Loan Default Probability Prediction

Jiayi Jiang

2023-04-09

Predicting Probability of Loan Default

Summary of the procedures:

Step 1: The first step is to pre-process the datasets since there are missing values and unbalanced train-test problems. All missing values are from "employment" variable and this is a categorical variable, so I choose to replace those missing values by the most frequent category "10+". In addition, I will convert/encode several binary category variables into 0 and 1 such as "initial_list_status" and "term" columns. Finally, factor all text data with more than 2 distinct values.

Step 2: Feature selection using Lasso with cross-validation, according to optimal lambda lambda.1se, we will choose total 13 variables: "reason", "n_collect", "interest", "initial_list_status", "term", "employment", "status", "quality", "volations", "fees_rec", "v1", "del", "req" because they are the variables with largest absolute value of coefficients after the lasso shrinking penalty. All variables importance plot are also shown below indicating the 13 variables are the most important ones.

Step 3: Train the data using Logistic model on the chosen subset of variables.

Step 4: Predict the raw risk score for the test data

Step 5: Probability calibration, after this step, I build a complete 600 risk score prediction (calibrated) for each customer in the test set. The result is shown below in the array "cali_pred1"

```
library(readr)
library(Hmisc)
```

Step 6: Calculate the MAE error of the prediction model: 0.1141 which is very low as desired!

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

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
## Loading required package: ggplot2

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
loan_train_final <- read_csv("~/Downloads/loan_train_final.csv")
```

```
## Rows: 2400 Columns: 31

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (6): initial_list_status, term, employment, status, reason, quality
## dbl (25): default, n_collect, credit_ratio, interest, recover, coll_fee, out...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
test_loan <- read_csv("~/Downloads/loan_test_final.csv")
```

```
## Rows: 600 Columns: 31
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (6): initial_list_status, term, employment, status, reason, quality
## dbl (25): default, n_collect, credit_ratio, interest, recover, coll_fee, out...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
########################## Step 1: Data Processing ##########################

####### Treat the missing values:
sum(is.na(loan_train_final))
```

```
## [1] 134
```

```
sum(is.na(test_loan))
```

```
## [1] 26
```

```
# describe(loan_train_final)
# describe(test_loan)

# describe(loan_train_final$employment)
# describe(test_loan$employment)

loan_train_final$employment[is.na(loan_train_final$employment)]<- "10+"
```

```
test_loan$employment[is.na(test_loan$employment)]<- "10+"



####### Treat the categorical data:
# summary(loan_train_final)
# summary(test_loan)

# describe(loan_train_final$reason)
# describe(test_loan$reason)

loan_train_final$initial_list_status[loan_train_final$initial_list_status == "a"] <- 0
loan_train_final$initial_list_status[loan_train_final$initial_list_status == "b"] <- 1
loan_train_final$initial_list_status <- as.numeric(loan_train_final$initial_list_status)

test_loan$initial_list_status[test_loan$initial_list_status == "a"] <- 0
test_loan$initial_list_status[test_loan$initial_list_status == "b"] <- 1
test_loan$initial_list_status <- as.numeric(test_loan$initial_list_status)

loan_train_final$term[loan_train_final$term == "3 yrs"] <- 0
loan_train_final$term[loan_train_final$term == "5 yrs"] <- 1
loan_train_final$term <- as.numeric(loan_train_final$term)

test_loan$term[test_loan$term == "3 yrs"] <- 0
test_loan$term[test_loan$term == "5 yrs"] <- 1
test_loan$term <- as.numeric(test_loan$term)

