ROC Curve for Classification Cardio Disease

Yan Gao

10 June, 2019

Data Source and Definitions

SOURCE: Cardiovascular Disease dataset I find on Kaggle [https://www.kaggle.com/sulianova/cardiovascular-disease-dataset].

This data set consists of 70 000 records of patients data, 11 features and a binary target variable to indicate whether the victim has an cardiovascular disease or not. This11 features could be clustered as 3 types:

(1) Objective: factual information. Features of this type are quite streight forward. The age(by day), height(by cm), weight(by cm) and gender(categorical) of the victims all belong to this category.
(2) Examination: results of medical examination. This includes the Systolic blood pressure(by mmHg), Diastolic blood pressure(by mmHg), Cholesterol(categorical) and Glucose(categorical).
(3) Subjective: information given by the patient. This includes Smoking(binary), Alcohol intake(binary) and Physical activity(binary)

Exploratory Data Analysis

This dataset includes both numeric data and categorical dat. So before applicating the method of ROC curve, I want to make sure no abnormal data in this dataset

Check for abnormal numeric data

```r
summary(dat.heart[c("age","height","weight","ap_hi","ap_lo")])
```

```
##       age            height           weight           ap_hi
##  Min.   :10798   Min.   : 55.0   Min.   : 10.00   Min.   : -150.0
##  1st Qu.:17664   1st Qu.:159.0   1st Qu.: 65.00   1st Qu.:  120.0
##  Median :19703   Median :165.0   Median : 72.00   Median :  120.0
##  Mean   :19469   Mean   :164.4   Mean   : 74.21   Mean   :  128.8
##  3rd Qu.:21327   3rd Qu.:170.0   3rd Qu.: 82.00   3rd Qu.:  140.0
##  Max.   :23713   Max.   :250.0   Max.   :200.00   Max.   :16020.0
##      ap_lo
##  Min.   :  -70.00
##  1st Qu.:   80.00
##  Median :   80.00
##  Mean   :   96.63
##  3rd Qu.:   90.00
##  Max.   :11000.00
```

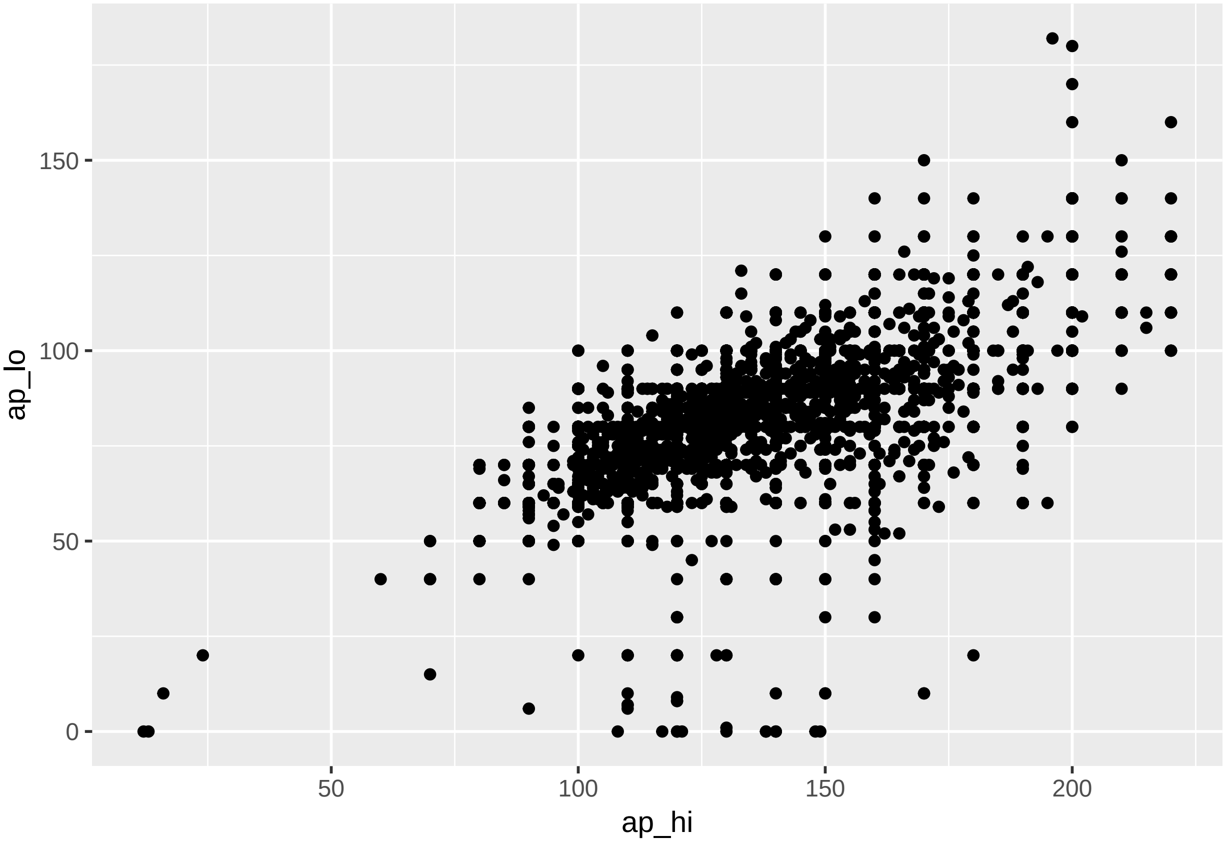
From the statistical summary, the data looks not so desiring:

(1) The age variable is counting by days, which could causes highly variance and is hard to read. It's better to change the unit to year.
(2) The minimuns of height, weight, Systolic blood pressure and Diastolic blood pressure is extremely low, which is not intended for human body. So do their maximums. But the 1st and 3rd quantile ranges are more likely to be human. The idea comes as to choose a proper quantile range to filter those abnormal data.

The Relation between the Systolic blood pressure and Diastolic blood pressure

The features are not all independent with each other. Take the relation betweeen the Systolic blood pressure and Diastolic blood pressure for example. So we should deal with the coefficient problem. An idea to achieve this is using lasso.

```
ggplot(data = dat.cardio,aes(x=ap_hi,y=ap_lo)) + geom_point()
```



Checking unbalanceness of class

If the proportions of classes are quite unbalanced, we need to be careful when spliting the data and interpreting the ROC result. From the result bellow, we can tell this is a quite balanced class variable.

```
mean(dat.cardio$cardio)
```

```
## [1] 0.4946506
```

Executive Summary

Research question: Could the given features somehow indicate whether the victim has a cardiovascular disease or not? Analyze which feature could have positive or negative influence on catching the cardiovascular. And how could we evaluate the goodness of our model.

Conclusion: I actually build a Logistic regression model with Lasso. It has some predicting ability, which reaches the AUC(area under curve) rate to be 79% with ROC curve. The effect of different features show that smoking, drinking alcohol or exercising seems lower the rate of catching the cardiovascular. This is unusually. Maybe this dataset is quite biased.

Research Method - AUC-ROC Curve

This dataset has both numeric and categorical explanatory variables and with a binary response. So, simple idea is to implement a Logistic regression to fit the model. But there could be overfitting problems, like the relation between the Systolic blood pressure and Diastolic blood pressure. The lasso regression can control the overfitting.

So after the model fitting of Logistic regression with Lasso, we usually use the fitting accuracy of the test data to denote how well the model behaves. It's intuitive if the accuracy itself is high enough. But when the accuracy is quite low, the AUC rate of ROC curve is more helpful to tell us what really happens.

ROC(Receiver Operating Characteristics) Curve

AUC-ROC curve is a performance measurement for classification problem at various thresholds settings. ROC is a probability curve and AUC represents degree or measure of separability. It tells how much model is capable of distinguishing between classes.

The ROC curve has the FPR(false positive rate) as the x-axis and TPR(true positive rate) as the y-axis. We only calculate the ROC curve from the test data and its predicted data. This means we could apply this method simply to all the classification model.

$$FPR = \frac{FP}{FP+TN}, \qquad TPR(recall) = \frac{TP}{TP+FN}$$

By control the specifity, which is the threshold between the predicted True and False. We usually set this value to be 0.5, which is a quite mindless set. The classification model actually predicts the probability distribution of different class. By tuning the specifity, we could find the best ones to seperate between different classes and that's the best evaluation specifity and corresponding accuaracy of our model. This could be applied to multiple class ocassion, which always compare the distingush ability between one class and all other classes.

