STOR565 Final Project

Tao Bian

April 14, 2018

Logistic Regression with LASSO:

```
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 3.4.3
## Loading required package: Matrix
## Loading required package: foreach
## Warning: package 'foreach' was built under R version 3.4.3
## Loaded glmnet 2.0-13
```

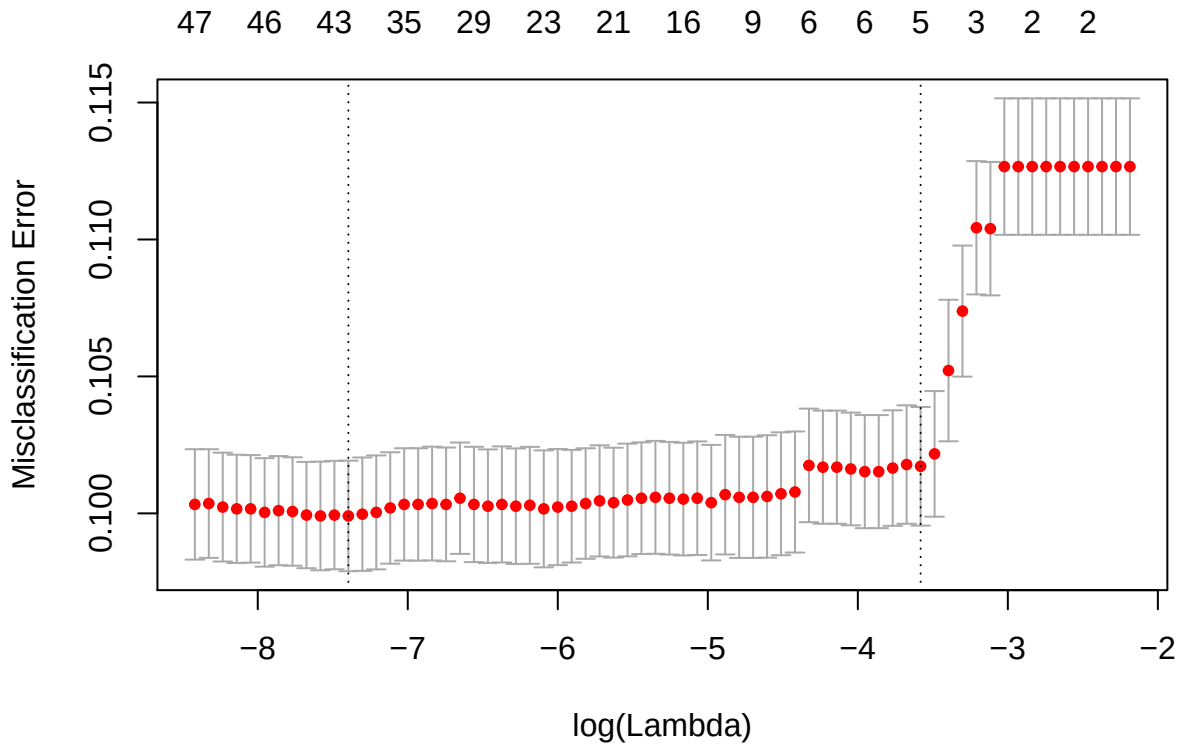
```
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 3.4.4
## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
## The following object is masked from 'package:glmnet':
##
##     auc
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
bank.train.mod <- read.csv("C:\\Users\\tbian\\Documents\\GitHub\\565project\\data\\modified_train.csv")
bank.test.mod <- read.csv("C:\\Users\\tbian\\Documents\\GitHub\\565project\\data\\modified_test.csv")
bank.train <- read.csv("C:\\Users\\tbian\\Documents\\GitHub\\565project\\data\\basic_train.csv")
bank.test <- read.csv("C:\\Users\\tbian\\Documents\\GitHub\\565project\\data\\basic_test.csv")
```

Step2. do the lASSO logistic model based on the basic training dataset:

```
set.seed(1005)
x.matrix.b<-model.matrix(~.,bank.train[,-20])[,-1]
x.test.b<-model.matrix(~.,bank.test[,-20])[,-1]
y.test.b=bank.test$y
foldid=sample(1:4,size=length(bank.train$y),replace=TRUE)
bank.lasso.b<-cv.glmnet(x.matrix.b,bank.train$y, family="binomial", type.measure="class", alpha=1)
plot(bank.lasso.b)
```




```
min(bank.lasso.b$cvm)
```

```
## [1] 0.09990612
```

Step 3. Fit the model with tuning lamda and generate test error and ROC Curve.

