MATH123 Final Project

Edbert Jao, Robert Kaplan

12/23/2021

#The following packages must be installed. Use install.packages("PACKAGE_NAME") if this computer does not have a package installed. library(tidyverse); library(ridge); library(glmnet); library(corrplot); library(ROSE); library(pls); library(xtable); library(stargazer);

```
#---------------------- Packages ---------------------------------------------

library(tidyverse);
```

```
## Warning: package 'tidyverse' was built under R version 4.0.5
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
## Warning: package 'tibble' was built under R version 4.0.5
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

```
## Warning: package 'readr' was built under R version 4.0.5
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
## Warning: package 'forcats' was built under R version 4.0.5
```

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

```
library(ridge);
```

```
## Warning: package 'ridge' was built under R version 4.0.5
```

```
library(glmnet);
```

```
## Warning: package 'glmnet' was built under R version 4.0.5
```

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

```
## Warning: package 'Matrix' was built under R version 4.0.5
```

```
## 
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
## 
##     expand, pack, unpack
```

```
## Loaded glmnet 4.1-3
```

```
library(corrplot);
```

```
## corrplot 0.90 loaded
```

```
library(ROSE);
```

```
## Warning: package 'ROSE' was built under R version 4.0.5
```

```
## Loaded ROSE 0.0-4
```

```
library(pls);
```

```
## Warning: package 'pls' was built under R version 4.0.5
```

```
##
## Attaching package: 'pls'
```

```
## The following object is masked from 'package:corrplot':
##
##     corrplot
```

```
## The following object is masked from 'package:stats':
##
##     loadings
```

```
library(knitr);
```

```
## Warning: package 'knitr' was built under R version 4.0.5
```

Set Directory to where you have stored the data csv.

```
#---------------------------- Data input --------------------------------------
setwd("C:\\Users\\edber\\OneDrive\\Desktop\\Road to PHD\\Fall 2021 @ Tufts\\Mathematical Aspects of Data Analysis\\Project
\\loans");

df = read.csv("accepted_2007_to_2018Q4.csv");
df$payment_status = ifelse(df$loan_status == "Charged Off" | df$loan_status == "Default", 1, 0);

select_trial = c("payment_status", "loan_amnt", "int_rate", "installment", "annual_inc", "dti",
           "delinq_2yrs", "fico_range_low", "fico_range_high", "open_acc", "pub_rec",
           "revol_bal", "revol_util", "total_acc", "out_prncp", "total_pymnt",
           "total_rec_prncp", "total_rec_int", "total_rec_late_fee",
           "last_fico_range_low", "last_fico_range_high", "tot_coll_amt", "tot_cur_bal",
           "open_acc_6m", "open_act_il", "open_il_12m", "open_il_24m", "mths_since_rcnt_il",
           "total_bal_il", "avg_cur_bal", "tot_hi_cred_lim", "inq_last_12m",
           "acc_open_past_24mths", "pct_tl_nvr_dlq");

loans_trial = df[select_trial];
rownames(loans_trial) = NULL;
loans_trial = na.omit(loans_trial);
```

```
#------------------------ Variable selection ----------------------------------

OLStrial <- lm(payment_status ~ ., loans_trial);
OLStrial %>% summary(); # now remove all non-significant variables: we remove ONLY tot_coll_amt and open_act_il
```

```
##
## Call:
## lm(formula = payment_status ~ ., data = loans_trial)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.31654 -0.05424 -0.01544  0.02574  1.24909
##
## Coefficients:
##                        Estimate Std. Error   t value Pr(>|t|)
## (Intercept)           3.765e-01  3.166e-02    11.893  < 2e-16 ***
## loan_amnt             4.339e-05  5.964e-08   727.545  < 2e-16 ***
## int_rate             -1.630e-03  3.202e-05   -50.907  < 2e-16 ***
## installment          -1.437e-05  1.448e-06    -9.919  < 2e-16 ***
## annual_inc           -1.146e-08  1.029e-09   -11.135  < 2e-16 ***
## dti                   1.763e-04  7.725e-06    22.825  < 2e-16 ***
## delinq_2yrs          -1.313e-03  1.594e-04    -8.239  < 2e-16 ***
## fico_range_low       -2.799e-02  7.903e-03    -3.542 0.000397 ***
## fico_range_high       2.851e-02  7.903e-03     3.608 0.000309 ***
## open_acc             -1.881e-04  4.338e-05    -4.337 1.44e-05 ***
## pub_rec               3.322e-03  2.254e-04    14.739  < 2e-16 ***
## revol_bal             1.761e-08  6.651e-09     2.647 0.008109 **
## revol_util            7.197e-05  6.852e-06    10.504  < 2e-16 ***
## total_acc             4.528e-04  1.644e-05    27.545  < 2e-16 ***
## out_prncp            -4.519e-05  4.335e-08 -1042.455  < 2e-16 ***
## total_pymnt           5.970e-06  2.618e-07    22.800  < 2e-16 ***
## total_rec_prncp      -5.068e-05  2.399e-07  -211.210  < 2e-16 ***
## total_rec_int        -7.083e-06  2.734e-07   -25.907  < 2e-16 ***
## total_rec_late_fee    7.400e-05  1.003e-05     7.380 1.59e-13 ***
## last_fico_range_low  -4.049e-04  2.418e-06  -167.425  < 2e-16 ***
## last_fico_range_high -7.842e-04  3.947e-06  -198.682  < 2e-16 ***
## tot_coll_amt         -5.944e-09  1.686e-08    -0.353 0.724348
## tot_cur_bal          -4.902e-08  4.272e-09   -11.474  < 2e-16 ***
## open_acc_6m           6.318e-04  1.357e-04     4.655 3.24e-06 ***
## open_act_il          -3.051e-05  6.039e-05    -0.505 0.613449
## open_il_12m           3.000e-03  2.112e-04    14.205  < 2e-16 ***
## open_il_24m          -7.172e-04  1.443e-04    -4.970 6.69e-07 ***
## mths_since_rcnt_il    4.580e-05  5.496e-06     8.332  < 2e-16 ***
## total_bal_il         -6.731e-08  3.770e-09   -17.853  < 2e-16 ***
## avg_cur_bal           5.715e-08  1.682e-08     3.399 0.000677 ***
## tot_hi_cred_lim       2.916e-08  3.532e-09     8.257  < 2e-16 ***
## inq_last_12m         -2.049e-04  5.905e-05    -3.471 0.000519 ***
## acc_open_past_24mths -3.247e-04  6.577e-05    -4.937 7.95e-07 ***
## pct_tl_nvr_dlq        4.903e-04  1.581e-05    31.014  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.1423 on 1347907 degrees of freedom
## Multiple R-squared:  0.743,  Adjusted R-squared:  0.743
## F-statistic: 1.181e+05 on 33 and 1347907 DF,  p-value: < 2.2e-16
```

```
select = c("payment_status", "loan_amnt", "int_rate", "installment", "annual_inc", "dti",
  "delinq_2yrs", "fico_range_low", "fico_range_high", "open_acc", "pub_rec",
  "revol_bal", "revol_util", "total_acc", "out_prncp", "total_pymnt",
  "total_rec_prncp", "total_rec_int", "total_rec_late_fee",
  "last_fico_range_low", "last_fico_range_high", "tot_cur_bal",
  "open_acc_6m", "open_il_12m", "open_il_24m", "mths_since_rcnt_il",
  "total_bal_il", "avg_cur_bal", "tot_hi_cred_lim", "inq_last_12m",
  "acc_open_past_24mths", "pct_tl_nvr_dlq");
loans_unbalanced = df[select];
loans_unbalanced = na.omit(loans_unbalanced);

# Randomly delete non-defaulters to balance the classes:
loans_balanced <- ovun.sample(payment_status ~ ., data = loans_unbalanced, method = "under", N = 232418, seed = 115)$data
loans = loans_balanced;
```

```
#------------------------- More data cleaning --------------------------------

# Standardize the data
loans[,-1] <- loans_balanced[,-1] %>% scale(center = TRUE, scale = TRUE);
loans <- as.data.frame((loans));
loans_sq <- loans;
# To generate squared terms without interactions:
for (i in seq_along(loans)) {
loans_sq[[i]] <- I(loans_sq[[i]]^2)
}
loans_sq$payment_statussq1 <- NULL;
loans_all <- cbind(loans, loans_sq);
names(loans_all) <- make.unique(names(loans_all), sep = "sq");
```

```
#------------------ Testing order of prediction model -------------------------

# Drawing a randomly generated sample
list = sample(1:232418, 12000, replace = FALSE);
training2 = loans_all[list[1:10000],]; # training data for use w/ linear and squared terms
testing2 = loans_all[list[10001:12000],]; # testing data for use w/ linear and squared terms
```

