350c

Goodness of Fit Measures for Binary Outcome Models

Justin Grimmer (Jens Hainmueller's Slides)

Stanford

Outline

Outline

Regression Models for Binary Outcomes

- Often we are interested in binary outcomes Y that take on two values 1 or 0 depending on whether an event of interest occurred or not (vote or not, war or peace, etc).
- Three widely used regression models for binary outcomes:
  - Linear Probability Model (LPM)
  - Probit regression
  - Logistic regression
- Goals
  - Model the probability of the event occurring as a function of the independent variables
  - Estimate the probability of the event occurring given certain values of the independent variables
  - Predict how the probability of the event occurring changes for different values of the independent variables
  - Classify the observations into categories based on the probability of the event occurring

Assessing Goodness of Fit

- How do we validate that our fitted model is any good? How well does our model fit the data?
- Two core concepts to assess model fit for binary outcome models:
- Calibration
    - A model is well calibrated if the observed risk matches the predicted risk (e.g. for observations with a predicted probability of .2 there should be approximately 20% of $Y = 1$ observed outcomes)
    - Well calibrated model ensures that in future samples the observed proportions of positives will be close to the estimated probabilities
- Discrimination
    - A model has high discrimination ability if it allows us to discriminate between low and high risk observations
    - This means observations with $Y = 1$ should have high and observations with $Y = 0$ should have low predicted probabilities $\hat{Y}$
    - But we don't want a lot of $\hat{Y}$ close to .5 if outcomes are all $Y = 1$ or $Y = 0$ (sharpness)

Outline

Calibration

- In a well calibrated model, the observed risk matches the predicted risk such that in future samples the observed proportion of positives will match the predicted probabilities
- Can check this for any model prediction using calibration plots:
  - Bin predicted probabilities $\hat{Y}$ into 10 equal interval bins $g_k \in ([0, .1], (.1, .2], ..., (.9, 1])$
  - In each bin g_k compute the average predicted probability $\bar{\hat{Y}}_{\in g_k}$ and the fraction of observed positives $\bar{Y}_{\in g_k}$
  - Plot the two vectors of averages against each other
- If the model is well calibrated, the binned averages should trace the identity line

