsLate <- 0
cs_train$Many_TwoToThreeMonthsLate[quantile(cs_train[,11],0.95)] <- 1

cs_train$NA_Dependents_are_Mean <- cs_train[,12] #Need to Change NA values to Mean before sorting throu
cs_train$NA_Dependents_are_Mean[is.na(cs_train[,12])] <- mean(cs_train[,12], na.rm = TRUE)
cs_train$Many_Dependents <- 0
cs_train$Many_Dependents[quantile(cs_train$NA_Dependents_are_Mean,0.95)] <- 1
```

```
train1 <-cs_train[cs_train[,3]< quantile(cs_train[,3],0.95),]
```

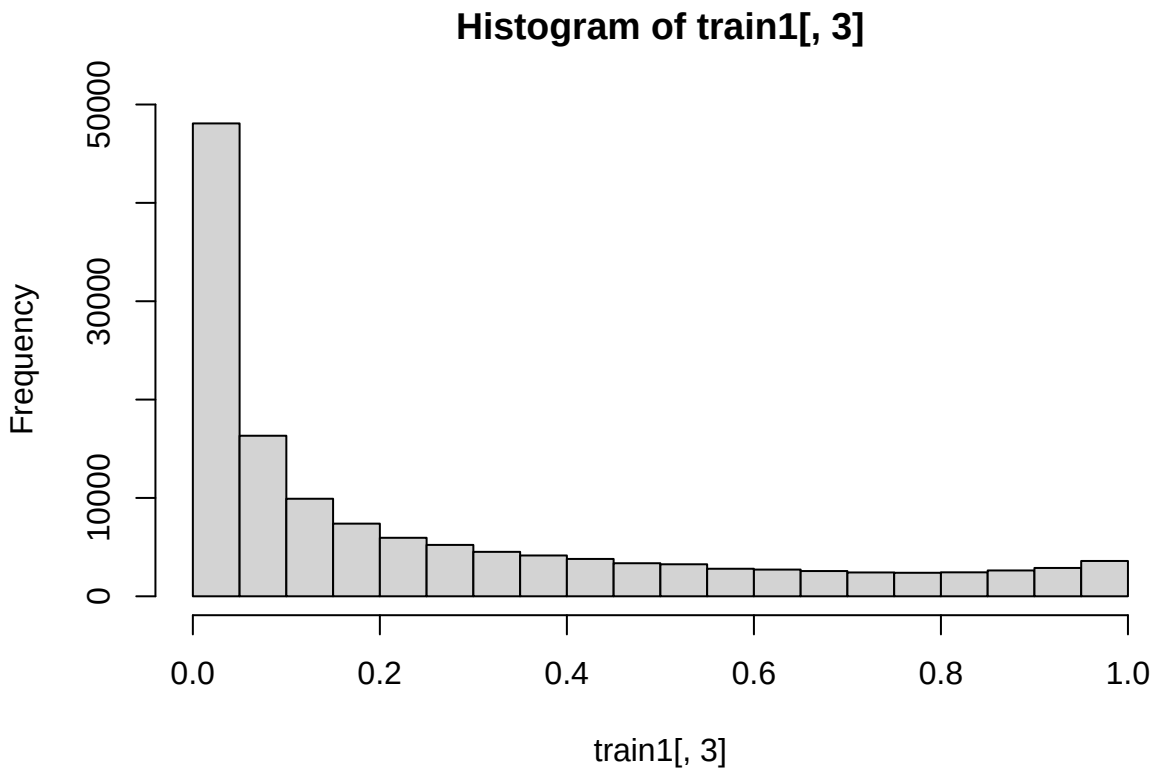
```
head(cs_train)
```

```
##   X SeriousDlqin2yrs RevolvingUtilizationOfUnsecuredLines age
## 1 1                1                            0.7661266  45
## 2 2                0                            0.9571510  40
## 3 3                0                            0.6581801  38
## 4 4                0                            0.2338098  30
## 5 5                0                            0.9072394  49
## 6 6                0                            0.2131787  74
##   NumberOfTime30.59DaysPastDueNotWorse  DebtRatio MonthlyIncome
## 1                                    2 0.80298213          9120
## 2                                    0 0.12187620          2600
## 3                                    1 0.08511338          3042
## 4                                    0 0.03604968          3300
## 5                                    1 0.02492570         63588
## 6                                    0 0.37560697          3500
##   NumberOfOpenCreditLinesAndLoans NumberOfTimes90DaysLate
## 1                              13                       0
## 2                               4                       0
## 3                               2                       1
## 4                               5                       0
## 5                               7                       0
## 6                               3                       0
##   NumberRealEstateLoansOrLines NumberOfTime60.89DaysPastDueNotWorse
## 1                            6                                    0
## 2                            0                                    0
## 3                            0                                    0
## 4                            0                                    0
## 5                            1                                    0
## 6                            1                                    0
##   NumberOfDependents NA_Indicator HighRevolving Many_LessThan2MonthsLate
## 1                  2            0             0                       0
## 2                  1            0             0                       1
## 3                  0            0             0                       0
## 4                  0            0             0                       0
## 5                  0            0             0                       0
## 6                  1            0             0                       0
##   HighDebtRatio Many_CurrentLoans Many_AtLeastThreeMonthsLate Many_HouseLoans
## 1             0                 0                           1               0
## 2             0                 0                           0               0
## 3             0                 0                           0               1
## 4             0                 0                           0               0
## 5             0                 0                           0               0
## 6             0                 0                           0               0
##   Many_TwoToThreeMonthsLate NA_Dependents_are_Mean Many_Dependents
## 1                         1                      2               0
## 2                         0                      1               0
## 3                         0                      0               1
## 4                         0                      0               0
## 5                         0                      0               0
## 6                         0                      1               0
```

Split the data to Testing/Training

```
# Remove NA cases otherwise cannot predict
raw_data <- cs_train
cs_train_omit <- na.omit(raw_data)   # 150k -> 120,269 obs

# Sample 60% of data for training Purpose
n = nrow(cs_train_omit)
na.idx = raw_data$X[-cs_train_omit$X]  # indexes of data with NA values that we removed
n.idx = sample(n, n*0.6) # Indexes for test train split

cs_train = cs_train_omit[n.idx,]  # Training data 72,161 obs
cs_test = cs_train_omit[-n.idx,]  # Testing data  48,108 obs.
cs_NA = raw_data[na.idx,]  # data with NA value  29,731 obs
```