####### Factor the text data:
loan_train_final$reason <- as.factor(loan_train_final$reason)
loan_train_final$employment <- as.factor(loan_train_final$employment)
loan_train_final$status <- as.factor(loan_train_final$status)
loan_train_final$quality <- as.factor(loan_train_final$quality)

test_loan$reason <- as.factor(test_loan$reason)
test_loan$employment <- as.factor(test_loan$employment)
test_loan$status <- as.factor(test_loan$status)
test_loan$quality <- as.factor(test_loan$quality)
```

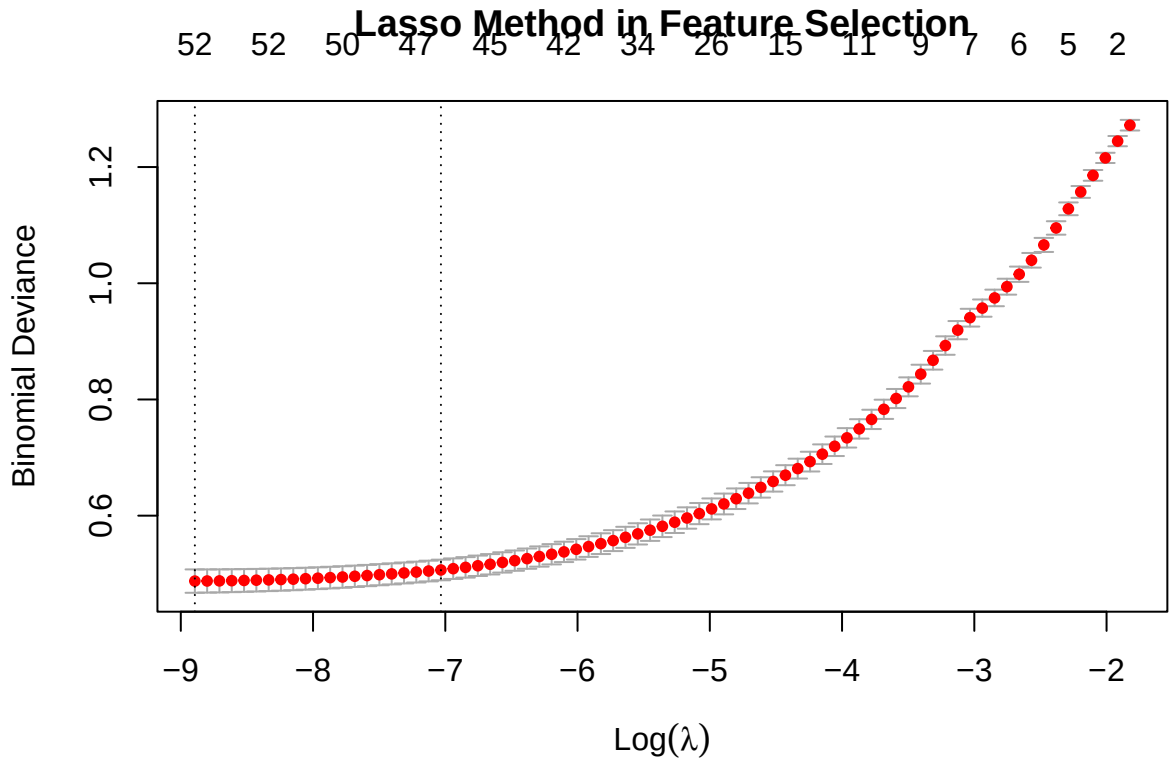
```
######################### Step 2: Feature Selection ##########################

