Classification Regression Problem

Jeremiah Joseph

7/12/2021

1. Introduction

This data set that I am using is the credit card record for various credit card applications. Using these two data sets I will merge the two to try to model which applicants would be most likely to be good candidates to have a credit card. The data has 438,557 observations. I will run various machine learning algorithms and give analysis and rank which works best on the data.

```
#packages involved in data clearning
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------------------------------
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

```
## -- Conflicts ---------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)

applicationRecord <- read.csv("application_record.csv")
creditRecord <- read.csv("credit_record.csv")
```

2. Data Cleaning

Link to Data: https://www.kaggle.com/rikdifos/credit-card-approval-prediction

There are some duplicate ids in the application record data set. This will make things difficult to merge. I am first removing these from the data

```
#removing duplicates from both data sets
applicationRecord <- applicationRecord[!duplicated(applicationRecord$ID), ]
creditRecord <- creditRecord[!duplicated(creditRecord$ID), ]
```

I am now row binding the credit record to the application record by ID. There are parts that do not overlap, meaning there is no data on specific ids. Due to this I will eliminate those as well from the data set.

```
#right joining the data sets
df <- right_join(applicationRecord, creditRecord, id="ID")
```

```
## Joining, by = "ID"
```

```
#removing those with no code gender
x <- is.na(df$CODE_GENDER)
df <- df[!x, ]

#removing those with no status
x <- is.na(df$STATUS)
df <- df[!x, ]
```

Now that we have the data set set up we can look at the general structure of the data

```
#getting structure
str(df)
```

```
## 'data.frame':    36457 obs. of  20 variables:
##  $ ID                 : int  5008804 5008805 5008806 5008808 5008809 5008810 5008811 5008812 5008813
##  $ CODE_GENDER        : chr  "M" "M" "M" "F" ...
##  $ FLAG_OWN_CAR       : chr  "Y" "Y" "Y" "N" ...
##  $ FLAG_OWN_REALTY    : chr  "Y" "Y" "Y" "Y" ...
##  $ CNT_CHILDREN       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AMT_INCOME_TOTAL   : num  427500 427500 112500 270000 270000 ...
##  $ NAME_INCOME_TYPE   : chr  "Working" "Working" "Working" "Commercial associate" ...
##  $ NAME_EDUCATION_TYPE: chr  "Higher education" "Higher education" "Secondary / secondary special" "
##  $ NAME_FAMILY_STATUS : chr  "Civil marriage" "Civil marriage" "Married" "Single / not married" ...
##  $ NAME_HOUSING_TYPE  : chr  "Rented apartment" "Rented apartment" "House / apartment" "House / apar
##  $ DAYS_BIRTH         : int  -12005 -12005 -21474 -19110 -19110 -19110 -19110 -22464 -22464 -22464 .
##  $ DAYS_EMPLOYED      : int  -4542 -4542 -1134 -3051 -3051 -3051 -3051 365243 365243 365243 ...
##  $ FLAG_MOBIL         : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ FLAG_WORK_PHONE    : int  1 1 0 0 0 0 0 0 0 0 ...
##  $ FLAG_PHONE         : int  0 0 0 1 1 1 1 0 0 0 ...
##  $ FLAG_EMAIL         : int  0 0 0 1 1 1 1 0 0 0 ...
##  $ OCCUPATION_TYPE    : chr  "" "" "Security staff" "Sales staff" ...
##  $ CNT_FAM_MEMBERS    : num  2 2 2 1 1 1 1 1 1 1 ...
##  $ MONTHS_BALANCE     : int  0 0 0 0 -22 0 0 -4 0 -1 ...
##  $ STATUS             : chr  "C" "C" "C" "0" ...
```

I will now convert predictors to proper data types or get rid of them if they seem unable to be used.

```
#converting gender to factor
df$CODE_GENDER <- as.factor(df$CODE_GENDER)

#converting car and realty ownership to factor
df$FLAG_OWN_CAR <- as.factor(df$FLAG_OWN_CAR)
df$FLAG_OWN_REALTY <- as.factor(df$FLAG_OWN_REALTY)
```

```
#converting and the type and status  to factor
df$NAME_INCOME_TYPE <- as.factor(df$NAME_INCOME_TYPE)
df$NAME_EDUCATION_TYPE <- as.factor(df$NAME_EDUCATION_TYPE)
df$NAME_FAMILY_STATUS <- as.factor(df$NAME_FAMILY_STATUS)
df$NAME_HOUSING_TYPE <- as.factor(df$NAME_HOUSING_TYPE)

#converting all the electronic ownership as factors
df$FLAG_MOBIL <- as.factor(df$FLAG_MOBIL)
df$FLAG_WORK_PHONE<- as.factor(df$FLAG_WORK_PHONE)
df$FLAG_PHONE <- as.factor(df$FLAG_PHONE)
df$FLAG_EMAIL <- as.factor(df$FLAG_EMAIL)

#Deleting occupation type as there are too many types to reasonably make predictor
df$OCCUPATION_TYPE <- NULL

#Deleting Days of birth and  as these are given in inconsistent formats and
#making days employed more readable
df$DAYS_BIRTH <- NULL
#days employed counts backwards so I am multiplying by -1
df$DAYS_EMPLOYED <- df$DAYS_EMPLOYED * -1
df$DAYS_EMPLOYED[df$DAYS_EMPLOYED < 0] <- 0
```

Next for the data cleaning we need to convert status to something that can be used for classification. Currently the definition of the column is; 0: 1-29 days past due 1: 30-59 days past due 2: 60-89 days overdue 3: 90-119 days overdue 4: 120-149 days overdue 5: Overdue or bad debts, write-offs for more than 150 days C: paid off that month X: No loan for the month. I will make C and X have the value of 1 for good credit candidates and 0 for bad credit candidates.

```
#finding values that are good credit candidates
x <- (df$STATUS == "C" | df$STATUS == "X")