```
train2 <- train1[train1[,5]< quantile(train1[,5],0.99),]
hist(train1[,3]) #Do we make transformation for predictors? Not Normal Distribution
```

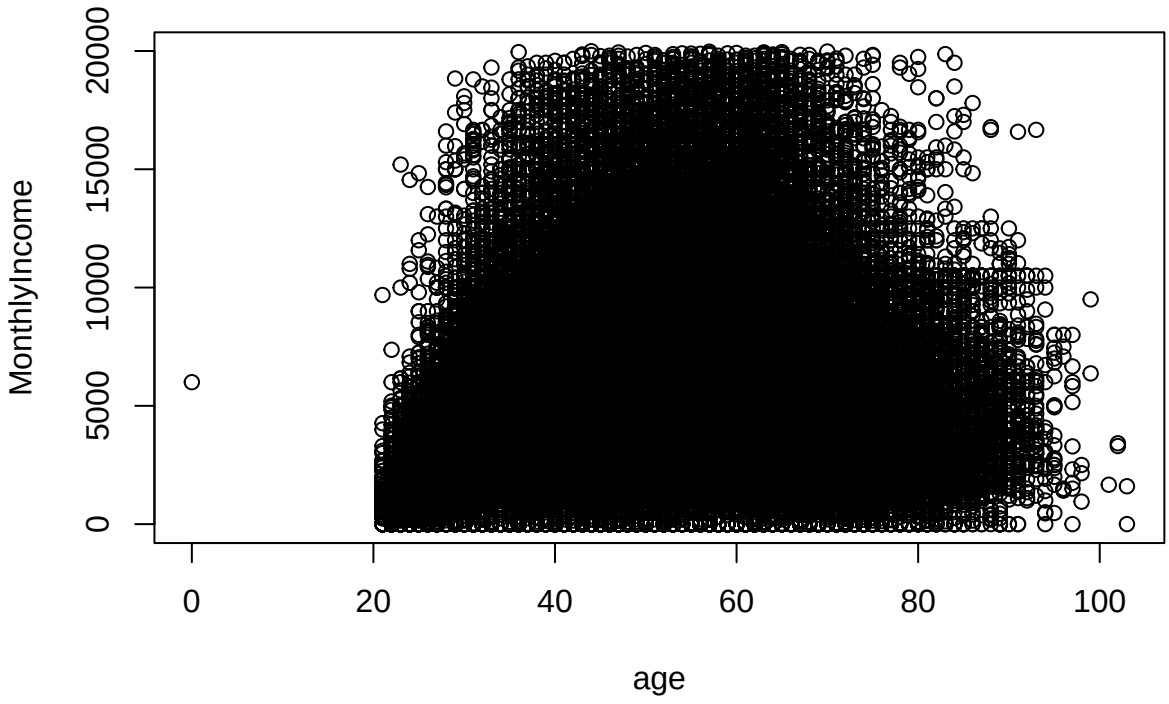


EDA

- monthly income = 0
- 30k monthly income = NA

- Dependent = NA 4k
- age 0 remove - only 1 point
- age very old
- group by age group etc
- 13 - 101 or older (maybe cut at 100)
- 80 or more

```
#plot(SeriousDlqin2yrs ~ age, data = cs_train_maj)
cs_train_maj_g1 <- cs_train_maj[cs_train_maj$MonthlyIncome<500000,] # less than 500k monthly income
cs_train_maj_g1 <- cs_train_maj[cs_train_maj$MonthlyIncome<20000,] # less than 20k monthly income
plot(MonthlyIncome ~ age, data = cs_train_maj_g1)
```



Question/Goal

- Comparing our model performance against paper's model
- Most important factors

Ensemble Learning

- Lasso Ensemble Algorithm
- Aggregating base learner: Weighted base=learner

Balancing Data

- Use clustering and pick one sub-group from majority
  - can try Age/Income
  - try Income/debt ratio
- Use bagging algorithm to create more minority data
- Use NA indicator 0 or 1 (maybe use mean/median)

```
# Cluster Grouping Majority Data (Try to cluster data by age/monthly income)
set.seed(1) # for reproducibility

head(cs_train_maj)
```

```
##   X SeriousDlqin2yrs RevolvingUtilizationOfUnsecuredLines age
## 2 2                0                              0.9571510  40
## 3 3                0                              0.6581801  38
## 4 4                0                              0.2338098  30
## 5 5                0                              0.9072394  49
## 6 6                0                              0.2131787  74
## 7 7                0                              0.3056825  57
##   NumberOfTime30.59DaysPastDueNotWorse    DebtRatio MonthlyIncome
## 2                                    0 1.218762e-01          2600
## 3                                    1 8.511338e-02          3042
## 4                                    0 3.604968e-02          3300
## 5                                    1 2.492570e-02         63588
## 6                                    0 3.756070e-01          3500
## 7                                    0 5.710000e+03            NA
##   NumberOfOpenCreditLinesAndLoans NumberOfTimes90DaysLlate
## 2                               4                        0
## 3                               2                        1
## 4                               5                        0
## 5                               7                        0
## 6                               3                        0
## 7                               8                        0
##   NumberRealEstateLoansOrLines NumberOfTime60.89DaysPastDueNotWorse
## 2                            0                                    0
## 3                            0                                    0
## 4                            0                                    0
## 5                            1                                    0
## 6                            1                                    0
## 7                            3                                    0
##   NumberOfDependents NA_Indicator
## 2                  1            0
## 3                  0            0
## 4                  0            0
## 5                  0            0
## 6                  1            0
## 7                  0            1
```

```r
# Test with smaller group based on monthly income range
cs_train_maj_g1 <- cs_train_maj[cs_train_maj$NA_Indicator==0,] # Filter out NA monthly income first
cs_train_maj_g1 <- cs_train_maj_g1[cs_train_maj_g1$MonthlyIncome<20000,]  #38035 obs
head(cs_train_maj_g1)
```

```
##     X SeriousDlqin2yrs RevolvingUtilizationOfUnsecuredLines age
## 2   2                0                            0.9571510  40
## 3   3                0                            0.6581801  38
## 4   4                0                            0.2338098  30
## 6   6                0                            0.2131787  74
## 8   8                0                            0.7544636  39
## 11 11                0                            0.6442260  30
##    NumberOfTime30.59DaysPastDueNotWorse  DebtRatio MonthlyIncome
## 2                                     0 0.12187620          2600
## 3                                     1 0.08511338          3042
## 4                                     0 0.03604968          3300
## 6                                     0 0.37560697          3500
## 8                                     0 0.20994002          3500
## 11                                    0 0.30947621          2500
##    NumberOfOpenCreditLinesAndLoans NumberOfTimes90DaysLate
## 2                                4                       0
## 3                                2                       1
## 4                                5                       0
## 6                                3                       0
## 8                                8                       0
## 11                               5                       0
##    NumberRealEstateLoansOrLines NumberOfTime60.89DaysPastDueNotWorse
## 2                             0                                     0
## 3                             0                                     0
## 4                             0                                     0
## 6                             1                                     0
## 8                             0                                     0
## 11                            0                                     0
##    NumberOfDependents NA_Indicator
## 2                   1            0
## 3                   0            0
## 4                   0            0
## 6                   1            0
## 8                   0            0
## 11                  0            0
```