```
#------------------ OLS with and without interaction terms --------------------
OLS_interactions = lm(payment_status ~ (loan_amnt+int_rate+installment+annual_inc+dti+delinq_2yrs+fico_range_low+fico_range_
high
                                        +open_acc+pub_rec+revol_bal+revol_util+total_acc+out_prncp+total_pymnt+total_rec_prn
cp
                                        +total_rec_int+total_rec_late_fee+last_fico_range_low+last_fico_range_high+tot_cur_b
al
                                        +open_acc_6m+open_il_12m+open_il_24m+mths_since_rcnt_il+total_bal_il+avg_cur_bal
                                        +tot_hi_cred_lim+inq_last_12m+acc_open_past_24mths+pct_tl_nvr_dlq)^2
                      + (loan_amntsq1+int_ratesq1+installmentsq1+annual_incsq1+dtisq1+delinq_2yrssq1+fico_range_lowsq1+fico_
range_highsq1
                         +open_accsq1+pub_recsq1+revol_balsq1+revol_utilsq1+total_accsq1+out_prncpsq1+total_pymntsq1+total_r
ec_prncpsq1
                         +total_rec_intsq1+total_rec_late_feesq1+last_fico_range_lowsq1+last_fico_range_highsq1+tot_cur_bals
q1
                         +open_acc_6msq1+open_il_12msq1+open_il_24msq1+mths_since_rcnt_ilsq1+total_bal_ilsq1+avg_cur_balsq1
                         +tot_hi_cred_limsq1+inq_last_12msq1+acc_open_past_24mthssq1+pct_tl_nvr_dlqsq1), training2); # OLS w
ith interaction terms
OLSpredicted_interactions <- predict(OLS_interactions, testing2);  # predict on test data
```

```
## Warning in predict.lm(OLS_interactions, testing2): prediction from a rank-
## deficient fit may be misleading
```

```
OLS_MSPE_interactions <- (1/length(testing2))*sum((testing2$payment_status - OLSpredicted_interactions)^2); # OLS w/ interac
tions: out-of-sample MSPE

OLSpredicted_inter_ins <-  predict(OLS_interactions, training2); # predict on training data
```

```
## Warning in predict.lm(OLS_interactions, training2): prediction from a rank-
## deficient fit may be misleading
```

```
OLS_MSPE_inter_ins <- (1/length(training2))*sum((training2$payment_status - OLSpredicted_inter_ins)^2); # OLS w/ interaction
s: in-sample MSPE



OLS = lm(payment_status ~ ., training2); # OLS without interaction terms
OLSpredicted <- predict(OLS, testing2); # predict on test data
```

```
## Warning in predict.lm(OLS, testing2): prediction from a rank-deficient fit may
## be misleading
```

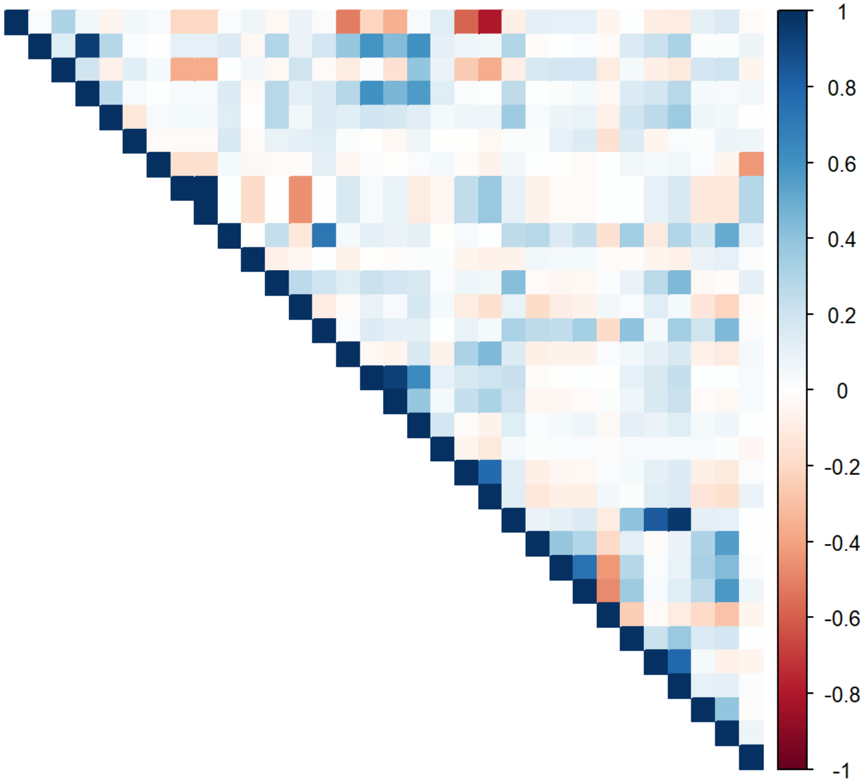
```
OLS_MSPE <- (1/length(testing2))*sum((testing2$payment_status - OLSpredicted)^2); # OLS: out-of-sample MSPE

OLSpredicted_ins <- predict(OLS, training2); # predict on training data
```

```
## Warning in predict.lm(OLS, training2): prediction from a rank-deficient fit may
## be misleading
```

```
OLS_MSPE_ins <- (1/length(testing2))*sum((testing2$payment_status - OLSpredicted_ins)^2); # OLS: in-sample MSPE
```

```
#------------------ RIDGE with and without interaction terms ------------------

Ridge_interactions = linearRidge(payment_status ~ (loan_amnt+int_rate+installment+annual_inc+dti+delinq_2yrs+fico_range_low+
fico_range_high
                                                   +open_acc+pub_rec+revol_bal+revol_util+total_acc+out_prncp+total_pymnt+to
tal_rec_prncp
                                                   +total_rec_int+total_rec_late_fee+last_fico_range_low+last_fico_range_hig
h+tot_cur_bal
                                                   +open_acc_6m+open_il_12m+open_il_24m+mths_since_rcnt_il+total_bal_il+avg_
cur_bal
                                                   +tot_hi_cred_lim+inq_last_12m+acc_open_past_24mths+pct_tl_nvr_dlq)^2
                                 + (loan_amntsq1+int_ratesq1+installmentsq1+annual_incsq1+dtisq1+delinq_2yrssq1+fico_range_l
owsq1+fico_range_highsq1
                                    +open_accsq1+pub_recsq1+revol_balsq1+revol_utilsq1+total_accsq1+out_prncpsq1+total_pymnt
sq1+total_rec_prncpsq1
                                    +total_rec_intsq1+total_rec_late_feesq1+last_fico_range_lowsq1+last_fico_range_highsq1+t
ot_cur_balsq1
                                    +open_acc_6msq1+open_il_12msq1+open_il_24msq1+mths_since_rcnt_ilsq1+total_bal_ilsq1+avg_
cur_balsq1
                                    +tot_hi_cred_limsq1+inq_last_12msq1+acc_open_past_24mthssq1+pct_tl_nvr_dlqsq1), training
2,
                                 lambda = cv.glmnet(y=training2$payment_status%>%as.matrix(),x=training2[,-1]%>%as.matrix(),
                                                    nfolds=10)$lambda.min); # Ridge with interactions
RIDGEpredicted_interactions <- predict(Ridge_interactions, testing2);  # predict on test data
RIDGE_MSPE_interactions <- (1/nrow(testing2))*sum((testing2$payment_status - RIDGEpredicted_interactions)^2); # Ridge w/ int
eractions: out-of-sample MSPE

RIDGEpredicted_interactions_ins <- predict(Ridge_interactions, training2); # predict on training data
RIDGE_MSPE_interactions_ins <- (1/nrow(training2))*sum((training2$payment_status - RIDGEpredicted_interactions_ins)^2); # Ri
dge w/ interactions: in-sample MSPE

Ridge = linearRidge(payment_status ~., training2,
                    lambda = cv.glmnet(y=training2$payment_status%>%as.matrix(),x=training2[,-1]%>%as.matrix(),
                                       nfolds=10)$lambda.min); # Ridge w/o interactions

RIDGEpredicted <- predict(Ridge, testing2); # predict on test data
RIDGE_MSPE <- (1/nrow(testing2))*sum((testing2$payment_status - RIDGEpredicted)^2); # Ridge: out-of-sample MSPE

RIDGEpredicted_ins <- predict(Ridge, training2); # predict on training data
RIDGE_MSPE_ins <- (1/nrow(training2))*sum((training2$payment_status - RIDGEpredicted_ins)^2); # Ridge: in-sample MSPE
```

```
#------------------------ Checking collinearity -----------------------------