###### Lasso with cross-validation:
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-4
```

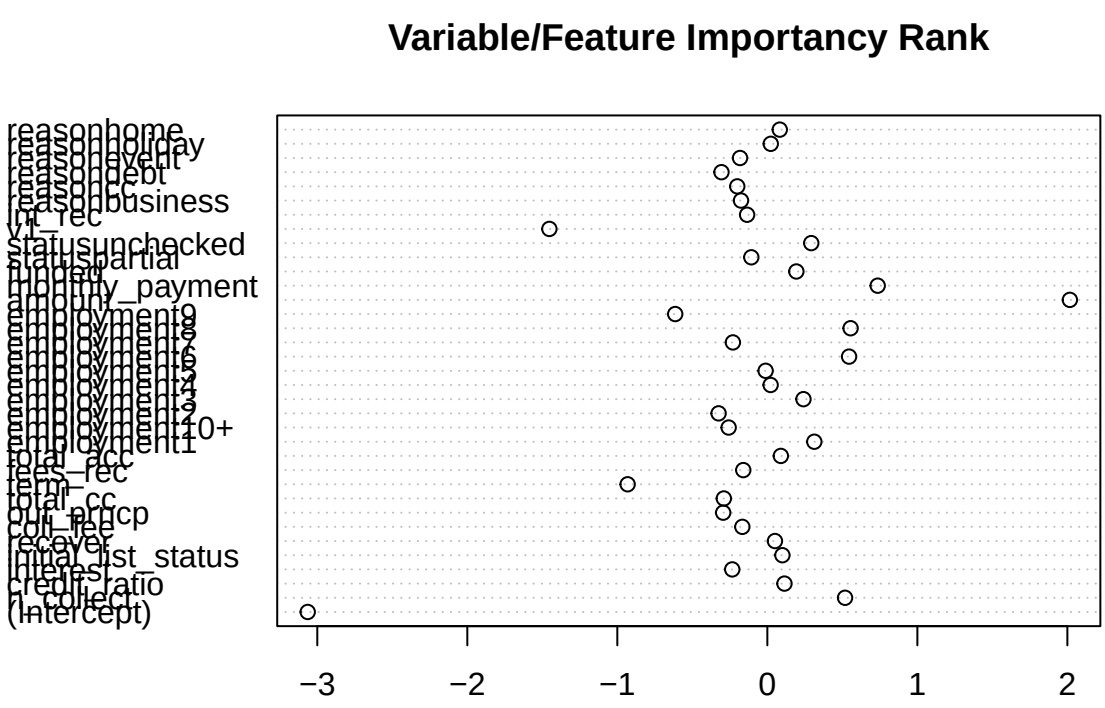
```
x_train <- model.matrix(default~., data = loan_train_final)[,-1]
x_test <- model.matrix(default~., data = test_loan)[,-1]
y_train <- loan_train_final$default
y_test <- test_loan$default
```

```
set.seed(8776)
cv.lasso <- cv.glmnet(x_train, y_train, family = 'binomial')
plot(cv.lasso, main = "Lasso Method in Feature Selection")
```




```
beta <- coef(cv.lasso, s = "lambda.1se")
label <- beta@Dimnames[[1]]
dotchart(beta[abs(beta)>0.01], labels = label, main = "Variable/Feature Importancy Rank")
```

