Apply for loans with strategy

Introduction

Nowadays, individuals and small businesses take out loans for many reasons. Most studies were done on the bank side, with existing data, scientists build models and predict the risk associated with each loan, but rarely people stand on the borrower's side and help them save the interests they don't have to pay. When taking out loans, we usually just tell the bank or the lending company the amount of money we need, and they provided an interest rate plan that we can either leave or take. Most borrowers don't understand the fanatical value behind the features of a loan such as term, and effect of the loan amount, and this could result user paying more interest than they need. Here I used the loan data over the past 8 years from the Leading Club (a peer to peer lending company), I will run regression and feature engineering to understand the basic mechanism with interest rate assignment. By selecting the relevant features to understand how interest rates are assigned, I will develop a recommendation program that help borrower save on interest.

Explore and Clean data

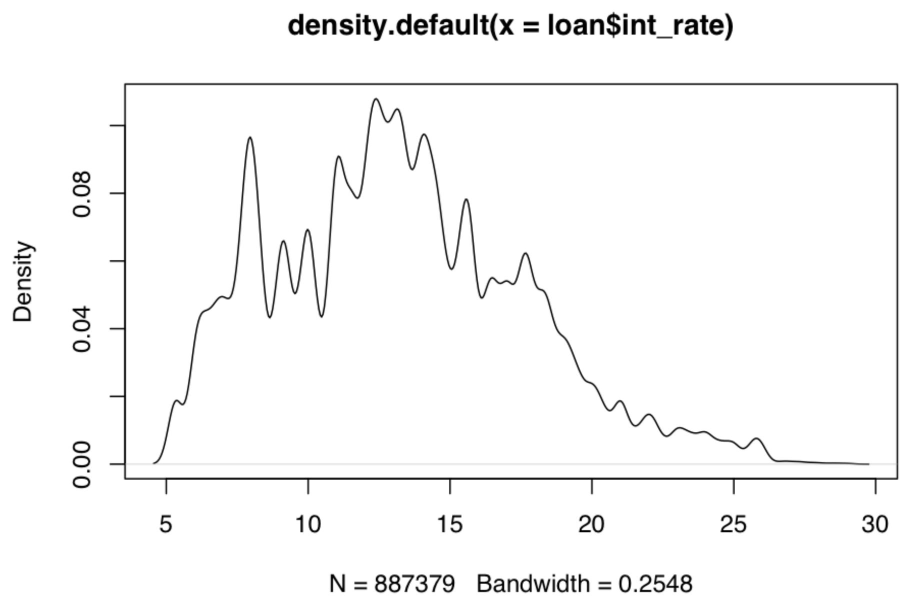
```
loan <- read.csv("loan.csv", stringsAsFactors = FALSE)
```

To clean up the data set, I drop all the meaningless column, and removed columns with more than 80% Na, for the rest I replace the missing value with the median:

```
num_NA <- sort(sapply(loan, function(x) {sum(is.na(x))}), decreasing=TRUE)
remain_col <- names(num_NA)[which(num_NA <= 0.8*dim(loan)[1])]
loan<- loan[,remain_col]
num_NA <- sort(sapply(loan, function(x) {sum(is.na(x))}), decreasing=TRUE)
for(i in names(num_NA)[which(num_NA >0)]){ loan[which(is.na(loan[,i])),i]<- median(loan[,i],na.rm = T)}
#meaningless <- c('id','member_id','verification_status','url', 'desc','title', 'zip_code')
#loan<- loan[, meaningless]
```

Since the interest rate is the most key we want to look at, let's see the the density plot of the int_rate:

```
##   10%   25%   50%   75%   90%
##  7.69  9.99 12.99 16.20 18.99
```



In the density plot we see the range of the int_rate is from 5% - 30%. I wonder how does this rate compare to the natinal aver rate at bank. I scrape data from the web and compare the lending club interest with the bank interest rate, and observed that: in general, the lending club has a interest 3 fold more than the bank, but the up and down does not quite follow the bank. (That's good for our project, since the overall interest trend is not within our control)

