MODEL ASSESSMENT

Dr. Aric LaBarr

Institute for Advanced Analytics

COMPARING MODELS

Purpose of Modeling

- Statistical models are created for two different purposes – estimation and prediction.
  - **Estimation:** Quantifying the expected change in response associated with predictors (relationships).
  - **Prediction:** Use the model to predict new response.

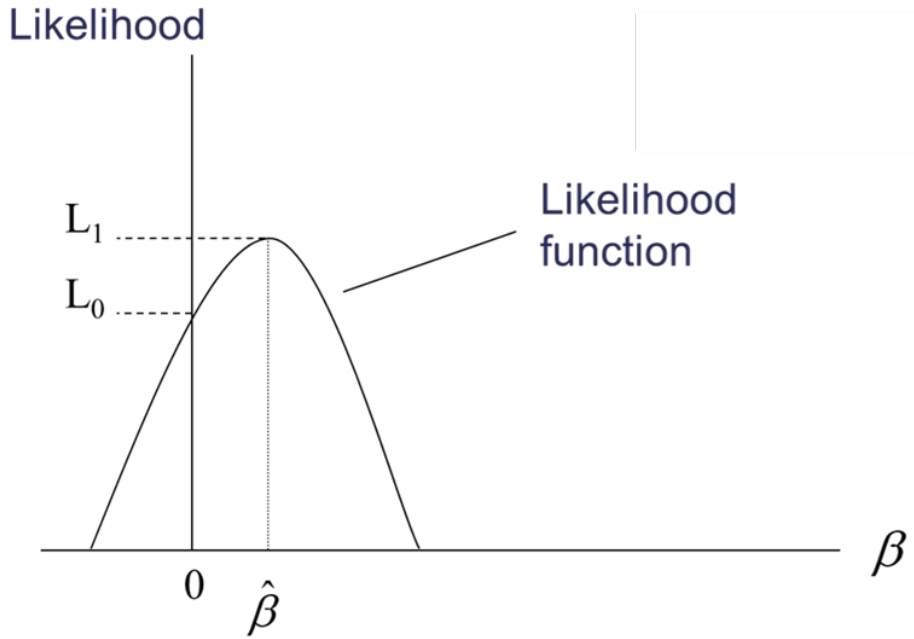
- Won't necessarily agree!

Deviance/Likelihood Measures

- AIC and BIC approximate out-of-sample prediction error by applying a penalty for model complexity:
  - AIC – crude, large-sample approximation of leave-one-out cross-validation.
  - BIC – favors smaller models/penalizes model complexity more.
- Lower values “better” than higher.
- No amount of lower is “better” enough.
- May not always agree, but neither is necessarily better.

Deviance/Likelihood Measures

- Number of "pseudo"-R^2 quantities for logistic regression.
- Higher values indicate "better" model.
- Generalized / Nagelkerke R^2 - how much better than intercept only model?
- Unlike linear regression, there is **no interpretation** on these.



$$R_G^2 = 1 - \left(\frac{L_0}{L_1}\right)^{\frac{2}{n}}$$

Deviance and Likelihood Measures

```
AIC(logit.model)

[1] 1287.964

BIC(logit.model)

[1] 1394.86

PseudoR2(logit.model, which = "Nagelkerke")

Nagelkerke 0.7075796
```

ASSESSING PREDICTIVE POWER

What is a Good Logistic Model?

- Logistic regression is a **model for probability of an event** – NOT the occurrence of an event.
- Logistic regression **can** be a classification model as well.

- Good model should reflect both of these, but importance of one over the other depends on the problem.

Discrimination vs. Calibration

- **Discrimination** – ability to separate the events from the non-events. How good is model at distinguishing the 1's from the 0's.
- **Calibration** – how well predicted probabilities agree with the actual frequency of the outcomes. Are predicted probabilities systematically too low/high?