## Is ridge appropriate? Check correlation matrix:
corr <- cor(loans);
corrplot::corrplot(corr, method = "color", type = "upper", order = "original",
         tl.col = "black", tl.pos = 'n');
```



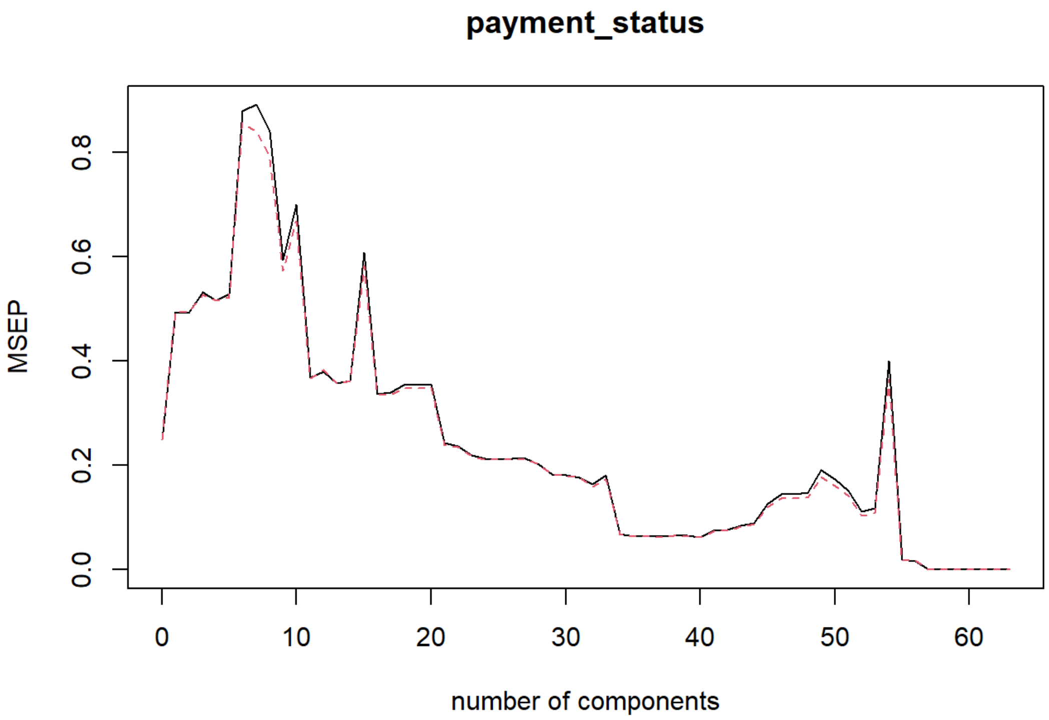
```
#--------------- Principal Components Regression -----------------------------

to_PCA <- training2;
to_PCA$payment_status <- NULL;
loans_PCA <- prcomp(to_PCA, center = FALSE, scale = FALSE) # As the data are already standardized.

# Check that projections on to PCs == projections from inbuilt
as.matrix(to_PCA) %*% as.matrix(loans_PCA$rotation) [,1:10] %>% head(1);
```

```
##               PC1       PC2       PC3       PC4       PC5      PC6       PC7
## 107506 -0.1825749 0.4756223 -1.988575 0.3890079 -3.341944 2.616729 -13.75661
##             PC8       PC9      PC10
## 107506 9.392443 -6.703958 0.3600787
```

```
loans_PCA$x [,1:10] %>% head(1);
```

```
##               PC1       PC2       PC3       PC4       PC5      PC6       PC7
## 107506 -0.1825749 0.4756223 -1.988575 0.3890079 -3.341944 2.616729 -13.75661
##             PC8       PC9      PC10
## 107506 9.392443 -6.703958 0.3600787
```

```
# Make dataframe to run OLS using projections:
training_projections <- as.data.frame((loans_PCA$x));
training_PCR <- cbind(training2$payment_status, training_projections);
names(training_PCR)[1] <- "payment_status";

testing_PCR <- testing2;
testing_PCR$payment_status <- NULL;
testing_projections <- as.matrix(testing_PCR) %*% as.matrix(loans_PCA$rotation) %>% as.data.frame();
testing_PCR <- cbind(testing2$payment_status, testing_projections);
names(testing_PCR)[1] <- "payment_status";

# Principal components regression: no interactions
# PCR_1 <- lm(payment_status ~ ., data = training_PCR);
# summary(PCR_1);


# PCR using package: no interactions
PCR_1 <- pcr(payment_status ~ ., data = training2, center = FALSE, scale = FALSE, validation = "CV", segments = 10); # cross
validate
validationplot(PCR_1, val.type="MSEP");
```



```
training_PCR_CV = training_PCR[,1:(PCR_1$validation$ncomp+1)];
PCR_2 <- lm(payment_status ~ ., data = training_PCR_CV); # PCR

PCRpredicted <- predict(PCR_2, testing_PCR); # predict on test data
```

```
## Warning in predict.lm(PCR_2, testing_PCR): prediction from a rank-deficient fit
## may be misleading
```

```
PCR_MSPE <- (1/nrow(testing_PCR))*sum((testing_PCR$payment_status - PCRpredicted)^2); # PCR: out-of-sample MSPE

PCRpredicted_ins <- predict(PCR_2, training_PCR); # predict on training data
```

```
## Warning in predict.lm(PCR_2, training_PCR): prediction from a rank-deficient fit
## may be misleading
```

```
PCR_MSPE_ins <- (1/nrow(training_PCR))*sum((training_PCR$payment_status - PCRpredicted_ins)^2); # PCR: in-sample MSPE
```

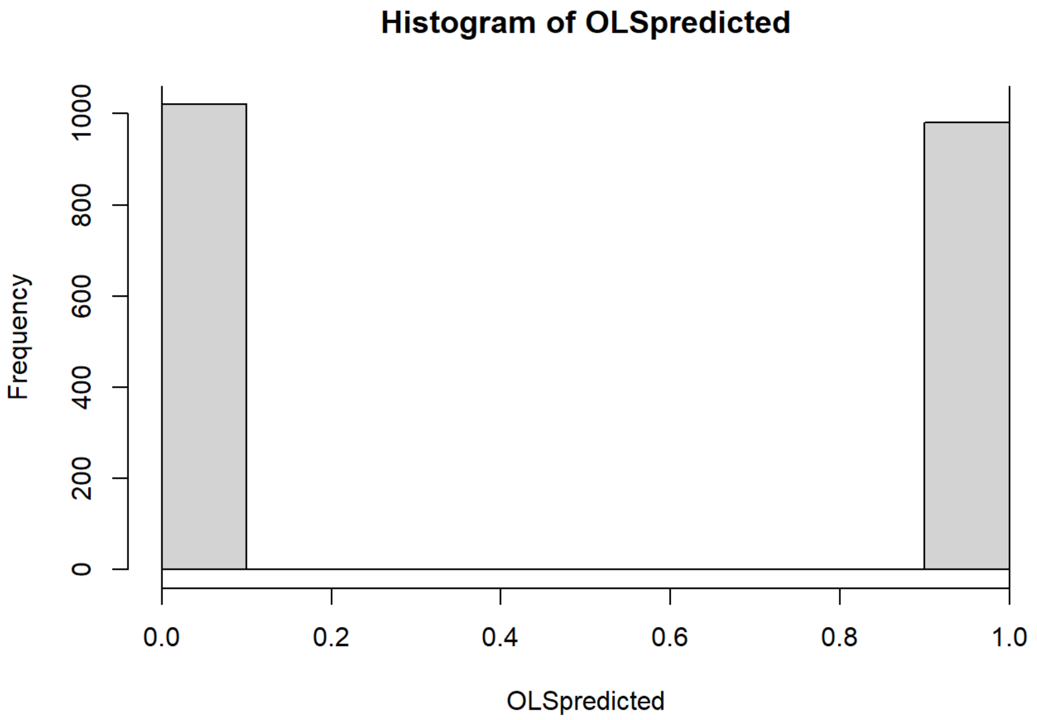
```
#---------------------- Lasso Regression --------------------------------------
# without interactions:
experiment_set = OLS$model;
LASSO1 = glmnet(x = experiment_set[,-1] %>% as.matrix(),
                y = experiment_set[,1] %>% as.matrix(),
                alpha = 1,
                lambda = cv.glmnet(x=experiment_set[,-1]%>%as.matrix(),y=experiment_set[,1]%>%as.matrix(), nfolds=10)$lambd
a.min,
                standardize = FALSE); # Lasso without ineractions

LASSO1predicted <- predict(LASSO1, as.matrix(testing2)[,-1]); # predict on test data
LASSO1_MSPE <- (1/nrow(testing2))*sum((testing2$payment_status - LASSO1predicted)^2); # Lasso: out-of-sample MSPE

LASSO1predicted_ins <- predict(LASSO1, as.matrix(training2)[,-1]); # predict on training data
LASSO1_MSPE_ins <- (1/nrow(training2))*sum((training2$payment_status - LASSO1predicted_ins)^2); # Lasso in-sample MSPE

# with interactions
### data preparation: lasso function cannot generate interaction terms automatically
y = training2[,1] %>% as.matrix();
f = as.formula(y ~ (loan_amnt+int_rate+installment+annual_inc+dti+delinq_2yrs+fico_range_low+fico_range_high
                    +open_acc+pub_rec+revol_bal+revol_util+total_acc+out_prncp+total_pymnt+total_rec_prncp
                    +total_rec_int+total_rec_late_fee+last_fico_range_low+last_fico_range_high+tot_cur_bal
                    +open_acc_6m+open_il_12m+open_il_24m+mths_since_rcnt_il+total_bal_il+avg_cur_bal
                    +tot_hi_cred_lim+inq_last_12m+acc_open_past_24mths+pct_tl_nvr_dlq)^2
               + (loan_amntsq1+int_ratesq1+installmentsq1+annual_incsq1+dtisq1+delinq_2yrssq1+fico_range_lowsq1+fico_range_h
ighsq1
                  +open_accsq1+pub_recsq1+revol_balsq1+revol_utilsq1+total_accsq1+out_prncpsq1+total_pymntsq1+total_rec_prnc
psq1
                  +total_rec_intsq1+total_rec_late_feesq1+last_fico_range_lowsq1+last_fico_range_highsq1+tot_cur_balsq1
                  +open_acc_6msq1+open_il_12msq1+open_il_24msq1+mths_since_rcnt_ilsq1+total_bal_ilsq1+avg_cur_balsq1
                  +tot_hi_cred_limsq1+inq_last_12msq1+acc_open_past_24mthssq1+pct_tl_nvr_dlqsq1));
x = model.matrix(f, training2)[,-1];
x_training = cbind(training2$payment_status, x);

