Data Science Challenge at Brightside

Haozhe Xu

Contents

1 Problem Description **1**

2 Data Description **1**

3 Data Pre-processing **1**
3.1 Transformation on rate, term, em_length, loan_type(purpose), verification 1
3.2 Data Cleaning 1
3.3 Correlation Among Attributes 2

4 Algorithm Executions (Classification) **3**
4.1 Decision Tree 3
4.2 Naive Bayes 3
4.3 Random Forest 3

5 Algorithm Result and Interpretation **3**
5.1 Decision Tree 3
5.2 Naive Bayes 4
5.3 Random Forest 4

6 Model Evaluation **4**
6.1 Conclusion 4

7 Algorithm Executions (Regression) **5**
7.1 Linear Regression 5

8 Model Evaluation **7**
8.1 Conclusion 7

9 Application **8**
9.1 Random Forest Model 8

10 Comments **8**

11 Appendix **8**
11.1 References 8

12 R code **8**

1 Problem Description

We have 2 years' worth of lending Club loan data and our goal is to inform investors on the best loans. Thus, I will use the predictive model algorithms to build a model that informs the user which loans they should invest in.

2 Data Description

our original dataset came from the Lending Club loan, which named " 2016Q1.csv". This dataset has around 13.3 thousand rows and contains 109 variables. And I only choose the 10 attributes. I tried to create a model to help users to invest in; therefore, only the attribute that users may care about before they decided to invest will be useful. Thous attributes like 'fund', 'term','rate','emp_length','verification', 'delinq_2yrs','fico_range_low','pct_tl_nvr_dlq','pub_rec_bankruptcies','purpose'.

3 Data Pre-processing

3.1 Transformation on rate, term, em_length, loan_type(purpose), verification

I transformed 4 attributes from string values to numerical values. They are rate, term, em_length, and verification. I also created a new attribute named 'loan_type' which transformed by purpose. Because purpose contains too many types and it will be very hard to predict, I just pick the most frequent type such as "credit_card", "debt_consolidation", "home_improvement", "major_purchase", then I set other types as "others".

3.2 Data Cleaning

The main dataset contains many missing values and these missing values as shown as no value or NA value. I used R functions to remove the missing values. the deletion reduces the total rows to 12.4 thousand rows.

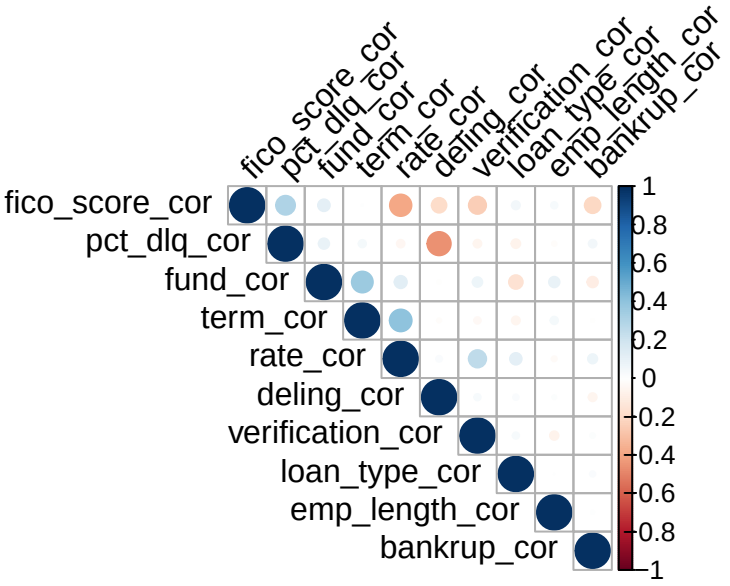
```
##       fund            term            rate         emp_length     verification
##  Min.   : 1000   Min.   :36.00   Min.   : 5.32   Min.   : 1.000   0:42158
##  1st Qu.: 9000   1st Qu.:36.00   1st Qu.: 8.49   1st Qu.: 3.000   1:82785
##  Median :15000   Median :36.00   Median :11.99   Median : 6.000
##  Mean   :15823   Mean   :42.95   Mean   :12.47   Mean   : 6.168
##  3rd Qu.:21000   3rd Qu.:60.00   3rd Qu.:15.31   3rd Qu.:10.000
##  Max.   :40000   Max.   :60.00   Max.   :28.99   Max.   :10.000
##      deling          fico_score       pct_dlq          bankrup
##  Min.   : 0.0000   Min.   :660.0   Min.   :  0.00   Min.   :0.0000
##  1st Qu.: 0.0000   1st Qu.:670.0   1st Qu.: 91.00   1st Qu.:0.0000
##  Median : 0.0000   Median :690.0   Median : 97.40   Median :0.0000
##  Mean   : 0.3482   Mean   :695.6   Mean   : 94.03   Mean   :0.1262
##  3rd Qu.: 0.0000   3rd Qu.:710.0   3rd Qu.:100.00   3rd Qu.:0.0000
##  Max.   :22.0000   Max.   :845.0   Max.   :100.00   Max.   :9.0000
##               loan_type
##  credit_card       :29973
##  debt_consolidation:71790
##  home_improvement  : 7967
##  major_purchase    : 2737
##  others            :12476
##
```