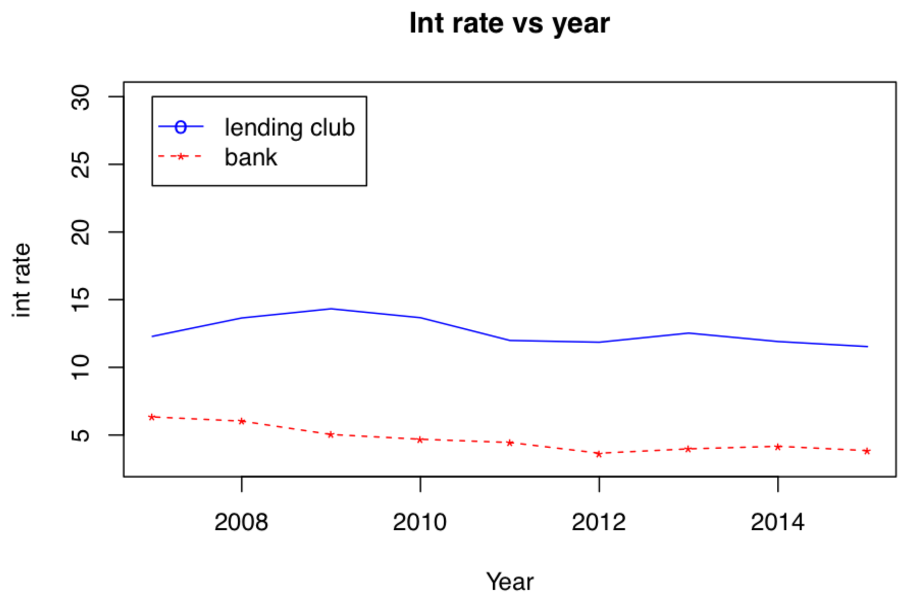
```
bank <- read.csv("History_loan_rate.csv", stringsAsFactors = FALSE)
```

```
bank <- bank[bank$Year >= 2007 ,]
bank <- bank[bank$Year <= 2015 ,]

bank$Lowest.Rate<-as.numeric(substr(bank$Lowest.Rate,start = 0, stop = 4))
bank$Highest.Rate<-as.numeric(substr(bank$Highest.Rate,start = 0, stop = 4))
bank$Average.Rate<-as.numeric(substr(bank$Average.Rate,start = 0, stop = 4))
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

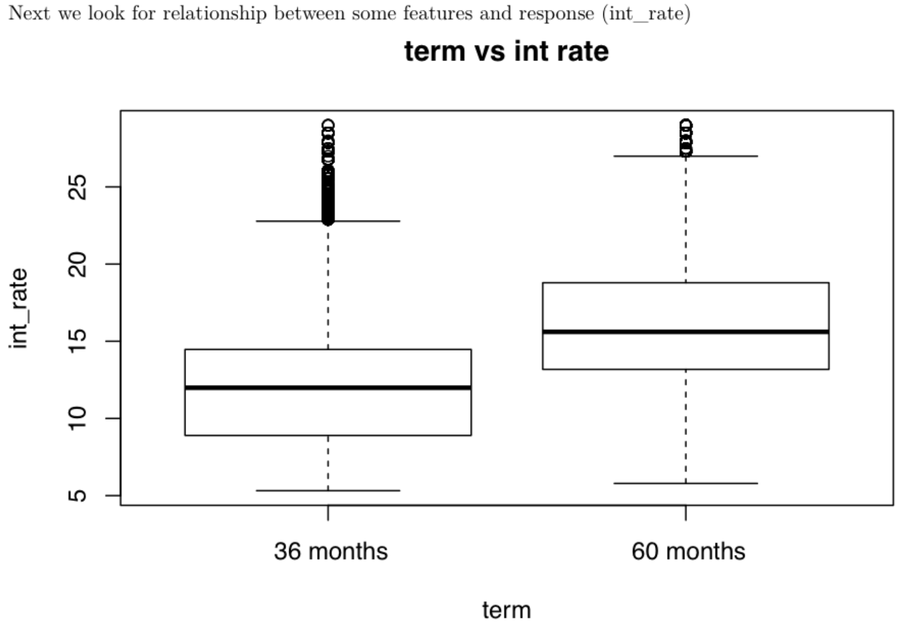
and plot the relation:

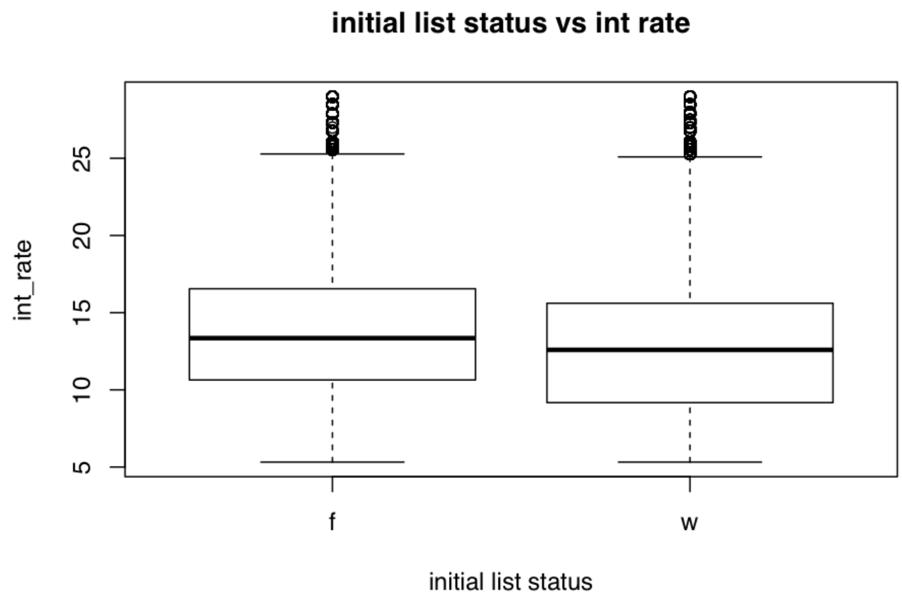
```
int.rate.by.year <- by(loan, loan$issue_year, function(x) {
    return(median(x$int_rate))})
y<-int.rate.by.year
plot(bank$Year, y,ylim = c(3,30),type = 'l',col = 'blue',main = 'Int rate vs year', xlab = 'Year', ylab=
points(bank$Year, bank$Average.Rate, col="red", pch="*")
lines(bank$Year, bank$Average.Rate, col="red",lty=2)
legend(2007,30,legend = c('lending club','bank'),pch = c('o','*'),lty = c(1,2),col =c('blue', 'red'),nc
```



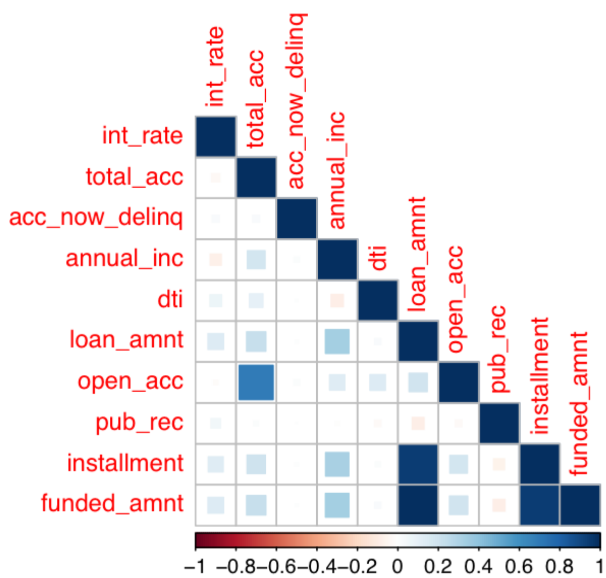
Features

Next we look for relationship between some features and response (int_rate)





And let's look at the correlations between feature, we can see a strong correlation between some features such as open account and total account, or loan amount and funded amount. Such it is inturitive to see correlation in these feature.



Term, initial list (Fraction or Whole loan) are important Feature that user can play with

Let's run the T test for term under the NULL hypothesis that there is no difference