y2 = testing2[,1] %>% as.matrix();
f2 = as.formula(y2 ~ (loan_amnt+int_rate+installment+annual_inc+dti+delinq_2yrs+fico_range_low+fico_range_high
                    +open_acc+pub_rec+revol_bal+revol_util+total_acc+out_prncp+total_pymnt+total_rec_prncp
                    +total_rec_int+total_rec_late_fee+last_fico_range_low+last_fico_range_high+tot_cur_bal
                    +open_acc_6m+open_il_12m+open_il_24m+mths_since_rcnt_il+total_bal_il+avg_cur_bal
                    +tot_hi_cred_lim+inq_last_12m+acc_open_past_24mths+pct_tl_nvr_dlq)^2
               + (loan_amntsq1+int_ratesq1+installmentsq1+annual_incsq1+dtisq1+delinq_2yrssq1+fico_range_lowsq1+fico_range_h
ighsq1
                  +open_accsq1+pub_recsq1+revol_balsq1+revol_utilsq1+total_accsq1+out_prncpsq1+total_pymntsq1+total_rec_prnc
psq1
                  +total_rec_intsq1+total_rec_late_feesq1+last_fico_range_lowsq1+last_fico_range_highsq1+tot_cur_balsq1
                  +open_acc_6msq1+open_il_12msq1+open_il_24msq1+mths_since_rcnt_ilsq1+total_bal_ilsq1+avg_cur_balsq1
                  +tot_hi_cred_limsq1+inq_last_12msq1+acc_open_past_24mthssq1+pct_tl_nvr_dlqsq1));
```

```
x2 = model.matrix(f2, testing2)[,-1];
x_testing = cbind(testing2$payment_status, x2);

LASSO1_interactions = glmnet(x, y, alpha=1, lambda=cv.glmnet(x, y, nfolds=10)$lambda.min, standardize=FALSE); # Lasso with interactions


LASSO1predicted_interactions <- predict(LASSO1_interactions, as.matrix(x_testing)[,-1]); # predict on test data
LASSO1_MSPE_interactions <- (1/nrow(testing2))*sum((testing2$payment_status - LASSO1predicted_interactions)^2); # Lasso: out-of-sample MSPE


LASSO1predicted_inter_ins <- predict(LASSO1_interactions, as.matrix(x_training)[,-1]); # predict on training data
LASSO1_MSPE_inter_ins <- (1/nrow(training2))*sum((training2$payment_status - LASSO1predicted_inter_ins)^2); # Lasso: in-sample MSPE
```

```
#---------------------- Classification / probability -----------------------------------------
summary(OLSpredicted);
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    0.00    0.49    1.00    1.00
```

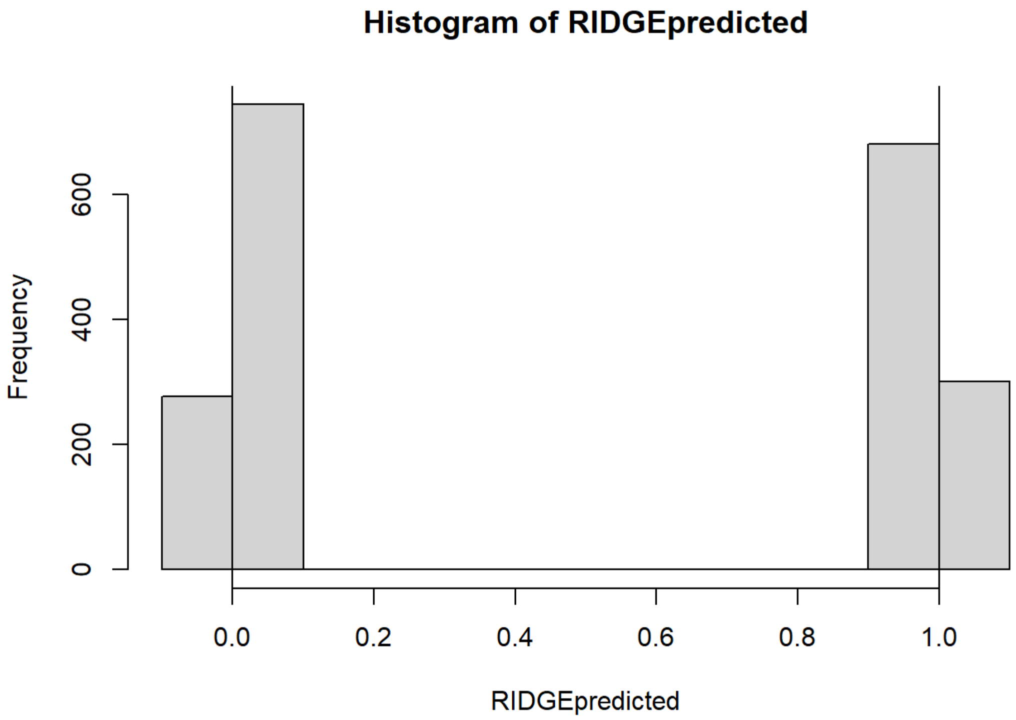
```
hist(OLSpredicted); abline(v = c(0, 1));
```



```
summary(RIDGEpredicted);
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
## -0.037777  0.006021  0.035042  0.489552  0.992379  1.030622
```

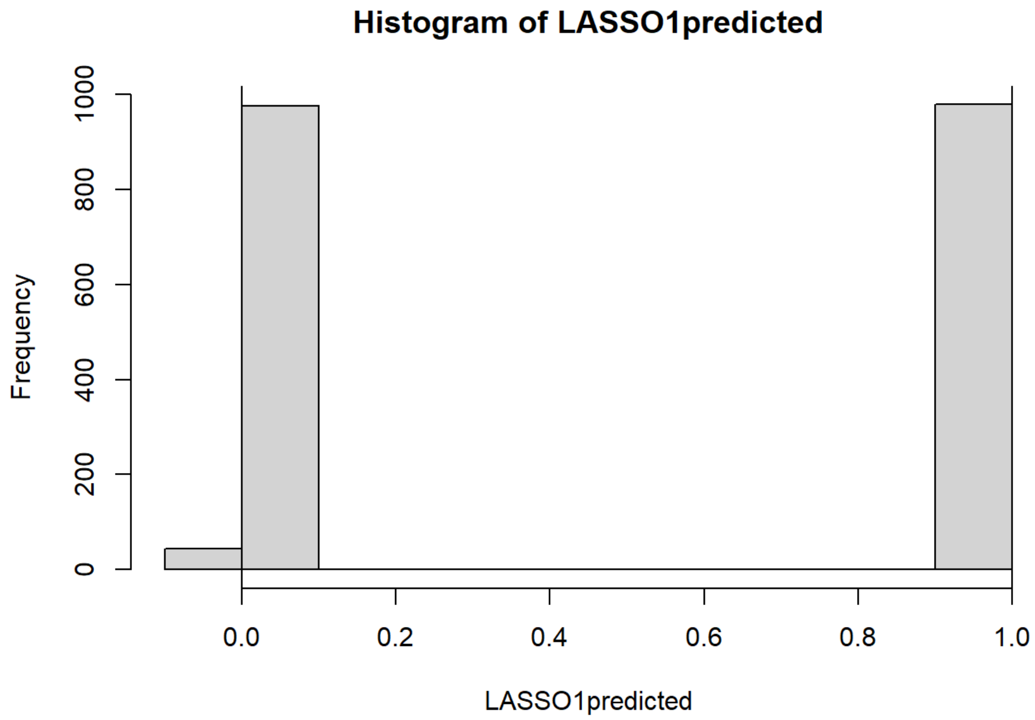
```
hist(RIDGEpredicted); abline(v = c(0, 1));
```



```
summary(LASSO1predicted);
```

```
##        s0
##  Min.   :-0.01549
##  1st Qu.: 0.02732
##  Median : 0.07131
##  Mean   : 0.48935
##  3rd Qu.: 0.97133
##  Max.   : 0.98899
```

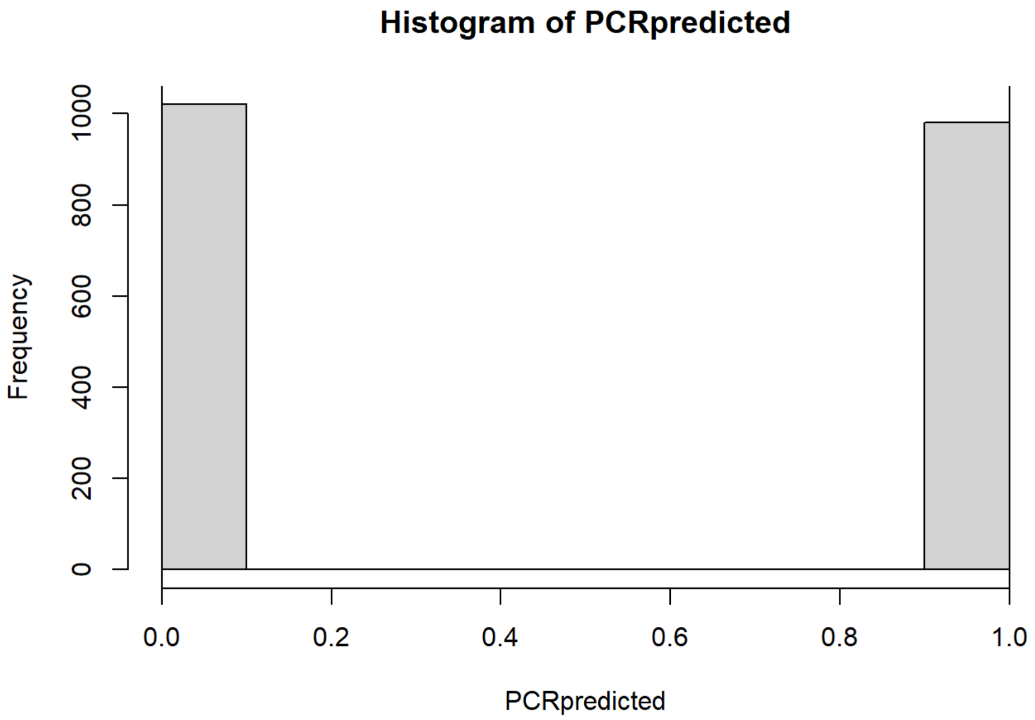
```
hist(LASSO1predicted); abline(v = c(0, 1));
```