And the types of attributes are

```
## 'data.frame':    124943 obs. of  10 variables:
##  $ fund        : int  8400 12000 28000 10000 20000 13625 9000 15000 18000 12000 ...
##  $ term        : num  36 36 36 36 36 60 36 60 36 60 ...
##  $ rate        : num  9.75 7.89 7.39 13.67 11.99 ...
##  $ emp_length  : num  2 3 1 10 10 1 9 1 10 10 ...
##  $ verification: Factor w/ 2 levels "0","1": 1 2 2 2 2 1 2 1 1 2 ...
##  $ deling      : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ fico_score  : int  670 700 790 705 675 745 675 675 700 660 ...
##  $ pct_dlq     : num  86.4 93.7 100 75 82.1 100 94.7 100 94.7 84.8 ...
##  $ bankrup     : int  0 0 0 1 0 0 1 0 0 0 ...
##  $ loan_type   : Factor w/ 5 levels "credit_card",..: 2 1 2 5 2 1 2 2 2 1 ...
```

3.3 Correlation Among Attributes

After the data transformation and data cleaning, I performed a correlation analysis in R to examine the relationship among attributes. As see in the below chart, blue colors represent positive correlation; red colors represent the negative correlation. From looking at the correlations with "loan_type" (DV), we may see "fund", "term","pct_dlq_cor" have negative correlation with "loan_type".



4 Algorithm Executions (Classification)

4.1 Decision Tree

In the decision tree, I set all the IV variables as predictors variables and put them into the decision tree function (C5.0.default). In the IV variables, "verification" is a binary variable and others are numeric value.

4.2 Naive Bayes

I use the naive Bayes() function to train our dataset and calculate the accuracy

4.3 Random Forest

The third algorithm I use is the Random Forest. Random Forest is a powerful machine learning algorithm. This algorithm creates multiple decision trees and then combining the output generated by each of the decision trees.
Random Forest works on the same underlying principle as Decision Trees. However, it does not select all the data points and variables in each of the trees. It randomly samples data points and variables in each of the trees that it creates and then combines the output at the end. Compared with the decision tree algorithm, the random forest generally has higher accuracy.

5 Algorithm Result and Interpretation

5.1 Decision Tree

After training my model, I predict the test value and compute the test set accuracy. based on the summary of the decision tree model, we can see that in the attribute usage part, "fund", "rate" and "fico_score" attributes are the most impactful variable.

The confusion matrix shows that accuracy is around 55%

5.2 Naive Bayes

After training my model, I predict the test value and compute the test set accuracy

Based on the confusion matrix, we can get the accuracy is around 54%.

5.3 Random Forest

After training my model, I predict the test value and compute the test set accuracy

I use the random forest model(mtry=8, ntree=500) to test my test dataset. the accuracy is around 56%.

6 Model Evaluation

6.1 Conclusion

After performing the Random Forest, Naive Bayes, and Decision tree, I get the following results.

Since the result of the three algorithms are very similar. other performance measures need to calculate and compare with the precision, recall, and F-score, because my model is a multi-class classification model, thus, based on the confusion matrix, I need to calculate precision and recall of each level, then find their average and calculate the F-score. And I use the Decision tree model as an example