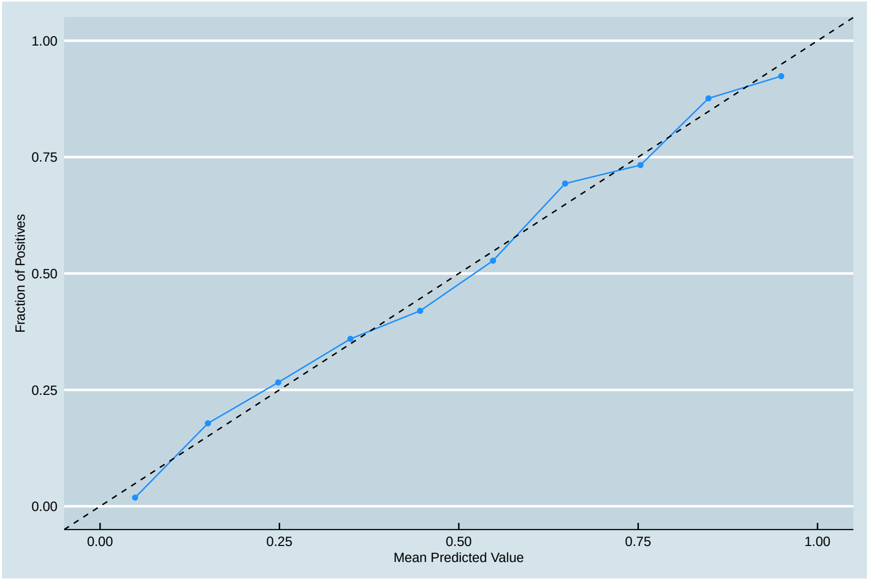
Calibration Plot: Correct Model

R Code

```
> n <- 1000
> x <- rnorm(n)
> xb <- 2*x
> pr <- exp(xb)/(1+exp(xb))
> y <- 1*(runif(n) < pr)
> d <-data.frame(y,x)
>
> logit <- glm(y~x, data=d, family=binomial(link="logit"))
> summary(logit)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.05078    0.08164  -0.622    0.534
x            1.99622    0.12797  15.599   <2e-16 ***
>
> # cut into 10 bins
> d$predprobs <- logit$fitted
> d$pptbin  <- cut(d$predprobs,seq(from=0,to=1,by=0.1),include.lowest = TRUE)
> # compute average predicted probs
> rel.logit <- (d %>% group_by(pptbin) %>% summarise(out = mean(y)))
> # compute observed averages
> mpv       <- (d %>% group_by(pptbin) %>% summarise(out = mean(predprobs)))
>
> pdf <- data.frame(fop = rel.logit$out,
+                   mpv = mpv$out)
>
> ggplot(pdf,aes(x=mpv,y=fop)) + geom_point(color="dodgerblue") + geom_line(color="dodgerblue") +
+   theme_economist(dkpanel = T) + geom_abline(intercept = 0,slope=1,linetype=2) +
+   xlim(0,1) + ylim(0,1) + xlab("Mean Predicted Value") + ylab("Fraction of Positives")
```