```
summary(PCRpredicted);
```

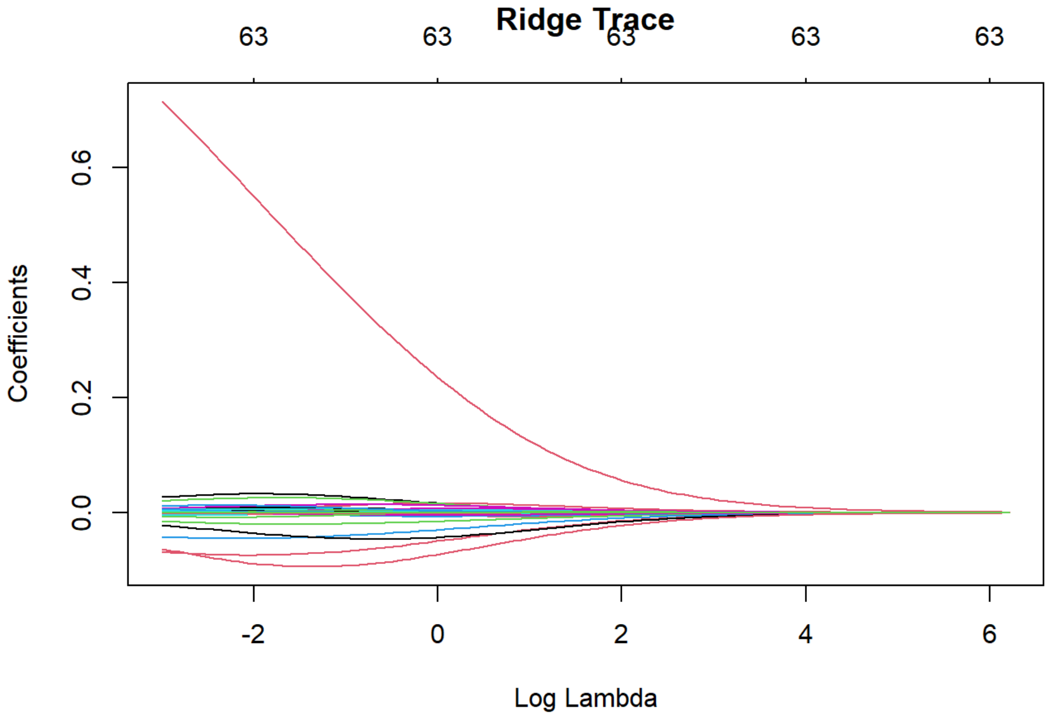
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    0.00    0.00    0.00    0.49    1.00    1.00
```

```
hist(PCRpredicted); abline(v = c(0, 1));
```



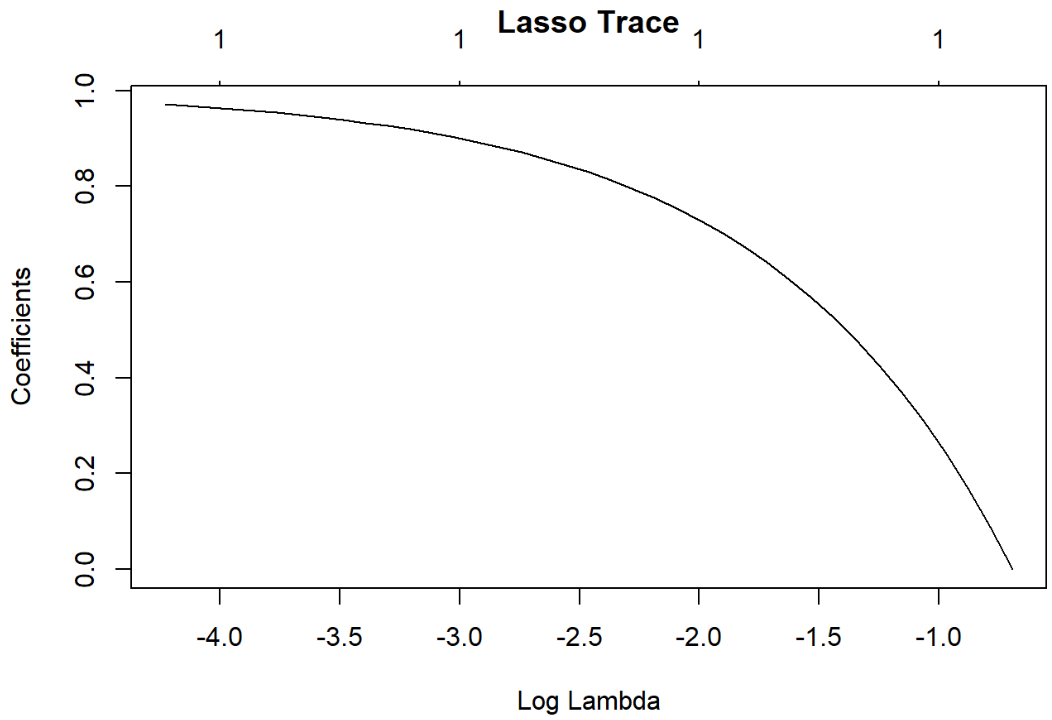
```
# Note that the standardized data have basically been moved to two new modes. Bounds are slightly larger.
```

```
#---------------------- Visualization ----------------------------------------

## Ridge Trace
ridge.fit = glmnet(x=experiment_set[,-1] %>% as.matrix(), y=experiment_set[,1]%>%as.matrix(),alpha=0);
plot(ridge.fit, xvar="lambda", main="Ridge Trace");
```

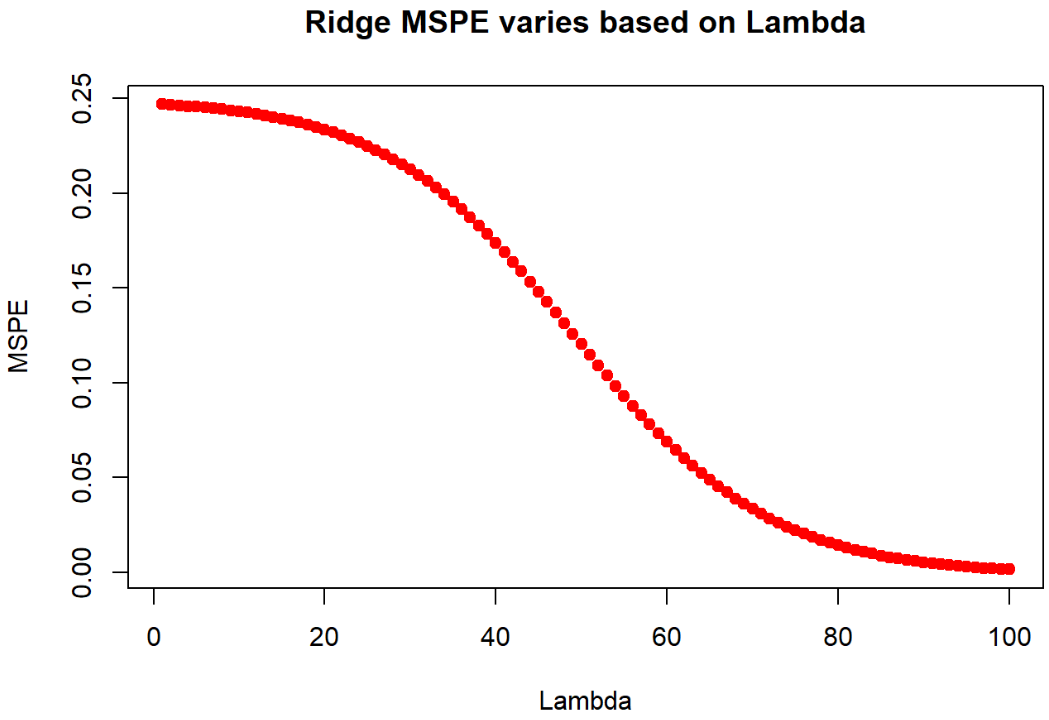


```
## Lasso Trace
lasso.fit = glmnet(x=experiment_set[,-1] %>% as.matrix(), y=experiment_set[,1]%>%as.matrix(),alpha=1);
plot(lasso.fit, xvar="lambda", main="Lasso Trace");
```