```
t.test(int_rate ~ term, data = loan)
```

```
##
##  Welch Two Sample t-test
##
## data:  int_rate by term
## t = -431.117, df = 467036, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -4.111525 -4.074310
## sample estimates:
## mean in group  36 months mean in group  60 months
##                 12.01868                 16.11160
```

The abs(t) score is really high therefore we reject the NULL hypothesis and conclude there is a significent difference between the two group. Furthermore, if possible just buy switching term plan the expected interest rate saving would be around 4.1%. Another general loan feature is the initial list status, This data point has two possible settings –"F" for fractional, "W" for whole.Certain institutional investors have indicated a preference to be able to purchase loans in their entirety to try and obtain legal and accounting treatment specific to their situation.

```
t.test(int_rate~initial_list_status,data = loan)
```

```
##
##  Welch Two Sample t-test
##
## data:  int_rate by initial_list_status
## t = 109.0349, df = 885516.7, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.9895826 1.0258105
## sample estimates:
## mean in group f mean in group w
##        13.73565        12.72795
```

So here looking at the t score, reject the null hypothesis, and 95% confident that haveing the account funded as a whole is saves about 1% interest.

Linear Regression

```
int_state <- by(loan, loan$addr_state, function(x) {return(mean(x$int_rate))})
loan$state_mean_int <- ifelse(loan$addr_state %in% names(int_state)[which(int_state <= quantile(int_sta
```

```
train_ind <- sample(1:dim(loan)[1],0.7*dim(loan)[1])
train<- loan[train_ind,]
test <- loan[-train_ind,]
```

Finding the most important feature, we do a linear regression on the numerical variable

```
Line_mod <- lm(int_rate ~ state_mean_int+home_ownership+annual_inc+dti+term+loan_amnt+total_acc +tot_cu
                , data = train)
summary(Line_mod)
```

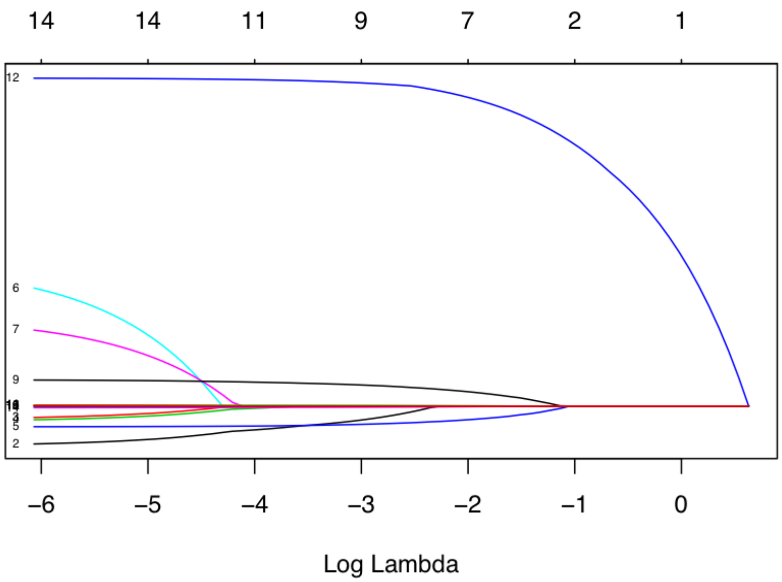
```
##
## Call:
## lm(formula = int_rate ~ state_mean_int + home_ownership + annual_inc +
##     dti + term + loan_amnt + total_acc + tot_cur_bal + open_acc +
##     pub_rec + installment + revol_bal, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -72.041  -2.339  -0.256   1.963  35.436
##
## Coefficients:
##                            Estimate Std. Error  t value Pr(>|t|)
## (Intercept)               1.094e+01  1.782e+00    6.141 8.19e-10 ***
## state_mean_intlow        -3.845e-01  1.717e-02  -22.398  < 2e-16 ***
## state_mean_intmediumhigh -1.259e-01  1.331e-02   -9.456  < 2e-16 ***
## state_mean_intmediumlow  -1.288e-01  1.225e-02  -10.517  < 2e-16 ***
## home_ownershipMORTGAGE   -6.531e-02  1.782e+00   -0.037   0.9708
## home_ownershipNONE        1.050e+00  1.857e+00    0.566   0.5717
## home_ownershipOTHER       1.429e+00  1.804e+00    0.792   0.4283
## home_ownershipOWN         1.487e-01  1.782e+00    0.083   0.9335
## home_ownershipRENT        3.783e-01  1.782e+00    0.212   0.8319
## annual_inc               -2.483e-06  6.783e-08  -36.602  < 2e-16 ***
## dti                       7.342e-03  2.008e-04   36.557  < 2e-16 ***
## term 60 months            1.230e+01  1.651e-02  744.918  < 2e-16 ***
## loan_amnt                -1.593e-03  2.745e-06 -580.319  < 2e-16 ***
## total_acc                -1.773e-02  4.745e-04  -37.363  < 2e-16 ***
## tot_cur_bal              -6.449e-07  3.532e-08  -18.257  < 2e-16 ***
## open_acc                  5.217e-03  1.041e-03    5.014 5.33e-07 ***
## pub_rec                   2.772e-01  6.859e-03   40.413  < 2e-16 ***
## installment               5.174e-02  8.615e-05  600.600  < 2e-16 ***
## revol_bal                -6.224e-07  1.994e-07   -3.121   0.0018 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.087 on 621146 degrees of freedom
## Multiple R-squared:  0.504,  Adjusted R-squared:  0.504
## F-statistic: 3.506e+04 on 18 and 621146 DF,  p-value: < 2.2e-16
```

As we see that most features are significent. The we run linear regression using LASSO to add penalty to redundent features

