GLMNET Models Analysis

12/11/2017

Data Initialization

```
set.seed(12345)
loan_data <- read.csv("/Users/niniliu/Documents/EECS6893/Project/EECS 6893 Project/integrated_data.csv"

#set up categorical variables
loan_data$FIRST_TIME_HOME_BUYER_FLAG.f <- factor(loan_data$FIRST_TIME_HOME_BUYER_FLAG)
loan_data$OCCUPANCY_STATUS.f <- factor(loan_data$OCCUPANCY_STATUS)
loan_data$CHANNEL.f <- factor(loan_data$CHANNEL)
loan_data$LOAN_PURPOSE.f <- factor(loan_data$LOAN_PURPOSE)
loan_data$SUPER_CONFORMING_FLAG.f <- factor(loan_data$SUPER_CONFORMING_FLAG)

#interpret HPI index
#loan_data['HPI_var'] <- (loan_data$HPI_MAX-loan_data$HPI_MIN)/loan_data$HPI_ORIG
loan_data['HPI_inc'] <- loan_data$HPI_MAX/loan_data$HPI_ORIG
loan_data['HPI_dec'] <- loan_data$HPI_ORIG/loan_data$HPI_MIN

#normalize variables
loan_select <- subset(loan_data,select=c(2,5,6,9:11,16,18,28,29))
scale_loan <- scale(loan_select)
loan_model <- cbind(scale_loan,subset(loan_data,select=c(23:27)),loan_data$IND_DEFAULT_2)
colnames(loan_model)[16] <- "DEFAULT_IND" #rename column
sample_size <- floor(0.8*nrow(loan_data))
train_index <- sample(seq_len(nrow(loan_data)), size = sample_size)
train <- loan_model[train_index,]
test <- loan_model[-train_index,]
```