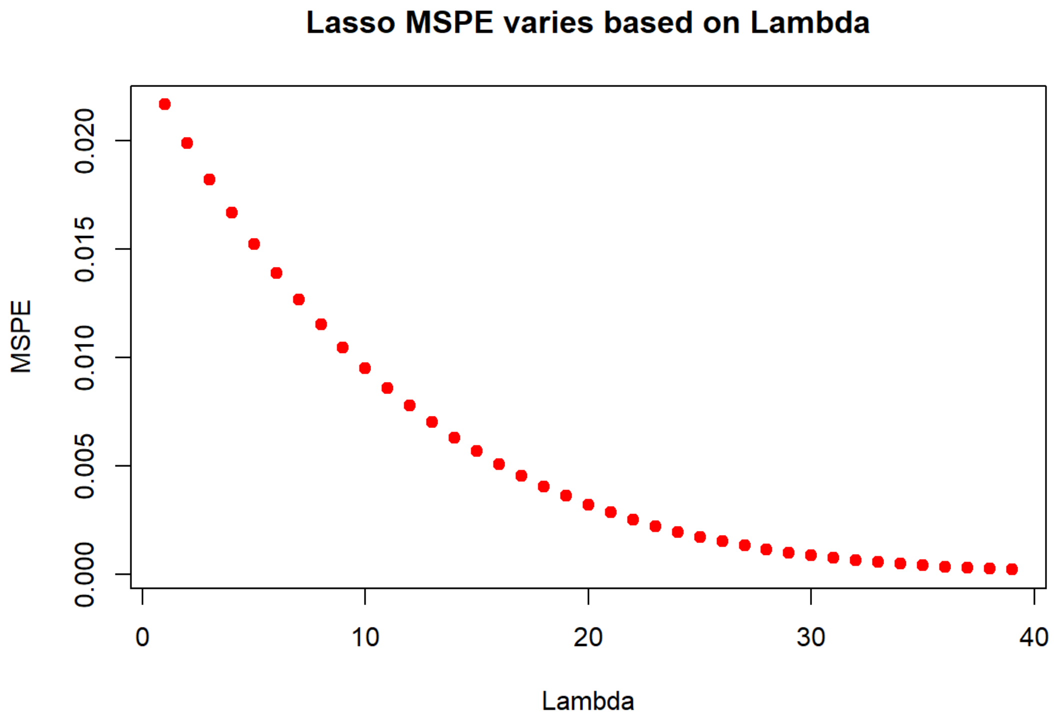
```
## Warning in plotCoef(x$beta, lambda = x$lambda, df = x$df, dev = x$dev.ratio, : 1
## or less nonzero coefficients; glmnet plot is not meaningful
```



```
## MSPE Curve
RidgeMSPE_curve = function(training, testing) {
  lambdas = glmnet(x=experiment_set[,-1] %>% as.matrix(), y=experiment_set[,1]%>%as.matrix(),alpha=0)$lambda;
  MSPE_list = rep(NA, length(lambdas));
  for(i in 1:length(lambdas)) {
    MSPE_list[i] = (1/nrow(testing))*sum((testing$payment_status -
                                            predict(linearRidge(payment_status ~., training, lambda = lambdas[i]), testing))
^2);
  }
  plot(MSPE_list, pch=19, col="red", xlab="Lambda", ylab="MSPE", main="Ridge MSPE varies based on Lambda");
};

RidgeMSPE_curve(training2, testing2);
```



```
LassoMSPE_curve = function(training, testing) {
  lambdas = glmnet(x=experiment_set[,-1] %>% as.matrix(), y=experiment_set[,1]%>%as.matrix(),alpha=1)$lambda;
  MSPE_list = rep(NA, length(lambdas));

  for(i in 1:length(lambdas)) {
    MSPE_list[i] = (1/nrow(testing))*sum((testing$payment_status -
                                        predict(linearRidge(payment_status ~., training, lambda = lambdas[i]), testing))
^2);
  }
  plot(MSPE_list, pch=19, col="red", xlab="Lambda", ylab="MSPE", main="Lasso MSPE varies based on Lambda");
};

LassoMSPE_curve(training2, testing2);
```



```
#--------------------- Tables and Figures -----------------------------------
library(xtable);
library(stargazer);
```

```
## Warning: package 'stargazer' was built under R version 4.0.3
```

```
##
## Please cite as:
```

```
##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.
```

```
##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
## Table showing class imbalance in dependent variable
counts_unbalanced <- table(loans_unbalanced$payment_status);
counts_balanced <- table(loans_balanced$payment_status);

table_imbalance <- rbind(counts_unbalanced, counts_balanced);
  rownames(table_imbalance) <- c("Unbalanced", "Balanced");
  colnames(table_imbalance) <- c("Non-defaulters", "Defaulters");
  table_imbalance <- xtable(table_imbalance, label = "classImbalance");
  print(table_imbalance);
```

```
## % latex table generated in R 4.0.2 by xtable 1.8-4 package
## % Tue Dec 21 07:33:48 2021
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrr}
##   \hline
##  & Non-defaulters & Defaulters \\
##   \hline
## Unbalanced & 1231732 & 116209 \\
##   Balanced & 116209 & 116209 \\
##    \hline
## \end{tabular}
## \label{classImbalance}
## \end{table}
```

```
## Tables showing how non-defaulters differ between unbalanced and balanced samples
##NOTE: the data are standardized now.
var_requests = c("payment_status", "loan_amnt", "int_rate", "annual_inc", "dti",
                   "delinq_2yrs", "fico_range_high", "revol_util",
                   "total_rec_prncp");

unbal_class0 <- subset(loans_unbalanced, subset = payment_status == 0, select = var_requests);
bal_class0 <- subset(loans_balanced, subset = payment_status == 0, select = var_requests);

stats_unbal_class0 <- stargazer(unbal_class0, digits = 3, summary.stat = c("mean", "sd"),
                              label = "unbalancedClass0stats");
```

```
##
## % Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
## % Date and time: Tue, Dec 21, 2021 - 7:33:50 AM
## \begin{table}[!htbp] \centering
##   \caption{}
##   \label{unbalancedClass0stats}
## \begin{tabular}{@{\extracolsep{5pt}}lcc}
## \\[-1.8ex]\hline
## \hline \\[-1.8ex]
## Statistic & \multicolumn{1}{c}{Mean} & \multicolumn{1}{c}{St. Dev.} \\
## \hline \\[-1.8ex]
## payment\_status & 0.000 & 0.000 \\
## loan\_amnt & 15,286.620 & 9,674.561 \\
## int\_rate & 12.686 & 4.956 \\
## annual\_inc & 81,512.120 & 129,412.800 \\
## dti & 19.458 & 16.853 \\
## delinq\_2yrs & 0.301 & 0.868 \\
## fico\_range\_high & 705.997 & 34.649 \\
## revol\_util & 47.037 & 24.778 \\
## total\_rec\_prncp & 8,053.515 & 7,900.010 \\
## \hline \\[-1.8ex]
## \end{tabular}
## \end{table}
```

```
stats_bal_class0 <- stargazer(bal_class0, digits = 3, summary.stat = c("mean", "sd"),
                              label = "balancedClass0stats");
```

```
##
## % Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
## % Date and time: Tue, Dec 21, 2021 - 7:33:50 AM
## \begin{table}[!htbp] \centering
##   \caption{}
##   \label{balancedClass0stats}
## \begin{tabular}{@{\extracolsep{5pt}}lcc}
## \\[-1.8ex]\hline
## \hline \\[-1.8ex]
## Statistic & \multicolumn{1}{c}{Mean} & \multicolumn{1}{c}{St. Dev.} \\
## \hline \\[-1.8ex]
## payment\_status & 0.000 & 0.000 \\
## loan\_amnt & 15,277.990 & 9,679.847 \\
## int\_rate & 12.678 & 4.940 \\
## annual\_inc & 81,568.630 & 85,594.490 \\
## dti & 19.391 & 15.691 \\
## delinq\_2yrs & 0.301 & 0.873 \\
## fico\_range\_high & 705.821 & 34.527 \\
## revol\_util & 47.146 & 24.713 \\
## total\_rec\_prncp & 8,011.836 & 7,862.078 \\
## \hline \\[-1.8ex]
## \end{tabular}
## \end{table}
```

```
## Table comparing key stats from balanced loans to sample we draw
for_bal_stats = loans[var_requests];
colnames(for_bal_stats) <- c("Payment status", "Loan amount", "Int. rate", "Annual inc.", "Debt-income ratio",
                                  "Delinquent w/in 2 yrs", "FICO score, u.b.", "Revolving util.",
                                  "Total princ. paid");
for_sample_stats = training2[var_requests];
colnames(for_sample_stats) <- c("Payment status", "Loan amount", "Int. rate", "Annual inc.", "Debt-income ratio",
                                  "Delinquent w/in 2 yrs", "FICO score, u.b.", "Revolving util.",
                                  "Total princ. paid");

stats_balanced <- stargazer(for_bal_stats, digits = 3, summary.stat = c("mean", "sd", "min", "max"),
                            label = "balancedStatTable");
```

```