```
## <sparse>[ <logic> ]: .M.sub.i.logical() maybe inefficient
```



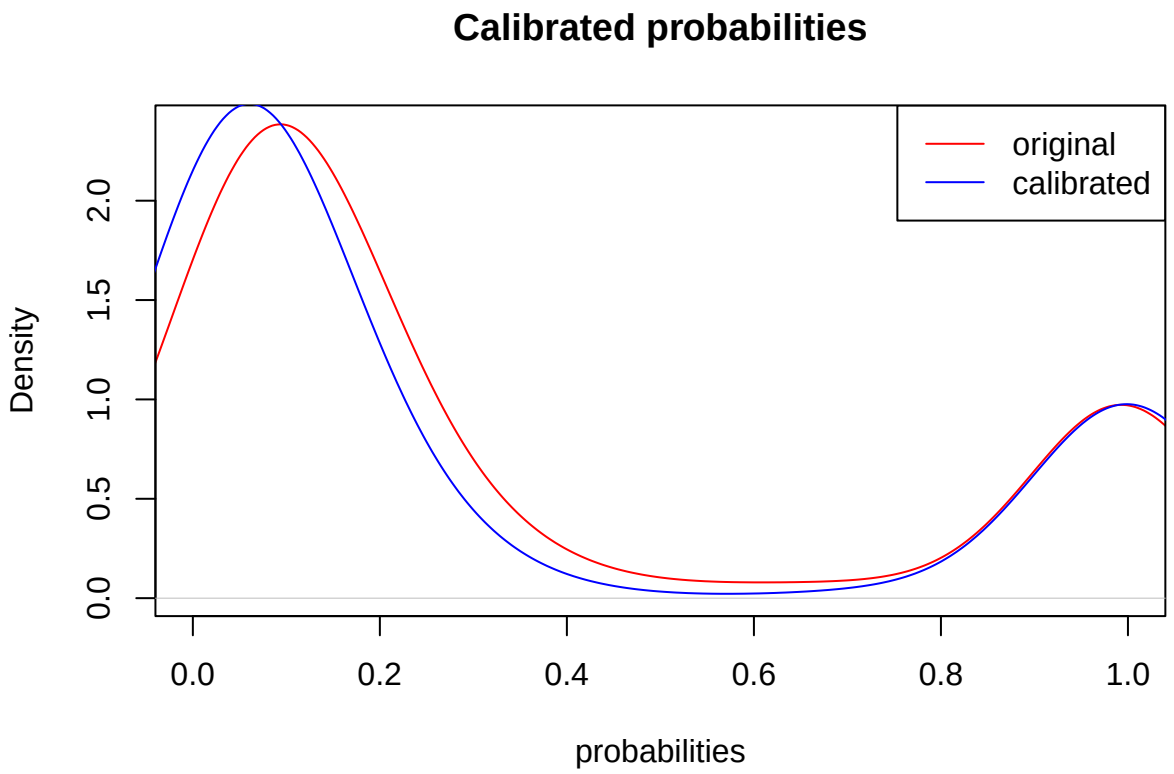

```
######################### Step 3: Train the logistic model ##########################
library(rfUtilities)

formal1 = as.formula(default~n_collect+interest+initial_list_status+term+fees_rec+employment+status+v1+

model1 <- glm(formal1, data = loan_train_final, family = "binomial")
summary(model1)
```

```
##
## Call:
## glm(formula = formal1, family = "binomial", data = loan_train_final)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2361  -0.8354  -0.5835   1.0551   2.3242
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -3.536796   0.481307  -7.348 2.01e-13 ***
## n_collect            -0.192087   0.444808  -0.432 0.665856
## interest              0.356571   0.037054   9.623  < 2e-16 ***
## initial_list_status  -0.481261   0.101057  -4.762 1.91e-06 ***
## term                 -0.186363   0.121651  -1.532 0.125536
## fees_rec              0.060537   0.011644   5.199 2.01e-07 ***
## employment1          -0.450230   0.249291  -1.806 0.070912 .
## employment10+        -0.597130   0.178852  -3.339 0.000842 ***
```

```
## employment2          -0.542846   0.226622  -2.395 0.016603 *
## employment3          -0.620575   0.227581  -2.727 0.006395 **
## employment4          -0.357628   0.241137  -1.483 0.138050
## employment5          -0.489746   0.248731  -1.969 0.048955 *
## employment6          -0.332747   0.260729  -1.276 0.201879
## employment7          -0.370426   0.252368  -1.468 0.142158
## employment8          -0.252181   0.267643  -0.942 0.346076
## employment9          -0.616314   0.280774  -2.195 0.028160 *
## statuspartial         0.037537   