```r
# Create a dat with the two predictors of interest
dat <- cs_train_maj_g1[,c(4,7)]  # Age and MonthlyIncome
head(dat)
```

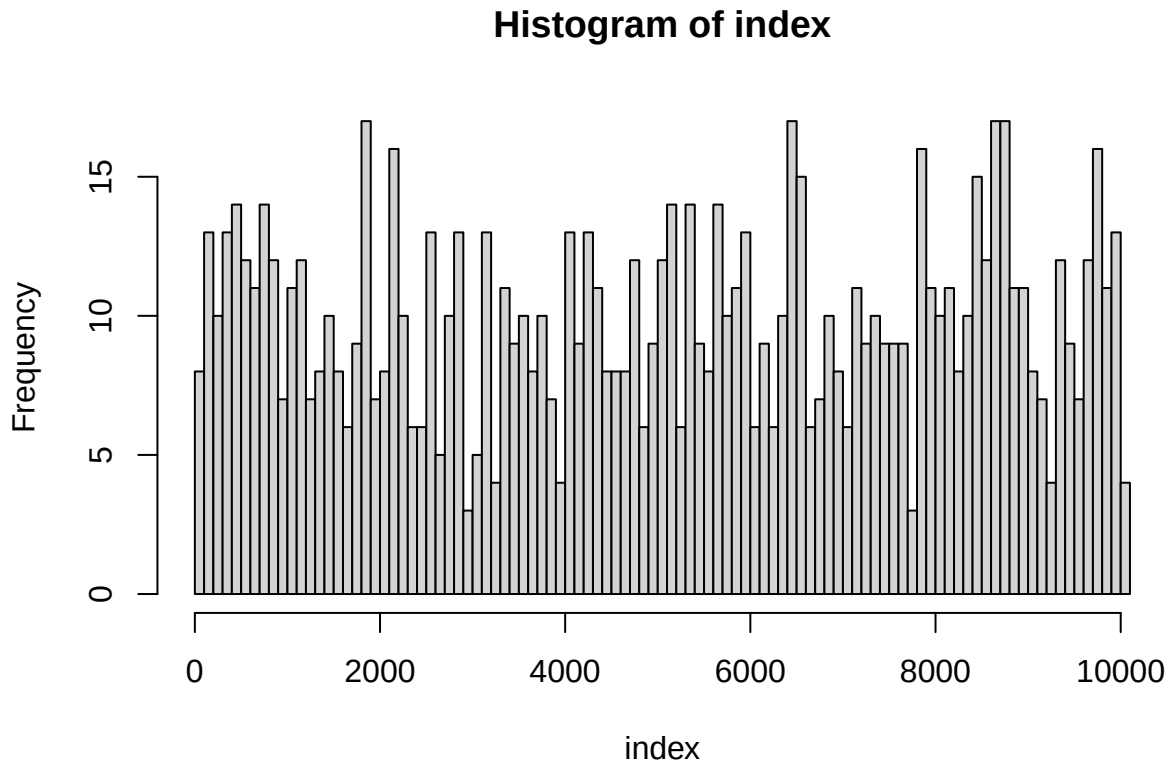
```
##    age MonthlyIncome
## 2   40          2600
## 3   38          3042
## 4   30          3300
## 6   74          3500
## 8   39          3500
## 11  30          2500
```

```
n_maj <- nrow(dat) # get number of rows

# Initial assignments to three groups that will need to update
assignments <- factor(sample(c(1,2,3), n_maj, replace = TRUE))
#plot(dat, col=assignments, xlim = c(0,110), asp=1)
#plot(dat, col=assignments)
```

```
# Boostrapping Minority Data
set.seed(1) # for reproducibility
# set number of minority data to reproduce
n_add <- 1000

n_min <- nrow(cs_train_min)
n_min
```

```
## [1] 10026
```

```
index <- sample(n_min, n_add, replace = TRUE)
#plot(density(index),  main="")  # show density curve of the index we randomized
hist(index, breaks = 100)
```




```
min(index)
```

```
## [1] 27
```

```
max(index)
```

```
## [1] 10014
```

```
length(index)
```

```
## [1] 1000
```

```
# We add the additional data for future analysis
cs_train_min_add <- cs_train_min[index,]
head(cs_train_min_add)
```

```
##             X SeriousDlqin2yrs RevolvingUtilizationOfUnsecuredLines age
## 15807   15807                1                            0.99999990  46
## 120332 120332                1                            0.04713019  45
## 71375   71375                1                            0.71069704  52
## 145782 145782                1                            0.99999990  52
## 127189 127189                1                            1.25614754  56
## 60636   60636                1                            0.41318696  35
##        NumberOfTime30.59DaysPastDueNotWorse    DebtRatio MonthlyIncome
## 15807                                     0    0.6327286          2700
## 120332                                    0    0.5847736          7000
## 71375                                     0    1.1807549          3761
## 145782                                    0    0.0000000          3200
## 127189                                    3    0.1938939          5600
## 60636                                     1 1203.0000000            NA
##        NumberOfOpenCreditLinesAndLoans NumberOfTimes90DaysLate
## 15807                                3                       1
## 120332                               8                       0
## 71375                               11                       0
## 145782                               0                       0
## 127189                               7                       2
## 60636                               10                       0
##        NumberRealEstateLoansOrLines NumberOfTime60.89DaysPastDueNotWorse
## 15807                             1                                    1
## 120332                            2                                    0
## 71375                             2                                    0
## 145782                            0                                    0
## 127189                            0                                    1
## 60636                             0                                    0
##        NumberOfDependents NA_Indicator
## 15807                   0            0
## 120332                  0            0
## 71375                   2            0
## 145782                  2            0
## 127189                  0            0
## 60636                   0            1
```

Which models to try

- Logistic regression (compare the different link functions)

- look maybe merge Lasso with logistic regression
- RF

Evaluating Model/Comparing results

- AUC

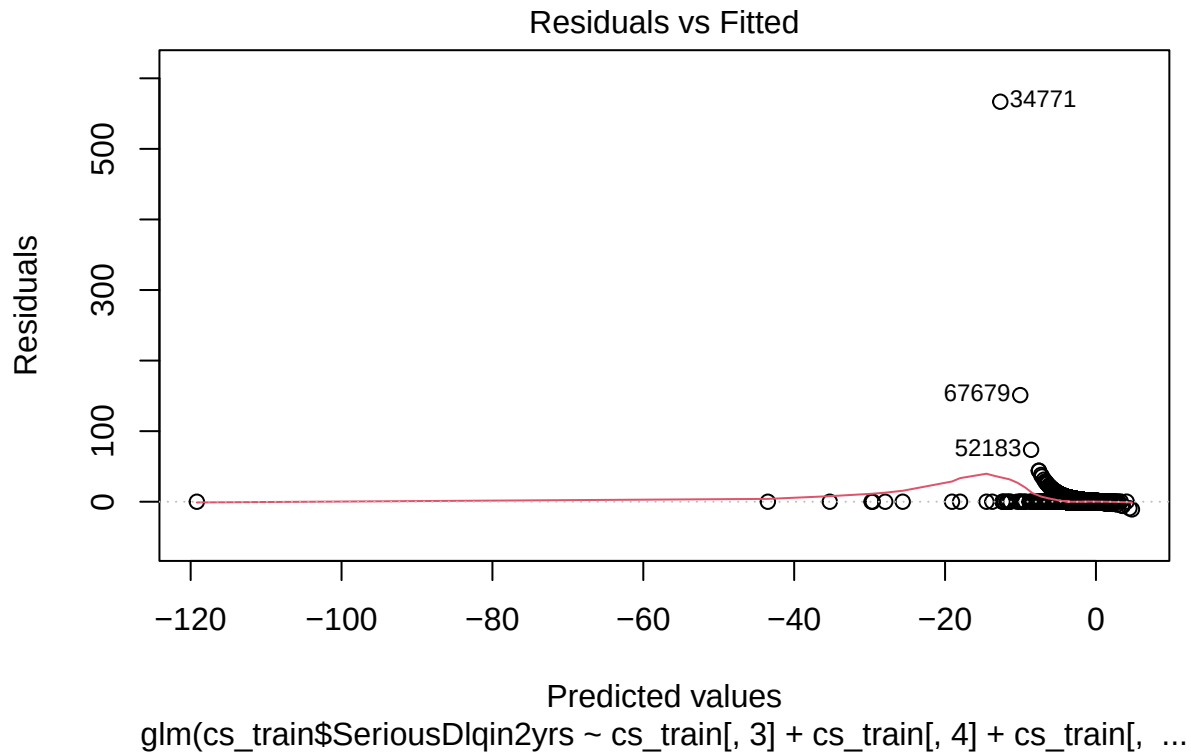
```
model <- glm(cs_train$SeriousDlqin2yrs ~ cs_train[,3] +cs_train[,4] +cs_train[,5] +cs_train[,6] +cs_tra
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
#To Change family link use family = quasi(variance = "mu^3", link = "log") change quasi
summary(model)
```