Calibration Plot: Correct Model

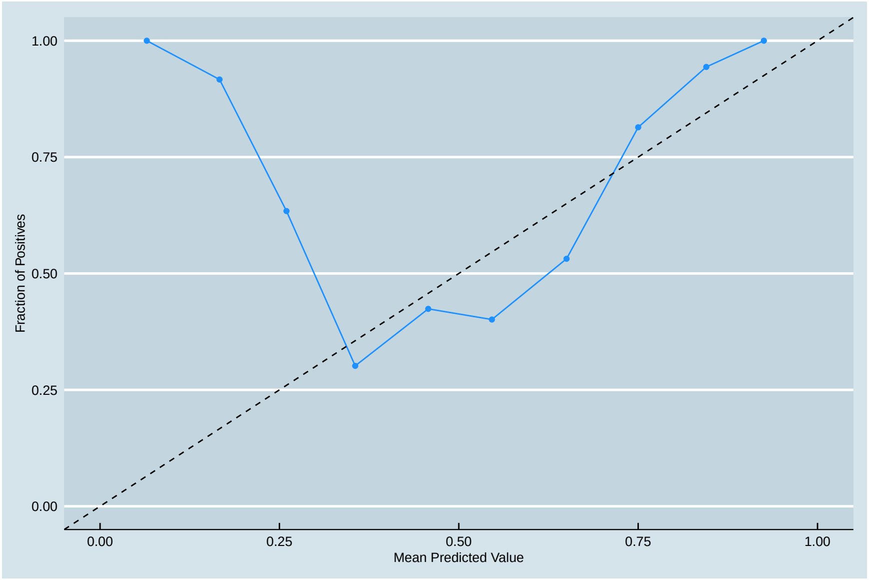


Calibration Plot: Miss-specified Model

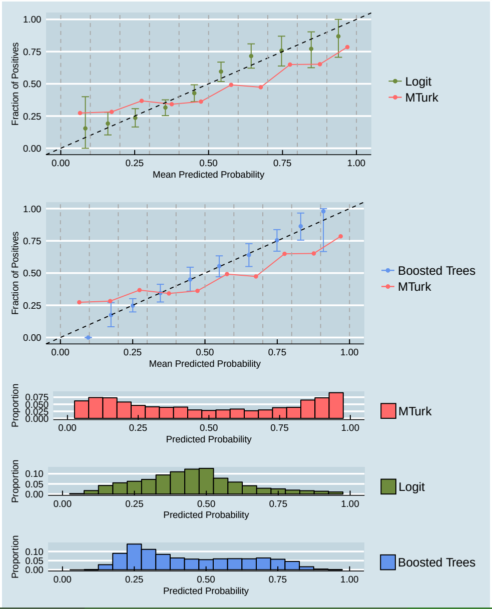
R Code

```
> xb <- -2*x + 1.25*x^2
> pr <- exp(xb)/(1+exp(xb))
> y <- 1*(runif(n) < pr)
> d <-data.frame(y,x)
>
> logit <- glm(y~x, data=d, family=binomial(link="logit"))
> summary(logit)
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.61150    0.07215   8.475   <2e-16 ***
x           -0.92570    0.08506 -10.883   <2e-16 ***
> d$predprobs <- logit$fitted
> head(d)
  y           x predprobs
1 1  0.07894658 0.6314440
2 0 -0.21551746 0.6923231
3 0 -0.25559249 0.7001684
4 1  0.77489539 0.4735684
5 0  2.62295669 0.1398448
6 0  0.50200285 0.5366323
>
> # cut predicted probabilitie sinto 10 bins
> d$pptbin  <- cut(d$predprobs,seq(from=0,to=1,by=0.1),include.lowest = TRUE)
> rel.logit <- (d %>% group_by(pptbin) %>% summarise(out = mean(y)))
> mpv       <- (d %>% group_by(pptbin) %>% summarise(out = mean(predprobs)))
>
> pdf <- data.frame(fop = rel.logit$out,
+                   mpv = mpv$out)
> ggplot(pdf,aes(x=mpv,y=fop)) + geom_point(color="dodgerblue") + geom_line(color="dodgerblue") +
+   theme_economist(dkpanel = T) + geom_abline(intercept = 0,slope=1,linetype=2) +
+   xlim(0,1) + ylim(0,1) + xlab("Mean Predicted Value") + ylab("Fraction of Positives")
```

Example with Miss-specified Model



Predicting Recidivism



Outline

Discrimination

- Good calibration of a model is not sufficient for model validation
  - A silly model that predicts the same predicted probability for every observation will have good calibration (in future samples the observed proportion will be close to the predicted probability)
  - However, such a model would not be helpful since it does not help us to discriminate between high and low risk observations
- Need to check discrimination ability of a model
  - Discrimination is high if $Y = 1$ observations correspond with high $\hat{Y}$ values and observations with $Y = 0$ correspond to low $\hat{Y}$ values
- Discrimination can be captured in various ways including classification accuracy, sensitivity, specificity, precision, recall, etc.
- Many metrics are used and different fields have different preferences

Outline

Classification accuracy

For classification we choose a cutoff value on the probability scale, say $c = .5$, and classify all predicted values above c as predicting a positive, and all below c as predicting a negative

The contingency table or confusion matrix for a binary classifier is given by:

	Observed	
	Positive ($Y = 1$)	Negative ($Y = 0$)
Predicted positive ($\hat{Y} > c$)	True Positive (TP)	False Positive (FP)
Predicted negative ($\hat{Y} < c$)	False Negative (FN)	True Negative (TN)
	Total Positive (P)	Total Negative (N)

- Accuracy: $ACC = (TP + TN)/(P + N)$
  - Percent correctly classified of all predictions

Recidivism Example: Classification accuracy

R Code

```
> source("http://pcwww.liv.ac.uk/~william/R/crosstab.r")
> d <- read.csv("bail.csv")
> d$y <- d$two_year_recid
> crosstab(d,row.vars="y")

y      Count Total %
  0    524.0    52.4
  1    476.0    47.6
  Sum 1000.0   100.0
>
> # model 1
> logit1  <- glm(y~sex+age+as.factor(race),data=d)
> summary(logit1)

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       0.653519   0.055209  11.837  < 2e-16 ***
sex              -0.162660   0.038255  -4.252 2.32e-05 ***
age              -0.006605   0.001296  -5.096 4.16e-07 ***
as.factor(race)2  0.162315   0.033086   4.906 1.09e-06 ***
as.factor(race)3 -0.012272   0.057799  -0.212    0.832
as.factor(race)4 -0.195496   0.183733  -1.064    0.288
as.factor(race)5 -0.475177   0.480880  -0.988    0.323
---

> d$yhat1 <- logit1$fitted
> d$ypred1 <- as.numeric(d$yhat1>.5)
```