```r
library(glmnet)
```

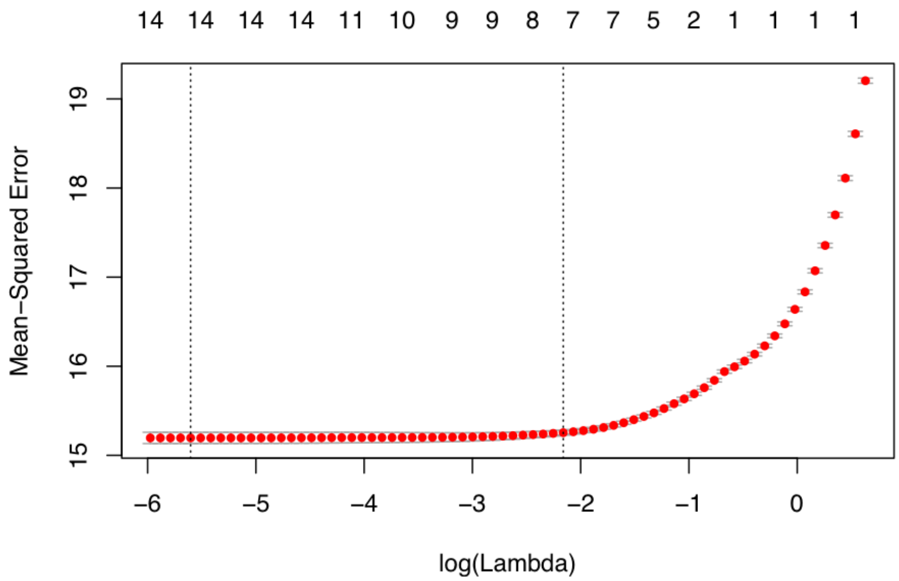
```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:base':
##
##     crossprod, tcrossprod

## Loading required package: foreach

## Loaded glmnet 2.0-3
```

```r
train_sub<- train[,c('int_rate', 'state_mean_int', 'home_ownership', 'annual_inc', 'dti', 'term', 'loan_
ind<- train_sub[,-1]
ind<- model.matrix(~. , ind)
dep<- train_sub[,1]
fit<- glmnet(x = ind, y=dep)
vnat<- coef(fit)
plot(fit,xvar='lambda', label = T, yaxt ='n', ylab = '')
```



From the above fit we can conclue the most important feature being, loan amount,

```r
cvfit <- cv.glmnet(ind, dep,alpha=1, nlambda = 100)
plot(cvfit)
```



Using the optimal lambda we retrive the following parameters:

```
fit1<- glmnet(x = ind, y=dep, lambda= cvfit$lambda.1se, alpha = 1)
fit1$beta[,1]
```

```
##              (Intercept)        state_mean_intlow state_mean_intmediumhigh
##             0.000000e+00             0.000000e+00             0.000000e+00
##  state_mean_intmediumlow   home_ownershipMORTGAGE       home_ownershipNONE
##             0.000000e+00            -1.882037e-01             0.000000e+00
##      home_ownershipOTHER        home_ownershipOWN       home_ownershipRENT
##             0.000000e+00             0.000000e+00             2.014912e-01
##               annual_inc                      dti            term 60 months
##            -2.045764e-06             5.477318e-03             3.984137e+00
##                loan_amnt                total_acc              tot_cur_bal
##             0.000000e+00            -8.703331e-03            -1.817252e-06
##                 open_acc
##             0.000000e+00
```

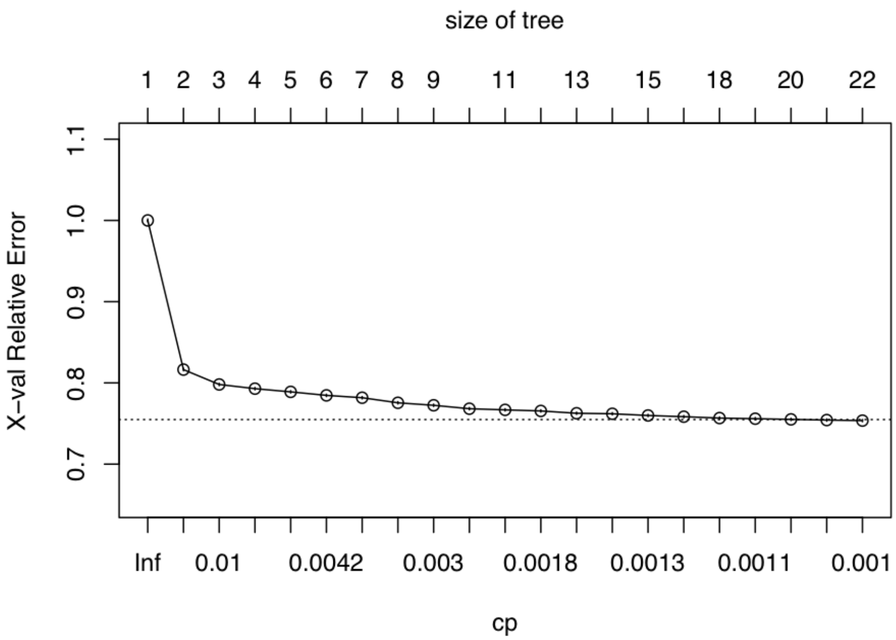
So we conclude that the important features for this model is: annual income, home ownership, term, total account, and total current balance.

Tree Based model

