Regression Model

Niharika D,Zachariah Alex,Rachanna Kurra,Tilak Kumar Bonala

2023-05-04

#Loading Libraries

```
library(caret)

library(dplyr)

library(corrplot)

library(glmnet)

library(tidyverse)

library(tidyr)

library(randomForest)
```

Reading the already cleaned data file

```
data<-read.csv('data_cleaned.csv')

# Create a new column called 'default' with a value of 1 is loss is above 0 and 0 is loss is 0


data$default <- ifelse(data$loss == 0, 0, 1)
data$default<- as.factor(data$default)


#Normalizing loss column by dividing with 100
data$loss <- (data$loss/100)


#Creating subset of customers who have defaulted (i.e loss > 0)

default_customers<- subset(data, data$default == 1)
```

Create a preprocessing model that eliminates near zero variance variables, highly correlated variables, and then does the imputation of missing values with the median

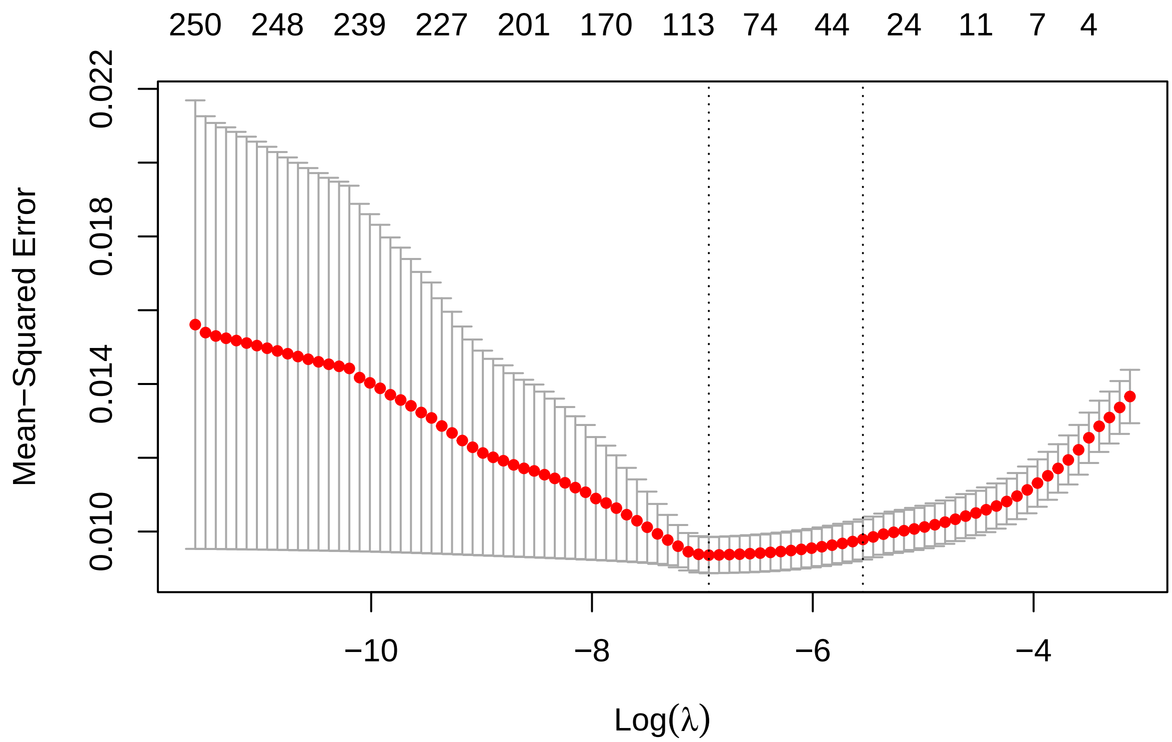
```
data1<-select(default_customers,-c(f736,f764))

preProcessModel <- preProcess(data1[,-c(701,702)], method = c("nzv", "corr", "medianImpute"))
Preprocessed_default <- predict(preProcessModel, data1)
```