Recidivism Example: Classification accuracy

R Code

```
> # model 2
> logit2   <- glm(y~sex+age+as.factor(race)+log(1+priors_count),data=d)
> summary(logit2)

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)            0.553299   0.053324  10.376  < 2e-16 ***
sex                   -0.104176   0.036774  -2.833  0.00471 **
age                   -0.007815   0.001237  -6.318 3.99e-10 ***
as.factor(race)2       0.089658   0.032197   2.785  0.00546 **
as.factor(race)3      -0.021096   0.054915  -0.384  0.70094
as.factor(race)4      -0.070564   0.174957  -0.403  0.68680
as.factor(race)5      -0.573363   0.456928  -1.255  0.20984
log(1 + priors_count)  0.166681   0.016016  10.407  < 2e-16 ***
---

> d$yhat2  <- logit2$fitted
> d$ypred2 <- as.numeric(d$yhat2>.5)
```

Recidivism Example: Classification accuracy

R Code

```
> # model 1
> crosstab(d,row.vars="ypred1",col.vars="y",type="frequency",percentages=F)
       y    0    1  Sum
ypred1
0         352  204  556
1         172  272  444
Sum       524  476 1000
> ACCm1 <- mean(d$ypred1 == d$y)
> ACCm1
[1] 0.624
>
> # model 2
> crosstab(d,row.vars="ypred2",col.vars="y",type="frequency",percentages=F)
       y    0    1  Sum
ypred2
0         378  178  556
1         146  298  444
Sum       524  476 1000
> ACCm2 <- mean(d$ypred2 == d$y)
> ACCm2
[1] 0.676
>
> # all negative model (null model)
> ACCneg <- mean(0 == d$y)
> ACCneg
[1] 0.524
>
> # percent increase in accuracy over null model
> ((ACCm2-ACCneg)/ACCneg)*100
[1] 29.00763
```

Classification accuracy (Cutpoint .85)

R Code

```
> # model 1
> d$ypred1 <- as.numeric(d$yhat1>.85)
>
> # model 2
> d$ypred2 <- as.numeric(d$yhat2>.85)
>
> # model 1
> crosstab(d,row.vars="ypred1",col.vars="y",type="frequency",percentages=F)
       y    0    1  Sum
ypred1
0         524  476 1000
Sum       524  476 1000
> ACCm1 <- mean(d$ypred1 == d$y)
> ACCm1
[1] 0.524
>
> # model 2
> crosstab(d,row.vars="ypred2",col.vars="y",type="frequency",percentages=F)
       y    0    1  Sum
ypred2
0         521  455  976
1           3   21   24
Sum       524  476 1000
> ACCm2 <- mean(d$ypred2 == d$y)
> ACCm2
[1] 0.542
>
> # all negative model (null model)
> ACCneg <- mean(0 == d$y)
> ACCneg
[1] 0.524
```

Is classification accuracy sufficient?

R Code

```
> library(foreign)
> d <- read.dta("repdata.dta")
> d <- d[d$year==1982,]
> d <- na.omit(d[,c("war","lpop","polity2","gdpen","ethfrac","lmtnest","Oil","cname")])
>
> d$y <- d$war
>
> crosstab(d,row.vars="y",type="frequency",percentages=F)
   y Count
   0   115
   1    21
 Sum   136
> crosstab(d,row.vars="y",type="column.pct",percentages=F)
   y    %
   0 0.85
   1 0.15
 Sum 1.00
```