Table 1: Classifiaction Result

Algorithm	Accuracy	Opinion
Decision Tree	55%	default
Naive Bayes	54%	default
Random Forest)	56%	mtry=8, ntree=500

$$P(debt) = \frac{12919}{12919 + 945 + 331 + 38 + 108} = 0.903 \tag{1}$$

$$P(credit) = \frac{643}{5137 + 643 + 138 + 15 + 64} = 0.107 \tag{2}$$

$$P(others) = \frac{291}{2011 + 145 + 291 + 14 + 36} = 0.117 \tag{3}$$

$$P(major) = 0 \tag{4}$$

$$P(home) = \frac{31}{1420 + 114 + 64 + 11 + 31} = 0.019 \tag{5}$$

$$P(average) = \frac{0.903 + 0.107 + 0.117 + 0.019}{5} = 0.229 \tag{6}$$

Thus, the average of precision is 0.229

$$R(debt) = \frac{12919}{12919 + 5137 + 2011 + 454 + 1420} = 0.589 \tag{7}$$

$$R(credit) = \frac{643}{945 + 643 + 145 + 34 + 114} = 0.342 \tag{8}$$

$$R(others) = \frac{291}{301 + 138 + 291 + 42 + 64} = 0.348 \tag{9}$$

$$R(major) = 0 \tag{10}$$

$$R(home) = \frac{31}{109 + 64 + 36 + 13 + 31} = 0.123 \tag{11}$$

$$R(average) = \frac{0.589 + 0.342 + 0.348 + 0.123}{5} = 0.28 \tag{12}$$

Thus, the average of recall is 0.28
Then we can calcaulte the F-score:

$$F - score = \frac{2 * 0.28 * 0.229}{0.28 + 0.229} = 0.252 \tag{13}$$

Thus, the average of F-score is 0.28
So, we can get:

Table 2: Evaluation Result

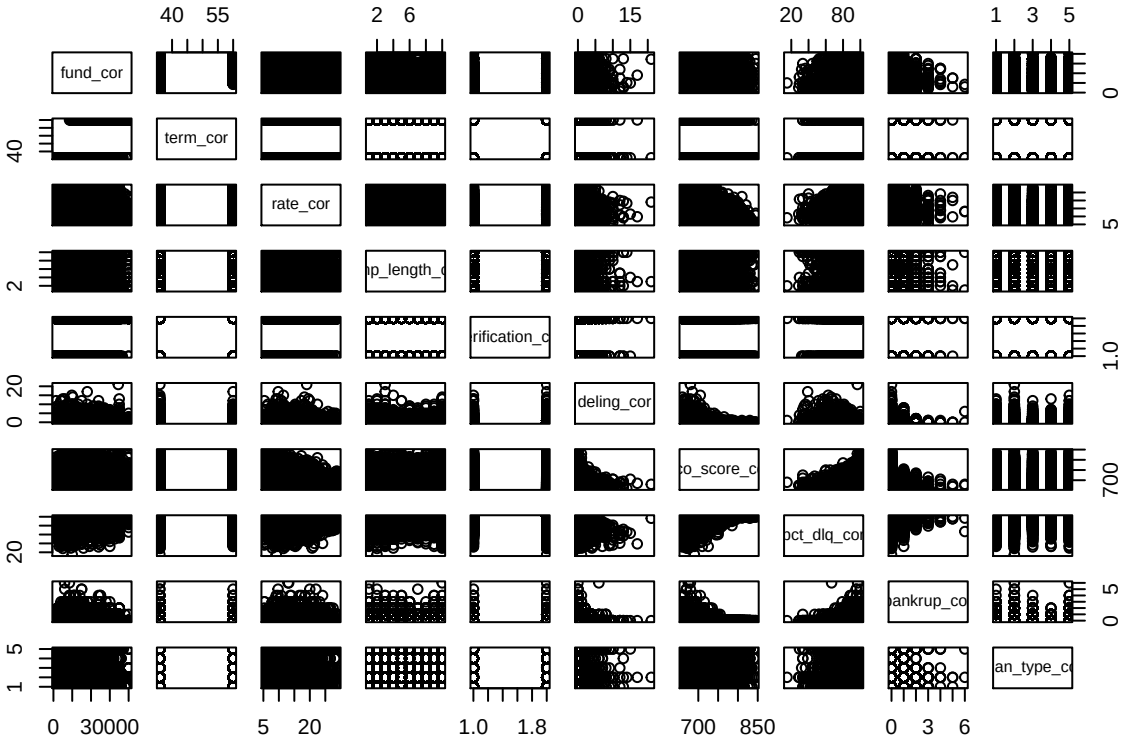
Algorithm	Precision	Recall	F-score
Decision Tree	0.229	0.28	0.252
Naive Bayes	0.225	0.279	0.249
Random Forest	0.295	0.229	0.258

According to the table, we can conclude that with the ntree=500 and ntry=8, we can get the random forest model is my best model, which can help us to predict the loan type before we want to invest in.