- **May not agree with each other!**

ASSESSING PREDICTIVE POWER

Probability Based Metrics

Coefficient of Discrimination

- Want model to assign a higher probability to events and lower probability to non-events.
- **Coefficient of discrimination** (or **discrimination slope**) is the difference in average predicted probability between 1's and 0's:

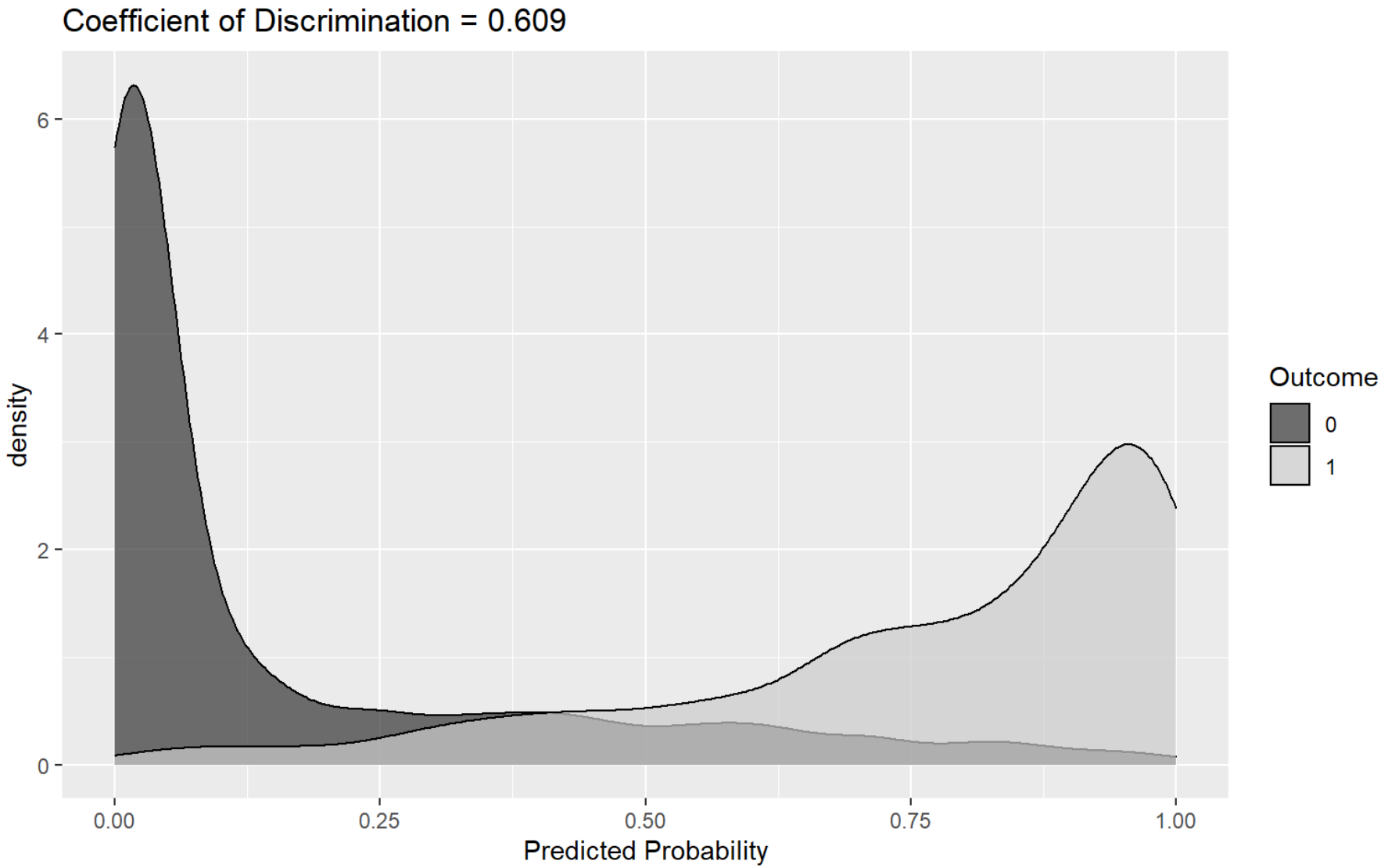
$$D = \bar{\hat{p}}_1 - \bar{\hat{p}}_0$$

- Able to compare with histograms as well.

Discrimination Slope

```
train$p_hat <- predict(logit.model, type = "response")
p1 <- train$p_hat[train$Bonus == 1]
p0 <- train$p_hat[train$Bonus == 0]

coef_discrim <- mean(p1) - mean(p0)

ggplot(train, aes(p_hat, fill = factor(Bonus)))
 + geom_density(alpha = 0.7)
 + scale_fill_grey()
 + labs(x = "Predicted Probability", fill = "Outcome",
        title = paste("Coefficient of Discrimination = ",
        round(coef_discrim, 3), sep = ""))
```

Discrimination Slope



Rank-order Statistics

- How well does the model order predictions?

- **Concordance:** for a pair of subjects with and without the event, the one **with the event** had the **higher** predicted probability.
- **Discordance:** for a pair of subjects with and without the event, the one **with the event** had the **lower** predicted probability.
- **Tied:** for a pair of subjects with and without the event, they both have the **same** predicted probability.