Accuracy Paradox

R Code

```
> logit1  <- glm(war~lpop+polity2+gdpen+ethfrac+lmtnest+Oil,data=d,family= binomial(link = "logit"))
> summary(logit1)
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.38555    1.95858  -3.260  0.00111 **
lpop         0.47944    0.20212   2.372  0.01769 *
polity2      0.04639    0.04466   1.039  0.29896
gdpen       -0.32338    0.14613  -2.213  0.02690 *
ethfrac      0.77218    0.97873   0.789  0.43014
lmtnest      0.37545    0.21782   1.724  0.08476 .
Oil         -0.23107    0.77083  -0.300  0.76435
---
> d$yhat1 <- logit1$fitted
> d$ypred1 <- as.numeric(d$yhat1>.5)
>
> crosstab(d,row.vars="ypred1",col.vars="y",type="frequency",percentages=F)
       y   0   1 Sum
ypred1
0        112  19 131
1          3   2   5
Sum      115  21 136
> ACCm1 <- mean(d$ypred1 == d$y)
> ACCm1
[1] 0.8382353
>
> # all negative model (null model)
> ACCneg <- mean(0 == d$y)
> ACCneg
[1] 0.8455882
```

Outline

Sensitivity and specificity

	Observed	
	Positive ($Y = 1$)	Negative ($Y = 0$)
Predicted positive ($\hat{Y} > c$)	True Positive (TP)	False Positive (FP)
Predicted negative ($\hat{Y} < c$)	False Negative (FN)	True Negative (TN)
	Total Positive (P)	Total Negative (N)

- Sensitivity: $TPR = TP/P$
  - Percent of positives that are correctly classified (also called recall or true positive rate)
  - For a screening test with higher sensitivity, fewer actual positives go undetected (lower type 2 error)
- Specificity: $SPC = TN/N$
  - Percent of negatives that are correctly classified (also called true negative rate)
  - For a screening test with higher specificity, fewer negatives are labeled as positive (lower type 1 error)
- In theory a classifier might achieve 100% on both, but in practice there are often tradeoffs and we do not attain 100%
- The higher c the lower the sensitivity, but the higher the specificity

Sensitivity and specificity

R Code

```
> # model 1
> crosstab(d,row.vars="ypred1",col.vars="y",type="frequency",percentages=F)
       y    0    1  Sum
ypred1
0         352  204  556
1         172  272  444
Sum       524  476 1000
> crosstab(d,row.vars="ypred1",col.vars="y",type="column.pct",percentages=F)
       y    0    1
ypred1
0        0.67 0.43
1        0.33 0.57
Sum      1.00 1.00
>
> # model 2
> crosstab(d,row.vars="ypred2",col.vars="y",type="frequency",percentages=F)
       y    0    1  Sum
ypred2
0         378  178  556
1         146  298  444
Sum       524  476 1000
> crosstab(d,row.vars="ypred2",col.vars="y",type="column.pct",percentages=F)
       y    0    1
ypred2
0        0.72 0.37
1        0.28 0.63
Sum      1.00 1.00
```

Receiver Operating Characteristic (ROC) Curve

- ROC curves plot the sensitivity against the specificity across all cutpoints *c* from 0 to 1
- High discrimination ability means high sensitivity and specificity simultaneously. For such models the ROC curve goes close to the top left corner
- Low discrimination ability means low sensitivity and specificity simultaneously. For such models the ROC curve goes close to the 45 degree diagonal line.
- Can plot ROC curves to compare discrimination ability across models.