Feature selection for regression(loss) using Lasso

```
set.seed(3456)

X1 <- as.matrix(Preprocessed_default[ ,-c(258,259)])
Y1 <- as.vector(Preprocessed_default$loss)


lasso_model <- cv.glmnet(X1, Y1, alpha = 1, family = "gaussian", nfolds = 10, type.measure = "mse")


summary(lasso_model)
```

```
##            Length Class  Mode
## lambda     92     -none- numeric
## cvm        92     -none- numeric
## cvsd       92     -none- numeric
## cvup       92     -none- numeric
## cvlo       92     -none- numeric
## nzero      92     -none- numeric
## call        7     -none- call
## name        1     -none- character
## glmnet.fit 12     elnet  list
## lambda.min  1     -none- numeric
## lambda.1se  1     -none- numeric
## index       2     -none- numeric
```

```
plot(lasso_model)
```




```r
#Finding the minimum value of lambda

lasso_model$lambda.min
```

```
## [1] 0.0009666836
```

```r
#Finding the coefficients at minimum lambda value

cv_lasso_coefs <- coef(lasso_model, s = "lambda.min")
cv_lasso_coefs
```

```
## 258 x 1 sparse Matrix of class "dgCMatrix"
##                          s1
## (Intercept)   2.108013e-01
## id            .
## f1            .
## f3            1.543983e-04
## f5            .
## f6            .
## f13           6.280944e-03
## f16           .
## f19          -1.270289e-03
## f25           .
## f26          -3.140426e-10
## f29           .
```

```
## f31         -3.358549e-06
## f32          .
## f43          .
## f44         -4.504861e-04
## f47         -8.786071e-03
## f54         -3.201996e-04
## f57         -2.153045e-02
## f64         -6.446780e-03
## f65          .
## f66          .
## f67          1.119668e-02
## f70         -5.121977e-02
## f71          7.613265e-03
## f73          .
## f76          6.923281e-04
## f80          .
## f81          2.525327e-03
## f82          .
## f90          5.656881e-03
## f92         -4.336391e-04
## f94          .
## f99          .
## f100         .
## f102         .
## f104         .
## f109         .
## f110        -1.014860e-02
## f112        -5.843114e-03
## f121         4.319189e-03
## f122         .
## f124        -7.741173e-02
## f129        -7.258677e-02
## f130         3.953401e-03
## f131         3.410484e-03
## f132         .
## f133         .
## f139         .
## f140        -2.473592e-03
## f143         3.161893e-04
## f144         9.774693e-04
## f146         .
## f148         .
## f149         .
## f150         .
## f151         6.678316e-03
## f153         9.388189e-03
## f158         .
## f159         1.961165e-04
## f161        -8.293225e-03
## f163        -4.565523e-03
## f168         .
## f170         .
## f171         .
## f173         .
```

```
## f178          .
## f180          .
## f181          .
## f183          .
## f188          .
## f189         -1.529752e-03
## f190          .
## f191          .
## f193          .
## f198         -1.298838e-02
## f199         -1.351648e-03
## f200          .
## f202          .
## f203          .
## f204          .
## f208          .
## f209          .
## f212          5.446143e-04
## f213          1.312700e-03
## f217          .
## f218          .
## f220          .
## f221          .
## f223          .
## f229          5.069314e-02
## f231          .
## f233          .
## f238          .
## f239          .
## f241         -5.023047e-03
## f243          .
## f248          .
## f249          .
## f251          .
## f259          .
## f261         -5.183299e-03
## f268         -1.209140e-01
## f269          .
## f270          5.675407e-02
## f272          .
## f277          .
## f278          .
## f280          .
## f281          2.080290e-03
## f287         -8.826718e-11
## f288         -5.894110e-04
## f289          .
## f314          1.000207e-02
## f316          8.962565e-03
## f320          .
## f321          .
## f322          .
## f324          .
## f329         -3.373098e-03
```

```