Once we draw the FPR-TPR curve with different specifity, the model capability we looking at is the area under the curve(AUC). The larger the AUC is, the better the model predicts with correct classification. If $AUC = 1$, the model should predict perfectly. If $0.7 < AUC < 1$, this means the mdoel predicts well. If $AUC = 0.5$, the model should have no predict ability. Interestingly, if $AUC = 0$, the model predicts exactly reverse result, which is still quite useful.

Data satisfaction of requirements of method

As the input of cv.glmnet function must be a matrix with double type variable, all the categorical features should be converted to independent dunmmy columns. In this model, as I already removed some outliers, the numeric data doesn't need to be scaled.

```
factornames<-c('cholesterol','gluc')
mat.factor<-matrix(predict(dummyVars(~factor(gender), data = dat.cardio), newdata = dat.cardio)[,-1],nc
mat.factor<-cbind(mat.factor,matrix(predict(dummyVars(~factor(smoke), data = dat.cardio), newdata = dat
mat.factor<-cbind(mat.factor,matrix(predict(dummyVars(~factor(alco), data = dat.cardio), newdata = dat.
mat.factor<-cbind(mat.factor,matrix(predict(dummyVars(~factor(active), data = dat.cardio), newdata = da
dimnames(mat.factor)<-list(NULL,c("factor(gender)2","factor(smoke)2","factor(alco)2","factor(active)2")
for(factor.this in factornames){
  fmla<-as.formula(str_c("~ factor(",factor.this,")"))
  col.this<-which(str_detect(names(dat.cardio),factor.this))
  omitlevel<-(round(median(unlist(dat.cardio[,col.this]))))
  mat.factor<-cbind(mat.factor,predict(dummyVars(fmla, data = dat.cardio), newdata = dat.cardio)[,-omit
}
dimnames(mat.factor)[[2]]<-str_replace_all(dimnames(mat.factor)[[2]],"\\)","\\.")
dimnames(mat.factor)[[2]]<-str_replace_all(dimnames(mat.factor)[[2]],"factor\\(","")
mat.base<-as.matrix(dplyr::select(dat.cardio,age,height,weight,ap_hi,ap_lo),nrow=nrow(dat.cardio))
x.mat<-cbind(mat.base,mat.factor)
```

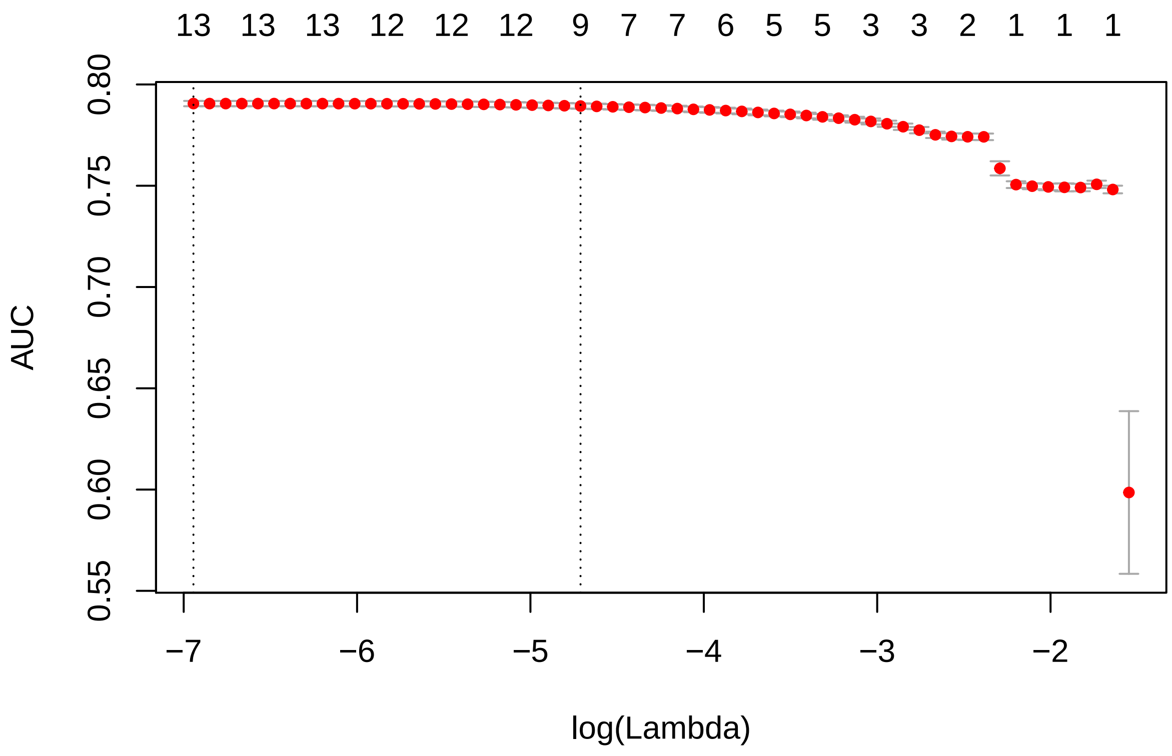
To test the model, the train-validate method is applied. So the data shoud be splitted to train and test set. As the number of disease and normal is quite balanced, normal sample without even percentage of classes should be fine.