# putting values for status
df$STATUS <- 0
df$STATUS[x] <- 1

#converting status to factor
df$STATUS <- as.factor(df$STATUS)
```

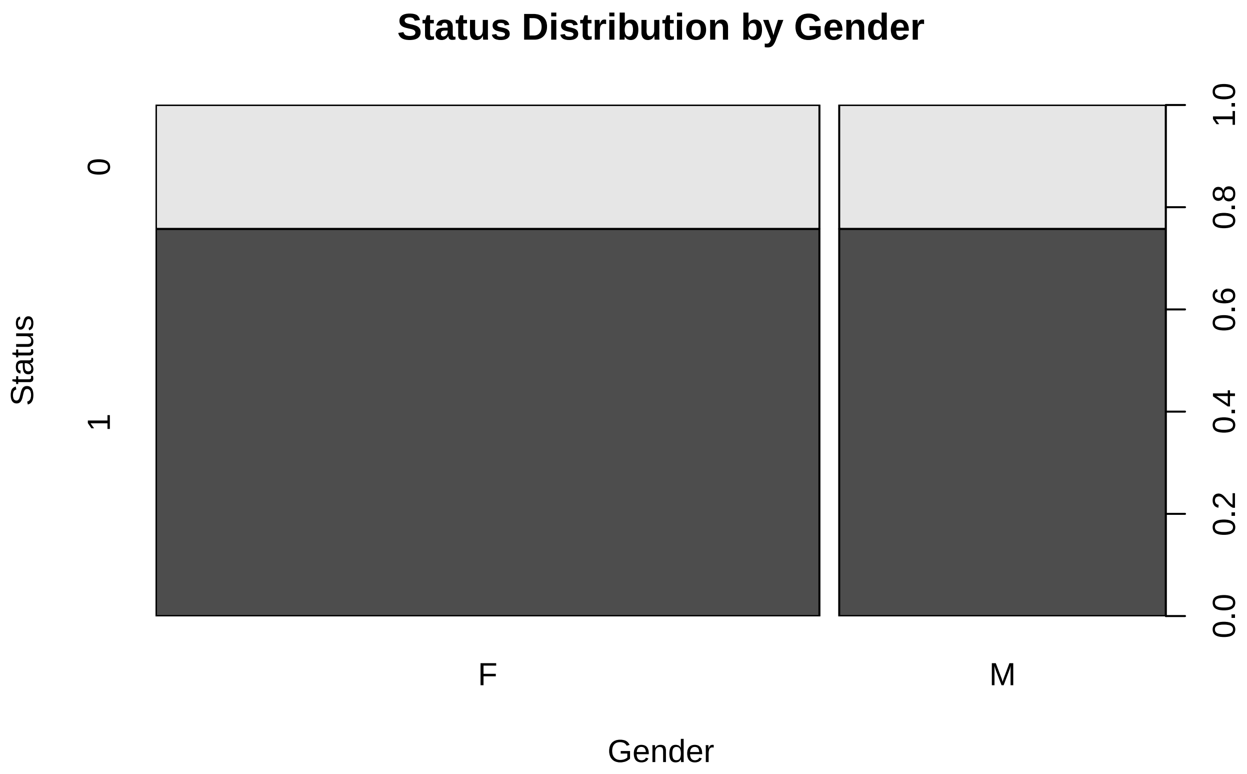
3. Data Exploration

The first thing I am going to do is see how gender affects the status. When looking at the summary we can see that there is about double the amount of females to men, so they are not evenly distributed. This is important to keep in mind when doing the modeling. Looking at the plot we can see that both genders have a very similar percentage of people who have good status. This gives evidence that gender will likley be a poor predictor of credit card status.

```
#seeing the distribution of males to females
summary(df[2])
```

```
##  CODE_GENDER
##  F:24430
##  M:12027
```

```r
#plotting how gender effects good candidates
plot(df$CODE_GENDER, df$STATUS, main= "Status Distribution by Gender"
     , xlab= "Gender", ylab="Status")
```



Next I will look at how ownership of cars various phones, cars, email, and realty change the status. First when looking at the car the summary shows us a disproportionate amount of people do not own cars. Looking at the sum of how people who do and do not own cars changes status, there is slightly a higher percentage of people who own cars with good status, but not significant.

```r
#LOOKING AT CAR

#getting distribution of cars
summary(df[3])
```

```
##  FLAG_OWN_CAR
##  N:22614
##  Y:13843
```

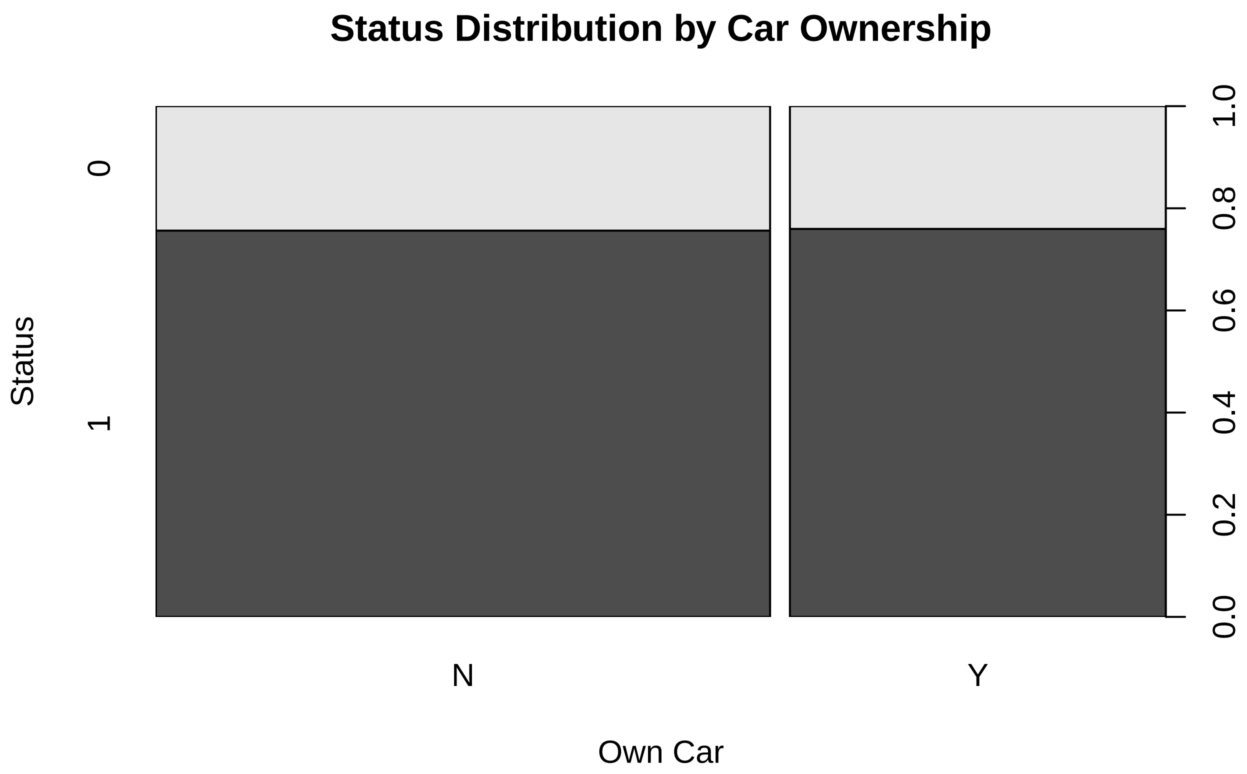
```r
#getting amount of good credit that do own car and do not own car.
sum(df$FLAG_OWN_CAR == "Y" & df$STATUS == 1)
```

```
## [1] 10516
```

```
sum(df$FLAG_OWN_CAR == "N" & df$STATUS == 1)
```

```
## [1] 17103
```

```
#plotting cars with status
plot(df$FLAG_OWN_CAR, df$STATUS, main= "Status Distribution by Car Ownership"
     , xlab= "Own Car", ylab="Status")
```



Next with realty, most people actually do own reality. From looking at the data most people owned realty and those who did not tend to on average have better status.

