Classification Using Logistic regression

Department of Statistics, University of South Carolina

Stat 705: Data Analysis II

Prediction Rules

$$\text{logit}(\pi_i) = \log\left(\frac{\pi_i}{1-\pi_i}\right) = \mathbf{X}\boldsymbol{\beta}$$

It seems intuitive to base prediction of an outcome given $\mathbf{x}$ upon the following rule:

$$If\, \hat{\pi}_{\mathbf{x}} > 0.5, \text{then } \hat{Y}_{\mathbf{x}} = 1; \text{ else } \hat{Y}_{\mathbf{x}} = 0.$$

We can create a 2-way table of $\hat{Y}_{\mathbf{x}}$ vs. $Y_{\mathbf{x}}$ for any given threshold value of $\hat{\pi}_{\mathbf{x}}$ and readily visualize two types of classification errors: $\hat{Y}_{\mathbf{x}} = 1$ when $Y_{\mathbf{x}} = 0$, and $\hat{Y}_{\mathbf{x}} = 0$ when $Y_{\mathbf{x}} = 1$. A best classification rule would minimize the sum of these classification errors.

Decision Boundary

To predict the outcome of a new input $x \in \mathbb{R}^p$, we form

$$\widehat{\pi_x} = \frac{e^{\mathbf{X}\beta}}{1 + e^{\mathbf{X}\beta}},$$

and then predict the associated class according

$$\widehat{f(x)} = \begin{cases} 0 & \widehat{\pi_x} \leq 0.5 \\ 1 & \widehat{\pi_x} > 0.5 \end{cases}$$

Equivalently,

$$\widehat{f(x)} = \begin{cases} 0 & \mathbf{X}\widehat{\beta} \leq 0 \\ 1 & \mathbf{X}\widehat{\beta} > 0 \end{cases}$$