## f330        -5.865412e-03
## f331         .
## f333         6.805882e-08
## f338         .
## f339         .
## f340         6.007584e-04
## f341        -1.818251e-03
## f357         .
## f358         .
## f361         5.188089e-06
## f366         .
## f367         .
## f374         .
## f378         .
## f382         1.723278e-12
## f383         .
## f384         1.595930e-04
## f385         .
## f391         8.893024e-46
## f393         .
## f398         .
## f402         1.307298e-02
## f403         .
## f411         .
## f412         .
## f413        -5.918445e-03
## f420         .
## f421         .
## f422         .
## f425         .
## f428         .
## f430         .
## f431        -4.599487e-04
## f432         .
## f433         .
## f436         .
## f441         .
## f442         .
## f444         .
## f448         .
## f451         .
## f458         .
## f461         .
## f468         .
## f470         .
## f471         .
## f472         .
## f479        -1.416773e-03
## f489         .
## f499        -1.129048e-03
## f509        -6.585700e-03
## f514        -2.323939e-04
## f516         2.409966e-04
## f518         .
```

```
## f522          4.509724e-03
## f523         -4.711547e-09
## f524          .
## f525          .
## f526         -5.155165e-11
## f530          3.685328e-14
## f533          .
## f536          .
## f546         -8.204474e-02
## f556          8.789095e-03
## f566          .
## f567          .
## f587          .
## f588         -3.708596e-03
## f589          .
## f591          .
## f598         -1.161620e-02
## f600          .
## f601          .
## f609          .
## f611          .
## f612          .
## f613          .
## f614          .
## f618         -7.296978e-05
## f621         -2.808117e-03
## f623          1.054737e-12
## f628          .
## f629          1.834701e-02
## f631          6.318250e-05
## f634          1.467287e-04
## f636          9.610007e-07
## f637         -7.099348e-03
## f638          .
## f639         -6.679340e-05
## f640          .
## f643          .
## f646          .
## f647         -4.018520e-04
## f648         -3.229055e-04
## f649         -5.639699e-05
## f650          .
## f651          .
## f652          1.157557e-05
## f653          2.130604e-05
## f654          1.546527e-04
## f656          .
## f659          .
## f660          .
## f661          .
## f663          6.822372e-05
## f664          .
## f669          1.217865e-04
## f671          2.868147e-02
```

```
## f672         .
## f673         .
## f674        -6.402492e-05
## f675         .
## f677        -4.021237e-04
## f679        -3.169236e-04
## f680         .
## f682         .
## f699         .
## f715         .
## f716         .
## f725         9.112264e-04
## f733         .
## f734        -7.801009e-04
## f735         .
## f739         1.071340e-03
## f740        -3.182254e-05
## f742         .
## f743         .
## f744         .
## f746         1.025915e-03
## f755         .
## f756         .
## f760         .
## f763        -5.493724e-03
## f765         1.619394e-03
## f766         1.442881e-01
## f768         9.313912e-02
## f774        -1.004009e-01
## f775         .
```

```r
#Converting coefficients obtained into a dataframe

cv_lasso_coefs <- data.frame(name = cv_lasso_coefs@Dimnames[[1]][cv_lasso_coefs@i + 1], coefficient = cv

head(cv_lasso_coefs)
```

```
##          name   coefficient
## 1 (Intercept)  2.108013e-01
## 2          f3  1.543983e-04
## 3         f13  6.280944e-03
## 4         f19 -1.270289e-03
## 5         f26 -3.140426e-10
## 6         f31 -3.358549e-06
```

```r
#Removing the intercept from the coefficient data frame

cv_lasso_coefs <- cv_lasso_coefs[-1, ]


#Converting the coefficient data frame to vector

cv_lasso_coefs <- as.vector(cv_lasso_coefs$name)
```

```r
#Adding loss variable back to the vector

cv_lasso_coefs1 <- c(cv_lasso_coefs,"loss")

cv_lasso_coefs1
```

```
##   [1] "f3"   "f13"  "f19"  "f26"  "f31"  "f44"  "f47"  "f54"  "f57"  "f64"
##  [11] "f67"  "f70"  "f71"  "f76"  "f81"  "f90"  "f92"  "f110" "f112" "f121"
##  [21] "f124" "f129" "f130" "f131" "f140" "f143" "f144" "f151" "f153" "f159"
##  [31] "f161" "f163" "f189" "f198" "f199" "f212" "f213" "f229" "f241" "f261"
##  [41] "f268" "f270" "f281" "f287" "f288" "f314" "f316" "f329" "f330" "f333"
##  [51] "f340" "f341" "f361" "f382" "f384" "f391" "f402" "f413" "f431" "f479"
##  [61] "f499" "f509" "f514" "f516" "f522" "f523" "f526" "f530" "f546" "f556"
##  [71] "f588" "f598" "f618" "f621" "f623" "f629" "f631" "f634" "f636" "f637"
##  [81] "f639" "f647" "f648" "f649" "f652" "f653" "f654" "f663" "f669" "f671"
##  [91] "f674" "f677" "f679" "f725" "f734" "f739" "f740" "f746" "f763" "f765"
## [101] "f766" "f768" "f774" "loss"
```

```r
#Combining the columns selected by lasso with variable selection and forming a new dataset

data_new<-select(default_customers,cv_lasso_coefs1)
```