```
# summary of reality
summary(df$FLAG_OWN_REALTY)
```

```
##     N     Y
## 11951 24506
```

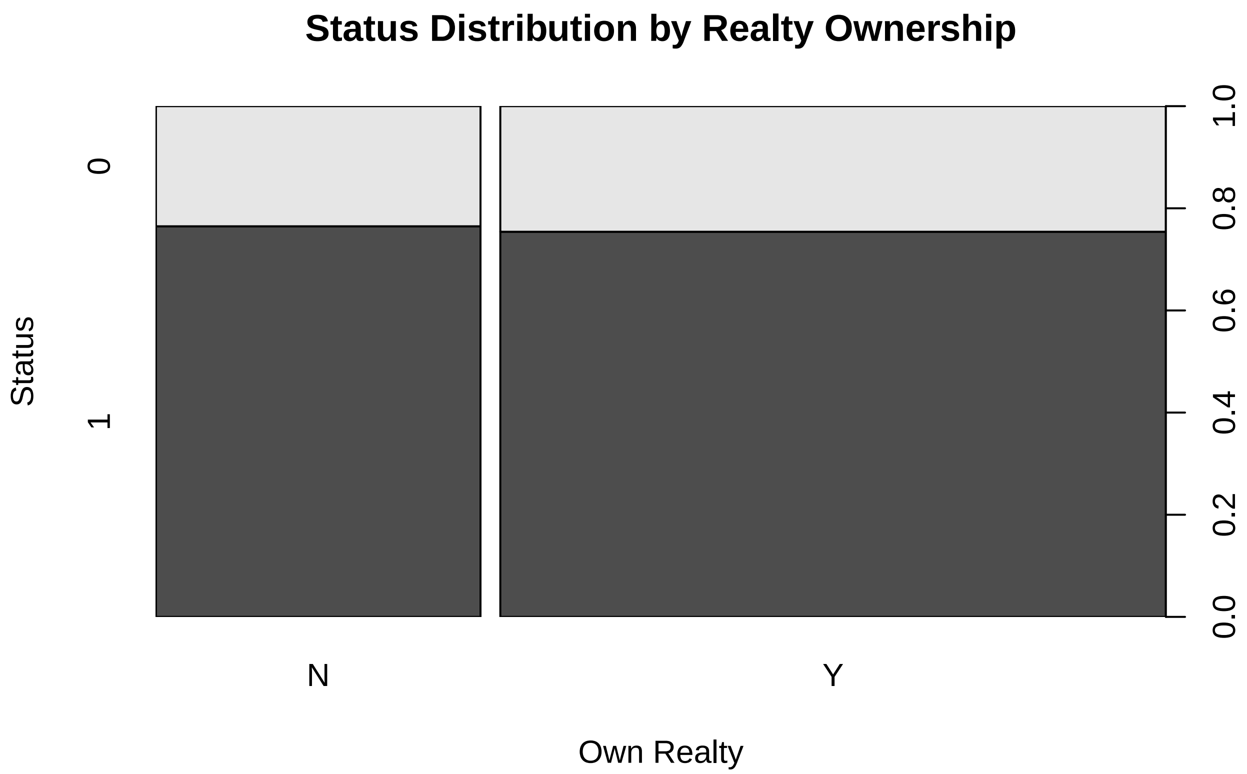
```
#getting amount of good credit that do own car and do not own car.
sum(df$FLAG_OWN_REALTY == "Y" & df$STATUS == 1)
```

```
## [1] 18479
```

```
sum(df$FLAG_OWN_REALTY == "N" & df$STATUS == 1)
```

```
## [1] 9140
```

```
#plotting cars with status
plot(df$FLAG_OWN_REALTY, df$STATUS, main= "Status Distribution by Realty Ownership"
     , xlab= "Own Realty", ylab="Status")
```



Next I will look at different technologies and there impact on credit. When looking at this the vast majority of people own all these. Because of that it will likly serve as bad predictors.

```
summary(df[12:15])
```

```
##  FLAG_MOBIL FLAG_WORK_PHONE FLAG_PHONE FLAG_EMAIL
##  1:36457    0:28235         0:25709    0:33186
##             1: 8222         1:10748    1: 3271
```

Next I will at how income impacts credit. Looking at the data the mean income of those who have good credit is actually lower than those who have bad credit. The plot shows there is a lot of outliers regardless. This is further shown by the high variance

```
#getting mean and variance
mean(df$AMT_INCOME_TOTAL)
```

```
## [1] 186685.7
```

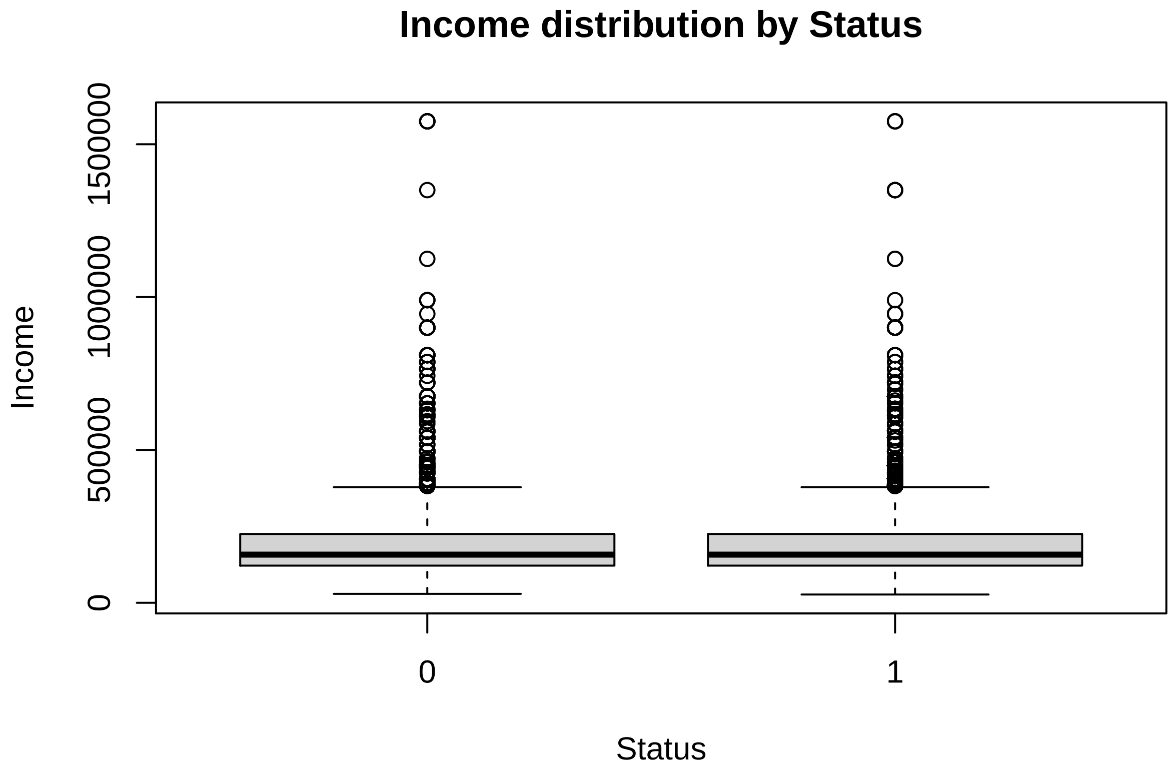
```
var(df$AMT_INCOME_TOTAL)
```