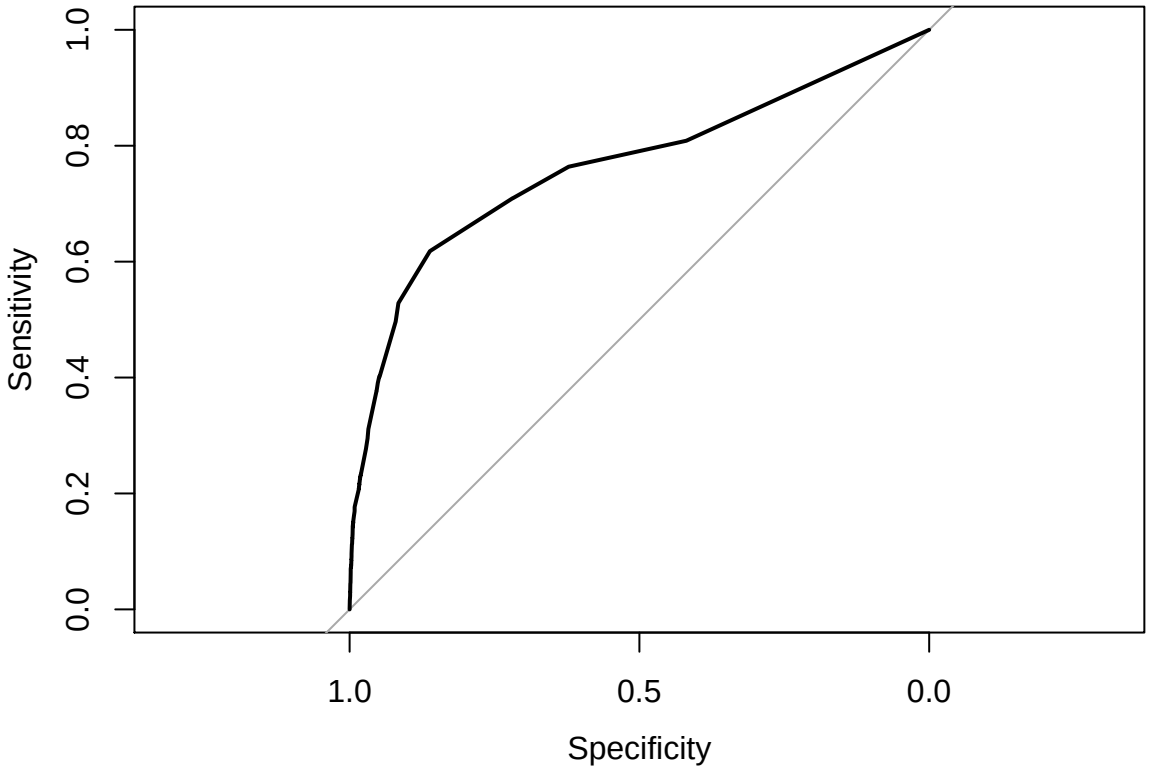
```
set.seed(1005)
fit.b<-glmnet(x.matrix.b,bank.train$y, family="binomial", alpha=1,lambda = bank.lasso.b$lambda.1se)
logistic.predict.b<-predict (fit.b, newx = x.test.b , type="response")
log.pre.b<-ifelse(logistic.predict.b<0.5,0,1)
y.test.b<-ifelse(y.test.b=='no',0,1)
table(y.test.b, log.pre.b)
```

```
##          log.pre.b
## y.test.b    0    1
##        0 9086   50
##        1  994  166
```

```
1-mean(y.test.b==log.pre.b)  #####[1] 0.1012044
```

```
## [1] 0.1013986
```

```
log.roc.b <- roc(y.test.b, as.numeric(logistic.predict.b))
plot(log.roc.b)
```