```
##
## Call:
## glm(formula = cs_train$SeriousDlqin2yrs ~ cs_train[, 3] + cs_train[,
##     4] + cs_train[, 5] + cs_train[, 6] + cs_train[, 7] + cs_train[,
##     8] + cs_train[, 9] + cs_train[, 10] + cs_train[, 11] + cs_train[,
##     12], family = "binomial", data = cs_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0921  -0.3978  -0.3272  -0.2669   5.0359
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -1.499e+00  5.980e-02 -25.058  < 2e-16 ***
## cs_train[, 3]  -1.400e-04  1.444e-04  -0.970   0.3321
## cs_train[, 4]  -2.424e-02  1.193e-03 -20.324  < 2e-16 ***
## cs_train[, 5]   5.204e-01  1.551e-02  33.560  < 2e-16 ***
## cs_train[, 6]  -1.386e-04  6.097e-05  -2.273   0.0230 *
## cs_train[, 7]  -3.880e-05  4.199e-06  -9.241  < 2e-16 ***
## cs_train[, 8]  -6.727e-03  3.532e-03  -1.905   0.0568 .
## cs_train[, 9]   4.249e-01  2.177e-02  19.517  < 2e-16 ***
## cs_train[, 10]  6.102e-02  1.425e-02   4.281 1.86e-05 ***
## cs_train[, 11] -9.086e-01  2.525e-02 -35.984  < 2e-16 ***
## cs_train[, 12]  1.054e-01  1.237e-02   8.518  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 36511  on 72160  degrees of freedom
## Residual deviance: 33703  on 72150  degrees of freedom
## AIC: 33725
##
## Number of Fisher Scoring iterations: 6
```

```
plot(model, which = 1) #Outliers skew residuals plot
```




```
stepAIC(model)
```

```
## Start:  AIC=33725.41
## cs_train$SeriousDlqin2yrs ~ cs_train[, 3] + cs_train[, 4] + cs_train[,
##     5] + cs_train[, 6] + cs_train[, 7] + cs_train[, 8] + cs_train[,
##     9] + cs_train[, 10] + cs_train[, 11] + cs_train[, 12]

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                   Df Deviance   AIC
## - cs_train[, 3]    1    33705 33725
## <none>                  33703 33725
## - cs_train[, 8]    1    33707 33727
## - cs_train[, 6]    1    33710 33730
## - cs_train[, 10]   1    33721 33741
## - cs_train[, 12]   1    33773 33793
## - cs_train[, 7]    1    33810 33830
## - cs_train[, 9]    1    34094 34114
## - cs_train[, 4]    1    34138 34158
## - cs_train[, 5]    1    34705 34725
## - cs_train[, 11]   1    34967 34987

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=33724.9
## cs_train$SeriousDlqin2yrs ~ cs_train[, 4] + cs_train[, 5] + cs_train[,
##     6] + cs_train[, 7] + cs_train[, 8] + cs_train[, 9] + cs_train[,
##     10] + cs_train[, 11] + cs_train[, 12]

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                   Df Deviance   AIC
## <none>                  33705 33725
## - cs_train[, 8]    1    33708 33726
## - cs_train[, 6]    1    33712 33730
## - cs_train[, 10]   1    33722 33740
## - cs_train[, 12]   1    33775 33793
## - cs_train[, 7]    1    33812 33830
## - cs_train[, 9]    1    34096 34114
## - cs_train[, 4]    1    34139 34157
## - cs_train[, 5]    1    34707 34725
## - cs_train[, 11]   1    34969 34987

##
```

```
## Call:  glm(formula = cs_train$SeriousDlqin2yrs ~ cs_train[, 4] + cs_train[,
##     5] + cs_train[, 6] + cs_train[, 7] + cs_train[, 8] + cs_train[,
##     9] + cs_train[, 10] + cs_train[, 11] + cs_train[, 12], family = "binomial",
##     data = cs_train)
##
## Coefficients:
##    (Intercept)   cs_train[, 4]   cs_train[, 5]   cs_train[, 6]   cs_train[, 7]
##     -1.499e+00      -2.424e-02       5.204e-01      -1.389e-04      -3.888e-05
##  cs_train[, 8]   cs_train[, 9]  cs_train[, 10]  cs_train[, 11]  cs_train[, 12]
##     -6.659e-03       4.250e-01       6.090e-02      -9.087e-01       1.054e-01
##
## Degrees of Freedom: 72160 Total (i.e. Null);  72151 Residual
## Null Deviance:       36510
## Residual Deviance: 33700     AIC: 33720
```

```
model2 <- glm(cs_train$SeriousDlqin2yrs ~ MonthlyIncome + cs_train[,13] + cs_train[,14] + cs_train[,15]
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
model2
```

```
##
## Call:  glm(formula = cs_train$SeriousDlqin2yrs ~ MonthlyIncome + cs_train[,
##     13] + cs_train[, 14] + cs_train[, 15] + cs_train[, 16] +
##     cs_train[, 17] + cs_train[, 18] + cs_train[, 19] + cs_train[,
##     20] + cs_train[, 22], family = "binomial", data = cs_train)
##
## Coefficients:
##    (Intercept)   MonthlyIncome  cs_train[, 13]  cs_train[, 14]  cs_train[, 15]
##     -2.321e+00      -4.457e-05              NA              NA      -8.130e+00
## cs_train[, 16]  cs_train[, 17]  cs_train[, 18]  cs_train[, 19]  cs_train[, 20]
##     -8.067e+00              NA              NA      -8.110e+00              NA
## cs_train[, 22]
##             NA
##
## Degrees of Freedom: 72160 Total (i.e. Null);  72156 Residual
## Null Deviance:       36510
## Residual Deviance: 36340     AIC: 36350
```

```
model2$rank # equals 7 which is how many variables are not NA
```

```
## [1] 5
```

```
#Certain Dummy Variables are a  Singular Matrix Meaning that some of our variables can be constructed u
#Not Sure what to do, but I'll remove these variables in the meantime
```

```
model3 <- glm(cs_train$SeriousDlqin2yrs ~ MonthlyIncome + cs_train[,15] + cs_train[,16] + cs_train[,17]
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
model3
```

```
##
## Call:  glm(formula = cs_train$SeriousDlqin2yrs ~ MonthlyIncome + cs_train[,
##     15] + cs_train[, 16] + cs_train[, 17] + cs_train[, 18] +
##     cs_train[, 19], family = "binomial", data = cs_train)
##
## Coefficients:
##    (Intercept)   MonthlyIncome  cs_train[, 15]  cs_train[, 16]  cs_train[, 17]
##     -2.321e+00      -4.457e-05      -8.130e+00      -8.067e+00              NA
## cs_train[, 18]  cs_train[, 19]
##             NA      -8.110e+00
##
## Degrees of Freedom: 72160 Total (i.e. Null);  72156 Residual
## Null Deviance:       36510
## Residual Deviance: 36340     AIC: 36350
```