7 Algorithm Executions (Regression)

7.1 Linear Regression

if our clients have already known the loan type that they want to invest, and they only want to know how much money they can get at the end. Then we can also use a regression method to build a new model.
I will use the same attributes, data transformation, and data cleaning, but at this time, I will treat "rate" as my dependent variables and other variables will be my independent variables. Before I form the linear regression, I need to plot the data first, because I need to check the predictor variables for any concerns with multicollinearity.



Based on this plot, it's really hard to answer this question. Thus, I use another method to check and it's called "Variance inflation factor" (VIF). I calculate the mean of VIF value if its value is far larger than 6.0, and there is a clear problem with multicollinearity here.

I get the mean of VIF value is 1.17 which is smaller than 6.0, so my predictor variables don't have a multicollinearity problem here. Then I will form a multiple- linear regression model.
Firstly, I use all the attributes to form the full linear regression model.

```
##
## Call:
## lm(formula = rate ~ fund + term + as.numeric(loan_type) + emp_length +
##     as.numeric(verification) + deling + fico_score + pct_dlq +
##     bankrup, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.5727  -2.7332  -0.6129   2.1980  19.2038
```

```
##
## Coefficients:
##                             Estimate Std. Error  t value Pr(>|t|)
## (Intercept)                3.824e+01  2.878e-01  132.868  < 2e-16 ***
## fund                       1.598e-05  1.364e-06   11.717  < 2e-16 ***
## term                       1.787e-01  1.077e-03  165.937  < 2e-16 ***
## as.numeric(loan_type)      6.844e-01  9.862e-03   69.396  < 2e-16 ***
## emp_length                -2.501e-02  3.043e-03   -8.220  < 2e-16 ***
## as.numeric(verification)   1.711e+00  2.403e-02   71.218  < 2e-16 ***
## deling                    -9.886e-02  1.338e-02   -7.391 1.46e-13 ***
## fico_score                -5.843e-02  3.903e-04 -149.679  < 2e-16 ***
## pct_dlq                    3.003e-02  1.448e-03   20.746  < 2e-16 ***
## bankrup                   -1.428e-01  2.957e-02   -4.828 1.38e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.831 on 124933 degrees of freedom
## Multiple R-squared:  0.3694, Adjusted R-squared:  0.3693
## F-statistic:  8131 on 9 and 124933 DF,  p-value: < 2.2e-16
```

Based on the summary of my linear regression model, we can get the expression of my model:

$$\begin{aligned} rate = 38.24 + (1.598e-05) * fund + 0.17878 * term + 0.6844 * loantype - \\ 0.025 * emplength + 1.711 * verification - 0.09886 * deling \\ -0.058 * ficoscore + 0.03 * pctdlq - 0.1428 * bankrup \end{aligned} \tag{14}$$

Secondly, I use the backward elimination with BIC control feature selection method to reduce my model.

```
##
## Call:
## lm(formula = rate ~ fund + as.numeric(loan_type) + emp_length +
##     term, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.487  -3.227  -0.612   2.704  18.674
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               3.742e+00  6.030e-02  62.062   <2e-16 ***
## fund                      1.298e-07  1.518e-06   0.085    0.932
## as.numeric(loan_type)     5.820e-01  1.114e-02  52.247   <2e-16 ***
## emp_length               -5.513e-02  3.458e-03 -15.944   <2e-16 ***
## term                      1.817e-01  1.221e-03 148.784   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.367 on 124938 degrees of freedom
## Multiple R-squared:  0.1805, Adjusted R-squared:  0.1805
## F-statistic:  6881 on 4 and 124938 DF,  p-value: < 2.2e-16
```

Based on the summary of my linear regression model, we can get the expression of my model:

$$rate = 3.742 + (1.298e-07) * fund + 0.58 * loantype - 0.05 * emplength + 0.18 * term \tag{15}$$

8 Model Evaluation

8.1 Conclusion

I will use the Coefficient of determination (R^2) value to evaluate this model. If the R^2 is closer to 1, which means that our model fits our data. At this time, my first linear regression model R^2 value is 0.37, which means that my model is not very perfect. But my second linear regression model R^2 value is 0.18, which is smaller than the first model. According to the Coefficient of determination, the full linear regression model is the better model.