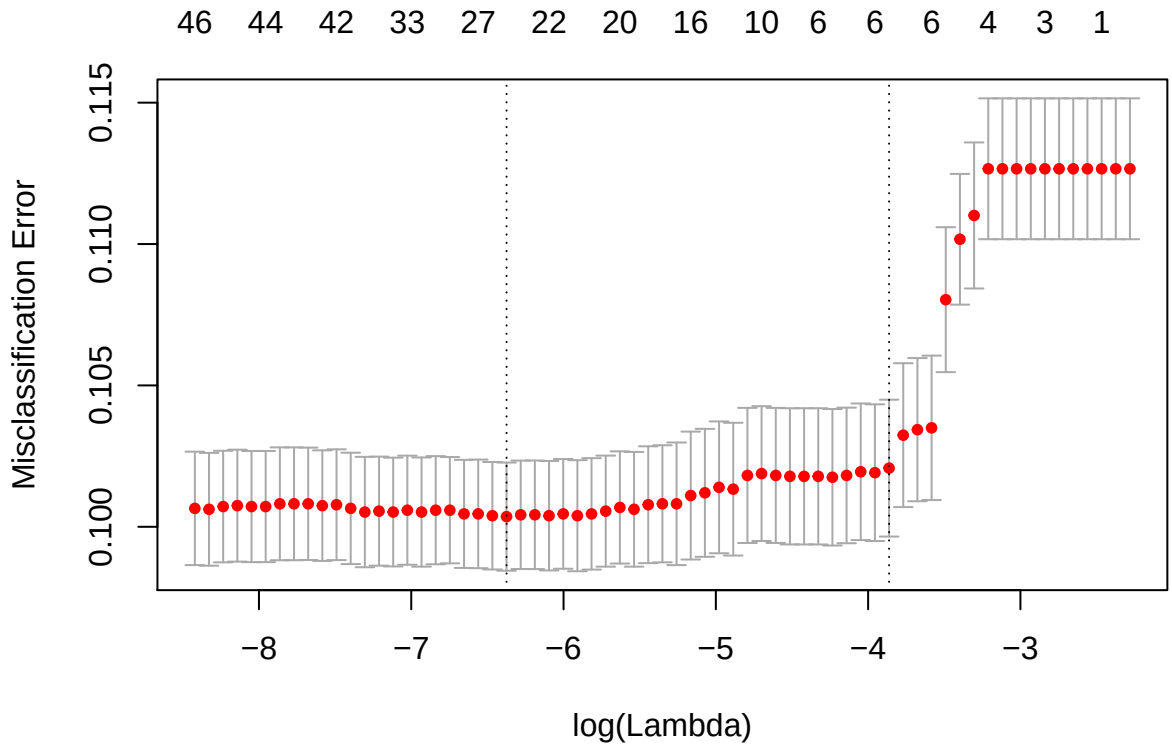



```
log.roc.b$auc
```

```
## Area under the curve: 0.764
```

Now, we will do the lasso logistic regression on the modified train dataset and test dataset:

```
set.seed(1005)
x.matrix.m<-model.matrix(~.,bank.train.mod[,-14])[,-1]
x.test.m<-model.matrix(~.,bank.test.mod[,-14])[,-1]
y.test.m=bank.test.mod$y
foldid.m=sample(1:4,size=length(bank.train.mod$y),replace=TRUE)
bank.lasso.m<-cv.glmnet(x.matrix.m,bank.train.mod$y, family="binomial", type.measure="class", alpha=1)
plot(bank.lasso.m)
```




```
min(bank.lasso.m$cvm)
```

```
## [1] 0.1003594
```

Check the test error and ROC curve:

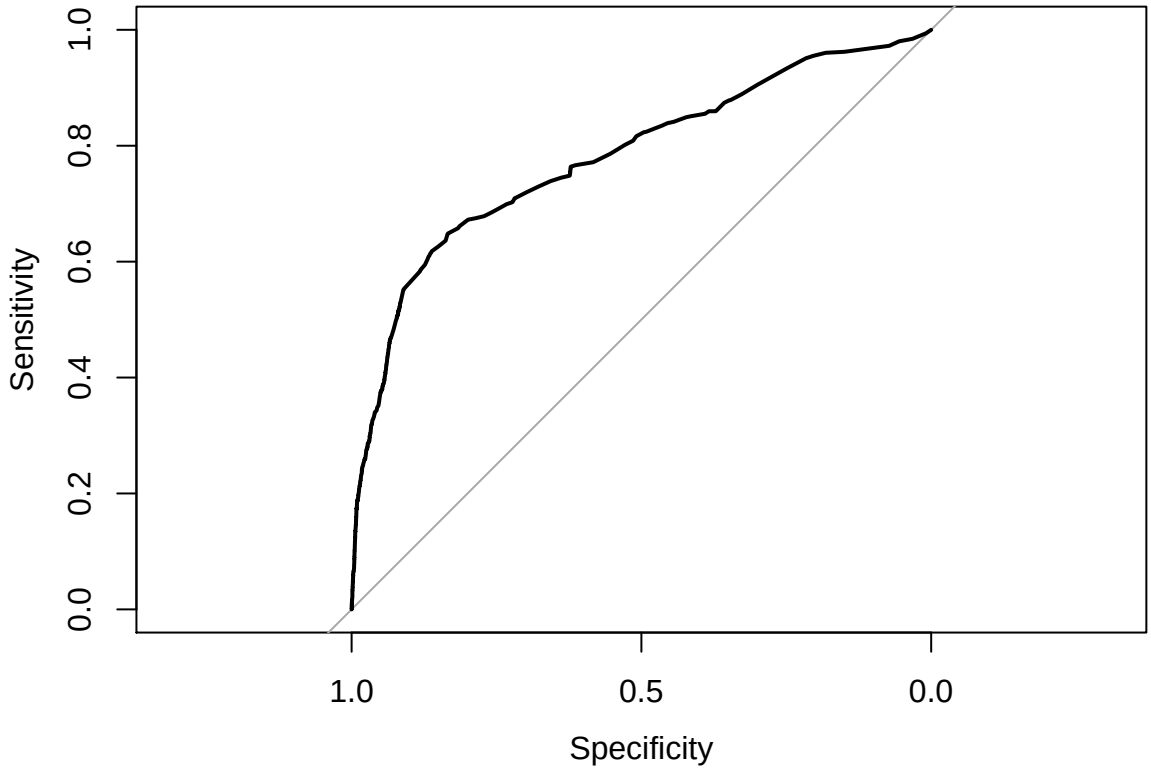
```
set.seed(1005)
fit.m<-glmnet(x.matrix.m,bank.train.mod$y, family="binomial", alpha=1,lambda = bank.lasso.m$lambda.1se)
logistic.predict.m<-predict (fit.m, newx = x.test.m , type="response")
log.pre.m<-ifelse(logistic.predict.m<0.5,0,1)
y.test.m<-ifelse(y.test.m=="no",0,1)
table(y.test.m, log.pre.m)
```

```
##          log.pre.m
## y.test.m    0    1
##        0 9062   74
##        1  973  187
```

```
1-mean(y.test.m==log.pre.m)   #####[1] 0.1007187
```

```
## [1] 0.10169
```

```
log.roc.m <- roc(y.test.m, as.numeric(logistic.predict.m))
plot(log.roc.m)
```




```
log.roc.m$auc
```

```
## Area under the curve: 0.7808
```