GLMNET Models

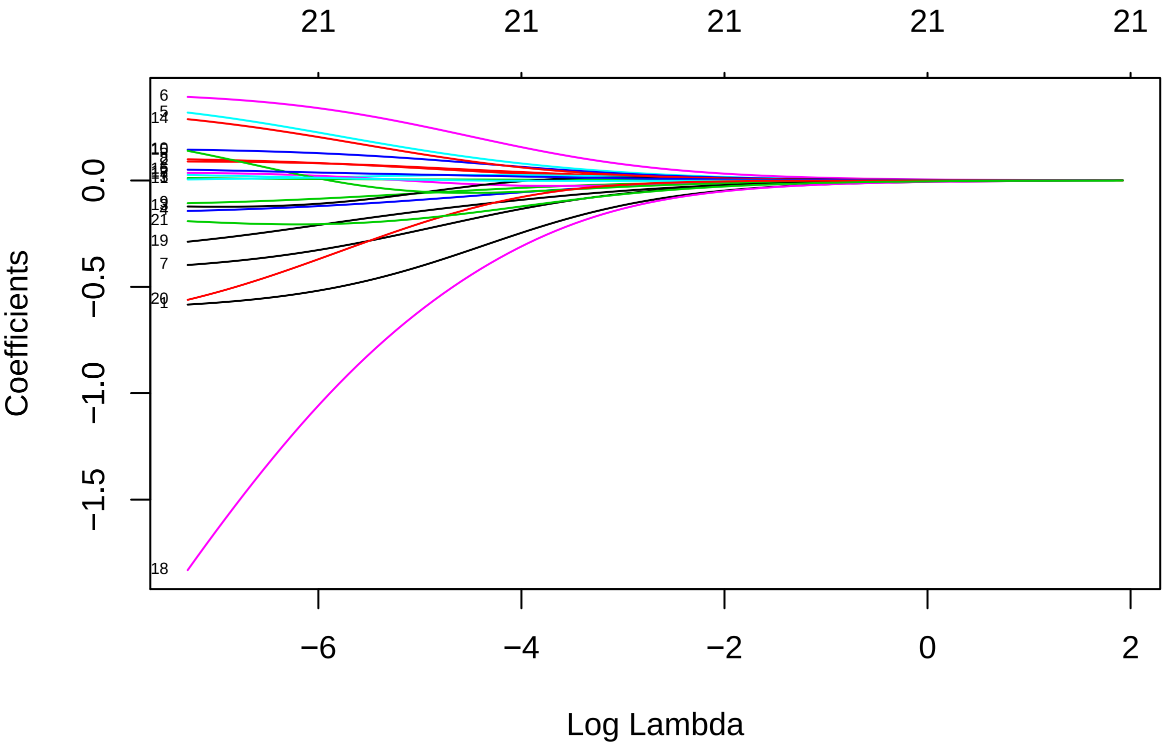
```
#glmnet model
library(glmnet)
```

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

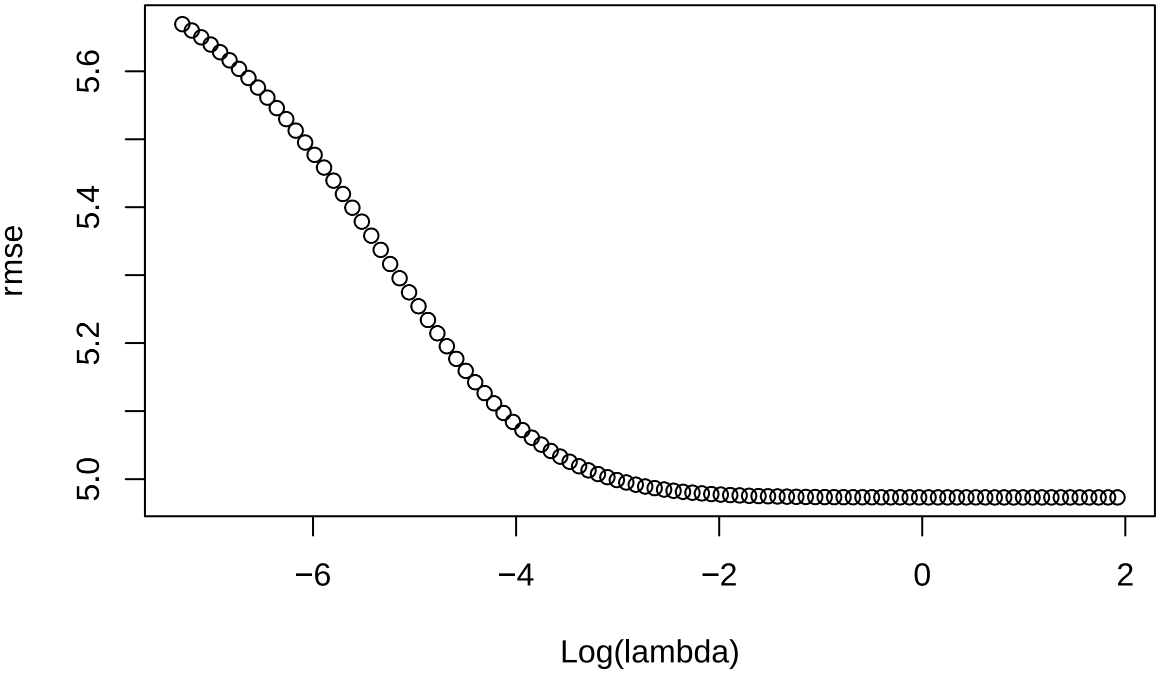
```
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-13
```

```
x = model.matrix(DEFAULT_IND ~ . -1, data = loan_model)
y = loan_model$DEFAULT_IND
glmnet.tr <- glmnet(x[train_index,],y[train_index],family="binomial",alpha=0)
plot(glmnet.tr, xvar = "lambda", label = TRUE)
```



```
pred_net <- predict(glmnet.tr, x[-train_index, ])
#pred <- ifelse(pred > 0.5,1,0)
rmse = sqrt(apply((y[-train_index] - pred_net)^2, 2, mean))
plot(log(glmnet.tr$lambda), rmse, type = "b", xlab = "Log(lambda)")
```