```
## [1] 10361046628
```

```
mean(df$AMT_INCOME_TOTAL[df$STATUS == 1])
```

```
## [1] 186248.4
```

```
#plotting graph
plot(df$STATUS,df$AMT_INCOME_TOTAL, main= "Income distribution by Status"
     , xlab= "Status", ylab= "Income")
```



Next I will check days employed. The days employeed in the data set is right skewed. Typically those with good credit status have been employed for longer.

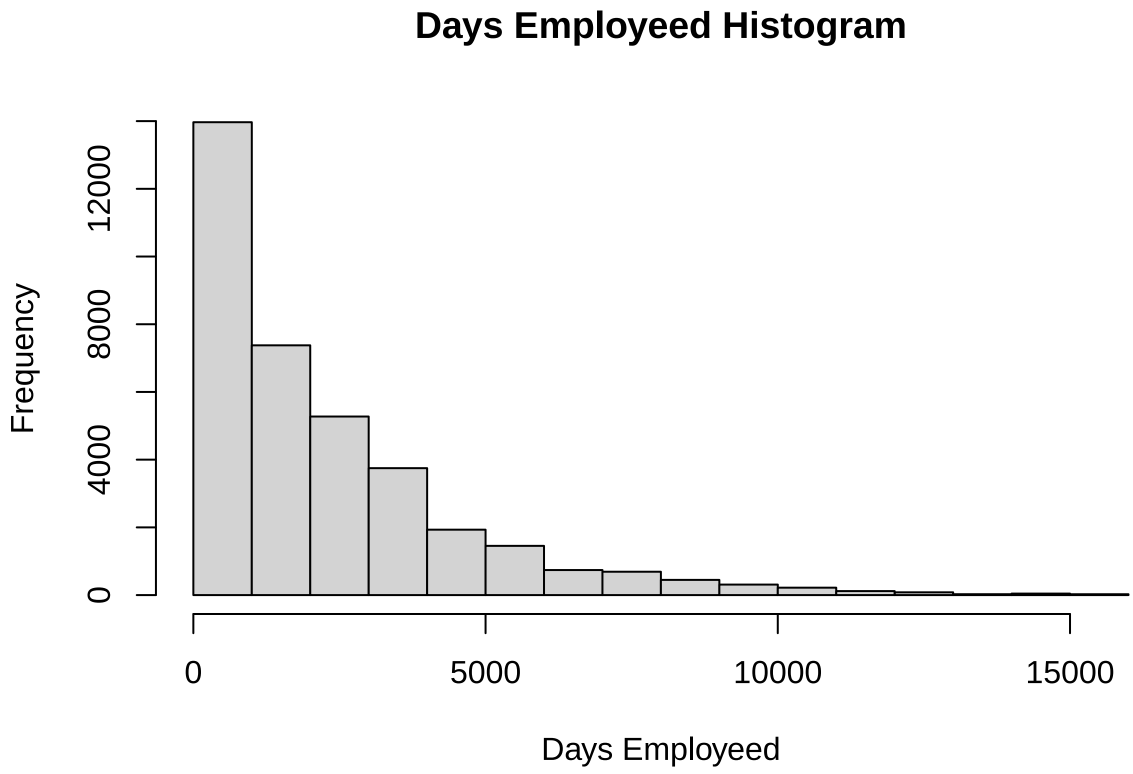
```
#getting two means.
mean(df$DAYS_EMPLOYED)
```

```
## [1] 2200.317
```

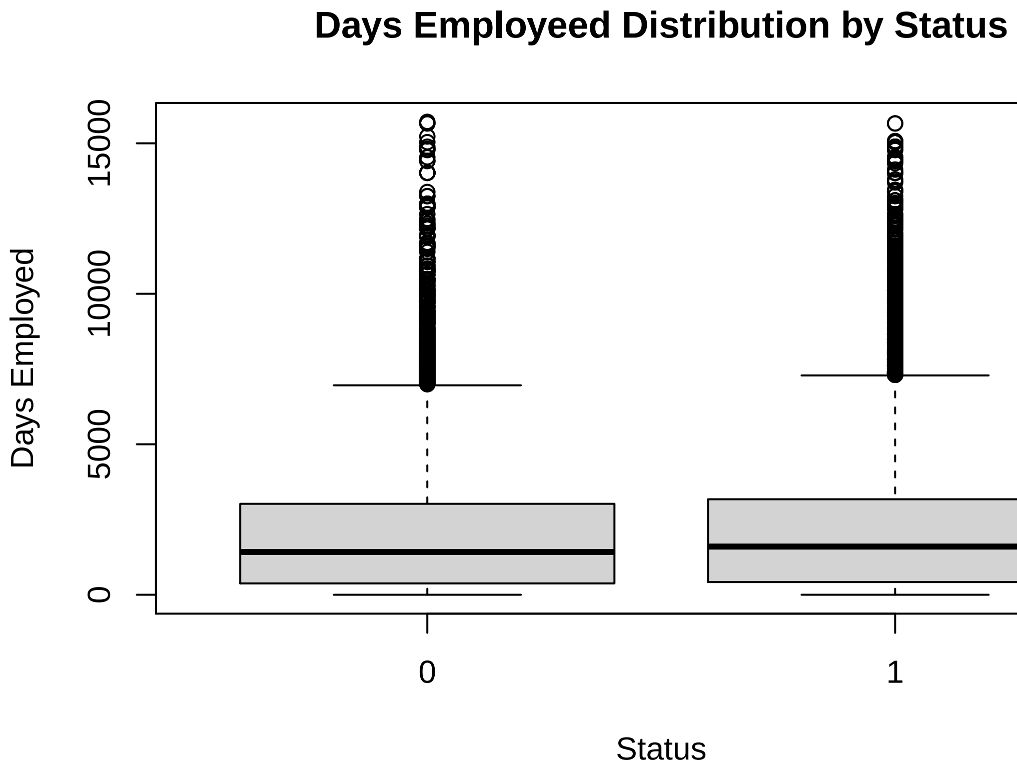
```r
mean(df$DAYS_EMPLOYED[df$STATUS == 1])
```

```
## [1] 2229.576
```

```r
#generating plots
hist(df$DAYS_EMPLOYED, main="Days Employeed Histogram", xlab = "Days Employeed")
```

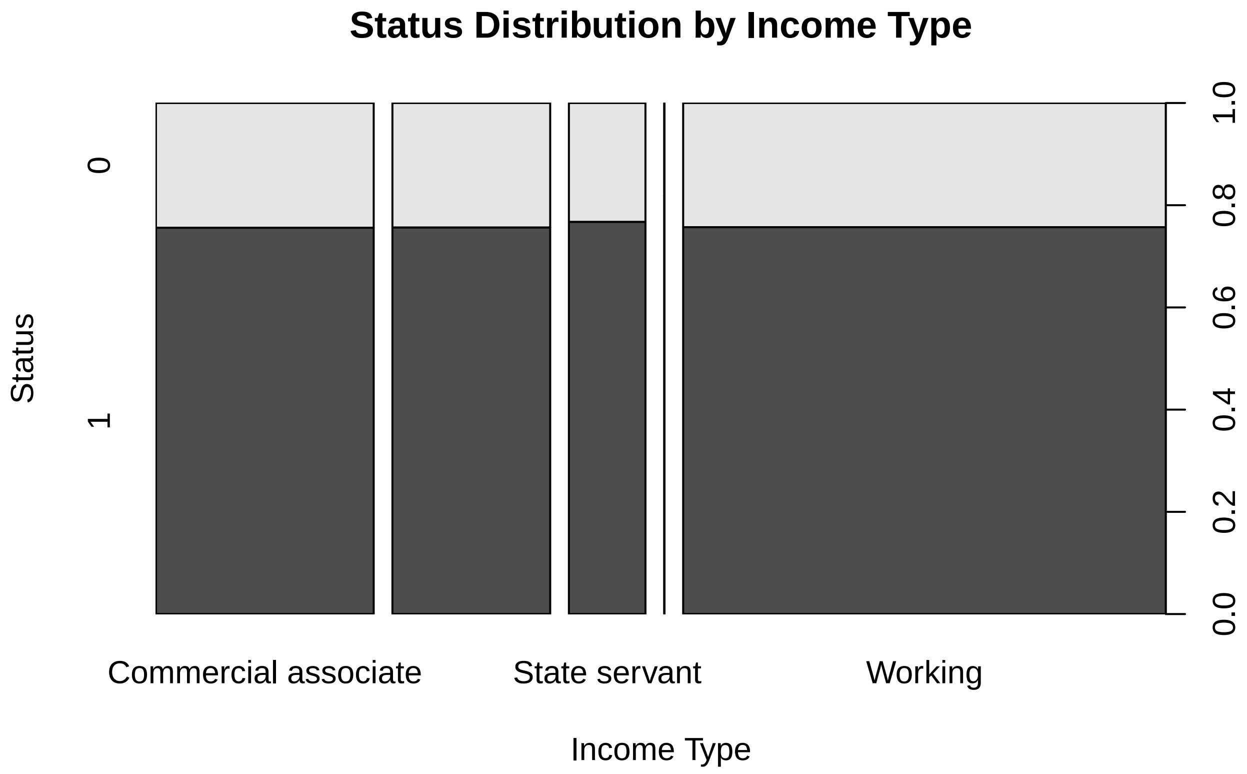