```
library(rpart)
```

```
## Warning: package 'rpart' was built under R version 3.1.3
```

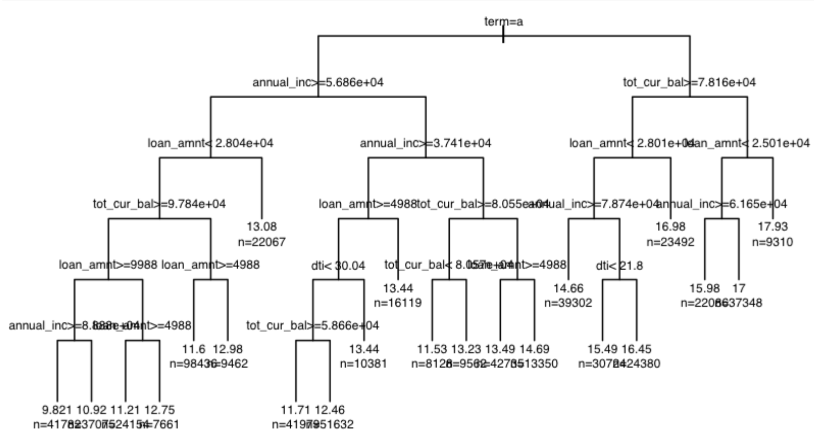
```
tree0 <- rpart(int_rate ~ state_mean_int+home_ownership+annual_inc+dti+term+loan_amnt+total_acc +tot_cu:
plotcp(tree0)
```




```r
bestcp <- tree0$cptable[which.min(tree0$cptable[, 'xerror']),'CP']
tree_pruned <- prune(tree0, cp = bestcp)
test_pred <-predict(tree0, test)
sqrt(sum(test_pred- test$int_rate)^2 / length(test_pred))
```

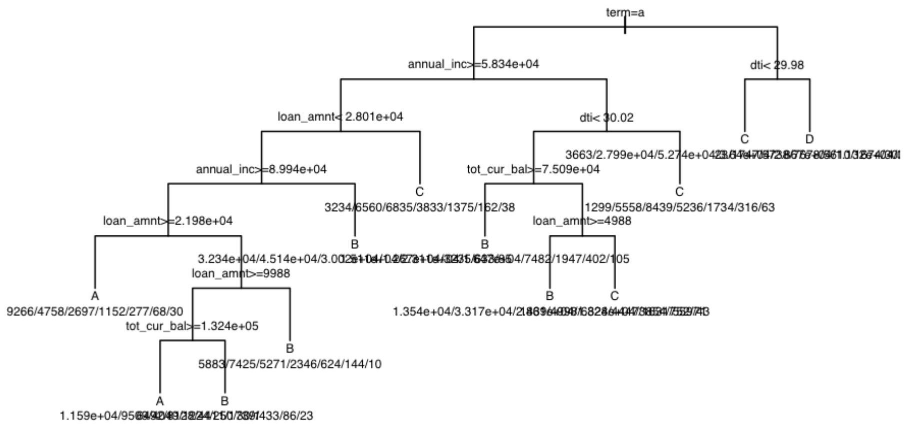
```
## [1] 4.341481
```

```r
plot(tree0 ,uniform = TRUE)
text(tree0, cex = 0.5, use.n = TRUE, xpd= TRUE)
```



We use the same model to predict the grade as an outcome:

```r
train_sub1<- train[,c('grade', 'state_mean_int', 'home_ownership', 'annual_inc', 'dti', 'term', 'loan_a
treeG <- rpart(grade ~ state_mean_int+home_ownership+annual_inc+dti+term+loan_amnt+total_acc +tot_cur_b
plot(treeG, uniform = T)
text(treeG, cex = 0.5, use.n = TRUE, xpd= TRUE)
```




```r
P<- predict(treeG, test, type = 'class')
table(test$grade, predicted = P)
```

```
##    predicted
##         A     B     C     D     E     F     G
##   A  8881 31402  4039     5     0     0     0
##   B  6200 50101 19401   738     0     0     0
##   C  3031 36283 31975  2460     0     0     0
##   D  1012 16510 21331  3246     0     0     0
##   E   238  4184 14221  2421     0     0     0
##   F    48   826  5273   748     0     0     0
##   G    15   139  1369   117     0     0     0
```

The above tree is not doing as good at the lower grades.

Random forest decision tree:

```r
library(randomForest)
```

```
## randomForest 4.6-12
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```r
train_sub$state_mean_int<- as.factor(train_sub$state_mean_int)
train_sub$home_ownership<- as.factor(train_sub$home_ownership)
train_sub$term<- as.factor(train_sub$term)
set.seed(2)
rf<- randomForest(x = train_sub[,-1], y = train_sub[,1], importance = TRUE, do.trace = TRUE, nodesize =
```