0.115052   0.326 0.744224
## statusunchecked       0.011074   0.125247   0.088 0.929545
## v1                   -0.005699   0.006055  -0.941 0.346599
## reasonbusiness        1.117324   0.465435   2.401 0.016368 *
## reasoncc             -0.116264   0.341440  -0.341 0.733472
## reasondebt            0.006220   0.329393   0.019 0.984935
## reasonevent           1.930428   1.261380   1.530 0.125915
## reasonholiday         0.017203   0.636384   0.027 0.978434
## reasonhome            1.502763   0.911647   1.648 0.099269 .
## reasonmedical         0.628096   0.570846   1.100 0.271206
## reasonmoving          0.517572   0.530234   0.976 0.329005
## reasonother           0.107596   0.375256   0.287 0.774321
## reasonrenovation     -0.271294   0.385584  -0.704 0.481687
## reasonsolar         -11.547807 228.799011  -0.050 0.959747
## reasontransport       0.030023   0.581190   0.052 0.958802
## qualityq2            -0.840911   0.240965  -3.490 0.000483 ***
## qualityq3            -1.642059   0.315610  -5.203 1.96e-07 ***
## qualityq4            -2.126720   0.411831  -5.164 2.42e-07 ***
## qualityq5            -3.058167   0.516809  -5.917 3.27e-09 ***
## qualityq6            -3.823126   0.653157  -5.853 4.82e-09 ***
## qualityq7            -3.698967   0.810675  -4.563 5.05e-06 ***
## violations           -0.311404   0.107886  -2.886 0.003897 **
## del                  -0.044171   0.052311  -0.844 0.398455
## req                   0.116841   0.041337   2.827 0.004706 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3058.0  on 2399  degrees of freedom
## Residual deviance: 2645.8  on 2360  degrees of freedom
## AIC: 2725.8
##
## Number of Fisher Scoring iterations: 11
```