```r
plot(df$STATUS,df$DAYS_EMPLOYED, main= "Days Employeed Distribution by Status"
     , xlab= "Status", ylab= "Days Employed")
```

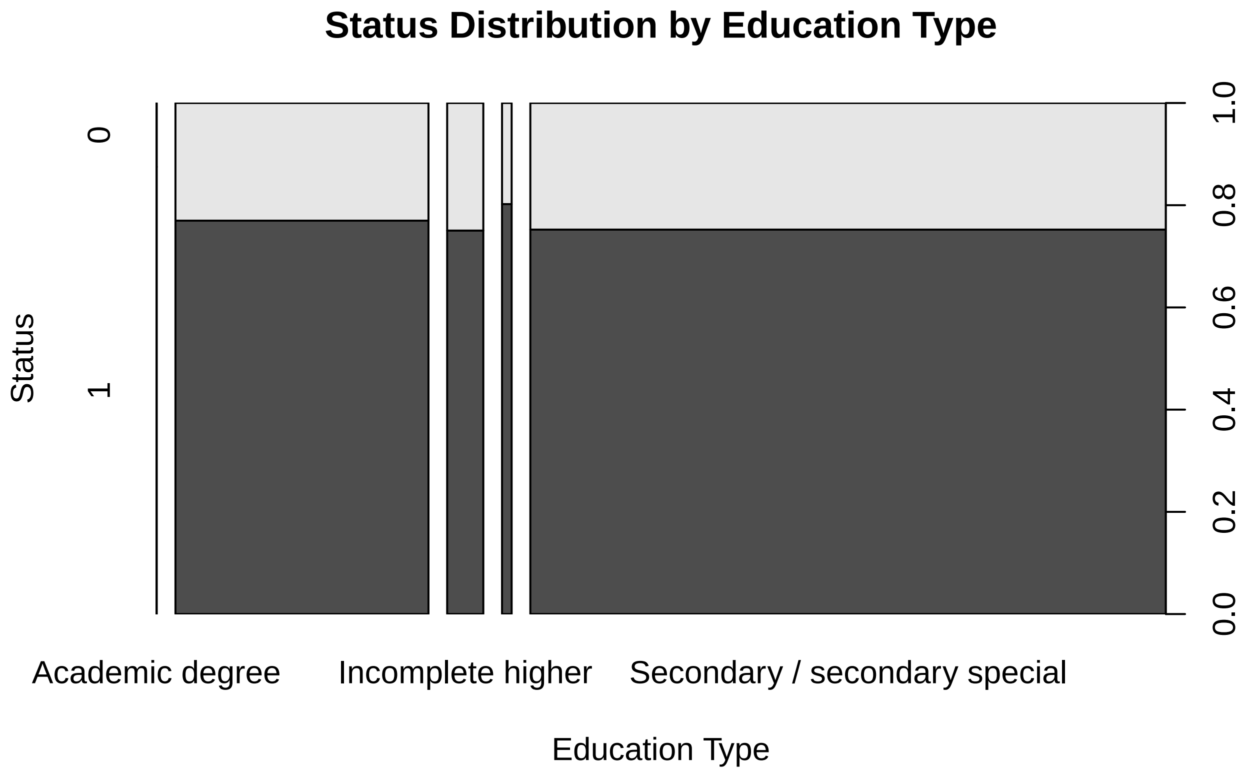


Next I will plot some of the more basic personal information. Looking at this some of these graphs tend to show some fluctuation in the distribution of the two statuses. The income type has very little difference across each level. For the education type however, higher education typically seem to be much better candidates. For family status, those in civil marriages and those who are separated tend to be better candidates. Finally with housing types those with co-op apartments, rented apartments, and those with parents tend to be better candidates than the others. Those in office apartments tend to be worse candidates.

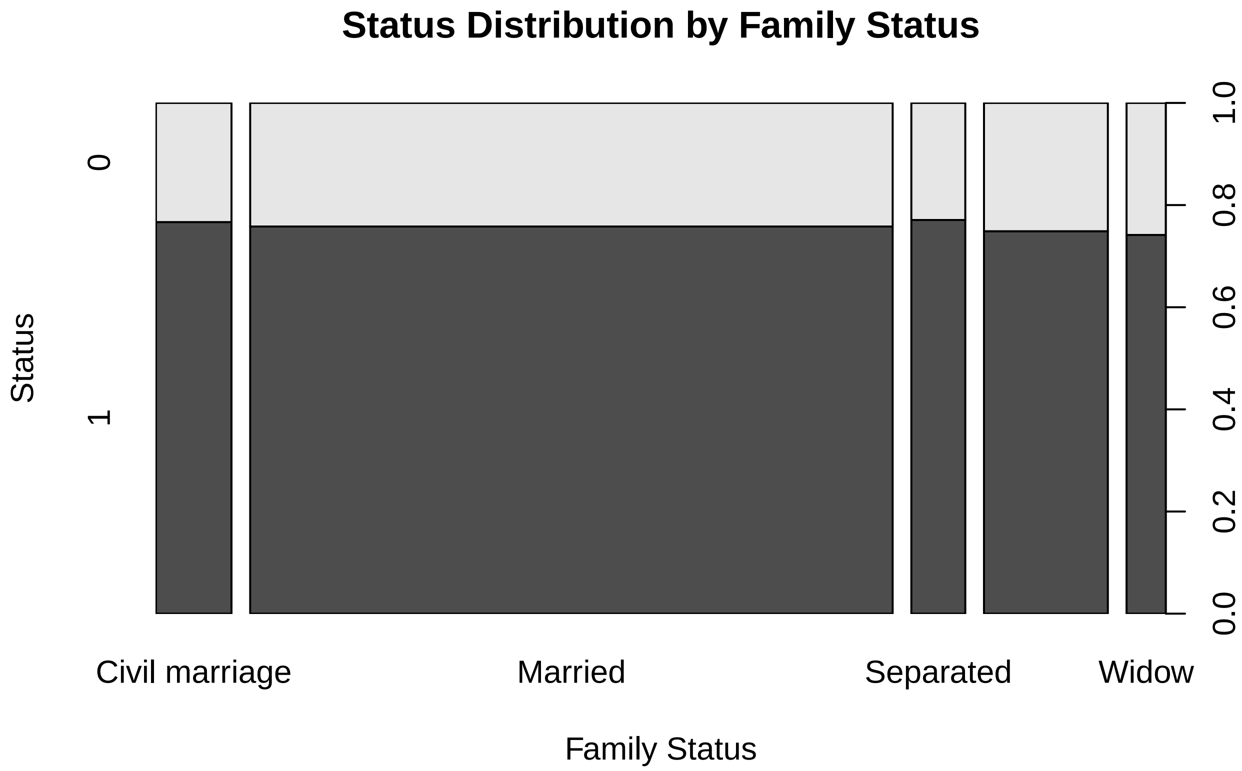
```r
#plotting income
plot(df$NAME_INCOME_TYPE, df$STATUS, main= "Status Distribution by Income Type",
     xlab= "Income Type", ylab="Status")
```



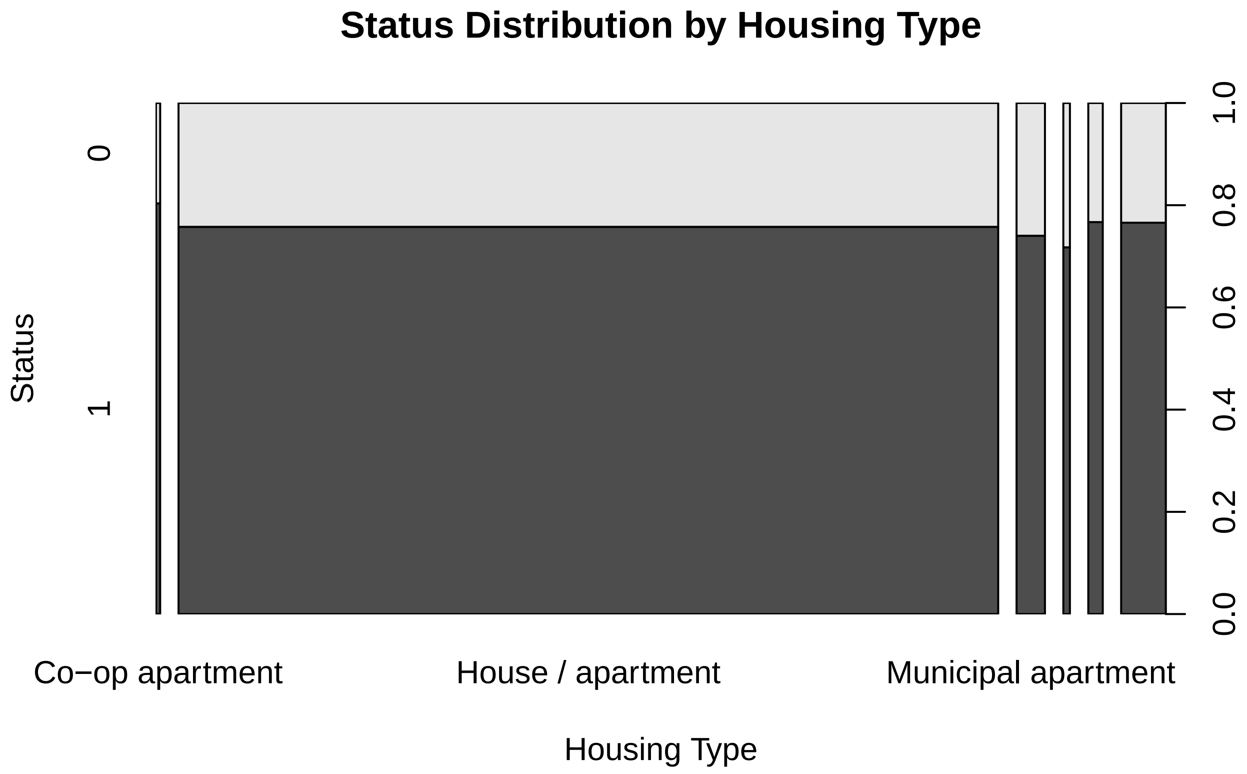
```
#plotting education type
plot(df$NAME_EDUCATION_TYPE, df$STATUS, main= "Status Distribution by Education Type"
     , xlab= "Education Type", ylab="Status")
```