9 Application

9.1 Random Forest Model

If we have an aggressive client who only has 1000 dollars and he/she liked 60 payments on the loan, and she/he wants the highest interest rate which is 28.99%. and the ower of this loan only need to have 1 years employment; this loan is not necessary to verify; there have 10 times past-due incidences of delinquency for the past 2 years, and the fico score can be 660 points; the percent of trades never delinquent is 90%; the number of public record bankruptcies is 8. Then we can put these data as predictor variable into my random forest model, the model predicts that she/he should choose the loan type is "debt_consolidation". So we can recommend her/him to invest in this type of loan.
if we have a conservative client who only has 1000 dollars and he/she liked 60 payments on the loan, and she/he wants the highest interest rate which is 15.32%. and the ower of this loan only need to have 10 years employment; this loan is necessary to verify; there have 0 times past-due incidences of delinquency for the past 2 years, and the fico score can be 790 points; the percent of trades never delinquent is 100%; the number of public record bankruptcies is 0. Then we can put these data as predictor variable into my random forest model, the model predicts that she/he should choose the loan type is "others". So we can recommend her/him to invest in this type of loan.

10 Comments

There are strengths and weaknesses in my project. The strengths include two types of models: Classification and Regression model. It can solve my business problem in two ways. one is to predict the loan type, another one is to predict the rate.
The weakness of my project is that due to the technique computing device limited, I only can handle one of the datasets. Because my dataset is small, my model may not have enough "materials" to properly learn. Secondly, my classification model's accuracy is not very high and my regression model's coefficient of determination is also not very closer to 1.

11 Appendix

11.1 References

1.) Bnebeker. (n.d.). bnebeker/data_challenge. Retrieved from https://github.com/bnebeker/data_challenge