Blow is the logistic regression training at smote dataset:

```
source("565_proj_func.R")
```

```
## Warning: package 'gbm' was built under R version 3.4.4
## Loading required package: survival
## Warning: package 'survival' was built under R version 3.4.4
## Loading required package: lattice
## Loading required package: splines
## Loading required package: parallel
## Loaded gbm 2.1.3
```

```
bank.dummy=augmented_dataframe(bank.train)
```

```
## Warning: package 'mlr' was built under R version 3.4.4
## Loading required package: ParamHelpers
## Warning: package 'ParamHelpers' was built under R version 3.4.4
```

```
set.seed(1005)
library(smotefamily)
```

```
## Warning: package 'smotefamily' was built under R version 3.4.4
```

```
bank.smote <- ADAS(bank.dummy[,2:ncol(bank.dummy)],bank.dummy[,1],K=5)
dim(bank.smote$data)
```

```
## [1] 54597    53
```

```
str(bank.smote$data)
```

```
## 'data.frame':    54597 obs. of  53 variables:
##  $ age                : num  31 53 27 36 32 38 28 42 31 70 ...
##  $ admin.             : num  1 1 0 0 1 0 0 1 0 0 ...
##  $ blue.collar        : num  0 0 0 0 0 0 1 0 0 0 ...
##  $ entrepreneur       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ housemaid          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ management         : num  0 0 0 1 0 0 0 0 0 0 ...
##  $ retired            : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ self.employed      : num  0 0 0 0 0 1 0 0 0 0 ...
##  $ services           : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ student            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ technician         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ unemployed         : num  0 0 1 0 0 0 0 0 0 0 ...
##  $ divorced           : num  0 0 0 0 0 1 0 0 0 0 ...
##  $ married            : num  0 1 0 0 0 0 1 1 1 1 ...
##  $ single             : num  1 0 1 1 1 0 0 0 0 0 ...
##  $ basic.4y           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ basic.6y           : num  0 0 0 0 0 0 1 0 1 0 ...
##  $ basic.9y           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ high.school        : num  1 0 0 0 0 1 0 0 0 0 ...
##  $ illiterate         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ professional.course: num  0 0 0 0 0 0 0 1 0 0 ...
##  $ university.degree  : num  0 1 1 1 1 0 0 0 0 0 ...
##  $ default_no         : num  1 1 1 1 1 1 1 1 0 1 ...
##  $ unknown            : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ housing_no         : num  0 1 0 0 1 1 1 1 1 0 ...
##  $ housing_unknown    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ loan_no            : num  1 0 0 1 1 1 1 0 1 1 ...
##  $ loan_unknown       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ contact_cellular   : num  1 0 1 1 1 1 0 1 0 1 ...
##  $ apr                : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ aug                : num  0 0 0 0 0 1 0 0 0 0 ...
##  $ dec                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ jul                : num  1 0 1 0 0 0 0 0 0 0 ...
##  $ jun                : num  0 1 0 1 1 0 1 0 0 0 ...
##  $ mar                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ may                : num  0 0 0 0 0 0 0 1 1 0 ...
##  $ nov                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ oct                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ fri                : num  0 0 0 0 1 1 0 0 0 0 ...
##  $ mon                : num  0 0 0 0 0 0 0 1 0 0 ...
##  $ thu                : num  0 1 1 0 0 0 0 0 0 1 ...
##  $ tue                : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ campaign           : num  1 10 2 2 1 3 5 1 4 2 ...
##  $ pdays              : num  999 999 999 3 999 999 999 999 999 999 ...
##  $ previous           : num  0 0 0 1 0 1 0 0 0 0 ...
##  $ failure            : num  0 0 0 0 0 1 0 0 0 0 ...
```

```
##  $ nonexistent        : num  1 1 1 0 1 0 1 1 1 1 ...
##  $ emp.var.rate       : num  1.4 -2.9 1.4 -2.9 -2.9 -2.9 1.4 -1.8 1.1 -1.8 ...
##  $ cons.price.idx     : num  93.9 93 93.9 93 93 ...
##  $ cons.conf.idx      : num  -42.7 -40.8 -42.7 -40.8 -40.8 -31.4 -41.8 -46.2 -36.4 -47.1 ...
##  $ euribor3m          : num  4.96 1.24 4.96 1.26 1.27 ...
##  $ nr.employed        : num  5228 5076 5228 5076 5076 ...
##  $ class              : chr  "1" "1" "1" "1" ...
```

```
#################### generate test dataset####################
bank.dummy.t=augmented_dataframe(bank.test)
dim(bank.dummy.t)
```

```
## [1] 10296    53
```