Method application and interpration

Fit Logistic Regression Model with Lasso

I fit the data to the model, which I tuned a bit. Th type of measurement is been changed to AUC rather than the binomial deviance by default.

```
glmnet_classifier<-cv.glmnet(x = dat.train, y = dat.cardio[train==0,]$cardio,
  family = 'binomial', alpha = 1,type.measure = "auc",nfolds = 10,
  thresh = 1e-3,maxit = 1e3)
plot(glmnet_classifier)
```



By looking at the plot, the model is not strongly overfitted and could be simplfied with lambda.1se in some sense. Both the two lambda have AUC greater than 0.7 but lower than 0.8. So, it's a great model with predicting capability, but not perfect.

Interpretion with ROC Curve

Now it's time to look at the model test result.

```
coef(glmnet_classifier, s = "lambda.1se")
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                            1
## (Intercept)   -10.307206585
## age             0.044773871
## height          .
## weight          0.006636005
## ap_hi           0.053206880
## ap_lo           0.008548577
## gender.2        .
## smoke.2        -0.048989952
## alco.2         -0.010658691
## active.2       -0.103824087
## cholesterol.2   0.258419550
## cholesterol.3   0.786399521
## gluc.2          .
## gluc.3          .
```

```
paste("mean of cross validation error:",round(max(glmnet_classifier$cvm),4),collapse = " ")
```

```
## [1] "mean of cross validation error: 0.7906"
```

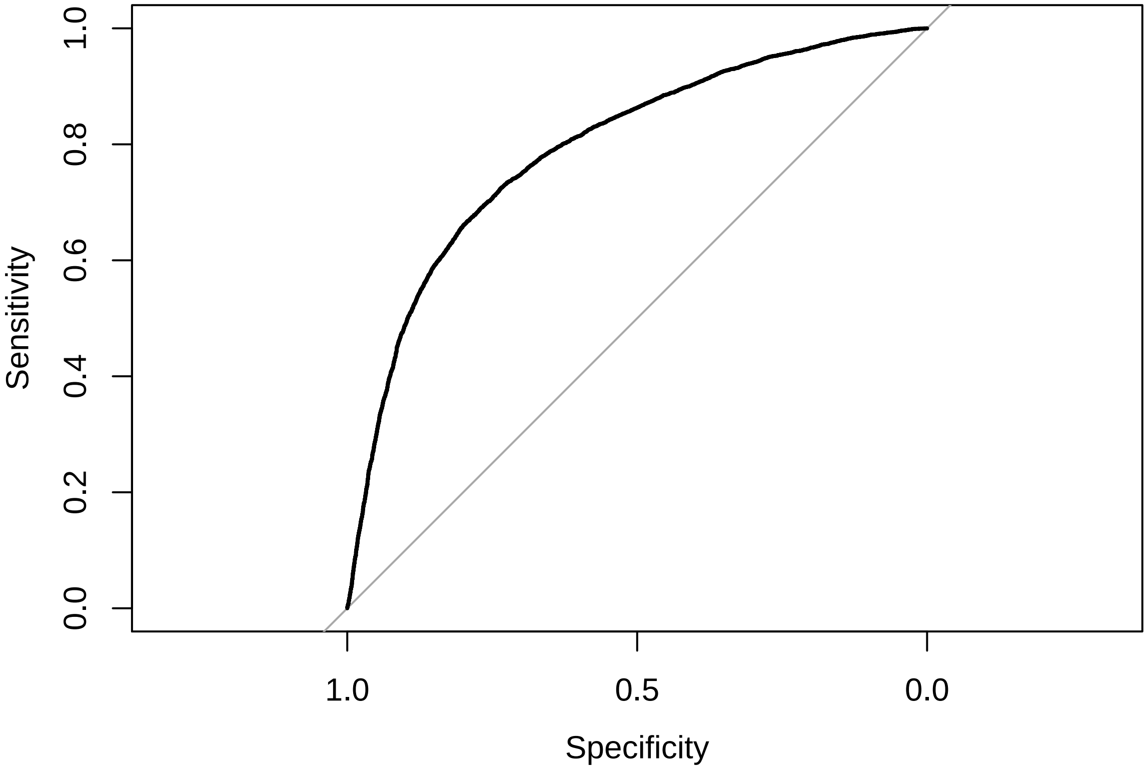
```
preds<-predict(glmnet_classifier, dat.test, type = 'response')[,1]
paste("accuracy:",round(mean((preds > 0.5) == dat.cardio[train==1,]$cardio),4),collapse = " ")
```

```
## [1] "accuracy: 0.7305"
```

```
paste("AUC:",round(glmnet:::auc(dat.cardio[train==1,]$cardio, preds),4),collapse = " ")
```

```
## [1] "AUC: 0.7932"
```

```
## ROC curve
plot(roc(dat.cardio[train==1,]$cardio ~ preds))
```