Concordance

- **Interpretation** – For all possible (1,0) pairs, the model assigned the higher predicted probability to the observation with the event *concordance%* of the time.
- Common metrics based on concordance:
  - c-statistic: $c = Concordance\ \% + \frac{1}{2} Tied\ \%$
  - Somers' D (Gini): $D_{xy} = 2c - 1$
  - Kendall's τ_a: $\tau_a = \frac{\#\text{concordant} - \#\text{discordant}}{\frac{n(n-1)}{2}}$

Rank-order Statistics – R

```
Concordance(train$Bonus, train$p_hat)

$Concordance
[1] 0.9428394

$Discordance
[1] 0.05716055

$Tied
[1] -5.551115e-17

$Pairs
[1] 1017240
```

```
somersD(train$Bonus, train$p_hat)

[1] 0.8856789
```

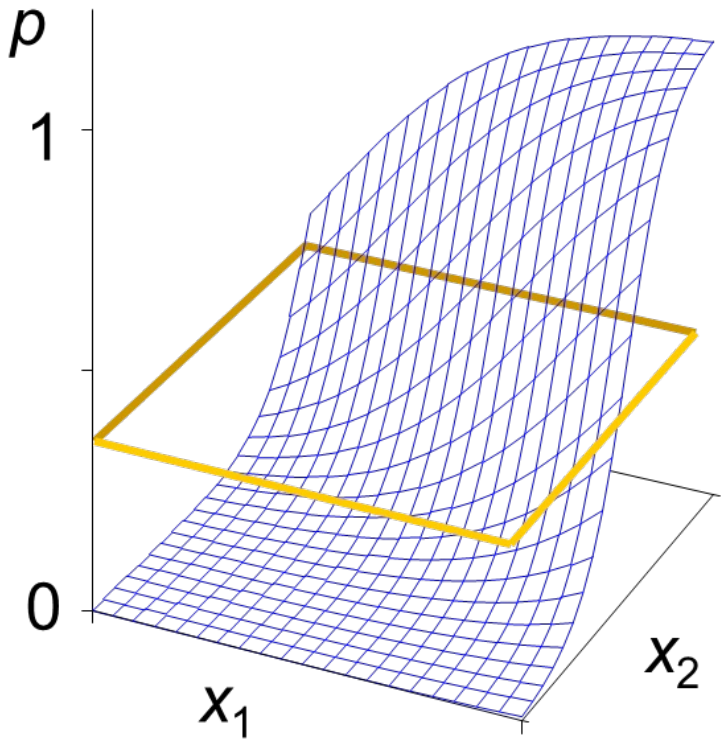
ASSESSING PREDICTIVE POWER

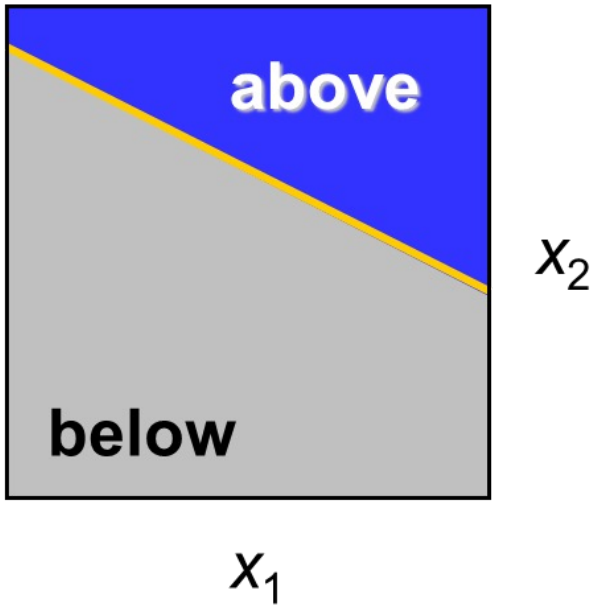
Classification Based Metrics

Classification

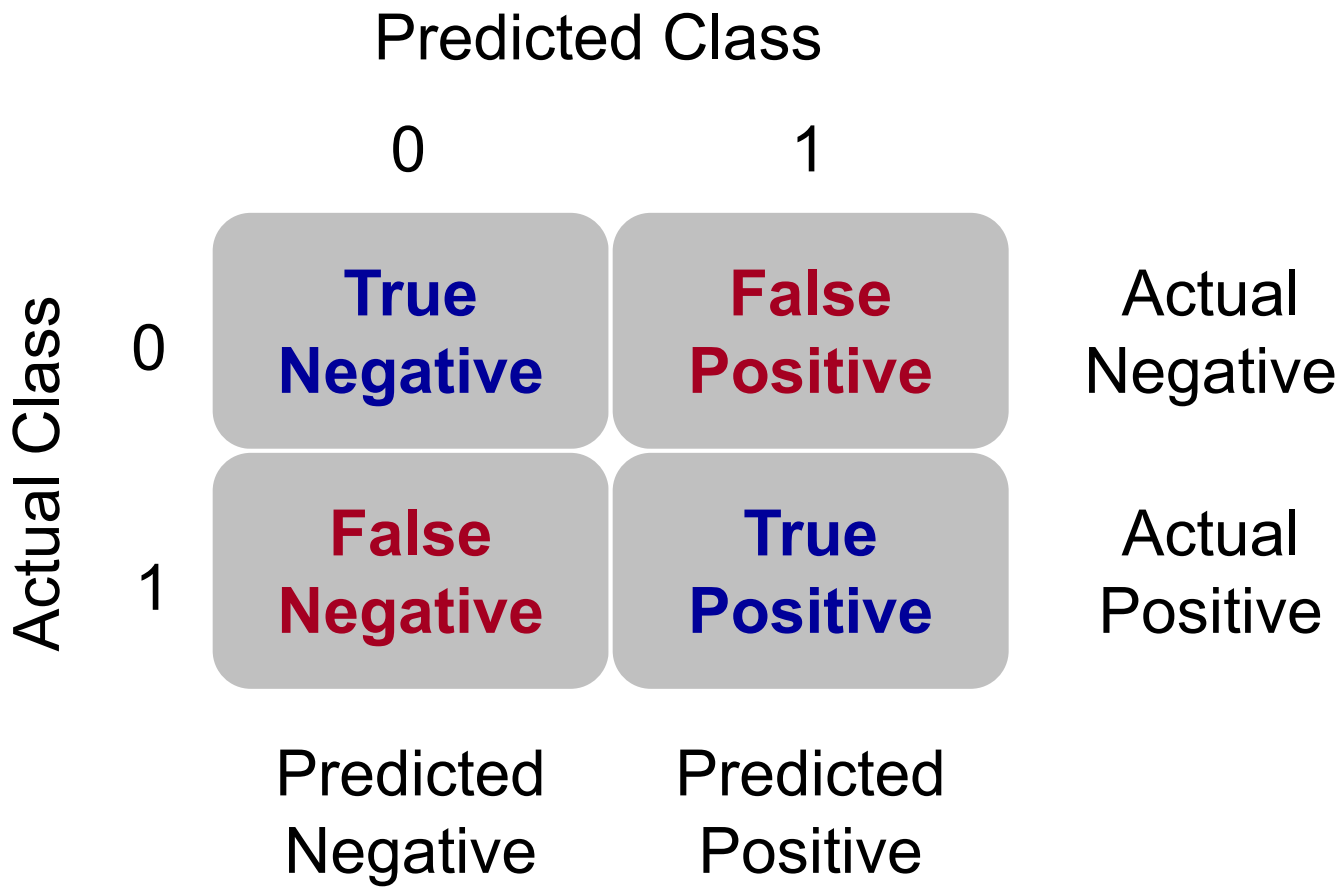
- Want model to correctly classify events and non-events.
- **Classification** forces the model to predict $\hat{y}_i = 1$ or $\hat{y}_i = 0$ based on whether the predicted probability exceeds some threshold – for example, $\hat{y}_i = 1$ if $\hat{p}_i > 0.5$.
- Strict classification-based measures completely discard any information about the actual quality of the model's predicted probabilities.