```
######################### Step 4: Predict the risk score  ########################

pred_model1 <- predict(model1,test_loan, type="response")
pred2<-predict(cv.lasso,x_test,type='response')


######################### Step 5: Probability Calibration  ########################
cali_pred1<-probability.calibration(test_loan$default, pred2, regularization = FALSE)

plot(density(pred2), col="red", xlim=c(0,1), ylab="Density", xlab="probabilities",
      main="Calibrated probabilities" )
```

```
lines(density(cali_pred1), col="blue")
legend("topright", legend=c("original","calibrated"),
                lty = c(1,1), col=c("red","blue"))
```




```
########################## Final Output: Loan Default Probability for Each Customer in Test Dataset ####
cali_pred1
```

```
##   [1] 5.943944e-02 6.018878e-02 4.297436e-02 1.000000e+00 1.000000e+00
##   [6] 4.294778e-02 7.047477e-02 2.868334e-02 1.792751e-02 6.916190e-02
##  [11] 5.040673e-02 4.122400e-02 1.021128e-02 1.000000e+00 1.193075e-02
##  [16] 6.619830e-02 1.000000e+00 1.000000e+00 5.582803e-02 7.662080e-03
##  [21] 7.145856e-02 5.420766e-02 2.545986e-01 1.000000e+00 2.551261e-02
##  [26] 4.631982e-02 1.000000e+00 6.038957e-02 6.589336e-02 1.000000e+00
##  [31] 1.835395e-01 5.516305e-02 7.315407e-02 7.276469e-02 1.582387e-01
##  [36] 2.654743e-02 7.213950e-02 1.000000e+00 2.308218e-02 1.900644e-02
##  [41] 2.557899e-02 6.131574e-02 1.325079e-01 4.944862e-02 1.738068e-01
##  [46] 2.117081e-02 1.000000e+00 1.486869e-01 5.408785e-02 6.358809e-02
##  [51] 2.127584e-01 6.305899e-02 1.000000e+00 9.359993e-03 1.472137e-03
##  [56] 3.203502e-02 1.230530e-01 6.791311e-02 2.922291e-02 5.499662e-02
##  [61] 3.716746e-03 1.000000e+00 6.529998e-02 7.113301e-01 2.729634e-01
##  [66] 2.124367e-01 1.849495e-01 0.000000e+00 2.719880e-01 3.211263e-02
##  [71] 1.357455e-01 1.000000e+00 2.275113e-01 3.161350e-02 1.005840e-01
##  [76] 1.000000e+00 1.000000e+00 1.000000e+00 8.571429e-01 6.304622e-02
##  [81] 5.839630e-02 7.495848e-01 4.187388e-02 5.862376e-02 1.000000e+00
##  [86] 4.658738e-02 2.314300e-02 1.195258e-02 1.000000e+00 4.178084e-02
```