```
model4 <- glm(cs_train$SeriousDlqin2yrs ~ MonthlyIncome + cs_train[,15] + cs_train[,16] + cs_train[,17]
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
model4
```

```
##
## Call:  glm(formula = cs_train$SeriousDlqin2yrs ~ MonthlyIncome + cs_train[,
##     15] + cs_train[, 16] + cs_train[, 17] + cs_train[, 18] +
##     cs_train[, 19], family = binomial(link = "probit"), data = cs_train)
##
## Coefficients:
##    (Intercept)   MonthlyIncome  cs_train[, 15]  cs_train[, 16]  cs_train[, 17]
##     -1.384e+00      -1.505e-05      -2.742e+00      -2.721e+00              NA
## cs_train[, 18]  cs_train[, 19]
##             NA      -2.736e+00
##
## Degrees of Freedom: 72160 Total (i.e. Null);  72156 Residual
## Null Deviance:       36510
## Residual Deviance: 36380     AIC: 36390
```

```
model5 <- glm(cs_train$SeriousDlqin2yrs ~ MonthlyIncome + cs_train[,15] + cs_train[,16] + cs_train[,17]
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
model5
```

```
##
## Call:  glm(formula = cs_train$SeriousDlqin2yrs ~ MonthlyIncome + cs_train[,
##     15] + cs_train[, 16] + cs_train[, 17] + cs_train[, 18] +
##     cs_train[, 19], family = binomial(link = "cloglog"), data = cs_train)
##
## Coefficients:
##    (Intercept)   MonthlyIncome  cs_train[, 15]  cs_train[, 16]  cs_train[, 17]
```

```
##     -2.362e+00      -4.376e-05      -8.000e+00      -7.939e+00              NA
## cs_train[, 18]  cs_train[, 19]
##             NA      -7.981e+00
##
## Degrees of Freedom: 72160 Total (i.e. Null);  72156 Residual
## Null Deviance:       36510
## Residual Deviance: 36330     AIC: 36340
```

```
anova(model,model2,model3,model4,model5) #model 2 is the one with NA values
```

```
## Analysis of Deviance Table
##
## Model 1: cs_train$SeriousDlqin2yrs ~ cs_train[, 3] + cs_train[, 4] + cs_train[,
##     5] + cs_train[, 6] + cs_train[, 7] + cs_train[, 8] + cs_train[,
##     9] + cs_train[, 10] + cs_train[, 11] + cs_train[, 12]
## Model 2: cs_train$SeriousDlqin2yrs ~ MonthlyIncome + cs_train[, 13] +
##     cs_train[, 14] + cs_train[, 15] + cs_train[, 16] + cs_train[,
##     17] + cs_train[, 18] + cs_train[, 19] + cs_train[, 20] +
##     cs_train[, 22]
## Model 3: cs_train$SeriousDlqin2yrs ~ MonthlyIncome + cs_train[, 15] +
##     cs_train[, 16] + cs_train[, 17] + cs_train[, 18] + cs_train[,
##     19]
## Model 4: cs_train$SeriousDlqin2yrs ~ MonthlyIncome + cs_train[, 15] +
##     cs_train[, 16] + cs_train[, 17] + cs_train[, 18] + cs_train[,
##     19]
## Model 5: cs_train$SeriousDlqin2yrs ~ MonthlyIncome + cs_train[, 15] +
##     cs_train[, 16] + cs_train[, 17] + cs_train[, 18] + cs_train[,
##     19]
##   Resid. Df Resid. Dev Df Deviance
## 1     72150      33703
## 2     72156      36337 -6 -2634.09
## 3     72156      36337  0     0.00
## 4     72156      36379  0   -41.35
## 5     72156      36334  0    44.37
```

```
#model 1 is the best but we can check model 5 using our dummy variables
```

```
x <- data.frame(cs_train$SeriousDlqin2yrs, cs_train$MonthlyIncome, cs_train[,15],cs_train[,16],cs_train
x <- na.omit(x) #Remember Monthly Income contains NA values
x_for_ridge <- data.matrix(x[,-1]) #Everything but Response Variable
ridge_model5 <- cv.glmnet(x_for_ridge,x[,1], alpha = 0, standardize = TRUE, nfolds = length(x))
ridge_model5
```

```
##
## Call:  cv.glmnet(x = x_for_ridge, y = x[, 1], nfolds = length(x), alpha = 0,      standardize = TRUE
##
## Measure: Mean-Squared Error
##
##     Lambda Index Measure        SE Nonzero
## min  0.749    21 0.06488 0.0008838       4
## 1se  4.812     1 0.06488 0.0008830       4
```

```
lasso_model5 <- cv.glmnet(x_for_ridge,x[,1], alpha = 1, standardize = TRUE, nfolds = length(x))
lasso_model5
```

```
##
## Call:  cv.glmnet(x = x_for_ridge, y = x[, 1], nfolds = length(x), alpha = 1,      standardize = TRUE
##
## Measure: Mean-Squared Error
##
##       Lambda Index Measure        SE Nonzero
## min 0.003317     5 0.06488 0.0008066       1
## 1se 0.004812     1 0.06488 0.0008115       0
```

Evaluate Final Model

- mmp plots to see if model fits the data, as well as the pearson Chi-square test
- checking for outliers and influential points
- Some measure of pesudo R-square and accuracy of the model
- Use confusion matrix, ROC/AUC curve, AIC to evaluate the different models

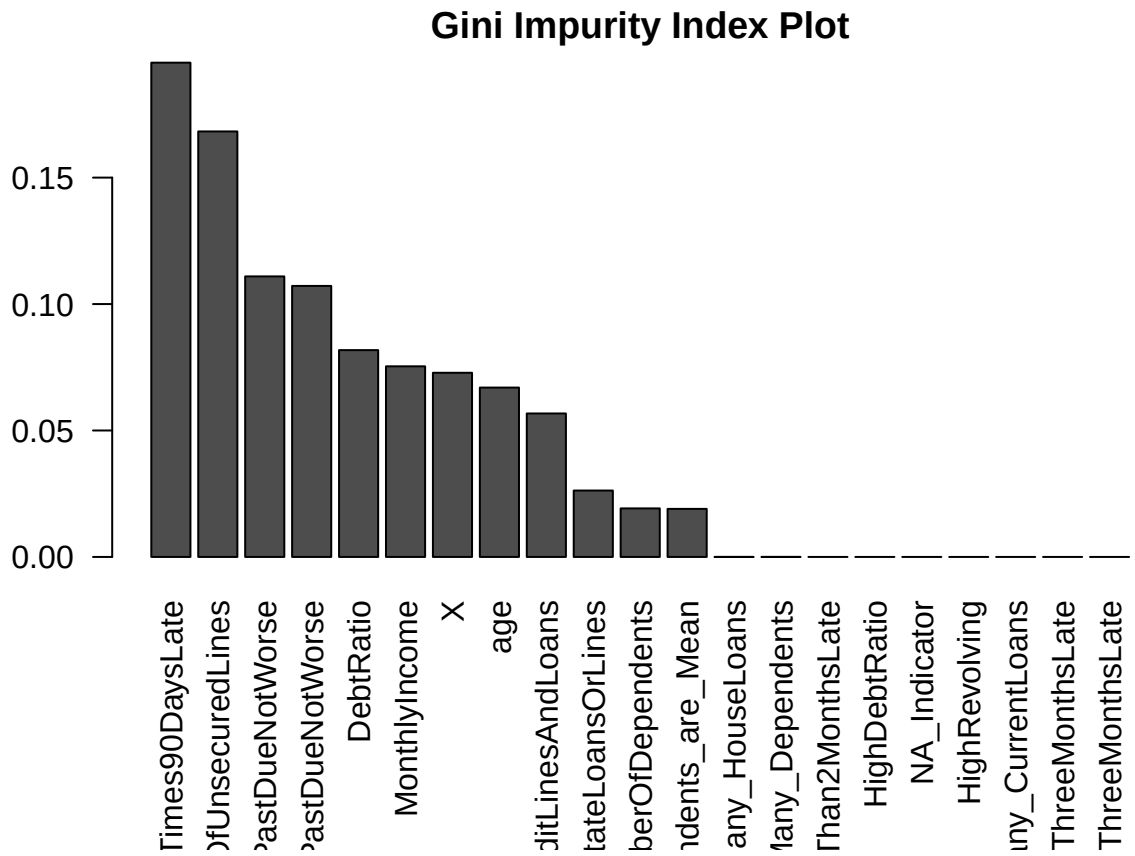
```
# Split x and y variables
train.x = cs_train[,-which(names(cs_train) == "SeriousDlqin2yrs")]
train.y = cs_train$SeriousDlqin2yrs
test.x = cs_test[,-which(names(cs_test) == "SeriousDlqin2yrs")]
test.y = cs_test$SeriousDlqin2yrs
```

Model after recoding Random Forest