```
#plotting family status
plot(df$NAME_FAMILY_STATUS, df$STATUS, main= "Status Distribution by Family Status"
     , xlab= "Family Status", ylab="Status")
```



```r
#plotting housing type
plot(df$NAME_HOUSING_TYPE, df$STATUS, main= "Status Distribution by Housing Type"
     , xlab= "Housing Type", ylab="Status")
```

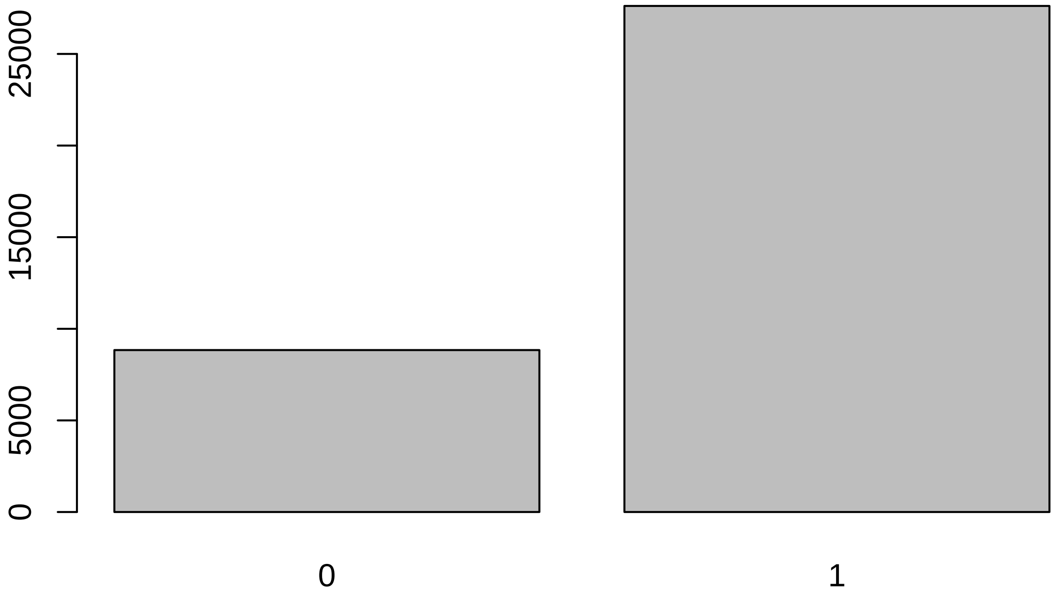


Finally I will look at the target itself. Looking at the summary and the plot the majority of targets tend to be good credit in this. We have around 75% of the data being of a certain status, so ideally are model will have a higher accuaracy than this.