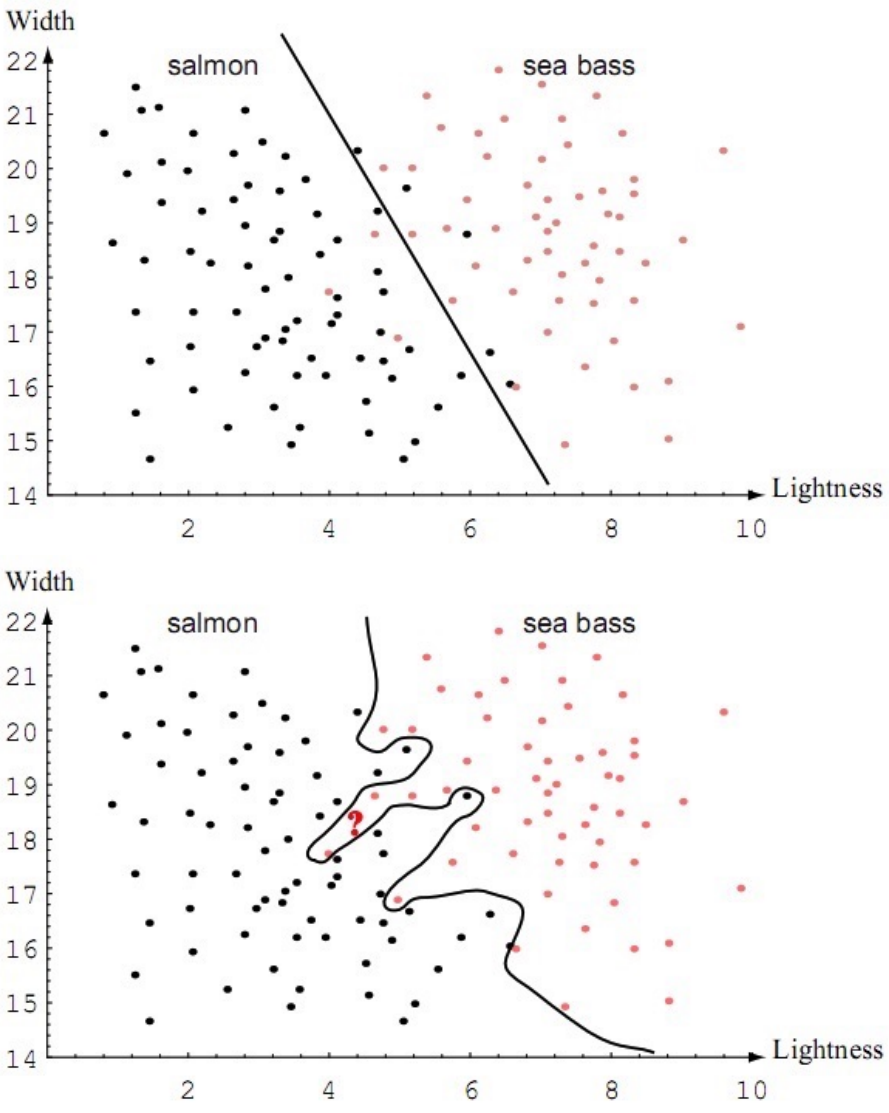
Decision Boundary

The decision boundary is the set of all $x \in \mathbb{R}^p$ such that

$$\mathbf{X}\widehat{\boldsymbol{\beta}} = \widehat{\beta}_0 + \widehat{\beta}_1 x_1 + \widehat{\beta}_2 x_2 + ... + \widehat{\beta_p} x_p = 0.$$

This is a point when $p = 1$; it is a line when $p = 2$, and in general it is a $(p-1)-$ dimensional subspace. We would therefore say that logistic regression has a *linear decision boundary*. This is because the above equation is linear in x.

Decision Boundary



Prediction rules–Example

Snoring Data (Agresti 2013)

Snoring (x)	$Y_x = 0$	$Y_x = 1$	$\hat{\pi}_x$
0	1355	24	0.0205
2	603	35	0.0443
4	192	21	0.0931
5	224	30	0.1324

Prediction rules–Example

Assume our threshold is 0.0205. Then if $\hat{\pi}_x > 0.0205$, $\hat{Y}_x = 1$; else $\hat{Y}_x = 0$.

	$Y_x = 0$	$Y_x = 1$	
$\hat{Y}_x = 0$	1355	24	1379
$\hat{Y}_x = 1$	603+192+224=1019	35+21+30=86	1105
	2374	110	

From the table, we can compute

$$\hat{P}(\hat{Y}_x = 0 | Y_x = 1) = \frac{24}{110} = 0.218 = 1 - \hat{P}(\hat{Y}_x = 1 | Y_x = 1)$$

$$\hat{P}(\hat{Y}_x = 1 | Y_x = 0) = \frac{1019}{2374} = 0.429 = 1 - \hat{P}(\hat{Y}_x = 0 | Y_x = 0)$$

Error Rate= $\frac{24+1019}{110+2374} \approx 0.42$

ROC Curve

ROC (Receiver Operating Characteristic) curves are created by plotting the *sensitivity* ($P(\hat{Y}_{\mathbf{x}} = 1 | Y_{\mathbf{x}} = 1)$) versus 1-*specificity* ($1 - P(\hat{Y}_{\mathbf{x}} = 0 | Y_{\mathbf{x}} = 0)$) over ordered unique values of $\hat{\pi}_{\mathbf{x}}$.

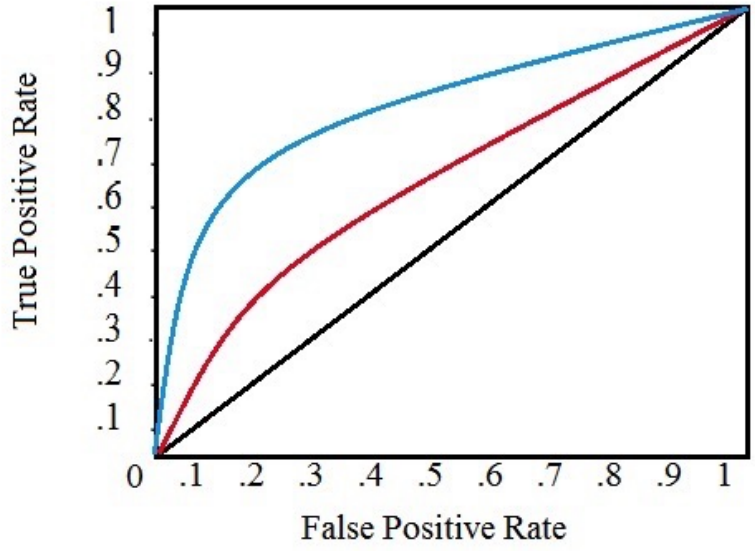
- The area under the ROC curve (AUC) is a measure of the model's predictive power.