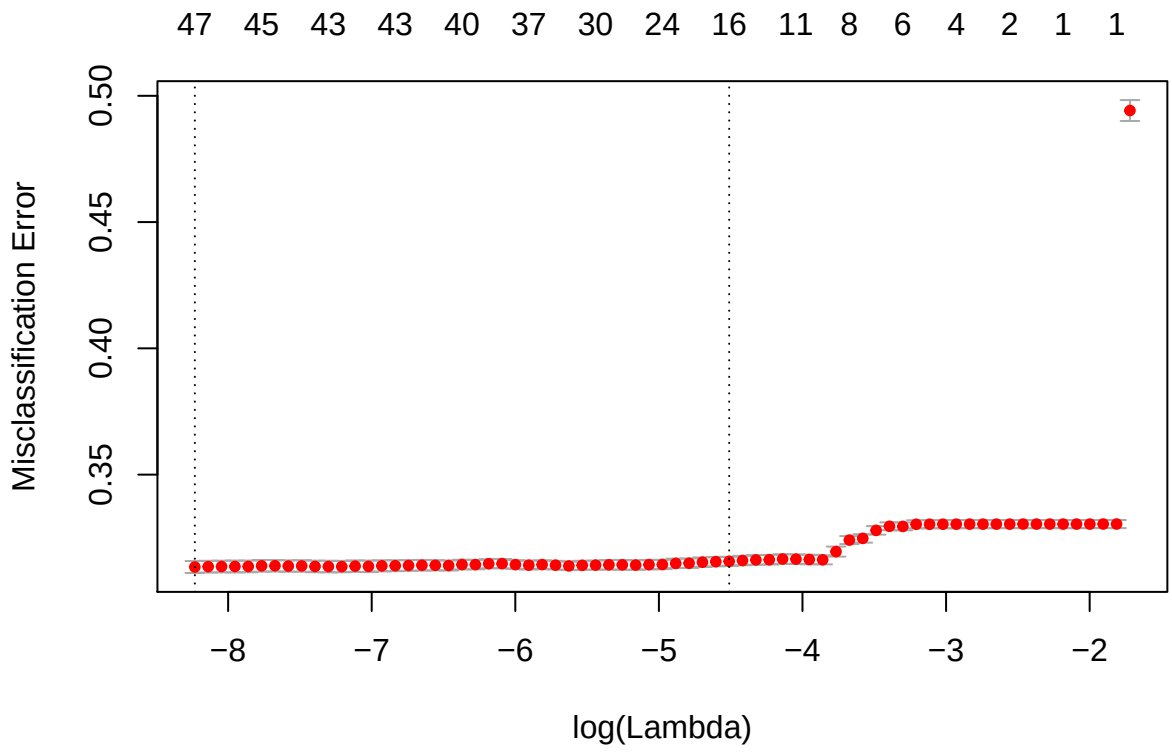
```
str(bank.dummy.t)
```

```
## 'data.frame':    10296 obs. of  53 variables:
##  $ y                  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ age                : int  56 57 59 55 34 41 57 54 55 42 ...
##  $ admin.             : num  0 0 1 0 0 1 1 1 0 0 ...
##  $ blue.collar        : num  0 0 0 0 0 0 0 0 1 1 ...
##  $ entrepreneur       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ housemaid          : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ management         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ retired            : num  0 0 0 1 0 0 0 0 0 0 ...
##  $ self.employed      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ services           : num  0 1 0 0 1 0 0 0 0 0 ...
##  $ student            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ technician         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ unemployed         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ divorced           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ married            : num  1 1 1 0 1 1 1 1 1 1 ...
##  $ single             : num  0 0 0 1 0 0 0 0 0 0 ...
##  $ basic.4y           : num  1 0 0 0 0 0 0 0 1 0 ...
##  $ basic.6y           : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ basic.9y           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ high.school        : num  0 1 0 1 1 0 0 1 0 0 ...
##  $ illiterate         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ professional.course: num  0 0 1 0 0 0 0 0 0 0 ...
##  $ university.degree  : num  0 0 0 0 0 1 1 0 0 0 ...
##  $ default_no         : num  1 0 1 1 1 1 1 1 0 0 ...
##  $ unknown            : num  0 1 0 0 0 0 0 0 1 1 ...
##  $ housing_no         : num  1 1 1 0 1 0 1 1 1 0 ...
##  $ housing_unknown    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ loan_no            : num  1 1 1 1 1 1 0 1 1 1 ...
##  $ loan_unknown       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ contact_cellular   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ apr                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ aug                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ dec                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ jul                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ jun                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ mar                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ may                : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ nov                : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
##  $ oct               : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ fri               : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ mon               : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ thu               : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ tue               : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ campaign          : int  1 1 1 1 1 1 1 1 2 1 ...
##  $ pdays             : int  999 999 999 999 999 999 999 999 999 999 ...