```
##      |      Out-of-bag   |
## Tree |      MSE  %Var(y) |
##    1 |    14.74    76.75 |
##    2 |    14.57    75.88 |
##    3 |    14.48    75.38 |
```

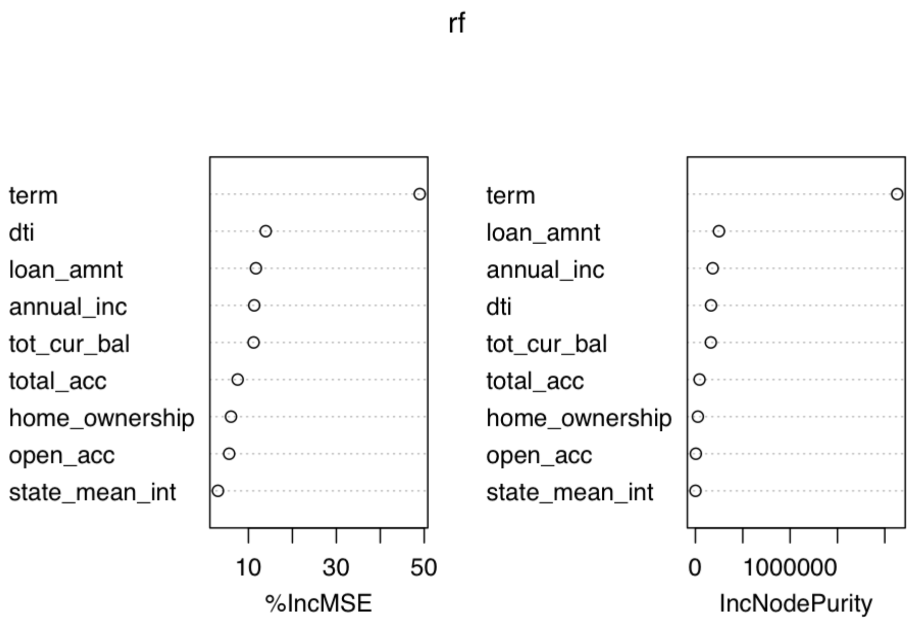
```
##     4 |     14.43     75.15 |
##     5 |     14.37     74.80 |
##     6 |     14.33     74.59 |
##     7 |      14.3     74.45 |
##     8 |     14.29     74.42 |
##     9 |     14.28     74.33 |
##    10 |     14.25     74.19 |
```

```
#getTree(rf,k=1,labelVar = TRUE)
```

```
train_sub1$grade <- as.factor(train_sub1$grade)
train_sub1$state_mean_int<- as.factor(train_sub1$state_mean_int)
train_sub1$home_ownership<- as.factor(train_sub1$home_ownership)
train_sub1$term<- as.factor(train_sub1$term)
rf1 <- randomForest(grade ~ ., data = train_sub1, nodesize = 6200, ntree = 10)
test$state_mean_int<- as.factor(test$state_mean_int)
test$home_ownership<- as.factor(test$home_ownership)
test$term<- as.factor(test$term)
test$grade<- as.factor(test$grade)
levels(test$term)<- levels(train_sub1$term)
levels(test$state_mean_int)<- levels(train_sub1$state_mean_int)
levels(test$home_ownership)<- levels(train_sub1$home_ownership)
levels(test$grade) <- levels(train_sub1$grade)
p1 <- predict(rf1, test,type = 'class')
table(test$grade, predict = p1)
```

```
##    predict
##         A     B     C     D     E     F     G
##   A  8616 31163  4540     8     0     0     0
##   B  5746 48043 21942   526   183     0     0
##   C  2799 33038 35304  2032   576     0     0
##   D   960 14691 22917  2894   637     0     0
##   E   213  3779 14043  2304   725     0     0
##   F    27   722  4999   752   395     0     0
##   G     6   140  1255   115   124     0     0
```

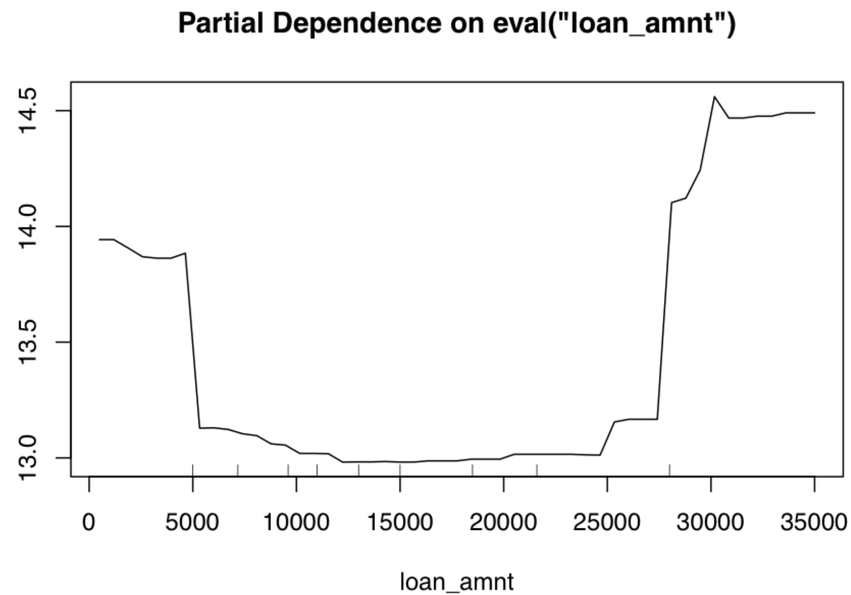
```
varImpPlot(rf)
```




```r
importance(rf, type = 1)
```

```
##                 %IncMSE
## state_mean_int  3.060083
## home_ownership  6.029248
## annual_inc     11.314554
## dti            13.936340
## term           48.991669
## loan_amnt      11.701491
## total_acc       7.543391
## tot_cur_bal    11.175048
## open_acc        5.609559
```

```r
importanceOrder = order(rf$importance[, '%IncMSE'],decreasing = T)
names<- rownames(rf$importance)[importanceOrder]
partialPlot(rf, train_sub, eval('loan_amnt'),xlab='loan_amnt')
```