ROC Curve

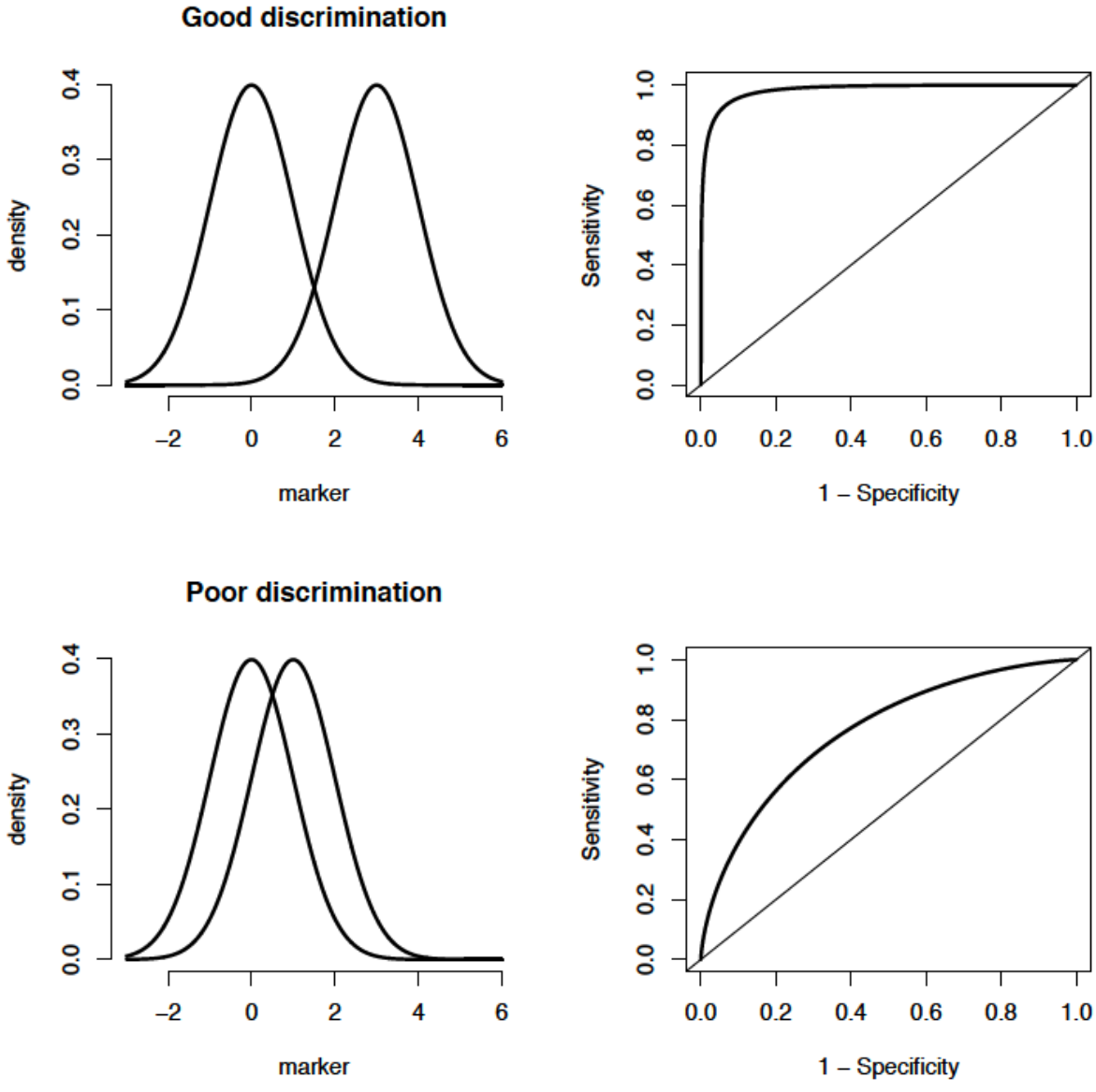
	Truth	
Test	Disease	No disease
Positive	TP	FP
Negative	FN	TN

$$\text{Sensitivity} = \frac{TP}{TP + FN} = P(\widehat{Y} = 1 | Y = 1)$$

$$\text{Specificity} = \frac{TN}{TN + FP} = P(\widehat{Y} = 0 | Y = 0)$$



- An ROC curve that is a diagonal line (sensitivity = 1-specificity) corresponds to a uninformative test where a positive test corresponds to flipping a coin with success probability equal to the sensitivity.
- The ROC curve always starts at (0, 0) and ends at (1,1)
- Curves that are higher represent better tests
- A marker with an ROC curve that is uniformly below the diagonal line is worse than guessing, and hence can be improved upon by taking the opposite decision