Logistic Discrimination





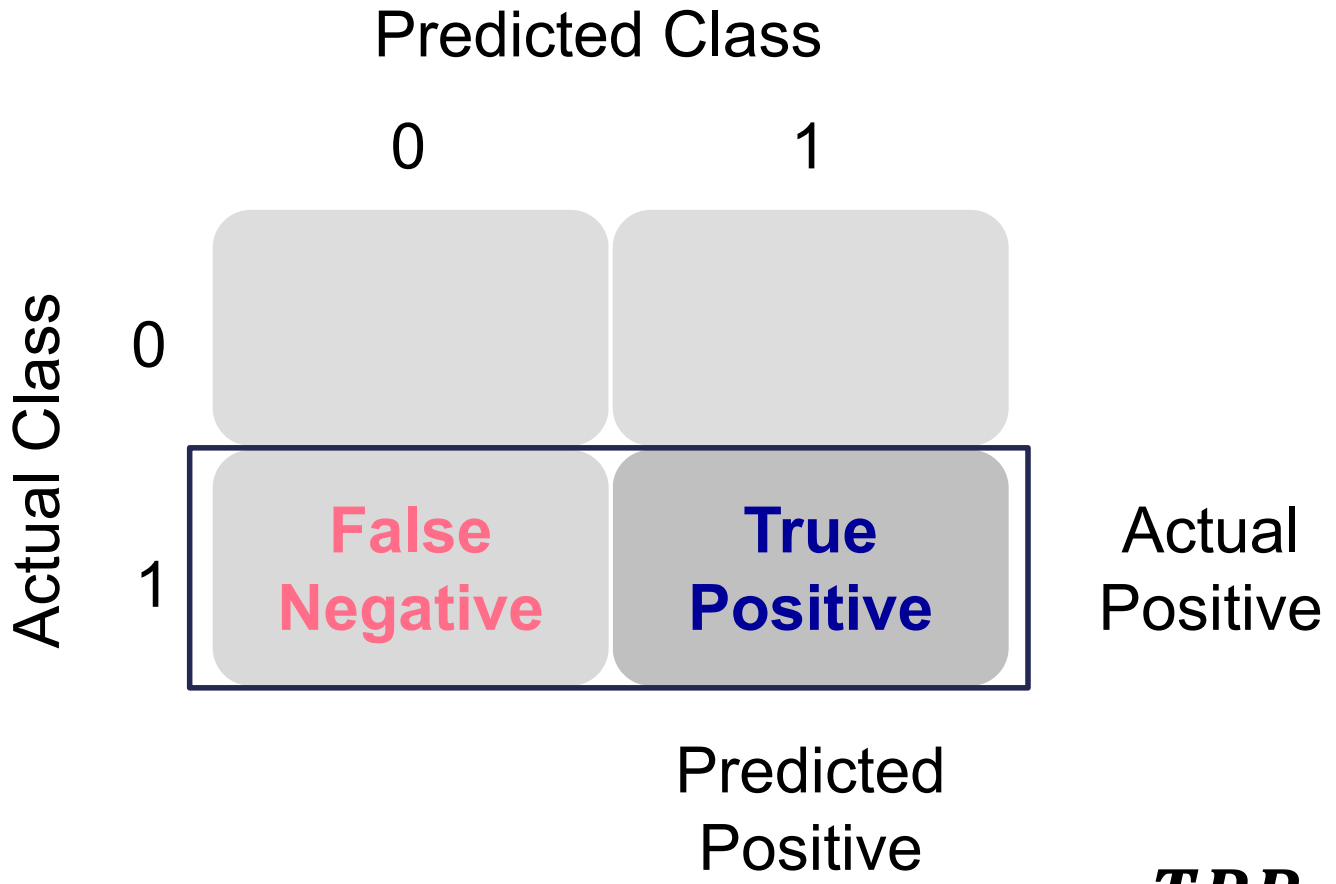
Classification Table



ASSESSING PREDICTIVE POWER

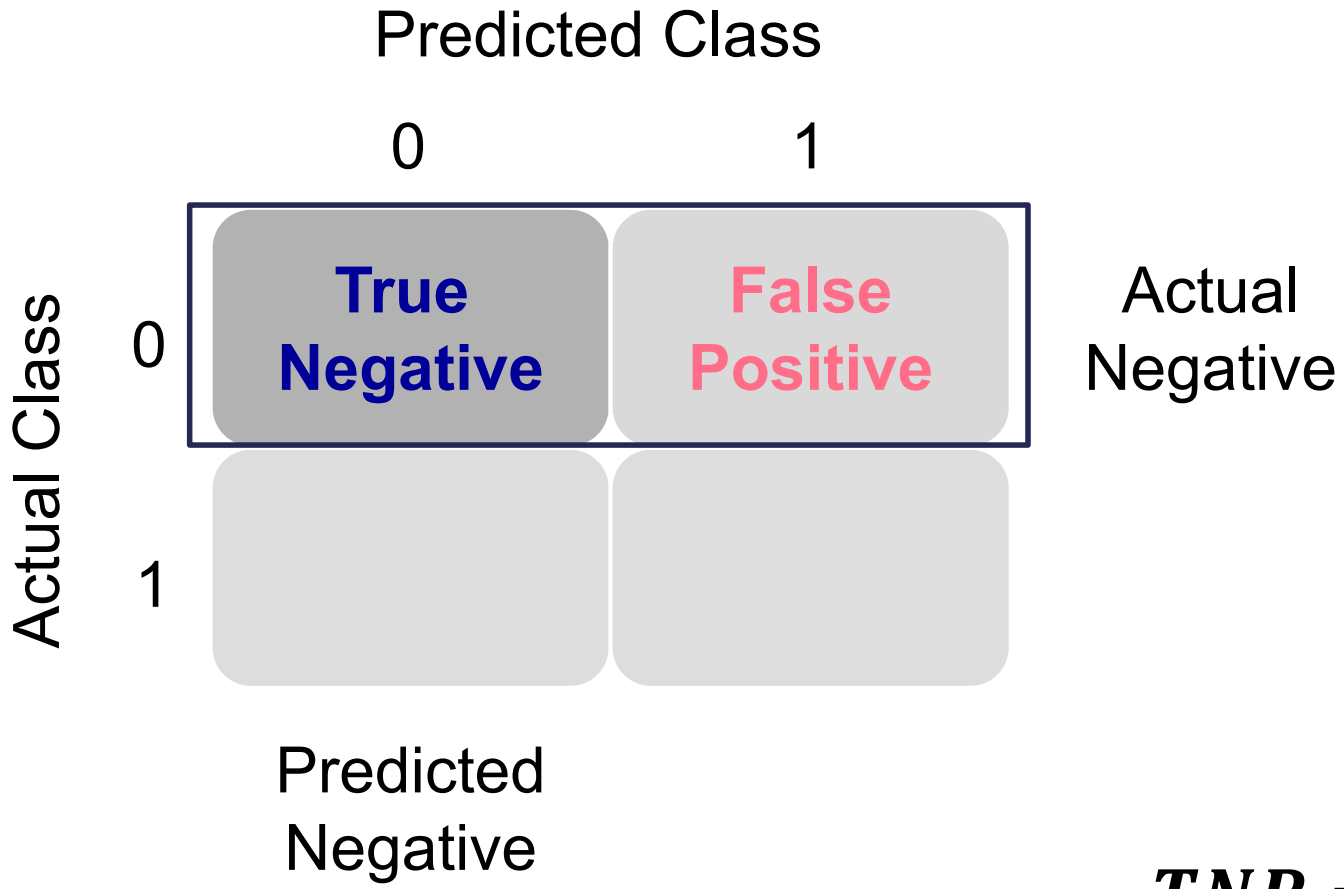
Sensitivity vs. Specificity

Sensitivity / Recall



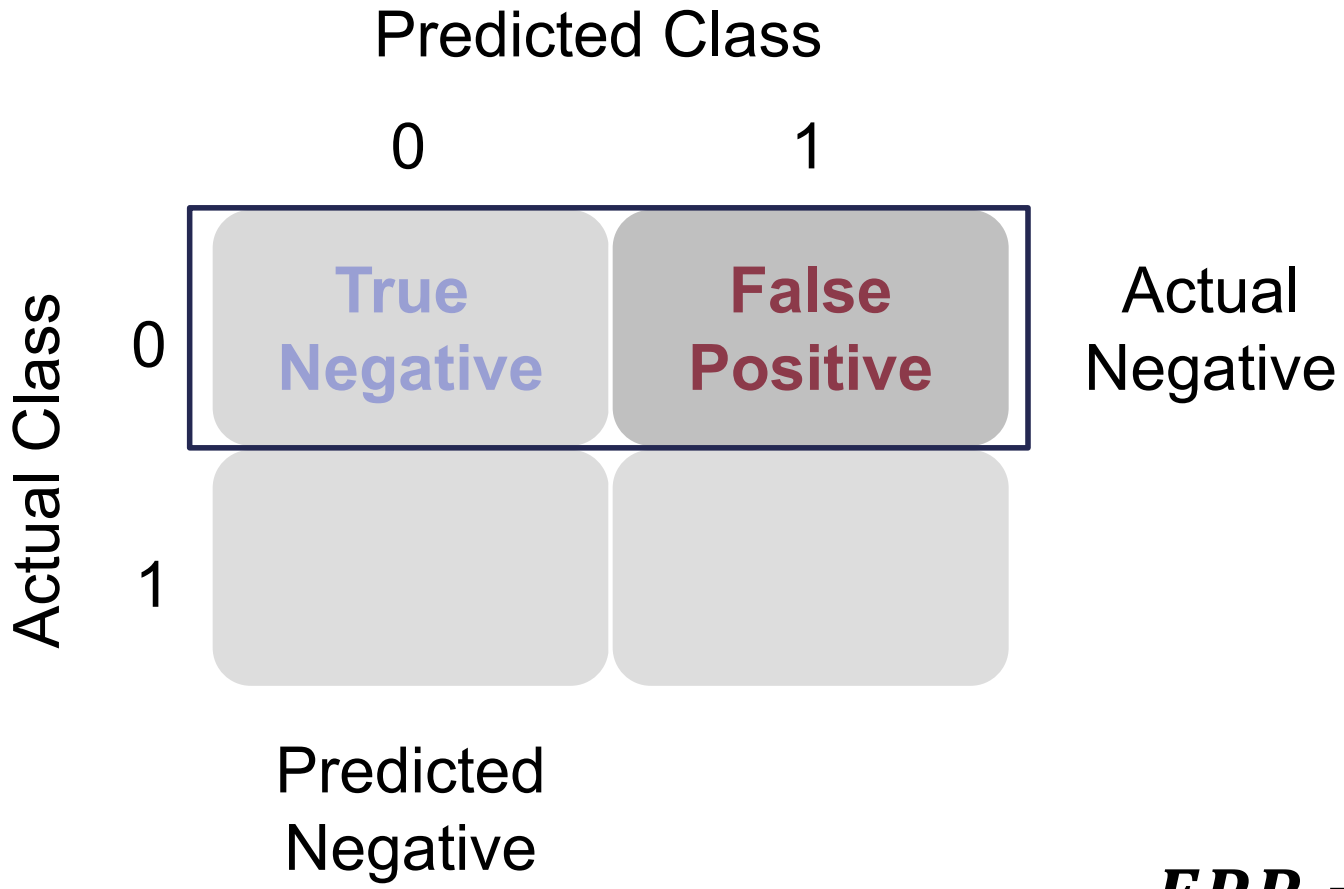
$$TPR = \frac{TP}{TP + FN}$$

Specificity



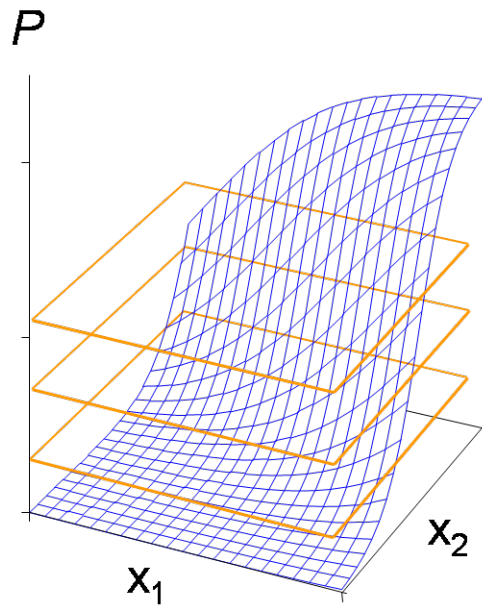
$$TNR = \frac{TN}{TN + FP}$$

1 – Specificity



$$FPR = \frac{FP}{TN + FP}$$

Classification Changes with Cut-off



response	$\hat{P}$	cutoff=.5	cutoff=.25
0	.32	0	1
1	.40	0	1
1	.92	1	1
0	.06	0	0
1	.52	1	1
1	.39	0	1
1	.22	0	0
0	.17	0	0
0	.13	0	0
⋮	⋮	⋮	⋮
1	.75	1	1

Best Cut-off?

- **Always** consider the cost of false positives and false negatives when doing classification.
- When **NOT** considering costs, many different techniques to "optimal" cut-off.
- **Youden J statistic** (or **Youden's index**):

$$J = \text{sensitivity} + \text{specificity} - 1$$

- "Optimal" – false positives and false negatives are weighed equally, so select cut-off that produces highest Youden *J* statistic.