```
summary(df$STATUS)
```

```
##     0     1
##  8838 27619
```

```
plot(df$STATUS)
```



4. Machine Learning Algorithms

4.0 Separting between train and test

```
#seed for reproducibility
set.seed(1234)

#getting 75% for train and rest for test
i <- sample(1:nrow(df), nrow(df) *.75, replace = FALSE)
train <- df[i,]
test <- df[-i,]
```

4.1 Logistic Regression

Feature

The features I am going to use for the logistic regression model is days employed, total income, housing type, family status, and education type. This is because days employed seems to be tending to be higher for those who have been employed longer according to the graph in section 3. I am also using total income as a feature. This is because there is a slight difference between the income median and mean incomes of those who are good candidates and those who are not. In terms of family status, those who are separated and in civil marriages seem to have higher percentages of success. In terms of housing type, those in office apartments seemed to be less likely than the others to be a good credit option. With education type those in higher degrees tend to be better candidates. Ultimately, though none of these predictors are very good, considering the vast majority of the data set in every catergory is a good credit option, these seem to have to most promising patterns.

Analysis of Model

Looking at the summary, the model does not seeem to be very good. Looking at the model the vast majority of the predictors had a high p value. That means that the significance level for most of the predictors is low. Really the only exceptions to this are total income, higher education, and municipal and office apartments. Also what is also concerning is the null deviance is not much higher than the residual deviance.

```r
#Making logistic regression model
glm1 <- glm(STATUS~AMT_INCOME_TOTAL+DAYS_EMPLOYED+NAME_EDUCATION_TYPE
            +NAME_FAMILY_STATUS+NAME_HOUSING_TYPE, data=train, family = "binomial")
summary(glm1)
```

```
## 
## Call:
## glm(formula = STATUS ~ AMT_INCOME_TOTAL + DAYS_EMPLOYED + NAME_EDUCATION_TYPE + 
##     NAME_FAMILY_STATUS + NAME_HOUSING_TYPE, family = "binomial", 
##     data = train)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.1906   0.6476   0.7353   0.7608   0.8993  
## 
## Coefficients:
##                                                     Estimate Std. Error z value
## (Intercept)                                        2.479e+00  6.562e-01   3.777
## AMT_INCOME_TOTAL                                  -2.588e-07  1.410e-07  -1.835
## DAYS_EMPLOYED                                      2.678e-05  6.215e-06   4.309
## NAME_EDUCATION_TYPEHigher education               -8.919e-01  6.132e-01  -1.454
## NAME_EDUCATION_TYPEIncomplete higher              -1.021e+00  6.168e-01  -1.655
## NAME_EDUCATION_TYPELower secondary                -7.004e-01  6.307e-01  -1.110
## NAME_EDUCATION_TYPESecondary / secondary special -9.882e-01  6.129e-01  -1.612
## NAME_FAMILY_STATUSMarried                         -5.112e-02  5.306e-02  -0.963
## NAME_FAMILY_STATUSSeparated                        3.805e-02  7.851e-02   0.485
## NAME_FAMILY_STATUSSingle / not married            -5.425e-02  6.354e-02  -0.854
## NAME_FAMILY_STATUSWidow                           -1.443e-01  8.350e-02  -1.729
## NAME_HOUSING_TYPEHouse / apartment                -3.419e-01  2.272e-01  -1.505
## NAME_HOUSING_TYPEMunicipal apartment              -4.949e-01  2.397e-01  -2.065
## NAME_HOUSING_TYPEOffice apartment                 -6.441e-01  2.742e-01  -2.349
## NAME_HOUSING_TYPERented apartment                 -3.431e-01  2.532e-01  -1.355
## NAME_HOUSING_TYPEWith parents                     -2.525e-01  2.360e-01  -1.070
##                                                   Pr(>|z|)    
## (Intercept)                                       0.000159 ***
## AMT_INCOME_TOTAL                                  0.066547 .  
## DAYS_EMPLOYED                                     1.64e-05 ***
## NAME_EDUCATION_TYPEHigher education               0.145833    
## NAME_EDUCATION_TYPEIncomplete higher              0.097825 .  
## NAME_EDUCATION_TYPELower secondary                0.266787    
## NAME_EDUCATION_TYPESecondary / secondary special 0.106903    
## NAME_FAMILY_STATUSMarried                        0.335362
## NAME_FAMILY_STATUSSeparated                      0.627900
## NAME_FAMILY_STATUSSingle / not married           0.393249
## NAME_FAMILY_STATUSWidow                          0.083870 .
## NAME_HOUSING_TYPEHouse / apartment               0.132303
## NAME_HOUSING_TYPEMunicipal apartment             0.038928 *
## NAME_HOUSING_TYPEOffice apartment                0.018840 *
## NAME_HOUSING_TYPERented apartment                0.175314
## NAME_HOUSING_TYPEWith parents                    0.284661
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 30307  on 27341  degrees of freedom
## Residual deviance: 30254  on 27326  degrees of freedom
## AIC: 30286
##
## Number of Fisher Scoring iterations: 4
```

When looking at the results with the test data the accuracy is 75.85%. Though this might not seem to bad, considering how unbalanced the data was, this is really poor results. When looking at the confusion matrix, the model predicted positive for everything, meaning the sensitivity was 0. The kappa value was 0, which is evidence of a poor model. Ultimately, the fact that the data itself was very unbalanced with respect to the target and the fact that there was very little patterns in the outcome between the various predictors made this model preform poorly.

```r
#Testing model with test data.
probslr <- predict(glm1, newdata = test, type="response")
predlr <- ifelse(probslr > 0.5, 1, 0)

#getting metrics through confusion matrix
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: ’caret’
```

```
## The following object is masked from ’package:purrr’:
##
##     lift
```

```r
confusionMatrix(as.factor(predlr), reference=test$STATUS)
```

```
## Warning in confusionMatrix.default(as.factor(predlr), reference = test$STATUS):
## Levels are not in the same order for reference and data. Refactoring data to
## match.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0    0    0
##          1 2201 6914
##
##                Accuracy : 0.7585
##                  95% CI : (0.7496, 0.7673)
##     No Information Rate : 0.7585
##     P-Value [Acc > NIR] : 0.5057
##
##                   Kappa : 0
##
##  Mcnemar’s Test P-Value : <2e-16
##
##             Sensitivity : 0.0000
##             Specificity : 1.0000
##          Pos Pred Value :    NaN
##          Neg Pred Value : 0.7585
##              Prevalence : 0.2415
##          Detection Rate : 0.0000
##    Detection Prevalence : 0.0000
##       Balanced Accuracy : 0.5000
##
##        ’Positive’ Class : 0
##
```

4.2 Naive Bayes

Features

The features I am going to use for the Naive Bayes model is days employed, total income, housing type, family status, and education type. This is because days employed seems to be tending to be higher for those who have been employed longer according to the graph in section 3. I am also using total income as a feature. This is because there is a slight difference between the income median and mean incomes of those who are good candidates and those who are not. In terms of family status, those who are separated and in civil marriages seem to have higher percentages of success. In terms of housing type, those in office apartments seemed to be less likely than the others to be a good credit option. With education type those in higher degrees tend to be better candidates. Ultimately, though none of these predictors are very good, considering the vast majority of the data set in every catergory is a good credit option, these seem to have to most promising patterns.

Analysis of Model

This is the results of building the model. As can be seen the model computed the A-priori probabilities and conditional probabilities using Bayes method and assuming independence.