```
##  [91] 4.059625e-02 2.834902e-03 1.000000e+00 6.378144e-02 1.000000e+00
##  [96] 1.047773e-01 1.094641e-01 3.065507e-02 2.487717e-02 1.000000e+00
## [101] 1.000000e+00 1.260230e-01 1.260615e-01 6.748735e-02 5.513467e-02
## [106] 3.564738e-02 5.250988e-02 6.456040e-02 4.201828e-02 1.000000e+00
## [111] 8.742083e-03 1.961468e-01 7.777778e-02 5.015959e-02 6.408368e-02
## [116] 3.802320e-02 3.509963e-02 1.053996e-02 1.000000e+00 8.479245e-04
## [121] 3.066821e-02 2.182537e-02 1.706541e-01 6.713700e-02 1.617021e-01
## [126] 2.346739e-01 7.201685e-02 2.177999e-01 6.855343e-02 6.759296e-02
## [131] 1.787075e-01 8.707147e-02 1.890745e-01 1.000000e+00 6.104609e-02
## [136] 1.461899e-02 2.590561e-02 1.425021e-01 3.253660e-02 9.724994e-02
## [141] 7.350470e-02 5.961046e-02 1.000000e+00 1.000000e+00 6.726810e-02
## [146] 8.455731e-02 1.000000e+00 3.157505e-03 2.014057e-01 1.556596e-01
## [151] 9.486549e-02 1.522918e-01 6.335271e-02 1.000000e+00 1.496311e-02
## [156] 1.154251e-02 1.756626e-02 6.899995e-02 1.421033e-01 1.000000e+00
## [161] 5.294198e-02 6.880631e-02 2.571925e-01 1.000000e+00 2.289473e-01
## [166] 2.826087e-01 6.113991e-02 6.722051e-02 1.000000e+00 1.000000e+00
## [171] 4.223930e-02 5.847642e-02 1.000000e+00 4.744999e-02 1.002837e-01
## [176] 4.041566e-02 2.150724e-02 5.805111e-02 1.000000e+00 3.287258e-03
## [181] 1.794304e-01 1.122037e-01 4.082766e-02 7.319721e-02 9.970997e-02
## [186] 1.133808e-01 1.000000e+00 5.940594e-02 5.469160e-03 9.838067e-02
## [191] 6.533446e-02 7.601595e-02 8.533192e-02 1.000000e+00 6.743340e-02
## [196] 6.117145e-02 1.000000e+00 4.862404e-02 1.000000e+00 4.955090e-02
## [201] 7.383930e-02 1.000000e+00 1.000000e+00 7.662175e-01 1.413594e-01
## [206] 1.332485e-01 2.836300e-01 1.000000e+00 7.740242e-02 2.013747e-02
## [211] 1.594004e-01 1.000000e+00 4.029895e-02 2.616275e-02 1.000000e+00
## [216] 1.406526e-02 3.517084e-02 1.000000e+00 7.043633e-02 1.000000e+00
## [221] 3.534016e-01 5.650140e-02 1.681988e-03 6.666667e-01 5.347497e-02
## [226] 2.976372e-02 4.214000e-02 2.883815e-01 1.535340e-02 1.000000e+00
## [231] 1.000000e+00 3.412229e-03 1.486833e-01 3.365444e-02 1.000000e+00
## [236] 2.382312e-02 4.798920e-02 6.175115e-02 6.158852e-03 1.000000e+00
## [241] 4.956119e-02 4.293299e-02 6.287447e-02 1.000000e+00 2.255126e-02
## [246] 1.000000e+00 1.439015e-01 1.000000e+00 2.714633e-03 1.174692e-01
## [251] 4.992116e-02 2.844366e-02 1.347581e-02 2.522341e-04 9.824700e-02
## [256] 1.116259e-01 3.086234e-02 3.475257e-02 2.082656e-03 1.000000e+00
## [261] 3.110887e-01 1.633498e-02 1.000000e+00 2.131518e-01 1.076004e-01
## [266] 5.650475e-02 1.000000e+00 3.565011e-04 1.000000e+00 6.256321e-02
## [271] 8.933502e-03 3.598300e-02 3.186492e-05 8.290498e-02 1.000000e+00
## [276] 1.173641e-01 1.466915e-01 5.060812e-02 1.236830e-01 1.643836e-01
## [281] 2.263335e-02 8.200412e-03 5.967021e-02 1.452088e-02 2.012810e-01
## [286] 1.000000e+00 6.408564e-02 1.000000e+00 2.069661e-01 6.028022e-02
## [291] 8.669447e-03 1.000000e+00 1.000000e+00 7.873297e-01 8.061715e-03
## [296] 4.236280e-04 1.000000e+00 6.327159e-02 1.000000e+00 1.000000e+00
## [301] 6.497126e-02 1.000000e+00 6.381172e-02 6.580611e-02 1.000000e+00
## [306] 7.292566e-02 1.000000e+00 2.437282e-02 2.303768e-02 1.864899e-02
## [311] 3.024827e-02 1.476215e-02 1.000000e+00 7.401722e-02 1.756263e-01
## [316] 7.201783e-02 1.000000e+00 5.317733e-02 1.000000e+00 4.563241e-02
## [321] 1.809393e-01 2.597934e-02 1.000000e+00 4.699020e-02 1.913004e-01
## [326] 1.000000e+00 7.199933e-02 1.000000e+00 1.974500e-02 1.000000e+00
## [331] 4.251103e-02 1.000000e+00 1.674907e-01 5.861407e-02 3.463135e-01
## [336] 1.000000e+00 3.840403e-02 4.780615e-02 2.302834e-02 1.861150e-02
## [341] 2.288275e-02 1.000000e+00 2.224123e-01 1.193257e-01 1.000000e+00
## [346] 6.328385e-02 1.000000e+00 4.911543e-02 1.000000e+00 5.367397e-02
## [351] 3.976424e-01 1.000000e+00 1.000000e+00 1.208294e-01 4.230791e-02
## [356] 3.671896e-02 1.000000e+00 9.719601e-02 1.000000e+00 6.252928e-02
```