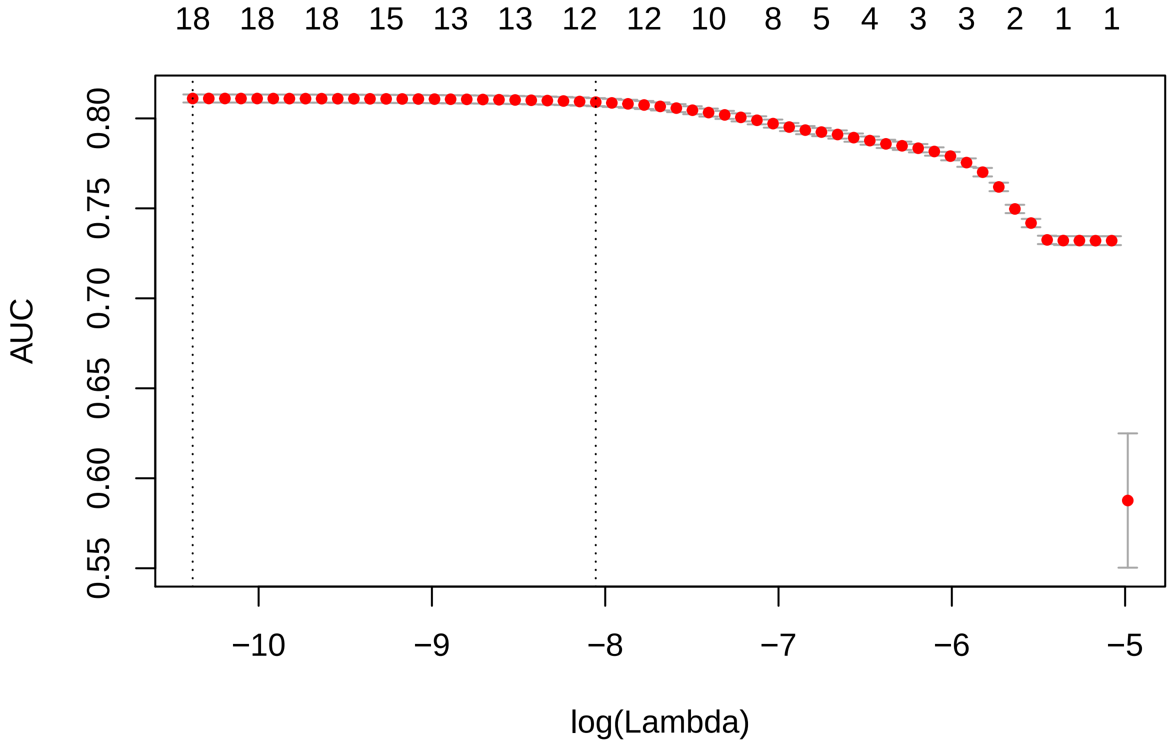


```
lam.best = glmnet.tr$lambda[order(rmse)[1]]
print(coef(glmnet.tr, s = lam.best))
```

```
## 22 x 1 sparse Matrix of class "dgCMatrix"
##                                              1
## (Intercept)                      -4.965586e+00
## FICO                             -1.914149e-03
## MORTGAGE_INSURANCE_PCT            3.939517e-04
## NUM_OF_UNITS                      4.715397e-05
## LOAN_SIZE                        -4.567327e-04
## ORGN_LTV                          6.938740e-04
## ORGN_RATE                         1.322353e-03
## NUM_OF_BORROWERS                 -1.026986e-03
## HPI_ORIG                          2.338836e-04
## HPI_inc                          -2.964399e-04
## HPI_dec                           5.694015e-04
## FIRST_TIME_HOME_BUYER_FLAG.fN     4.168576e-04
## FIRST_TIME_HOME_BUYER_FLAG.fU    -5.969613e-04
## FIRST_TIME_HOME_BUYER_FLAG.fY     5.242986e-04
## OCCUPANCY_STATUS.fO               3.790955e-04
## OCCUPANCY_STATUS.fS              -5.520860e-04
## CHANNEL.fC                        2.085163e-04
## CHANNEL.fR                       -5.996166e-05
## CHANNEL.fT                       -1.941031e-03
## LOAN_PURPOSE.fN                  -9.505862e-04
## LOAN_PURPOSE.fP                  -5.371943e-05
## SUPER_CONFORMING_FLAG.fY         -1.196322e-03
```

```
pred.best <- predict(glmnet.tr,x[-train_index,],s = lam.best,type='response')
#print(pred.best)
#pred.best <- ifelse(pred.best > 0.05,1,0)

#cross-validation
glmnet.cv <- cv.glmnet(x[train_index,],y[train_index],family="binomial",type.measure="auc")
plot(glmnet.cv)
```



```
pred.cv <- predict(glmnet.cv, x[-train_index, ])
print(coef(glmnet.cv,glmnet.cv$lambda.min))
```

```
## 22 x 1 sparse Matrix of class "dgCMatrix"
##                                         1
## (Intercept)                  -5.576863279
## FICO                         -0.611363870
## MORTGAGE_INSURANCE_PCT        0.086539710
## NUM_OF_UNITS                  0.009272885
## LOAN_SIZE                    -0.153866455
## ORGN_LTV                      0.383073761
## ORGN_RATE                     0.410240591
## NUM_OF_BORROWERS             -0.430524662
## HPI_ORIG                      0.104558988
## HPI_inc                      -0.113272720
## HPI_dec                       0.147704753
## FIRST_TIME_HOME_BUYER_FLAG.fN  .
## FIRST_TIME_HOME_BUYER_FLAG.fU  0.018429081
## FIRST_TIME_HOME_BUYER_FLAG.fY -0.090856369
## OCCUPANCY_STATUS.fO           0.284350783
## OCCUPANCY_STATUS.fS           0.175644621
## CHANNEL.fC                    0.023079477
## CHANNEL.fR                     .
## CHANNEL.fT                     .
## LOAN_PURPOSE.fN              -0.333538494
## LOAN_PURPOSE.fP              -0.693485082
## SUPER_CONFORMING_FLAG.fY     -0.110653748
```

```
#rmsecv = sqrt(apply((y[-train_index] - pred.cv)^2, 2, mean))
#plot(log(glmnet.cv$lambda), rmsecv, type = "b", xlab = "Log(lambda)")
pred.cvbest <- predict(glmnet.cv,x[-train_index,],s = "lambda.min")
```