12 R code

```
knitr::opts_chunk$set(echo = FALSE, warning=FALSE, message=FALSE)
library(C50)
library(e1071)
library(caret)
library(randomForest)
library(car)
library(corrplot)
library(leaps)
setwd("/Users/haozhexu/Desktop/Brightside")
Q1_2016 <- read.csv('2016Q1.csv',na.strings = c("", "NA"))
ALL_data <- Q1_2016

# feature selection
loan_type <- as.character(ALL_data[,14])
fund <- ALL_data[,2]
term <- as.character(ALL_data[,3])
rate <- as.character(ALL_data[,4])
emp_length<- as.character(ALL_data[,8])
verification <- as.character(ALL_data[,11])
deling <- ALL_data[,19]
fico_score <- ALL_data[,21]
pct_dlq <- ALL_data[,89]
bankrup <- ALL_data[,91]

# form the target dataset
loan_data <- cbind(fund,term,rate,emp_length,verification,deling,fico_score,pct_dlq,bankrup,loan_type)
loan_data[is.null(loan_data)]<-""
write.csv(loan_data,"loans_data.csv")
# read the dataset
data <- read.csv("loans_data.csv",na.strings = "NA",stringsAsFactors=FALSE)
data <- data[complete.cases(data),]
# data transformation
data$rate <- as.numeric(gsub("[\\%,]","",data$rate))
data$term <- as.numeric(gsub("[\\months,]","",data$term))
data$emp_length <- gsub("[\\years,]","",data$emp_length)
data$emp_length <- gsub("[\\+,]","",data$emp_length)
data$emp_length <- as.numeric(gsub("[\\<,]","",data$emp_length))
data <- data[2:11]

for (i in 1:length(data$loan_type)) {
  if(data$loan_type[i]=='debt_consolidation')
    data$loan_type[i]='debt_consolidation'
  else if(data$loan_type[i]=='credit_card')
    data$loan_type[i]='credit_card'
  else if(data$loan_type[i]=='home_improvement')
    data$loan_type[i]='home_improvement'
  else if(data$loan_type[i]=='major_purchase')
    data$loan_type[i]='major_purchase'
  else
    data$loan_type[i]='others'
}
```

```
for (i in 1:length(data$verification)) {
  if(data$verification[i]=='Source Verified'|data$verification[i]=='Verified')
    data$verification[i]=1
  else
    data$verification[i]=0
}
data$loan_type<-as.factor(data$loan_type)
data$verification <- as.factor(data$verification)
# data cleaning
data[ data == "?"] <- NA
colSums(is.na(data))
data <- data[!(data$emp_length %in% c(NA)),]
summary(data)
str(data)
# check the correlation among attributes
fund_cor <- data$fund
term_cor <- data$term
rate_cor <- data$rate
emp_length_cor <- data$emp_length
verification_cor <- as.numeric(data$verification)
deling_cor <- data$deling
fico_score_cor <- data$fico_score
pct_dlq_cor <- data$pct_dlq
bankrup_cor<- data$bankrup
loan_type_cor <- as.numeric(data$loan_type)
data_cor <- data.frame(fund_cor,term_cor,rate_cor,emp_length_cor,verification_cor,deling_cor,fico_score
r <- cor(data_cor)
corrplot(r,type = "upper",order = "hclust", tl.col = "black",tl.srt = 45)
# Split data into train and test dataset
colnames(data)<- c('fund','term','rate','emp_length','verification','deling','fico_score','pct_dlq','ba
sample_size <- floor(0.8 * nrow(data))
training_index <- sample(seq_len(nrow(data)),size = sample_size)
train <- data[training_index,]
test <- data[-training_index,]
# Decision tree
predictors <- c('fund','term','rate','emp_length','verification','deling','fico_score','pct_dlq','bankr
model_ds <- C5.0.default(x=train[,predictors],y=train$loan_type)
pred <- predict(model_ds,newdata = test)
u_ds <- union(pred,test$loan_type)
t_ds <- table(factor(pred,u_ds),factor(test$loan_type,u_ds))
print(confusionMatrix(t_ds))
# Naive Bayes
model_NB <- naiveBayes(loan_type~., data = train)
NB.pred <- predict(model_NB,test)
u <- union(NB.pred,test$loan_type)
t <- table(factor(NB.pred,u),factor(test$loan_type,u))
print(confusionMatrix(t))
# Random Forest
model_rf <- randomForest(loan_type ~., data = train,ntree=500,ntry=8)
pred <- predict(model_rf, newdata=test[-10])
cm <- table(test[,10], pred)
confusionMatrix(cm)
# plot the sample data
```

```
sample <- floor(0.2 * nrow(data))
sample_l <- sample(seq_len(nrow(data)),size = sample)
linear_data <- data[sample_l,]
fund_cor <- linear_data$fund
term_cor <- linear_data$term
rate_cor <- linear_data$rate
emp_length_cor <- linear_data$emp_length
verification_cor <- as.numeric(linear_data$verification)
deling_cor <- linear_data$deling
fico_score_cor <- linear_data$fico_score
pct_dlq_cor <- linear_data$pct_dlq
bankrup_cor<- linear_data$bankrup
loan_type_cor <- as.numeric(linear_data$loan_type)
data_linear <- data.frame(fund_cor,term_cor,rate_cor,
                          emp_length_cor,verification_cor,deling_cor,fico_score_cor,pct_dlq_cor,bankrup
pairs(data_linear)
# check the multicollearity
mean(vif(lm(rate ~ fund+term+as.numeric(loan_type)+emp_length+
              as.numeric(verification)+deling+fico_score+pct_dlq+bankrup,data = data)))
linear_model<-lm(rate ~ fund+term+as.numeric(loan_type)+
                   emp_length+as.numeric(verification)+deling+fico_score+pct_dlq+bankrup,data = data)
summary(linear_model)
# backward elimination with BIC control
n<- length(rate)
step(linear_model,direction = "backward",k=log(n))
linear_model_adj<-lm(rate ~ fund+as.numeric(loan_type)+emp_length+term,data = data)
summary(linear_model_adj)
# Application
# a  Aggressive client
# fund=1000,term=60 months,rate=28.99%,
# emp_length=1,verification=0,deling=10,fico_score=660,pct_dlq=95,bankrup=8,'loan_type'
newdat <- test
newdat_app_1 <- data.frame(fund=1000,term=60,rate=28.99,emp_length=1,
                           verification=0,deling=10,fico_score=660,pct_dlq=90,bankrup=8,loan_type='NA')
newdat <- rbind(newdat,newdat_app_1)
pred.rf_app <- predict(model_rf,newdata = newdat[-10])
pred.rf_app[24990]
# a  Conservative client
# fund=12000,term=36 months,rate=7.88%,
# emp_length=10,verification=1,deling=0,fico_score=790,pct_dlq=100,bankrup=0,'loan_type'
newdat <- test
newdat_app_2 <- data.frame(fund=1000,term=60,rate=15.32,
                           emp_length=10,verification=1,deling=0,fico_score=790,pct_dlq=100,bankrup=0,l
newdat <- rbind(newdat,newdat_app_2)
pred.rf_app <- predict(model_rf,newdata = newdat[-10])
pred.rf_app[24990]
```