```
# Fits Random Forest
model.rf = randomForest(y = as.factor(train.y), x=train.x, xtest=test.x, ytest=as.factor(test.y), mtry =

model.rf # Output shows confusion matrix for both train and test
```

```
##
## Call:
##  randomForest(x = train.x, y = as.factor(train.y), xtest = test.x,      ytest = as.factor(test.y), m
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 6.79%
## Confusion matrix:
##       0   1 class.error
## 0 67024 104 0.001549279
## 1  4793 240 0.952314723
##                 Test set error rate: 6.72%
## Confusion matrix:
##       0   1 class.error
## 0 44721  63 0.001406752
## 1  3171 153 0.953971119
```

```
# Random Forest Output
var.imp = data.frame(importance(model.rf, type=2))
var.imp$Variables = row.names(var.imp)
varimp = var.imp[order(var.imp$MeanDecreaseGini,decreasing = T),]
par(mar = c(7.5,3,2,2))
giniplot = barplot(t(varimp[-2]/sum(varimp[-2])),las=2,cex.names=1, main="Gini Impurity Index Plot")
```

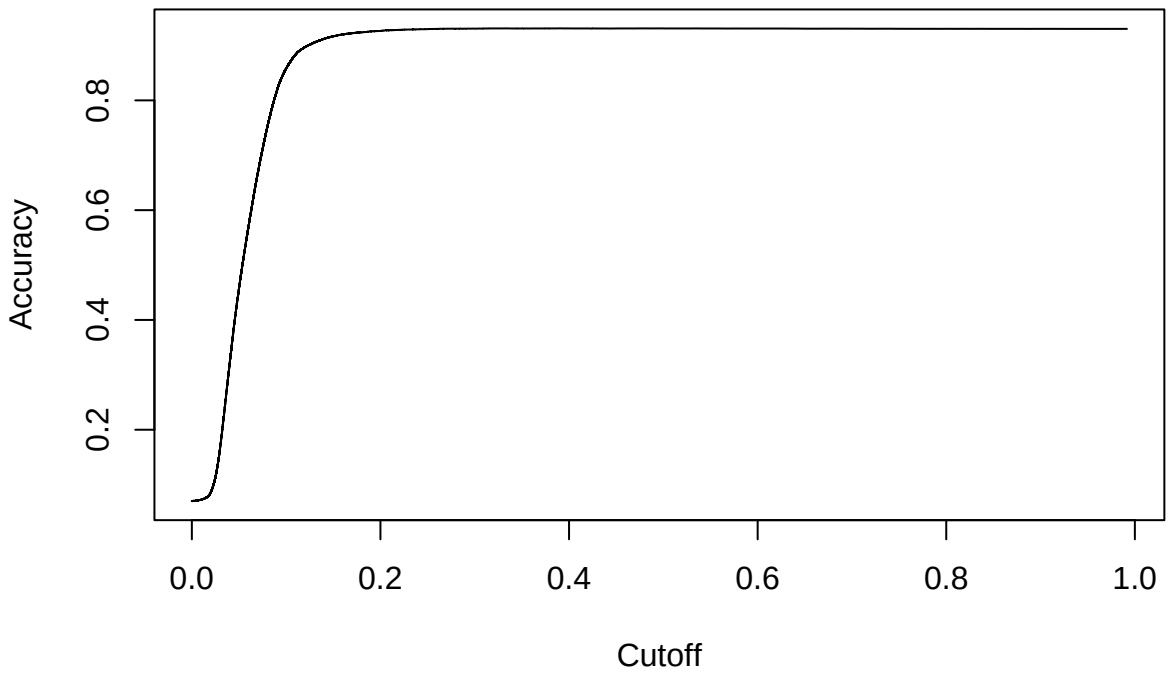


Evaluation on Final Model using Training Data

```
## Set model as final model
model.final <- model

## Evaluation on Final model using Training Data
train_preds = predict(model.final, newdata=train.x, type="response")
head(train_preds[is.na(train_preds)])
```

```
## named numeric(0)
```

```
#train.x[c(7,9),]
#train_preds[is.na(train_preds)]

pred_compare = prediction(train_preds, train.y)
plot(performance(pred_compare, "acc"))
```