Creating training and test partition with 70% for training and 30% for test

```r
set.seed(6782)

Split_data <- createDataPartition(data_new$loss,p=.7,list=FALSE,times=1)
Training <- data_new[Split_data,]
Validation <- data_new[-Split_data,]
```

Building Bagged Decision Tree model using Random Forest

```r
num_trees <- 100 #number of trees
sample_size <- 50 #size of the bootstrap sample used to grow each tree

# Building the Bagged Decision Tree Regression model

bagged_model <- randomForest(loss ~ ., data = Training,
                             ntree = num_trees,
                             mtry = 10,
                             sampsize = sample_size,
                             replace = TRUE)

summary(bagged_model)
```

```
##                 Length Class  Mode
## call               7   -none- call
## type               1   -none- character
## predicted       5167   -none- numeric
## mse              100   -none- numeric
## rsq              100   -none- numeric
```

```
## oob.times       5167   -none- numeric
## importance       103   -none- numeric
## importanceSD       0   -none- NULL
## localImportance    0   -none- NULL
## proximity          0   -none- NULL
## ntree              1   -none- numeric
## mtry               1   -none- numeric
## forest            11   -none- list
## coefs              0   -none- NULL
## y               5167   -none- numeric
## test               0   -none- NULL
## inbag              0   -none- NULL
## terms              3   terms  call
```

```
Predictions<- predict(bagged_model, Validation)
```

#Calculating MAE for the model

```
MAE<-MAE(Predictions,Validation$loss,na.rm=TRUE)
MAE
```

```
## [1] 0.06240627
```

Reading and preprocessing Test Data

```
data10<-read.csv("new_defaulted_test_customers.csv")

#Replacing null values with zeroes

data11 <- data10 %>% mutate_all(funs(replace_na(.,0)))

null_percent <- apply(data11 == 0, 2, mean)

#Removing columns having more than 30% null values

cols <- names(null_percent[null_percent <= 0.3])

new_test_file <- data11[, cols]


#Check if the columns with more than 30% null values are deleted

Sums<-(colSums(new_test_file==0)/nrow(new_test_file))*100

#Combining the variables from lasso model with test data to obtain a new test data with selected variab

Test_data_new<-select(new_test_file,cv_lasso_coefs)
```

Running the model on test data

```r
Test_loss_Predictions<-predict(bagged_model, Test_data_new)


write.csv(Test_loss_Predictions,file="Final_Predictions.csv")
```