```
## [361] 4.030550e-02 5.575946e-02 7.741128e-03 1.267421e-01 1.606924e-01
## [366] 4.349711e-02 2.984655e-02 1.204188e-01 6.524046e-02 1.603990e-01
## [371] 5.564236e-02 3.388099e-02 2.142786e-02 1.523857e-01 1.000000e+00
## [376] 1.000000e+00 8.172490e-02 1.000000e+00 4.728731e-02 1.412459e-02
## [381] 6.161562e-02 1.000000e+00 1.853966e-01 8.083645e-02 1.815855e-02
## [386] 1.000000e+00 7.717142e-02 5.732969e-02 1.080786e-01 1.718186e-01
## [391] 1.036164e-02 1.598772e-03 1.000000e+00 7.535476e-02 3.466481e-02
## [396] 1.196955e-03 5.924538e-02 6.814242e-02 1.000000e+00 2.270358e-02
## [401] 2.745258e-02 1.000000e+00 1.967889e-01 5.113334e-01 1.794285e-01
## [406] 2.689874e-02 3.333333e-02 9.066132e-02 6.966190e-02 4.008583e-02
## [411] 9.013892e-02 1.000000e+00 2.052868e-01 5.871738e-02 1.000000e+00
## [416] 1.013303e-01 3.073100e-01 5.027800e-02 1.000000e+00 1.000000e+00
## [421] 2.073384e-01 1.000000e+00 8.433613e-01 1.262121e-01 1.000000e+00
## [426] 1.489947e-01 4.128787e-02 9.880185e-02 3.300629e-02 3.645112e-02
## [431] 1.000000e+00 2.124125e-01 5.085771e-05 9.215018e-02 3.518469e-02
## [436] 8.968419e-04 1.924914e-02 8.031218e-01 6.166122e-02 2.172559e-02
## [441] 7.108138e-02 6.587258e-02 7.166490e-02 1.104636e-02 4.473687e-02
## [446] 2.521774e-02 6.150127e-02 2.401998e-01 6.220531e-02 1.000000e+00
## [451] 1.140444e-01 6.615455e-02 1.779803e-01 1.167435e-01 1.000000e+00
## [456] 4.791661e-02 3.706580e-02 1.075820e-01 1.966337e-02 1.000000e+00
## [461] 2.527883e-02 6.232983e-02 1.000000e+00 8.437744e-02 1.707018e-02
## [466] 1.409181e-02 6.536919e-02 6.820555e-02 1.000000e+00 1.000000e+00
## [471] 1.426217e-04 2.601859e-02 2.964745e-02 7.271098e-02 1.732615e-01
## [476] 1.000000e+00 2.402815e-01 5.631533e-02 3.599218e-02 6.781132e-02
## [481] 1.000000e+00 1.000000e+00 1.000000e+00 5.211011e-02 6.057300e-02
## [486] 1.986555e-01 1.000000e+00 4.737179e-02 7.530596e-02 1.370754e-01
## [491] 3.750000e-01 3.865218e-02 1.000000e+00 1.000000e+00 1.000000e+00
## [496] 1.843309e-01 3.712739e-03 7.307372e-02 1.000000e+00 4.457132e-02
## [501] 1.000000e+00 5.683244e-02 4.814726e-02 1.000000e+00 1.000000e+00
## [506] 5.018085e-02 1.000000e+00 5.085624e-02 7.017376e-02 5.184998e-02
## [511] 2.363381e-02 1.627272e-01 6.141572e-02 1.025221e-02 4.712926e-02
## [516] 7.313163e-03 4.475248e-02 1.000000e+00 1.416238e-02 4.712225e-02
## [521] 3.510790e-02 3.422820e-02 2.445225e-01 1.000000e+00 2.351568e-02
## [526] 2.818509e-02 1.000000e+00 1.869191e-02 1.000000e+00 1.000000e+00
## [531] 7.273403e-02 1.667290e-01 1.190983e-02 1.000000e+00 2.776451e-02
## [536] 1.754005e-02 9.681209e-02 1.754111e-02 1.000000e+00 6.116986e-02
## [541] 8.889577e-02 2.988909e-01 1.343132e-01 6.383351e-02 6.680302e-02
## [546] 1.000000e+00 9.970759e-02 1.000000e+00 1.000000e+00 4.794820e-02
## [551] 4.447669e-02 1.374602e-01 1.706636e-01 1.000000e+00 1.000000e+00
## [556] 9.861191e-02 1.377556e-03 8.946553e-02 8.379980e-04 1.027271e-01
## [561] 1.262989e-04 6.292886e-02 2.170048e-03 4.570129e-02 1.000000e+00
## [566] 1.000000e+00 6.679706e-02 1.000000e+00 4.407401e-02 1.000000e+00
## [571] 7.069280e-02 1.490567e-01 1.000000e+00 1.000000e+00 2.463013e-01
## [576] 2.178265e-02 2.555326e-02 6.044510e-02 1.241473e-01 4.002770e-02
## [581] 2.177237e-02 1.000000e+00 1.000000e+00 2.364023e-02 1.000000e+00
## [586] 2.344940e-01 1.000000e+00 1.000000e+00 4.537166e-02 1.000000e+00
## [591] 8.923466e-04 4.922574e-02 2.437965e-02 1.981692e-02 1.000000e+00
## [596] 4.078800e-02 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00
```

```
######################### Step 6: Evaluation: Calculate MAE error of the prediction  ###################
```

```
mean(abs(y_test-cali_pred1))
```

```
## [1] 0.1141143
```