Good discrimination
density
0.0
0.1
0.2
0.3
0.4
−2
0
2
4
6
marker
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1 – Specificity
Poor discrimination
density
marker
Sensitivity
1 – Specificity

AUC

- Better markers (tests/models) have higher AUCs
- The largest possible AUC is 1, the smallest (for an informative test) is .5
- If the ROC curve for one test dominates another, then its AUC will also be larger
- The converse is not true, a larger AUC does not imply a uniformly better test

Fitting logistic regression models (pp. 564–565)

The data are $(\mathbf{x}_j, Y_{\cdot j})$ for $j = 1, \ldots, c$.
The model is

$$Y_{\cdot j} \sim \text{bin}\left(n_j, \frac{e^{\boldsymbol{\beta}'\mathbf{x}_j}}{1 + e^{\boldsymbol{\beta}'\mathbf{x}_j}}\right).$$

The pmf of $Y_{\cdot j}$ in terms of $\boldsymbol{\beta}$ is

$$p(y_j; \boldsymbol{\beta}) = \begin{pmatrix} n_j \\ y_{\cdot j} \end{pmatrix} \left[\frac{e^{\boldsymbol{\beta}'\mathbf{x}_j}}{1 + e^{\boldsymbol{\beta}'\mathbf{x}_j}}\right]^{y_{\cdot j}} \left[1 - \frac{e^{\boldsymbol{\beta}'\mathbf{x}_j}}{1 + e^{\boldsymbol{\beta}'\mathbf{x}_j}}\right]^{n_j - y_{\cdot j}}.$$

The likelihood is the product of all N of these and the log-likelihood simplifies to

$$L(\boldsymbol{\beta}) = \sum_{k=1}^{p} \beta_k \sum_{j=1}^{c} y_{\cdot j} x_{jk} - \sum_{j=1}^{c} \log\left[1 + \exp\left(\sum_{k=1}^{p} \beta_k x_{jk}\right)\right] + \text{constant}.$$

German Credit Data Set

Data from Dr. Hans Hofmann of the University of Hamburg. These data have two classes for the credit worthiness: good or bad. There are predictors related to attributes, such as: checking account status, duration, credit history, purpose of the loan, amount of the loan, savings ...

```
>library(caret)
>data(GermanCredit)
> str(GermanCredit[, 1:10])
'data.frame': 1000 obs. of  10 variables:
 $ Duration                : int  6 48 12 42 24 36 24 36 12 30 ...
 $ Amount                  : int  1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 ...
 $ InstallmentRatePercentage: int  4 2 2 2 3 2 3 2 2 4 ...
 $ ResidenceDuration       : int  4 2 3 4 4 4 4 2 4 2 ...
 $ Age                     : int  67 22 49 45 53 35 53 35 61 28 ...
 $ NumberExistingCredits   : int  2 1 1 1 2 1 1 1 1 2 ...
 $ NumberPeopleMaintenance : int  1 1 2 2 2 2 1 1 1 1 ...
 $ Telephone               : num  0 1 1 1 1 0 1 0 1 1 ...
 $ ForeignWorker           : num  1 1 1 1 1 1 1 1 1 1 ...
 $ Class                   : Factor w/ 2 levels "Bad","Good": 2 1 2 2 1 2 2 2 2 1 ...
```