##
## % Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
## % Date and time: Tue, Dec 21, 2021 - 7:33:51 AM
## \begin{table}[!htbp] \centering
##   \caption{}
##   \label{balancedStatTable}
## \begin{tabular}{@{\extracolsep{5pt}}lcccc}
## \\[-1.8ex]\hline
## \hline \\[-1.8ex]
## Statistic & \multicolumn{1}{c}{Mean} & \multicolumn{1}{c}{St. Dev.} & \multicolumn{1}{c}{Min} & \multicolumn{1}{c}{Max}
\\
## \hline \\[-1.8ex]
## Payment status & 0.500 & 0.500 & 0 & 1 \\
## Loan amount & $-$0.000 & 1.000 & $-$1.531 & 2.581 \\
## Int. rate & $-$0.000 & 1.000 & $-$1.639 & 3.025 \\
## Annual inc. & $-$0.000 & 1.000 & $-$1.018 & 104.440 \\
## Debt-income ratio & $-$0.000 & 1.000 & $-$1.376 & 63.596 \\
## Delinquent w/in 2 yrs & 0.000 & 1.000 & $-$0.364 & 30.855 \\
## FICO score, u.b. & $-$0.000 & 1.000 & $-$1.112 & 4.752 \\
## Revolving util. & $-$0.000 & 1.000 & $-$2.009 & 4.599 \\
## Total princ. paid & $-$0.000 & 1.000 & $-$0.886 & 5.292 \\
## \hline \\[-1.8ex]
## \end{tabular}
## \end{table}
```

```
stats_sample <- stargazer(for_sample_stats, digits = 3, summary.stat = c("mean", "sd", "min", "max"),
                          label = "subsampleStatTable");
```

```
##
## % Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
## % Date and time: Tue, Dec 21, 2021 - 7:33:52 AM
## \begin{table}[!htbp] \centering
##   \caption{}
##   \label{subsampleStatTable}
## \begin{tabular}{@{\extracolsep{5pt}}lcccc}
## \\[-1.8ex]\hline
## \hline \\[-1.8ex]
## Statistic & \multicolumn{1}{c}{Mean} & \multicolumn{1}{c}{St. Dev.} & \multicolumn{1}{c}{Min} & \multicolumn{1}{c}{Max}
\\
## \hline \\[-1.8ex]
## Payment status & 0.494 & 0.500 & 0 & 1 \\
## Loan amount & $-$0.006 & 0.993 & $-$1.531 & 2.581 \\
## Int. rate & 0.001 & 1.000 & $-$1.639 & 3.025 \\
## Annual inc. & 0.012 & 1.240 & $-$0.993 & 102.132 \\
## Debt-income ratio & $-$0.018 & 0.857 & $-$1.311 & 47.231 \\
## Delinquent w/in 2 yrs & $-$0.008 & 1.011 & $-$0.364 & 16.860 \\
## FICO score, u.b. & 0.013 & 1.017 & $-$1.112 & 4.752 \\
## Revolving util. & $-$0.009 & 0.998 & $-$2.009 & 3.351 \\
## Total princ. paid & 0.009 & 1.004 & $-$0.886 & 5.292 \\
## \hline \\[-1.8ex]
## \end{tabular}
## \end{table}
```

```
# Table of all stats for subsample

var_names1 <- colnames(training2)[1:8];
var_names2 <- colnames(training2)[9:16];
var_names3 <- colnames(training2)[17:24];
var_names4 <- colnames(training2)[25:32];

vars_table <- matrix(c(var_names1, var_names2, var_names3, var_names4), ncol = 4, nrow = 8);

stats_subsample <- stargazer(training2, digits = 2, label = "subsampleStatTableAll",
                             font.size = "tiny", column.sep.width = "0pt");
```

```
##
## % Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
## % Date and time: Tue, Dec 21, 2021 - 7:33:52 AM
## \begin{table}[!htbp] \centering
##   \caption{}
##   \label{subsampleStatTableAll}
## \tiny
## \begin{tabular}{@{\extracolsep{0pt}}lccccccc}
## \\[-1.8ex]\hline
## \hline \\[-1.8ex]
## Statistic & \multicolumn{1}{c}{N} & \multicolumn{1}{c}{Mean} & \multicolumn{1}{c}{St. Dev.} & \multicolumn{1}{c}{Min} &
\multicolumn{1}{c}{Pctl(25)} & \multicolumn{1}{c}{Pctl(75)} & \multicolumn{1}{c}{Max} \\
## \hline \\[-1.8ex]
## payment\_status & 10,000 & 0.49 & 0.50 & 0 & 0 & 1 & 1 \\
## loan\_amnt & 10,000 & $-$0.01 & 0.99 & $-$1.53 & $-$0.79 & 0.57 & 2.58 \\
## int\_rate & 10,000 & 0.001 & 1.00 & $-$1.64 & $-$0.70 & 0.57 & 3.03 \\
## installment & 10,000 & $-$0.01 & 1.00 & $-$1.54 & $-$0.73 & 0.54 & 4.31 \\
## annual\_inc & 10,000 & 0.01 & 1.24 & $-$0.99 & $-$0.40 & 0.21 & 102.13 \\
## dti & 10,000 & $-$0.02 & 0.86 & $-$1.31 & $-$0.48 & 0.36 & 47.23 \\
## delinq\_2yrs & 10,000 & $-$0.01 & 1.01 & $-$0.36 & $-$0.36 & $-$0.36 & 16.86 \\
## fico\_range\_low & 10,000 & 0.01 & 1.02 & $-$1.11 & $-$0.80 & 0.46 & 4.72 \\
## fico\_range\_high & 10,000 & 0.01 & 1.02 & $-$1.11 & $-$0.80 & 0.46 & 4.75 \\
## open\_acc & 10,000 & $-$0.01 & 1.00 & $-$1.87 & $-$0.67 & 0.52 & 9.07 \\
## pub\_rec & 10,000 & $-$0.002 & 0.95 & $-$0.38 & $-$0.38 & $-$0.38 & 14.04 \\
## revol\_bal & 10,000 & 0.002 & 0.98 & $-$0.73 & $-$0.47 & 0.16 & 24.65 \\
## revol\_util & 10,000 & $-$0.01 & 1.00 & $-$2.01 & $-$0.75 & 0.73 & 3.35 \\
## total\_acc & 10,000 & $-$0.01 & 1.00 & $-$1.76 & $-$0.76 & 0.49 & 5.97 \\
## out\_prncp & 10,000 & $-$0.01 & 0.97 & $-$0.51 & $-$0.51 & 0.08 & 4.96 \\
## total\_pymnt & 10,000 & 0.01 & 1.01 & $-$1.11 & $-$0.70 & 0.37 & 5.32 \\
## total\_rec\_prncp & 10,000 & 0.01 & 1.00 & $-$0.89 & $-$0.64 & 0.25 & 5.29 \\
## total\_rec\_int & 10,000 & 0.01 & 1.01 & $-$0.96 & $-$0.66 & 0.32 & 9.04 \\
## total\_rec\_late\_fee & 10,000 & $-$0.01 & 0.90 & $-$0.21 & $-$0.21 & $-$0.21 & 29.47 \\
## last\_fico\_range\_low & 10,000 & 0.01 & 0.98 & $-$3.41 & $-$0.28 & 0.60 & 1.40 \\
## last\_fico\_range\_high & 10,000 & 0.01 & 1.00 & $-$1.52 & $-$0.90 & 0.84 & 2.43 \\
## tot\_cur\_bal & 10,000 & 0.005 & 1.03 & $-$0.89 & $-$0.69 & 0.40 & 17.00 \\
## open\_acc\_6m & 10,000 & $-$0.001 & 1.00 & $-$0.86 & $-$0.86 & 0.77 & 8.11 \\
## open\_il\_12m & 10,000 & $-$0.002 & 1.00 & $-$0.79 & $-$0.79 & 0.22 & 14.33 \\
## open\_il\_24m & 10,000 & $-$0.001 & 1.00 & $-$1.05 & $-$0.45 & 0.15 & 11.46 \\
## mths\_since\_rcnt\_il & 10,000 & 0.01 & 1.03 & $-$0.78 & $-$0.55 & 0.12 & 14.58 \\
## total\_bal\_il & 10,000 & $-$0.01 & 0.99 & $-$0.85 & $-$0.61 & 0.20 & 13.37 \\
## avg\_cur\_bal & 10,000 & 0.01 & 1.05 & $-$0.83 & $-$0.63 & 0.29 & 20.50 \\
## tot\_hi\_cred\_lim & 10,000 & 0.005 & 1.02 & $-$0.98 & $-$0.69 & 0.41 & 16.45 \\
## inq\_last\_12m & 10,000 & 0.01 & 1.01 & $-$0.90 & $-$0.51 & 0.26 & 10.70 \\
## acc\_open\_past\_24mths & 10,000 & 0.001 & 1.00 & $-$1.47 & $-$0.60 & 0.56 & 6.92 \\
## pct\_tl\_nvr\_dlq & 10,000 & 0.01 & 1.01 & $-$8.24 & $-$0.31 & 0.67 & 0.67 \\
## payment\_statussq1 & 10,000 & 0.49 & 0.50 & 0 & 0 & 1 & 1 \\
## loan\_amntsq1 & 10,000 & 0.99 & 1.30 & 0.0000 & 0.14 & 1.23 & 6.66 \\
## int\_ratesq1 & 10,000 & 1.00 & 1.50 & 0.001 & 0.09 & 1.34 & 9.15 \\
## installmentsq1 & 10,000 & 1.00 & 1.58 & 0.0000 & 0.13 & 1.16 & 18.57 \\
## annual\_incsq1 & 10,000 & 1.54 & 104.39 & 0.00 & 