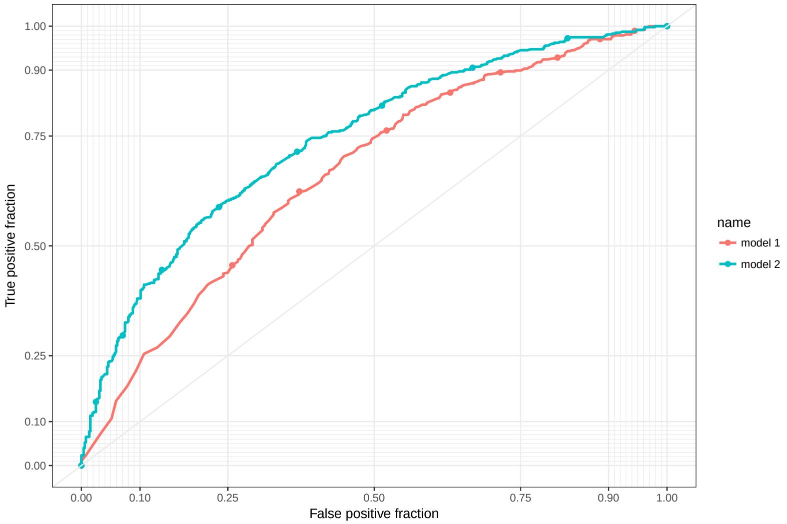
Receiver Operating Characteristic (ROC) Curve

R Code

```
> library(plotROC)
> dd <- data.frame(y=rep(d$y,2),
+                 pred=c(d$yhat1,
+                        d$yhat2),
+                 name=rep(c("model 1","model 2"),each=nrow(d))
+                 )
>
> head(dd)
  y      pred    name
1 0 0.3030943 model 1
2 0 0.5318072 model 1
3 1 0.5119914 model 1
4 1 0.4091240 model 1
5 0 0.3163049 model 1
6 1 0.6837284 model 1

> ggroc <- ggplot(dd, aes(d = y, m = pred, color = name)) + geom_roc(labels = FALSE) + style_roc()
>
> pdf("ROC.pdf",width=9,height=6)
> ggroc
> dev.off()
```

Receiver Operating Characteristic (ROC) Curve



Area Under ROC Curve

- We can summarize the discrimination ability of a model by computing the area under the estimated ROC curve (AUC).
- Maximum discrimination ability means $AUC = 1$
- Minimum discrimination ability means $AUC = 0.5$
- Can show that AUC is the probability a random pair of observations, one with $Y = 1$ and one with $Y = 0$, the observation with $Y = 1$ is correctly ranked as having higher predicted probability than the other.

Receiver Operating Characteristic (ROC) Curve

R Code

```
> ggroc <- ggplot(dd, aes(d = y, m = pred, color = name)) + geom_roc(labels = FALSE) + style_roc()
>
> calc_auc(ggroc)
  PANEL group       AUC
1     1     1 0.6654632
2     1     2 0.7413461

> library(pROC)
> roccurve <- roc(d$y ~ d$yhat1)
> auc(roccurve)
Area under the curve: 0.6655
```

Outline

Brier Score

To summarize the accuracy of binary classifiers we use the Brier score:

$$BS = \frac{1}{N}\sum_{i}^{N}(\hat{Y}_i - Y_i)^2$$

- If $\hat{Y}_i = 1$ and $Y_1 = 1$, then $BS = 0$, the best score achievable.
- If $\hat{Y}_i = 1$ and $Y_1 = 0$, then $BS = 1$, the worst score achievable.
- Note that this measure is based on relative deviations ($\hat{Y}_i = .51$ pays a much lager cost than $\hat{Y}_i = .99$ for $Y_i = 1$), rather than binary deviations (such as correctly or incorrectly classified for a given cutoff c)
- Note that this is proper score function:
  - If $\hat{Y}$ is the vector of predictions and you use a scoring function to give a reward of $S(\hat{Y}, i)$ if the i-th event occurs
  - The highest expected reward is obtained by reporting the true probability distribution so you encourages the forecaster to be honest to maximize the expected reward

Brier Score

BS can be decomposed as the sum of a measure of calibration and refinement:

$$BS = \frac{1}{N}\sum_{i}^{N}(\hat{Y}_i - Y_i)^2 = \frac{1}{N}\sum_{k}^{K}(\hat{Y}_k - \bar{Y}_k)^2 + \frac{1}{N}\sum_{k}^{K} n_k(\bar{Y}_k(1-\bar{Y}_k))$$