Training Data

```
> samp<-sample(1:nrow(GermanCredit), 0.6*nrow(GermanCredit))
>
> train<-GermanCredit[samp,]
> testing<-GermanCredit[-samp,]
>
> fit1<- glm(Class ~ Age + ForeignWorker + Property.RealEstate + Housing.Own +
+ CreditHistory.Critical, data=train, family="binomial")
>
> fit2<- glm(Class ~ Age + ForeignWorker, data=train, family="binomial")
>
>
> library(lmtest)
> lrtest(fit1, fit2)
Likelihood ratio test

Model 1: Class ~ Age + ForeignWorker + Property.RealEstate + Housing.Own +
    CreditHistory.Critical
Model 2: Class ~ Age + ForeignWorker
  #Df  LogLik Df  Chisq Pr(>Chisq)
1   6 -346.84
2   3 -367.14 -3 40.608  7.917e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Variable Importance

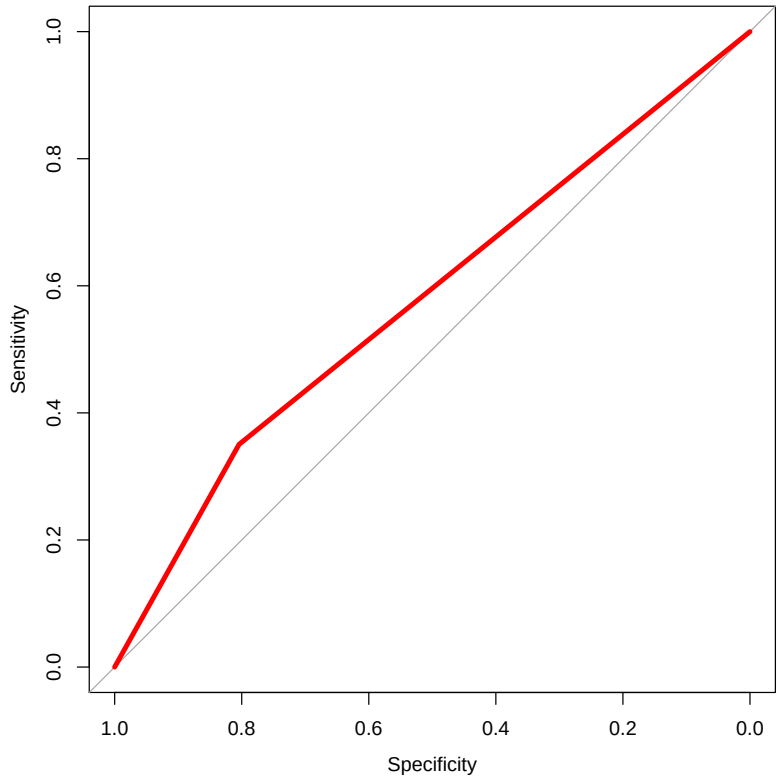
```
> library(e1071)
>
> mod_fit <- train(Class ~ Age + ForeignWorker + Property.RealEstate + Housing.
+ CreditHistory.Critical, data=train, method="glm", family="binomial")
> varImp(mod_fit)
glm variable importance

                       Overall
Housing.Own             100.00
Property.RealEstate      91.23
CreditHistory.Critical   57.99
Age                      14.36
ForeignWorker             0.00
```

R Code

```
> pred = predict(fit1, newdata=testing)
> accuracy <- table(pred, testing[,"Class"])
>
> pred = predict(fit1, newdata=testing)
> predb<-ifelse(pred > 0.5, 1, 0)
> Class<-factor(ifelse(testing$Class=="Good", 1, 0))
> confusionMatrix(factor(predb, levels=0:1), Class)
Confusion Matrix and Statistics
          Reference
Prediction   0   1
         0  51  95
         1  60 194

               Accuracy : 0.6125
                 95% CI : (0.5628, 0.6605)
    No Information Rate : 0.7225
    P-Value [Acc > NIR] : 0.999999

                  Kappa : 0.1192
 Mcnemar's Test P-Value : 0.006315
            Sensitivity : 0.4595
            Specificity : 0.6713
         Pos Pred Value : 0.3493
         Neg Pred Value : 0.7638
             Prevalence : 0.2775
         Detection Rate : 0.1275
   Detection Prevalence : 0.3650
      Balanced Accuracy : 0.5654