```
table(train.y, train_preds>0.2) # accurary on train
```

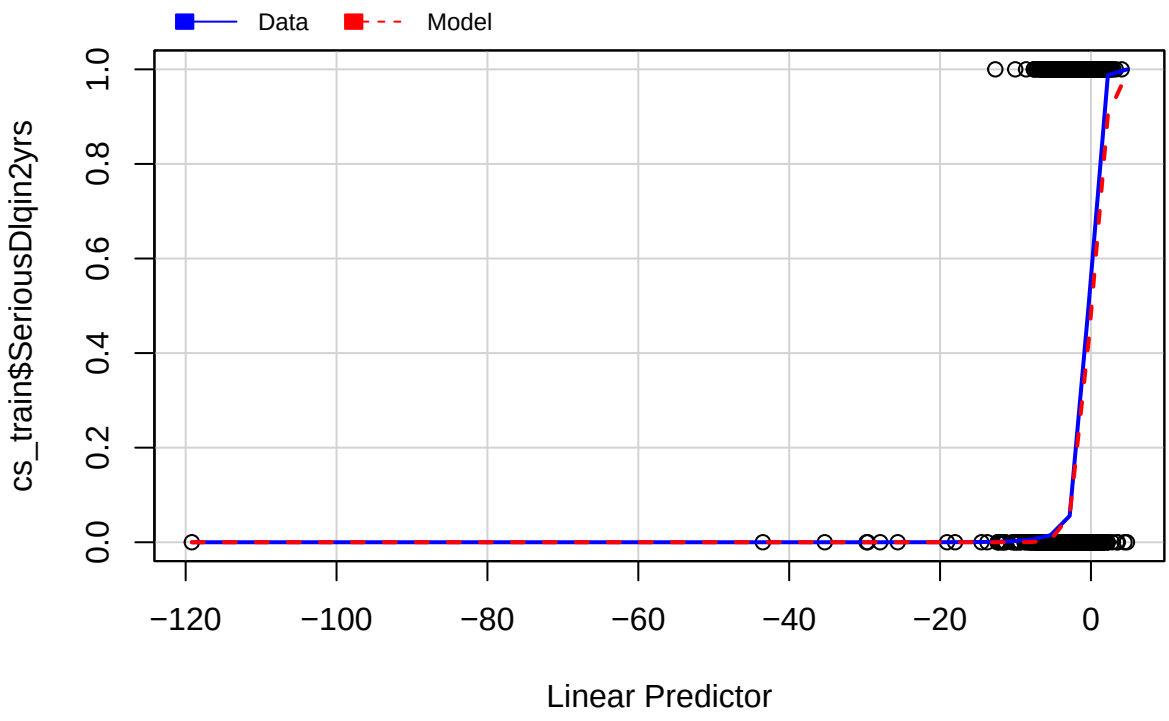
```
##
## train.y FALSE  TRUE
##       0 66009  1119
##       1  4185   848
```

Evaluating Model/Comparing results, Plots, Tests, chi-square test, influential points

Need to Update!!

```
# residual plots
#residualPlots(model.final)

# Marginial Model Plots
library(car)
mmp(model.final)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```



Goodness of Fit using Hosmer-Lemeshow Test

The p-value is 0, meaning that we want to reject the null hypothesis that the model is adequate.

```
# Goodness of Fit using Hosmer-Lemeshow Test
linpred=predict(model.final)

cs_train_m <- mutate(cs_train, predprob=predict(model.final, type="response"))  # cal p^_i
gdf <- group_by(cs_train_m, ntile(linpred, 1000)) # group up the data by eta_x into 100 groups
hldf <- summarise(gdf, y=sum(SeriousDlqin2yrs==1), ppred=mean(predprob), count=n())
head(hldf)
```

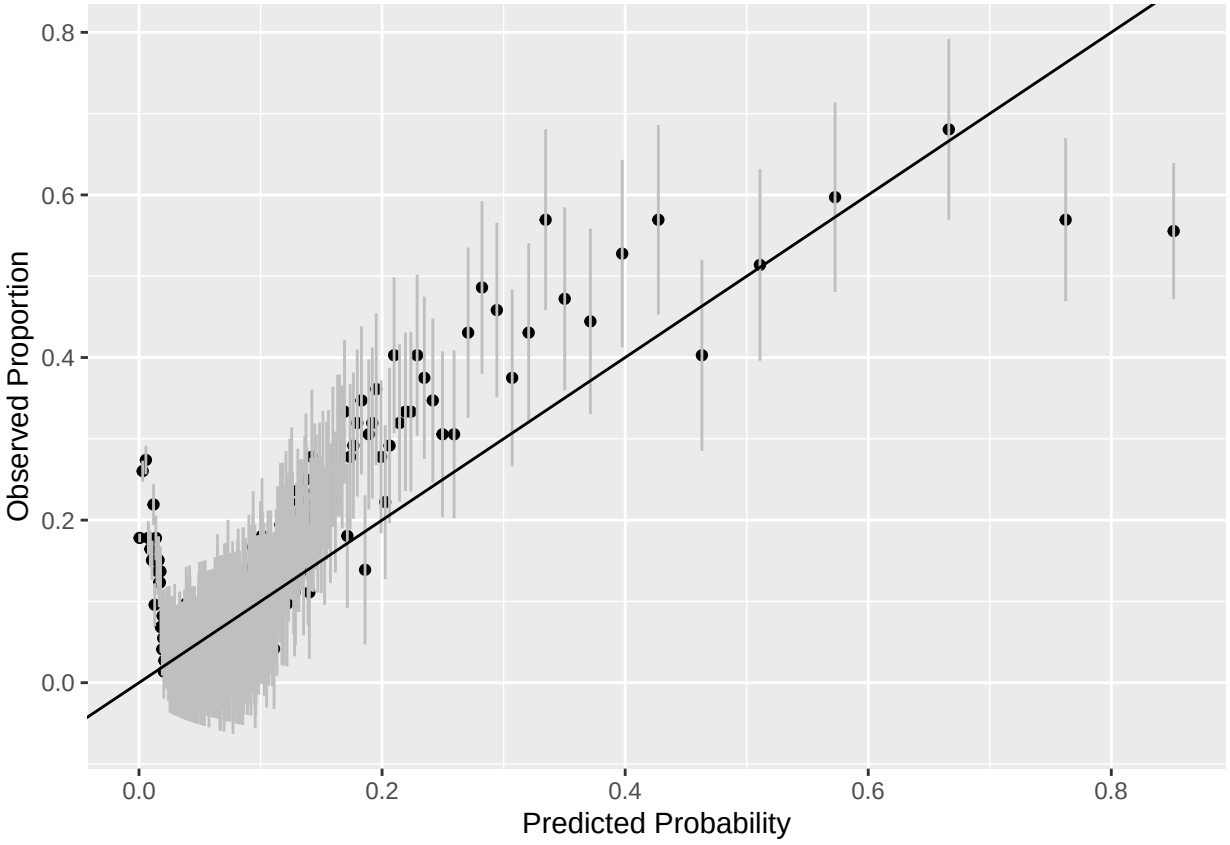
```
## # A tibble: 6 x 4
##   `ntile(linpred, 1000)`     y    ppred count
##                    <int> <int>    <dbl> <int>
## 1                      1    13 0.000517    73
## 2                      2    19 0.00302     73
## 3                      3    20 0.00564     73
## 4                      4    13 0.00771     73
## 5                      5    12 0.00935     73
## 6                      6    11 0.0107      73
```

```
# We adjust the size of the bins until there's only one group with less than 5
hldf[hldf$count<5,]
```

```
## # A tibble: 0 x 4
```

```
## # ... with 4 variables: ntile(linpred, 1000) <int>, y <int>, ppred <dbl>,
## #   count <int>
```

```
# Observed Proportion Confidence Interval vs Predicted Probability
hldf <- mutate(hldf, se.fit=sqrt(ppred*(1-ppred)/count))

ggplot(hldf,aes(x=ppred,y=y/count,ymin=y/count-2*se.fit, ymax=y/count+2*se.fit))+
  geom_point()+geom_linerange(color=grey(0.75))+
  geom_abline(intercept = 0,slope = 1)+
  xlab("Predicted Probability")+
  ylab("Observed Proportion")
```




```
# Hosmer-Lemeshow statistics
hlstat <- with(hldf, sum((y-count*ppred) ^2/(count * ppred * (1-ppred))))
c(hlstat, nrow(hldf))
```

```
## [1] 9894.036 1000.000
```

```
# The p-value is given by:
1-pchisq(hlstat, nrow(hldf)-2)
```

```
## [1] 0
```

AUC

Need to fix the Prediction function!!!