```
test$state_mean_int<- as.factor(test$state_mean_int)
test$home_ownership<- as.factor(test$home_ownership)
test$term<- as.factor(test$term)
sqrt(sum(test_pred-test$int_rate)^2)/length(test_pred)
```

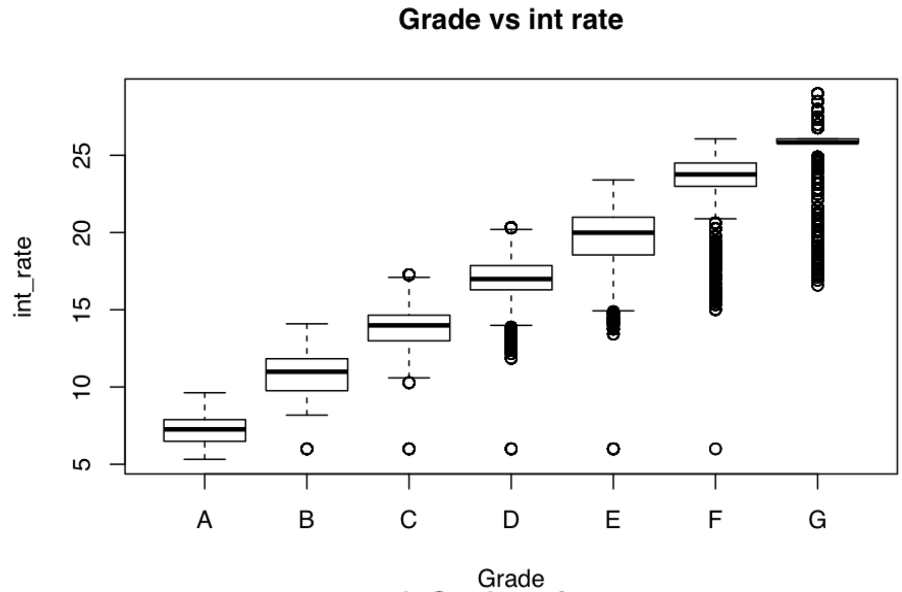
```
## [1] 0.008414386
```

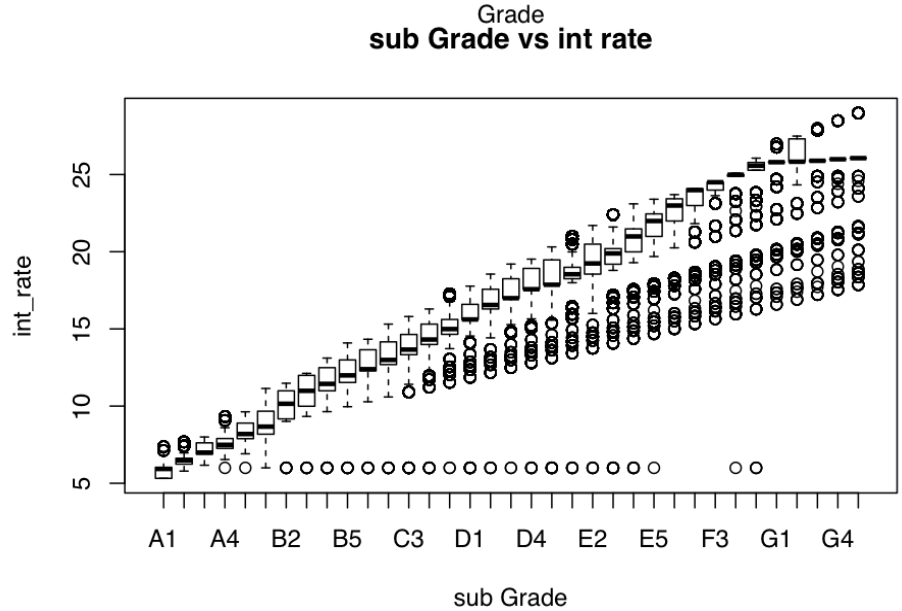
Recomendation

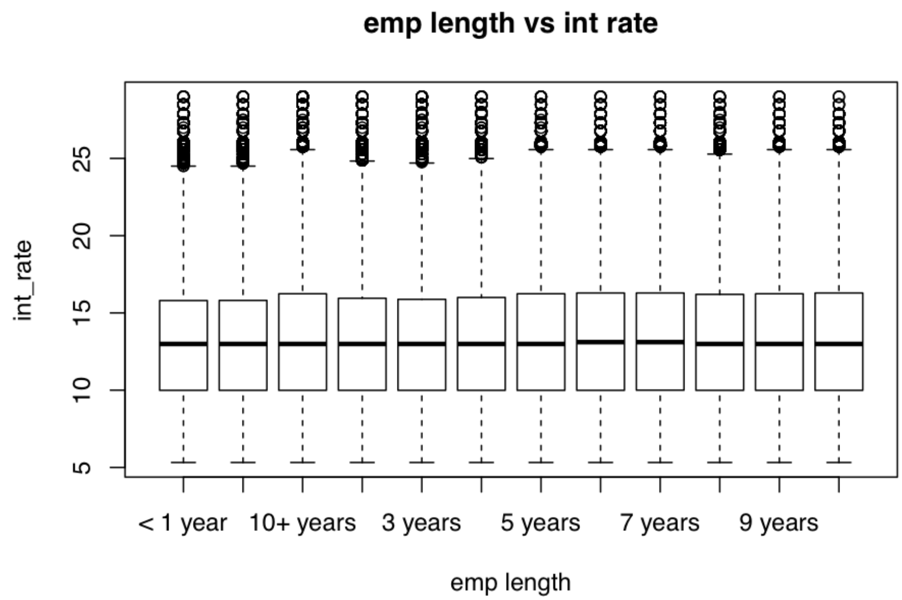
In general, user show pick the 36 month term instead of 60, and and try to lower loan amount for each loan. (If you need a big amount of money try to break it down)

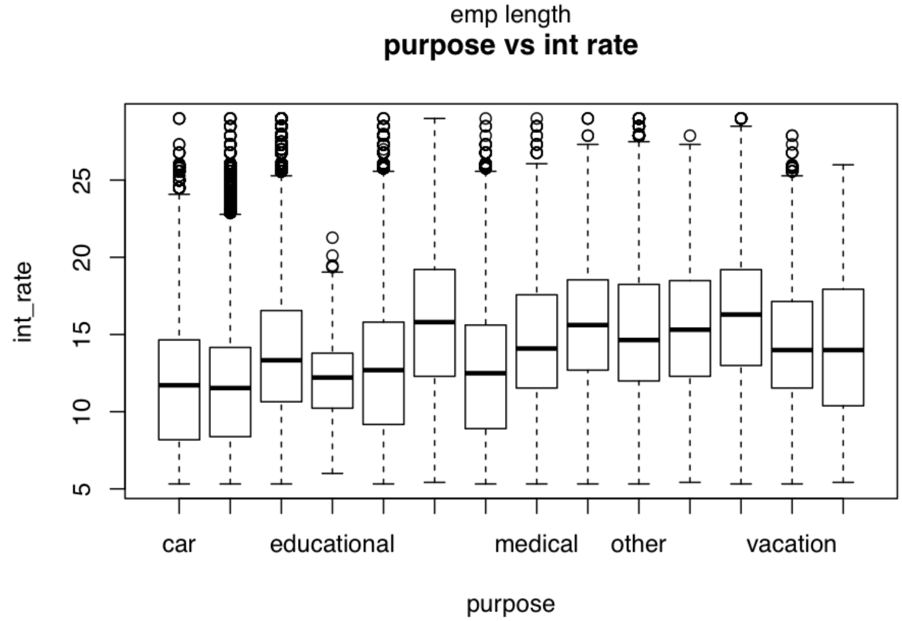
Appendix

The grade has a strong relationship with the int rate





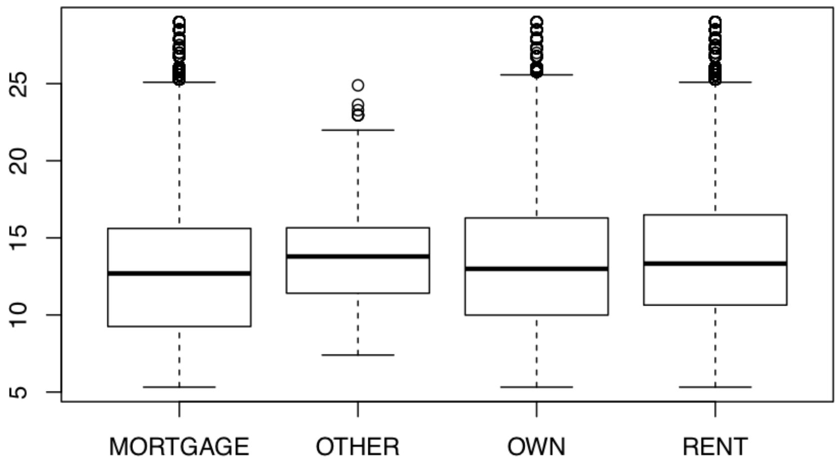




It is expected that he grade and int rate follows a highly linear relationship since both int rate and grade are assigned by the lending club, it doesn't suprise me that they are highly corrlated.

Home ownership