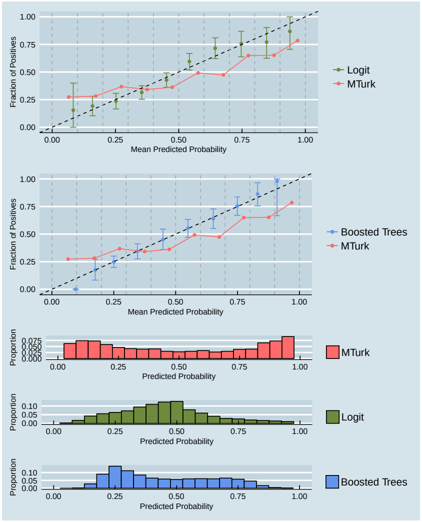
- K is the # of unique predicted probabilities, n_k is the # of predictions in predicted probability category k (s.t. $N = \sum_k n_k$), $\bar{Y}_k$ is the observed frequency in k given the predicted probability $\hat{Y}_k$
- The first term captures calibration
  - How much do observed frequency of positives in k vary from predicted fraction $\hat{Y}_k$
- The second term refinement (related to the area under the ROC curve)
  - How uniform is occurrence of outcomes within each group k

Brier Score

R Code

```
> mean((d$ypred1-d$y)^2)
[1] 0.1617647
> mean((0-d$y)^2)
[1] 0.1544118
```

Predicting Recidivism



Brier scores are: Mturk .24, Logit = .21 and Boosted Trees = .19 (all three model are similar on AUC)

Cross-Validation

- Often it is useful to validate model looking at out of sample fit instead of in-sample fit
- Most common method is $k-$fold Cross Validation
- Cut the dataset randomly into say 10 folds
- Start a loop, pick the first fold as the hold out, fit the model on the remaining 9 folds and predict outcomes for hold out fold
- Repeat while using next fold as hold out
- In the end you have an out of sample prediction for each observation and then examine model accuracy using this prediction
- Common is 10 folds, but can use other values

Cross-Validation

R Code

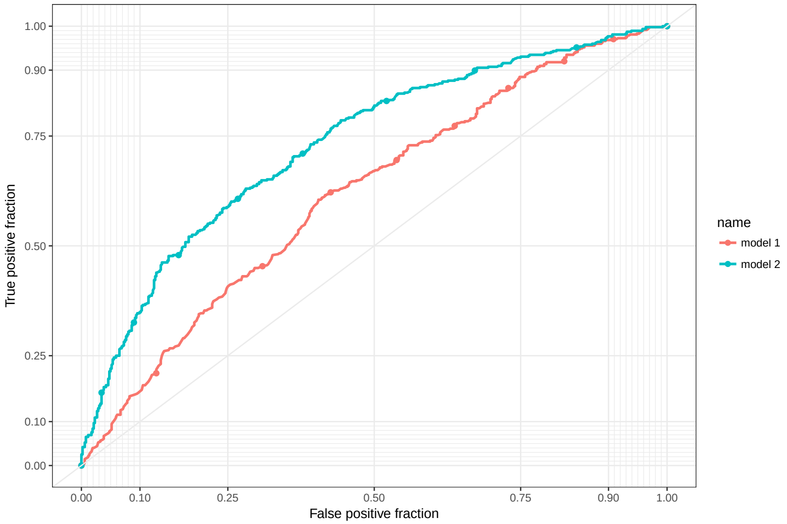
```
> ## bail example cross validation
> rm(list=ls())
> d <- read.csv("bail.csv")
> d$y <- d$two_year_recid
>
> Foldsize <- nrow(d)/10
> Foldsize
[1] 100
>
> # fold var
> d$fold <- sample(rep(1:10,each=Foldsize),nrow(d),replace=F)
>
> d$yhat1 <- NA
> d$yhat2 <- NA
>
> for(i in 1:10){
+
+ # model 1
+ logit1                <- glm(y~sex+age,data=d[d$fold!=i,])
+ d$yhat1[d$fold==i]    <- predict(logit1,newdata=d[d$fold==i,])
+
+ # model 2
+ logit2   <- glm(y~sex+age+log(1+priors_count),data=d[d$fold!=i,])
+ d$yhat2[d$fold==i]    <- predict(logit2,newdata=d[d$fold==i,])
+
+ }
```