Predict -> Currently gives 120k, expecting 80k

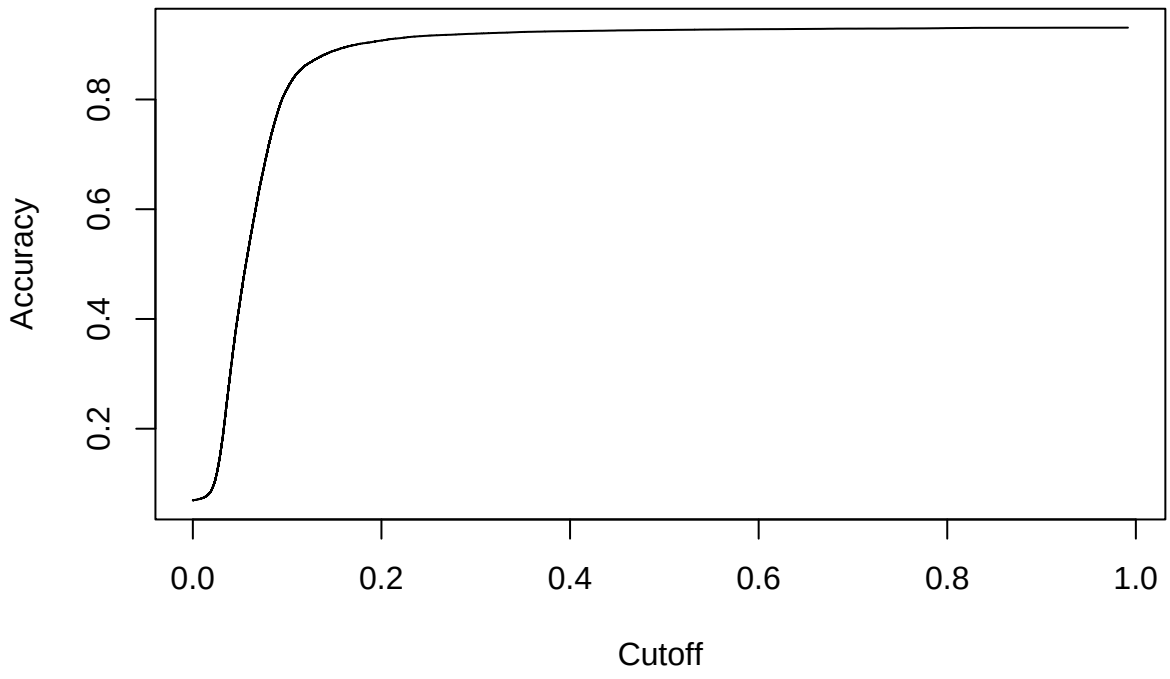
```
#Final Model(w/ interaction term ReugularMedicine * PhysicallyActive)
result_m2 = predict(model.final, newdata=test.x, type="response")
```

```
## Warning: 'newdata' had 48108 rows but variables found have 72161 rows
```

```
# Fix Later!!! Forced the length to be the same
result_m2 <- result_m2[1:length(test.y)]
head(test.y)
```

```
## [1] 1 0 0 0 0 1
```

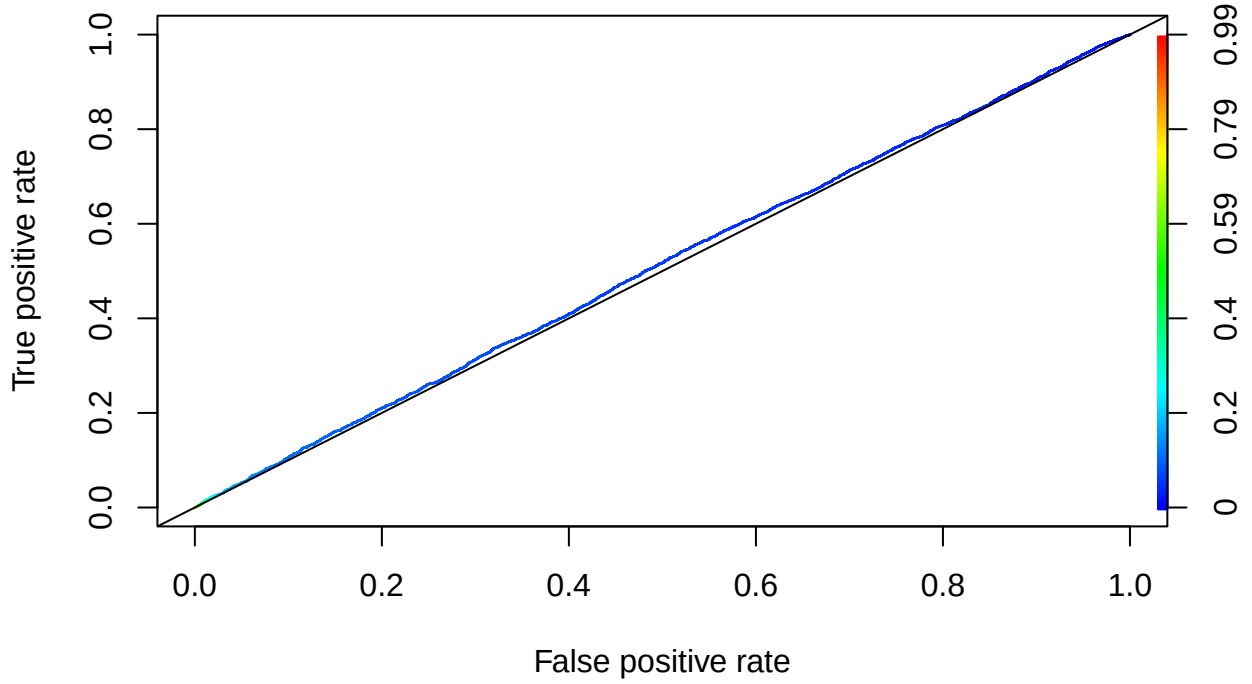
```
pred_m2 = prediction(result_m2, test.y)


plot(performance(pred_m2, "acc")) #It seems like 0.52 cutoff has the highest accuracy
```



```
table(test.y, result_m2>0.2)
```

```
##
## test.y FALSE  TRUE
##      0 43554  1230
##      1  3227    97
```

```
#Accuracy :
#Sensitivity :
#Specificity :
#The Specificity and accuracy improved a bit compared to the previous model without interaction term, s
plot(performance(pred_m2,"tpr","fpr"), colorize=T)
abline(0,1)
```




```
#Now we calculate the area under the curve (AUC) and accuracy of the model given above (glmModel2)
auc_ROCR2 <- performance(pred_m2, measure = "auc")
auc_ROCR2@y.values[[1]]
```

```
## [1] 0.5101429
```

Not Sure if we use this!!

```r
# Check how the model fits the data
# From model.final, we can calculate the difference in the two deviances from the summary(model): 987.2
#Number of regressors in the model: 813-802=11
pchisq(473.75,12)
```

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

```r
#The area below 473.75 is one which means the area above it is almost zero. This means that our model h
print(paste("Pearson's X^2 =",round(sum(residuals(model.final,type="pearson")^2),3)))
```

```
## [1] "Pearson's X^2 = 447306.933"
```

```r
qchisq(0.95,802)
```

```
## [1] 868.9936
```

```r
#781.61<868.99, so we fail to reject the null hypothesis and conclude that the logistic model fits the
```