```
loan$home_ownership<- ifelse(loan$home_ownership %in% c('ANY', 'NONE', 'OTHER'),'OTHER', loan$home_owne:
boxplot(int_rate~home_ownership,loan)
```



Good news, not that obvious, we see a slightly higher in the Other catagoary. So the recomendation here is don't leave it blank.

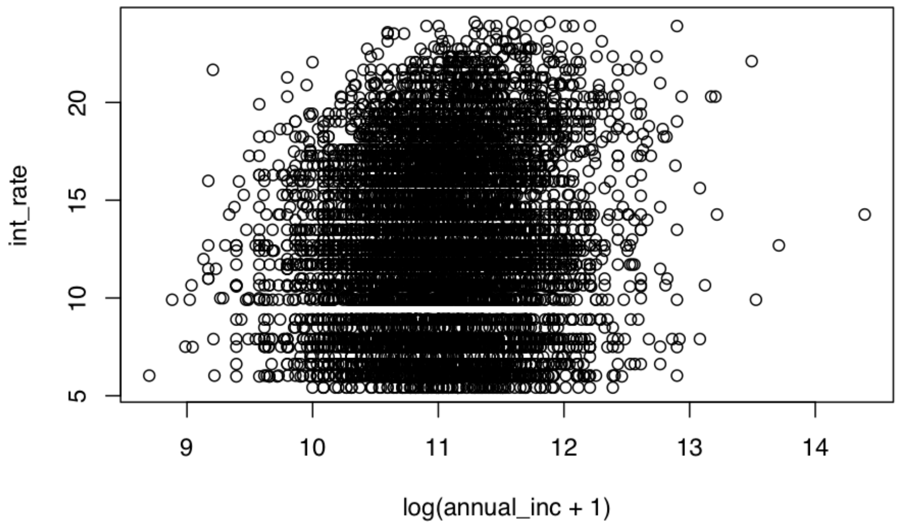
Application type: t test regarding application type.

```
t.test(int_rate ~ application_type, data = loan)
```

```
##
##  Welch Two Sample t-test
##
## data:  int_rate by application_type
## t = -10.1387, df = 510.612, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.299387 -1.552913
## sample estimates:
## mean in group INDIVIDUAL      mean in group JOINT
##                 13.24563                 15.17178
```

Althouth the t score is significent, but it might not provide causality, because the ones that need a joint account might not able to apply for one him self.

```
with(loan[1:10000, ], plot(log(annual_inc + 1), int_rate))
```



Here I want to check what's the percentage that people don't pay off the loan. I hope the recommendation system doesn't hurt the investor side of the story.

```
loan$loan_status <- gsub('Does not meet the credit policy. Status:',
                         '', loan$loan_status)
sort(table(loan$loan_status))
```

```
##
##            Default  Late (16-30 days)    In Grace Period
##               1219               2357               6253
##             Issued Late (31-120 days)        Charged Off
##               8460              11591              46009
##         Fully Paid            Current
##             209711             601779
```

```
loan$loan_status_1 <- with(loan, ifelse(loan_status %in% c('Current', 'Fully Paid', 'Issued'),
                                        1, 0))
table(loan$loan_status_1)
```

```
##
##      0      1
##  67429 819950
```