##  $ previous          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ failure           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ nonexistent       : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ emp.var.rate      : num  1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 ...
##  $ cons.price.idx    : num  94 94 94 94 94 ...
##  $ cons.conf.idx     : num  -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 ...
##  $ euribor3m         : num  4.86 4.86 4.86 4.86 4.86 ...
##  $ nr.employed       : num  5191 5191 5191 5191 5191 ...
```

```
x.matrix=model.matrix(~.,bank.smote$data[,-53])[,-1]
x.test=model.matrix(~.,bank.dummy.t[,-1])[,-1]
y.test=bank.dummy.t$y

################################## logistic ##################
foldid=sample(1:2,size=length(bank.smote$class),replace=TRUE)
bank.lasso<-cv.glmnet(x.matrix,bank.smote$data$class, family="binomial", type.measure="class", alpha=1)
plot(bank.lasso)
```




```
fit<-glmnet(x.matrix,bank.smote$data$class, family="binomial", alpha=1,lambda = bank.lasso$lambda.1se)
logistic.predict<-predict (fit, newx = x.test , type="response")
```

```r
log.pre<-ifelse(logistic.predict>=0.5,1,0)
table(y.test, log.pre)
```

```
##       log.pre
## y.test    0    1
##      0 7128 2008
##      1  373  787
```

```r
1-mean(y.test==log.pre)
```

```
## [1] 0.2312549
```

```r
fit$beta
```

```
## 52 x 1 sparse Matrix of class "dgCMatrix"
##                                s0
## age                             .
## admin.                          .
## blue.collar                     .
## entrepreneur                    .
## housemaid                       .
## management                      .
## retired                6.889729e-02
## self.employed                   .
## services                        .
## student                         .
## technician                      .
## unemployed                      .
## divorced              -4.006985e-02
## married                         .
## single                 3.825517e-02
## basic.4y                        .
## basic.6y                        .
## basic.9y                        .
## high.school                     .
## illiterate                      .
## professional.course             .
## university.degree               .
## default_no             2.715593e-01
## unknown                         .
## housing_no                      .
## housing_unknown       -9.442216e-02
## loan_no                         .
## loan_unknown          -5.625403e-14
## contact_cellular       2.694862e-01
## apr                             .
## aug                             .
## dec                             .
## jul                             .
## jun                             .
## mar                    2.923396e-01
## may                   -5.273469e-01
## nov                   -1.941615e-01
## oct                             .
## fri                             .
```

```
## mon                -1.001962e-01
## thu                 .
## tue                 .
## campaign           -3.720788e-02
## pdays              -5.200559e-04
## previous            .
## failure            -4.783658e-01
## nonexistent         .
## emp.var.rate       -6.550653e-02
## cons.price.idx      .
## cons.conf.idx       .
## euribor3m           .
## nr.employed        -7.078472e-03
```