From the model coefficient, we can tell that older people are more likely to have cardiovascular disease. However, smoking, drinking alcohol and exercising could lower the chance. This is unusual. An reason for this is the dataset could be biased. The Subjective features offered by victims may not be true. Also the record age is not the actual time they got the disease, but the time they knew they got the disease. But the results of medical examination do tell us how to get the pattern of this disease intuitively. Cholesterol must be the most important feature we should concern.

The accuracy is about 73% which is great. But a 79% AUC-ROC actually tell us how great the model is to distinguish between different classes.

Summary

ROC curve is actually a really good way to access the model seperate capability among different classed. Accuracy can tell us the model behavior with certain specifity, but the ROC curve can tell us the model performance with different specifity, which is great. It's a good accessment for classification model. It only requires test data and predicted data, which is easy to implement.

Reference

[1] *Understanding AUC - ROC Curve*, Sarang Narkhede, https://towardsdatascience.com/understanding-auc-roc-curve-68b2303

[2] *Understanding Understanding Confusion Matrix*, Sarang Narkhede, https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

[3] *Some R Packages for ROC Curve*, Joseph Rickert, https://rviews.rstudio.com/2019/03/01/some-r-packages-for-roc-curves/

Appendix

A good ROC curve example with small dataset

Data Source: https://www.kaggle.com/ronitf/heart-disease-uci

Target: Train a model to predict whether the patients have heart disease from the given patients' information

Method: ROC curve, Logistic regression, Lasso

```
dat.heart<-read.csv("./heart.csv")
names(dat.heart)[1]<-"age"

factornames<-c('restecg','thal')

mat.factor<-matrix(predict(dummyVars(~factor(sex), data = dat.heart), newdata = dat.heart)[,-1],ncol=1)
mat.factor<-cbind(mat.factor,matrix(predict(dummyVars(~factor(fbs), data = dat.heart), newdata = dat.hea
mat.factor<-cbind(mat.factor,matrix(predict(dummyVars(~factor(exang), data = dat.heart), newdata = dat.
dimnames(mat.factor)<-list(NULL,c("factor(sex)2","factor(fbs)2","factor(exang)2"))
for(factor.this in factornames){
  fmla<-as.formula(str_c("~ factor(",factor.this,")"))
  col.this<-which(str_detect(names(dat.heart),factor.this))
  omitlevel<-(round(median(unlist(dat.heart[,col.this]))))
  mat.factor<-cbind(mat.factor,predict(dummyVars(fmla, data = dat.heart), newdata = dat.heart)[,-omitle
}
dimnames(mat.factor)[[2]]<-str_replace_all(dimnames(mat.factor)[[2]],"\\)","\\.")
dimnames(mat.factor)[[2]]<-str_replace_all(dimnames(mat.factor)[[2]],"factor\\(","")
mat.base<-as.matrix(dplyr::select(dat.heart,age,cp,trestbps,chol,oldpeak,slope,ca,thalach),nrow=nrow(da
x.mat<-cbind(mat.base,mat.factor)

set.seed(19960909)
n<-nrow(x.mat)
train<-sample(rep(0:1,c(round(n*.8),n-round(n*.8))),n)
dat.train<-x.mat[train==0,]
dat.test<-x.mat[train==1,]
```

```
glmnet_classifier<-cv.glmnet(x = dat.train, y = dat.heart[train==0,]$target,
                             family = 'binomial',
                             alpha = 1,
                             type.measure = "auc",
                             nfolds = 3,
                             #thresh = 1e-3,
                             maxit = 1e3
                            )
```

```
plot(glmnet_classifier)
round(max(glmnet_classifier$cvm), 4)
```

```
preds<-predict(glmnet_classifier, dat.test, type = 'response')[,1]
mean((preds > 0.5) == dat.heart[train==1,]$target)
glmnet:::auc(dat.heart[train==1,]$target, preds)
```

```
g <- roc(dat.heart[train==1,]$target ~ preds)
plot(g)
```