       'Positive' Class : 0
```

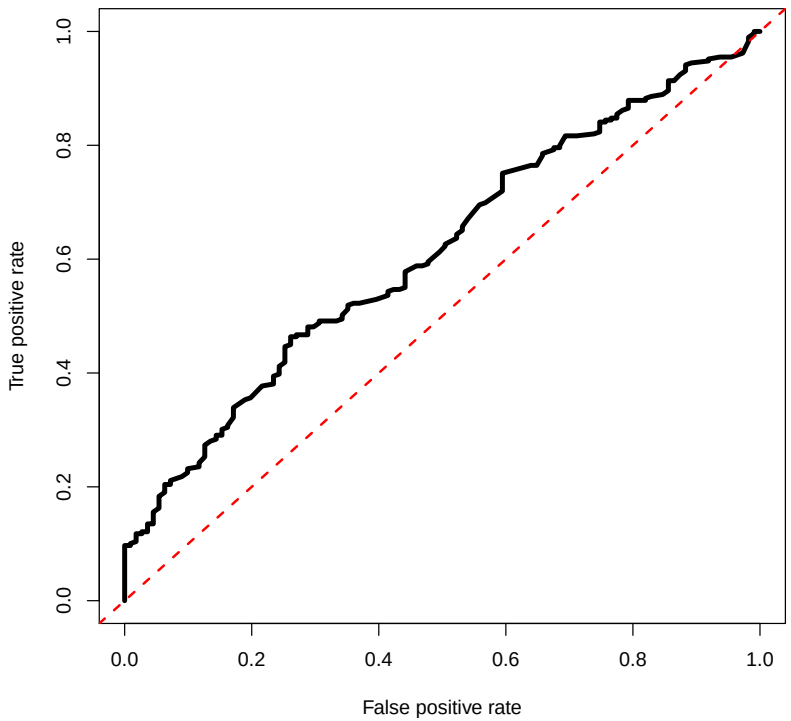
ROC curve

```
> library(pROC)
> f1<-roc(Class ~ CreditHistory.Critical, data=train)
> plot(f1, col="red")
> f1
Call:
roc.formula(formula = Class ~ CreditHistory.Critical, data = train)
Data: CreditHistory.Critical in 189 controls (Class Bad) < 411 cases (Class Good).
Area under the curve: 0.5773
```



R Code

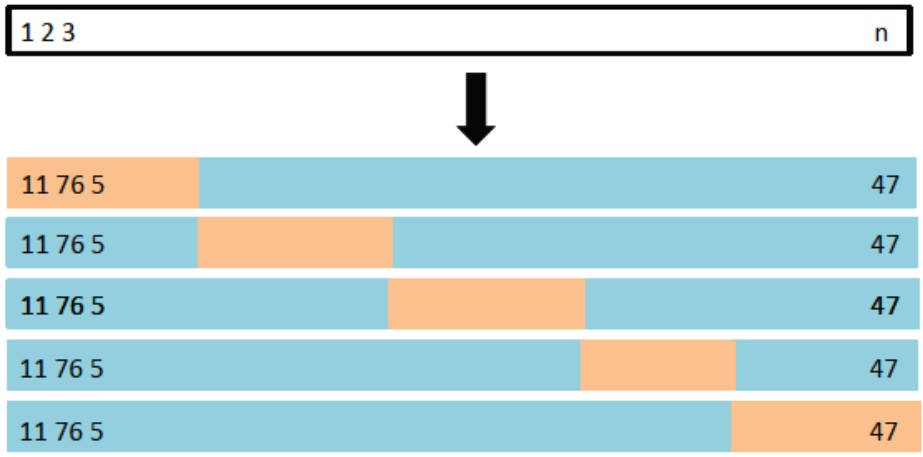
```
> library(ROCR)
> prob<-predict(fit1, newdata=testing, type="response")
> pred<-prediction(prob, testing$Class)
> perf<-performance(pred, measure="tpr", x.measure="fpr")
> auc<-performance(pred, measure="auc")
> auc<-auc@y.values[[1]]
> auc
[1] 0.6122074
> plot(perf)
>abline(a=0, b=1, lty=2, col=2)
```



Optimal Cutoff

```
> opt.cut = function(perf, pred){
+     cut.ind = mapply(FUN=function(x, y, p){
+         d = (x - 0)^2 + (y-1)^2
+         ind = which(d == min(d))
+         c(sensitivity = y[[ind]], specificity = 1-x[[ind]],
+             cutoff = p[[ind]])
+     }, perf@x.values, perf@y.values, pred@cutoffs)
+ }
> print(opt.cut(perf, pred))
                  [,1]
sensitivity 0.4809689
specificity 0.7117117
cutoff      0.7156623
```

Cross-Validation



- Calculate cross-validated error rate
- Calculate cross-validated AUC