```
#making model
library(e1071)
nb1 <- naiveBayes(STATUS~AMT_INCOME_TOTAL+DAYS_EMPLOYED+NAME_EDUCATION_TYPE
            +NAME_FAMILY_STATUS+NAME_HOUSING_TYPE, data=train)
nb1
```

```
##
```

```
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##         0         1
## 0.2427401 0.7572599
##
## Conditional probabilities:
##    AMT_INCOME_TOTAL
## Y       [,1]     [,2]
##   0 187609.8 105722.3
##   1 186371.2 100985.8
##
##    DAYS_EMPLOYED
## Y       [,1]     [,2]
##   0 2088.046 2330.672
##   1 2225.504 2368.336
##
##    NAME_EDUCATION_TYPE
## Y   Academic degree Higher education Incomplete higher Lower secondary
##   0    0.0004520115     0.2573451861      0.0396263372    0.0084375471
##   1    0.0011591403     0.2749094422      0.0374788698    0.0108186428
##    NAME_EDUCATION_TYPE
## Y   Secondary / secondary special
##   0                  0.6941389182
##   1                  0.6756339049
##
##    NAME_FAMILY_STATUS
## Y   Civil marriage    Married  Separated Single / not married      Widow
##   0     0.07849932 0.68811210 0.05378936           0.13289137 0.04670785
##   1     0.08171939 0.68703212 0.05863318           0.13141753 0.04119778
##
##    NAME_HOUSING_TYPE
## Y   Co-op apartment House / apartment Municipal apartment Office apartment
##   0     0.003616092       0.891366581         0.034654211      0.009040229
##   1     0.005071239       0.892682927         0.029751268      0.006858247
##    NAME_HOUSING_TYPE
## Y   Rented apartment With parents
##   0      0.015820401  0.045502486
##   1      0.015503502  0.050132818
```

When looking at testing metrics of our model, the model was still not very good. The accuracy was .7586, but the model still overwhelmingly picked picked positive. This is shown in the sensitivity being approximately .007. This however was an improvement from the logistic regression. The Kappa value was also higher than the logistic regression though it is still extreamly low.

```
#getting predictions and outputing confusion matrix
prednb <- predict(nb1, newdata = test, type = "class")
confusionMatrix(prednb, test$STATUS)
```

```
## Confusion Matrix and Statistics
```

```
##
##           Reference
## Prediction    0    1
##          0   16   15
##          1 2185 6899
##
##                Accuracy : 0.7586
##                  95% CI : (0.7497, 0.7674)
##     No Information Rate : 0.7585
##     P-Value [Acc > NIR] : 0.496
##
##                   Kappa : 0.0077
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.007269
##             Specificity : 0.997830
##          Pos Pred Value : 0.516129
##          Neg Pred Value : 0.759467
##              Prevalence : 0.241470
##          Detection Rate : 0.001755
##    Detection Prevalence : 0.003401
##       Balanced Accuracy : 0.502550
##
##        'Positive' Class : 0
##
```

4.3 KNN

Setting up labels

```
#Getting the test and the traing labels
trainLabels <- df[i, 18]
testLabels <- df[-i, 18]
```

Features

The features I am going to use for the KNN model is days employed, total income, housing type, family status, and education type. This is because days employed seems to be tending to be higher for those who have been employed longer according to the graph in section 3. I am also using total income as a feature. This is because there is a slight difference between the income median and mean incomes of those who are good candidates and those who are not. In terms of family status, those who are separated and in civil marriages seem to have higher percentages of success. In terms of housing type, those in office apartments seemed to be less likely than the others to be a good credit option. With education type those in higher degrees tend to be better candidates. Ultimately, though none of these predictors are very good, considering the vast majority of the data set in every catergory is a good credit option, these seem to have to most promising patterns.

Analysis of Model

We converted the factors to integers to work with the knn algorithm. Though this is not ideal, these were really the only predictors that seemed to be of any use so it is likely worth the negative impacts on model.

```
library(class)
#for knn to work we need to convert factors to
train$NAME_EDUCATION_TYPE <- as.integer(train$NAME_EDUCATION_TYPE)
train$NAME_FAMILY_STATUS <- as.integer(train$NAME_FAMILY_STATUS)
train$NAME_HOUSING_TYPE <- as.integer(train$NAME_HOUSING_TYPE)
test$NAME_EDUCATION_TYPE <- as.integer(test$NAME_EDUCATION_TYPE)
test$NAME_FAMILY_STATUS <- as.integer(test$NAME_FAMILY_STATUS)
test$NAME_HOUSING_TYPE <- as.integer(test$NAME_HOUSING_TYPE)

#getting knn model with k=3
knnPred <- knn(train=train[,c(6,8,9,10,11)], test=test[,c(6,8,9,10,11)],
               cl=trainLabels, k=3)
```

Looking at the results from the model this had the lowest accuracy. This however had by far the highest percentages of negatives that have been counted correctly. Due to this, though the accuracy was smaller and the correct positive percentage went down, this model was able to recognize negative cases at a much better rate than the other algorithms.

```
#getting accuracy
results <- (knnPred == testLabels)
acc <- length(which(results==TRUE)) / length(results)

#printing table and accuracy
table(results, knnPred)
```

```
##        knnPred
## results    0    1
##   FALSE  708 1707
##   TRUE   494 6206
```

```
acc
```

```
## [1] 0.7350521
```

5. Results Analysis

When ranking the algorithms I would rank the KNN the best, then I would rank the Naive Bayes second best and then I would rank the logistic regression the worst. This may be initially surprising as the the KNN had the worst accuracy of the models at approximately .735, but what distinguishes it from the other models is that sensitivity is much higher. The other algorithms worked well because the data set is very unbalanced, but the low sensitivity makes me feel that the under a more balanced data set, the models would preform much more poorly. The reason why I have the Naive Bayes model is second is because it had the highest accuracy but also able to pick up on some of the negative outcomes, though at a very small percentage. The logistic regression to me was the worst. This model though it had a fairly high accuracy, computed everything to be a positive outcome. Looking at the summary listed in the R code below, the minimum probability was .6674, meaning no observation was even really that close to being predicted as a negative value. Due to this, I felt this model was extremely bad, and was a victim of having its coefficients computed from a very unbalanced data set. Under more unbalanced data, this model would likely preform very poorly.

The reason why I felt that the KNN was able to preform a lot better in recognizing negative values is because the nature of the data. The data was very unbalanced and there was not very clear predictors. Due to this, because KNN looked at the nearest neighbors it was able to recognize the subtitles in the data such as office-homes ownership being less likely to produce a good credit card owner, and higher education being more likly. This likely allowed it to recognize very subtle patterns that were not as obvious, and thus produce better results.

In terms of all the models, none of the models worked very well. None of the models did significantly better than always picking good credit option. Through the data exploration we were able, however, to see cases that there were subtle cases where there were less likely to have good credit ownership like income, certain family statuses, certain housing types, certain education levels, and certain days working. The KNN model also showed results that could be studied more. For future research it would be advantageous to try to collect more balanced data, and try to focus on some of the possible indicators mentioned in the report.

```r
summary(probslr)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.6674  0.7455  0.7559  0.7574  0.7673  0.8879
```