Cross-Validation

R Code

```
> dd <- data.frame(y=rep(d$y,2),
+                  pred=c(d$yhat1,
+                         d$yhat2),
+                  name=rep(c("model 1","model 2"),each=nrow(d))
+ )
>
> head(dd)
  y      pred    name
1 0 0.2103555 model 1
2 0 0.4504963 model 1
> dd$name <- factor(dd$name)
>
> ggroc <- ggplot(dd, aes(d = y, m = pred, color = name)) + geom_roc(labels = FALSE) + style_roc()
> pdf("ROC2.pdf",width=9,height=6)
> ggroc
> dev.off()
null device
          1
>
> # AUC
> calc_auc(ggroc)
  PANEL group       AUC
1     1     1 0.6183747
2     1     2 0.7311746
>
> # BS
> mean((d$yhat1-d$y)^2)
[1] 0.2383642
> mean((d$yhat2-d$y)^2)
[1] 0.2106819
```

Cross-Validation ROC Curve



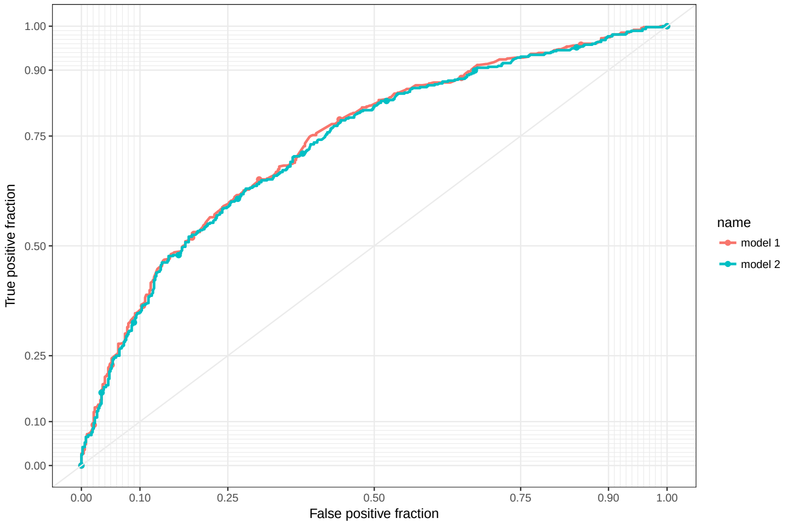
Cross-Validation

R Code

```
> logit2       <- glm(y~sex+age+log(1+priors_count),data=d)
> d$yhat2in    <- predict(logit2,newdata=d)
>
> dd <- data.frame(y=rep(d$y,2),
+                  pred=c(d$yhat2in,
+                         d$yhat2),
+                  name=rep(c("model 1","model 2"),each=nrow(d))
+ )
> ggroc <- ggplot(dd, aes(d = y, m = pred, color = name)) + geom_roc(labels = FALSE) + style_roc()

> # AUC
> calc_auc(ggroc)
  PANEL group       AUC
1     1     1 0.7369259
2     1     2 0.7311746
>
>
> # BS
> mean((d$yhat2in-d$y)^2)
[1] 0.2087114
> mean((d$yhat2-d$y)^2)
[1] 0.2106819
```

Cross-Validation ROC Curve


True positive fraction
False positive fraction
name
model 1
model 2