An bad ROC curve example

Data Source: https://www.kaggle.com/aaron7sun/stocknews

Target: Train a model to predict the relation of Dow Jones Industrial Average from the Reddit WorldNews Channel headlines.

Method: ROC curve, Logistic regression, PCA, text2vec.

```
rm(list=ls())
dat.djia<-read.csv("./DJIA_table.csv")
dat.djia$Open<-as.numeric(dat.djia$Open)
dat.djia$Close<-as.numeric(dat.djia$Close)
dat.djia$y<-sapply(dat.djia$Close >= dat.djia$Open, FUN=function(x){if (x) 1 else 0})

dat.headline<-read.csv("./RedditNews.csv")
dat.headline<-dat.headline %>% group_by(Date)
dat.headlines<-dat.headline %>% summarise(
  News = paste(News,collapse=" ")
)
dat<-merge(dat.djia,dat.headlines,by="Date")[c("y","News")]
rm(dat.djia,dat.headline,dat.headlines)

set.seed(19960909)
n<-nrow(dat)
train<-sample(rep(0:1,c(round(n*.8),n-round(n*.8))),n)
dat.train<-dat[train==0,]
dat.test<-dat[train==1,]
```

```
fun_tokenize<-function(df){
  itoken(df$News, preprocessor = tolower, tokenizer = word_tokenizer, progressbar = FALSE)
}
train.tokenize<-fun_tokenize(dat.train)
test.tokenize<-fun_tokenize(dat.test)
stwords<-c("a", "about", "above", "above", "across", "after", "afterwards", "again", "against", "all",
vocab<-create_vocabulary(train.tokenize,stopwords = stwords)

vectorizer<-vocab_vectorizer(vocab)

train.mtx<-create_dtm(train.tokenize, vectorizer)
test.mtx<-create_dtm(test.tokenize,vectorizer)
# define tfidf model
tfidf<-TfIdf$new()
# fit model to train data and transform train data with fitted model
train.tfidf<-fit_transform(train.mtx, tfidf)
# tfidf modified by fit_transform() call!
# apply pre-trained tf-idf transformation to test data
test.tfidf<-create_dtm(test.tokenize, vectorizer)
test.tfidf<-transform(test.tfidf, tfidf)
require(caret)
require(e1071)
```

```
trans = preProcess(as.data.frame(as.matrix(train.tfidf)),
                   method=c("BoxCox", "center",
                            "scale", "pca"))
PC = predict(trans, as.data.frame(as.matrix(train.tfidf)))
```

```
PC2 = predict(trans, as.data.frame(as.matrix(test.tfidf)))
```

```
glmnet_classifier<-cv.glmnet(x = as.matrix(PC), y = dat.train$y,
                              family = 'binomial',
                              alpha = 1,
                              type.measure = "auc",
                              #nfolds = 10,
                              #thresh = 1e-3,
                              #maxit = 1e3
                             )
```

```
plot(glmnet_classifier)
round(max(glmnet_classifier$cvm), 4)
```

```
preds<-predict(glmnet_classifier, as.matrix(PC2), type = 'response')[,1]

glmnet:::auc(dat.test$y, preds)
```

```
g <- roc(dat.test$y ~ preds)
plot(g)
```