0.03 & 0.31 & 10,430.89 \\
## dtisq1 & 10,000 & 0.74 & 22.96 & 0.00 & 0.04 & 0.50 & 2,230.80 \\
## delinq\_2yrssq1 & 10,000 & 1.02 & 7.85 & 0.13 & 0.13 & 0.13 & 284.28 \\
## fico\_range\_lowsq1 & 10,000 & 1.03 & 2.04 & 0.0001 & 0.10 & 0.91 & 22.29 \\
## fico\_range\_highsq1 & 10,000 & 1.03 & 2.04 & 0.0001 & 0.10 & 0.91 & 22.58 \\
## open\_accsq1 & 10,000 & 1.00 & 2.37 & 0.0001 & 0.11 & 1.03 & 82.24 \\
## pub\_recsq1 & 10,000 & 0.91 & 5.51 & 0.15 & 0.15 & 0.15 & 197.07 \\
## revol\_balsq1 & 10,000 & 0.96 & 10.03 & 0.00 & 0.04 & 0.34 & 607.62 \\
## revol\_utilsq1 & 10,000 & 1.00 & 1.11 & 0.0000 & 0.13 & 1.53 & 11.23 \\
## total\_accsq1 & 10,000 & 0.99 & 1.84 & 0.0001 & 0.12 & 1.19 & 35.69 \\
## out\_prncpsq1 & 10,000 & 0.95 & 2.55 & 0.0000 & 0.26 & 0.26 & 24.62 \\
## total\_pymntsq1 & 10,000 & 1.02 & 2.42 & 0.0000 & 0.11 & 0.82 & 28.28 \\
## total\_rec\_prncpsq1 & 10,000 & 1.01 & 2.76 & 0.00 & 0.11 & 0.60 & 28.01 \\
## total\_rec\_intsq1 & 10,000 & 1.02 & 3.29 & 0.0000 & 0.10 & 0.69 & 81.68 \\
## total\_rec\_late\_feesq1 & 10,000 & 0.82 & 10.66 & 0.04 & 0.04 & 0.04 & 868.20 \\
## last\_fico\_range\_lowsq1 & 10,000 & 0.97 & 2.71 & 0.0000 & 0.06 & 0.47 & 11.60 \\
## last\_fico\_range\_highsq1 & 10,000 & 1.01 & 0.90 & 0.0000 & 0.26 & 1.52 & 5.89 \\
## tot\_cur\_balsq1 & 10,000 & 1.06 & 4.65 & 0.0000 & 0.18 & 0.65 & 289.01 \\
## open\_acc\_6msq1 & 10,000 & 1.01 & 2.51 & 0.002 & 0.002 & 0.75 & 65.83 \\
## open\_il\_12msq1 & 10,000 & 0.99 & 3.06 & 0.05 & 0.05 & 0.62 & 205.31 \\
## open\_il\_24msq1 & 10,000 & 1.00 & 2.79 & 0.02 & 0.20 & 1.09 & 131.37 \\
## mths\_since\_rcnt\_ilsq1 & 10,000 & 1.05 & 4.84 & 0.0000 & 0.06 & 0.39 & 212.48 \\
## total\_bal\_ilsq1 & 10,000 & 0.97 & 4.43 & 0.00 & 0.07 & 0.61 & 178.69 \\
## avg\_cur\_balsq1 & 10,000 & 1.11 & 6.77 & 0.0000 & 0.13 & 0.52 & 420.39 \\
## tot\_hi\_cred\_limsq1 & 10,000 & 1.04 & 4.48 & 0.00 & 0.15 & 0.68 & 270.57 \\
## inq\_last\_12msq1 & 10,000 & 1.03 & 3.28 & 0.02 & 0.07 & 0.81 & 114.49 \\
## acc\_open\_past\_24mthssq1 & 10,000 & 1.00 & 2.12 & 0.0005 & 0.10 & 1.29 & 47.91 \\
## pct\_tl\_nvr\_dlqsq1 & 10,000 & 1.01 & 3.02 & 0.0000 & 0.14 & 0.45 & 67.89 \\
## \hline \\[-1.8ex]
## \end{tabular}
## \end{table}
```

```
# Regression table

colnamesreg <- c("OLS", "Ridge", "Lasso", "PCR", "OLS w/ interactions", "Ridge w/ interactions", "Lasso w/ interactions", "PCR w/ interactions");
rownamesreg <- c("$lambda$ / PCs", "In-sample MSPE", "Out-of-sample MSPE",
              "No. nonzero coefficients", "% of $y notin [0,1]");
      lambdas <- rep(NA,8);
          lambdas[1] <- "NA";
          lambdas[2] <- cv.glmnet(y=training2$payment_status%>%as.matrix(),
                                  x=training2[,-1]%>%as.matrix(), nfolds=10)$lambda.min;
          lambdas[3] <- cv.glmnet(x=experiment_set[,-1]%>%as.matrix(),
                                  y=experiment_set[,1]%>%as.matrix(), nfolds=10)$lambda.min;
          lambdas[4] <- PCR_1$ncomp;
          lambdas[5] <- "NA";
          lambdas[6] <- cv.glmnet(y=training2$payment_status%>%as.matrix(),
                                  x=training2[,-1]%>%as.matrix(), nfolds=10)$lambda.min;
          lambdas[7] <- cv.glmnet(x, y, nfolds=10)$lambda.min;
          lambdas[8] <- "NA";
      ismspe <- rep(NA,8);
          ismspe[1]   <- OLS_MSPE_ins;
          ismspe[2]   <- RIDGE_MSPE_ins;
          ismspe[3]   <- LASSO1_MSPE_ins;
          ismspe[4]   <- PCR_MSPE_ins;
          ismspe[5]   <- OLS_MSPE_inter_ins;
          ismspe[6]   <- RIDGE_MSPE_interactions_ins;
          ismspe[7]   <- LASSO1_MSPE_inter_ins;
          ismspe[8]   <- "NA";
      oosmspe <- rep(NA,8);
          oosmspe[1] <- OLS_MSPE;
          oosmspe[2] <- RIDGE_MSPE;
          oosmspe[3] <- LASSO1_MSPE;
          oosmspe[4] <- PCR_MSPE;
          oosmspe[5] <- OLS_MSPE_interactions;
          oosmspe[6] <- RIDGE_MSPE_interactions;
          oosmspe[7] <- LASSO1_MSPE_interactions;
          oosmspe[8] <- "NA";
      nzcoef <- rep(NA,8);
          nzcoef[1] <- "NA";
          nzcoef[2] <- Ridge$coef[Ridge$coef == 0] %>% length();
          nzcoef[3] <- (LASSO1$beta %>% length()) - LASSO1$df;
          nzcoef[4] <- "NA";
          nzcoef[5] <- "NA";
          nzcoef[6] <- Ridge_interactions$coef[Ridge_interactions$coef == 0] %>% length();
          nzcoef[7] <- (LASSO1_interactions$beta %>% length()) - LASSO1_interactions$df;
          nzcoef[8] <- "NA";
      pcnty <- rep(NA,8);
          pcnty[1] <- length(OLSpredicted[OLSpredicted > 1 | OLSpredicted < 0])/length(OLSpredicted);
          pcnty[2] <- length(RIDGEpredicted[RIDGEpredicted > 1 | RIDGEpredicted < 0])/length(RIDGEpredicted);
          pcnty[3] <- length(LASSO1predicted[LASSO1predicted > 1 | LASSO1predicted < 0])/length(LASSO1predicted);
          pcnty[4] <- length(PCRpredicted[PCRpredicted > 1 | PCRpredicted < 0])/length(PCRpredicted);
          pcnty[5] <- length(OLSpredicted_interactions[OLSpredicted_interactions>1|OLSpredicted_interactions<0])/length(OLSp
redicted_interactions);
          pcnty[6] <- length(RIDGEpredicted_interactions>1|RIDGEpredicted_interactions<0)/length(RIDGEpredicted_interaction
s);
          pcnty[7] <- length(LASSO1predicted_interactions[LASSO1predicted_interactions>1|LASSO1predicted_interactions<0])/le
ngth(LASSO1predicted_interactions);
          pcnty[8] <- "NA";


reg_table <- as.data.frame(rbind(lambdas,ismspe,oosmspe,nzcoef,pcnty), col.names = colnamesreg, row.names = rownamesreg);

kable(reg_table);
```

	V1	V2	V3	V4	V5	V6	V7
$lambda$ / PCs	NA	0.0145742846572847	0.0145742846572847	63	NA	0.0145742846572847	0.001387970
In-sample MSPE	77.84375	0.000229260586496013	0.0010911354883647	1.66719256823428e-28	3.00096284057368	0.0213045979734849	0.019968565
Out-of-sample MSPE	3.0161126435807e-30	0.0002315734188899	0.00111134981303019	1.65514810527821e-28	0.668514169806999	0.023310203495327	0.021211628
No. nonzero coefficients	NA	0	58	NA	NA	0	262
% of $y notin [0,1]	0.625	0.2885	0.022	0.51	0.463	1	0.461