ROC Curves

```
library(ROCR)
```

```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'
```

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

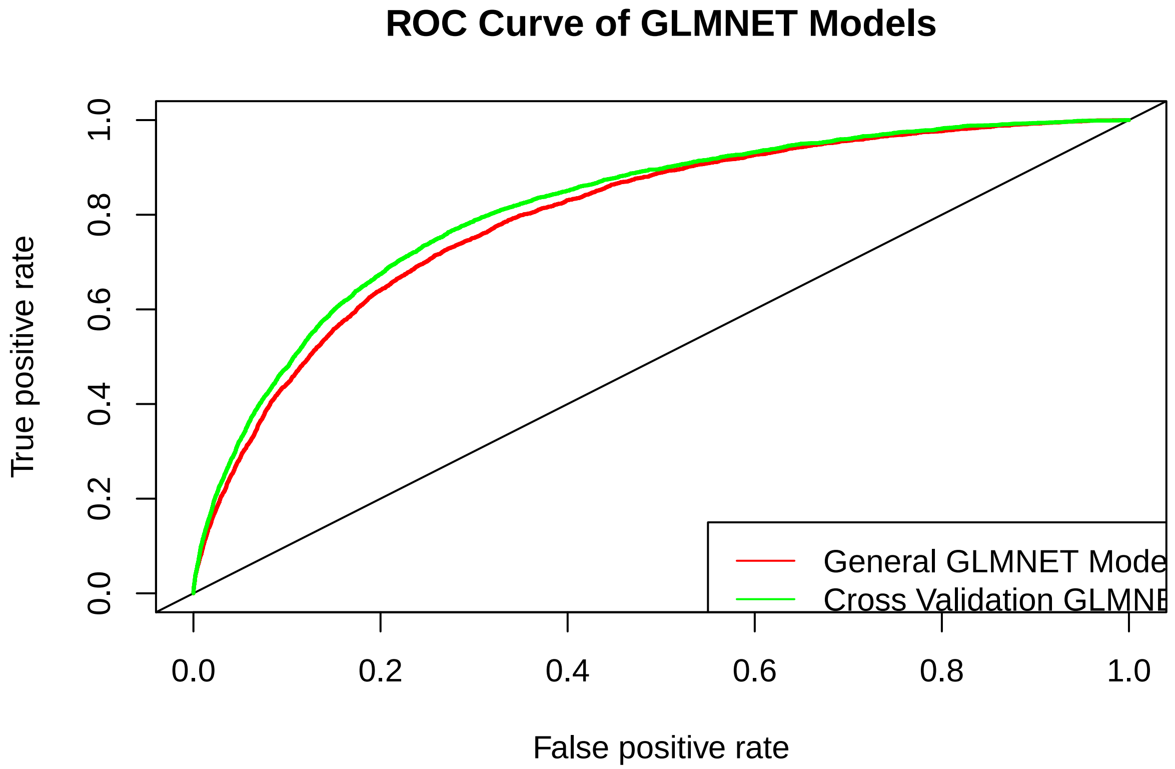
```
#ROC Curve of glmnet
pred2 <- prediction(pred.best,test$DEFAULT_IND)
perf2 <- performance(pred2,"tpr","fpr")
plot(perf2,col="red",lty=1,lwd=2,main="ROC Curve of GLMNET Models")
abline(0,1)
auc2 <- performance(pred2,"auc")
auc2 <- unlist(slot(auc2, "y.values"))
print(auc2)
```

```
## [1] 0.7979705
```

```
#cross validation
pred3 <- prediction(pred.cvbest,test$DEFAULT_IND)
perf3 <- performance(pred3,"tpr","fpr")
lines(perf3@x.values[[1]],perf3@y.values[[1]],col="green",lty=1,lwd=2)
auc3 <- performance(pred3,"auc")
auc3 <- unlist(slot(auc3, "y.values"))
print(auc3)
```

```
## [1] 0.8149484
```

```
legend(0.55,0.15, legend=c("General GLMNET Model", "Cross Validation GLMNET Model"),col=c("red", "green
```


ROC Curve of GLMNET Models
True positive rate
False positive rate
🟥 General GLMNET Mode
🟩 Cross Validation GLMNE