Classification Table

```
confusionMatrix(train$Bonus, train$p_hat, threshold = 0.5)
```

```
      0   1
0 1062 127
1  149 713
```

Youden Index

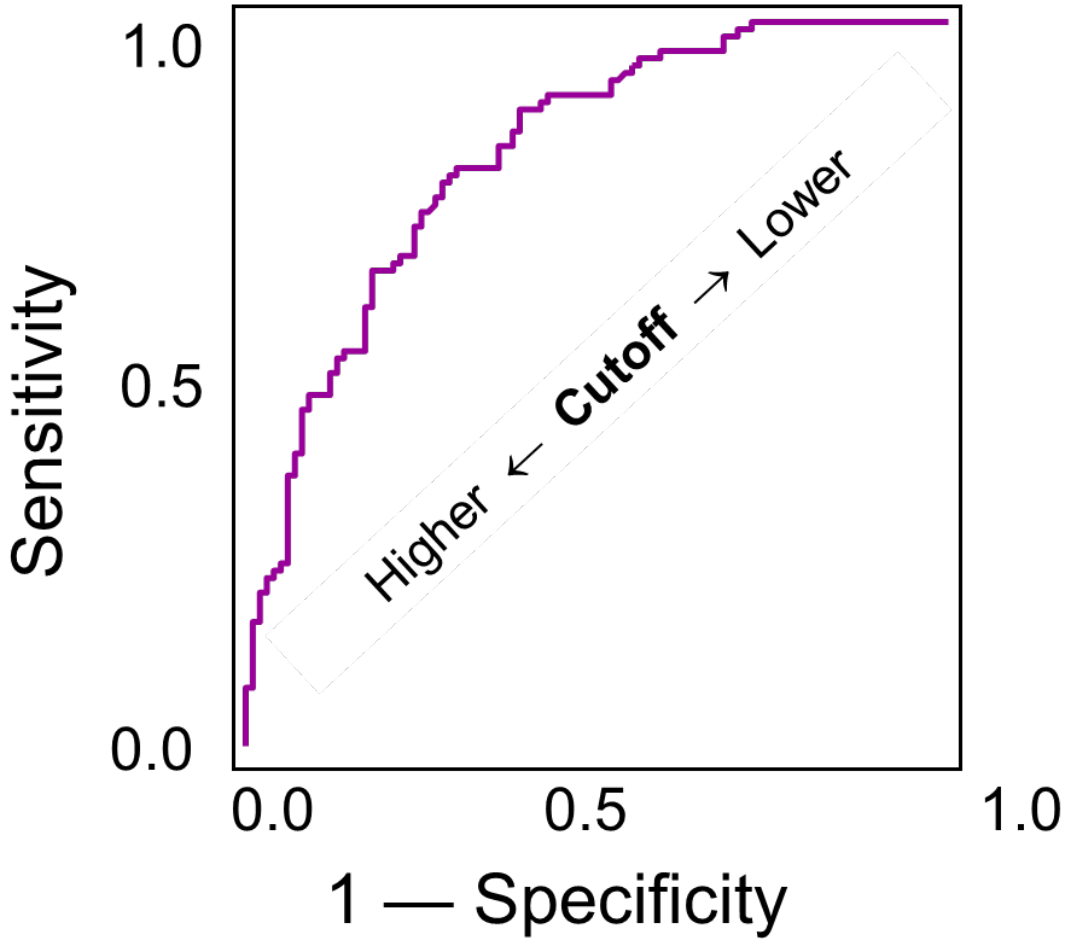
```
sens <- NULL
spec <- NULL
youden <- NULL
cutoff <- NULL

for(i in 1:49){
  cutoff = c(cutoff, i/50)
  sens <- c(sens, sensitivity(train$Bonus, train$p_hat, threshold = i/50))
  spec <- c(spec, specificity(train$Bonus, train$p_hat, threshold = i/50))
  youden <- c(youden, youdensIndex(train$Bonus, train$p_hat, threshold = i/50))
}

ctable <- data.frame(cutoff, sens, spec, youden)

print(ctable[order(-youden),])
```

Youden Index

	cutoff	sens	spec	youden
21	0.42	0.8916667	0.8406276	0.7322942
18	0.36	0.9178571	0.8142031	0.7320603
23	0.46	0.8690476	0.8629232	0.7319708
20	0.40	0.8976190	0.8340215	0.7316405
17	0.34	0.9261905	0.8042940	0.7304844
22	0.44	0.8773810	0.8530140	0.7303950
14	0.28	0.9511905	0.7778695	0.7290600
16	0.32	0.9345238	0.7943848	0.7289086
19	0.38	0.9083333	0.8199835	0.7283168
15	0.30	0.9428571	0.7844756	0.7273328
24	0.48	0.8583333	0.8687036	0.7270369
13	0.26	0.9571429	0.7696119	0.7267547
25	0.50	0.8488095	0.8769612	0.7257707

⋮

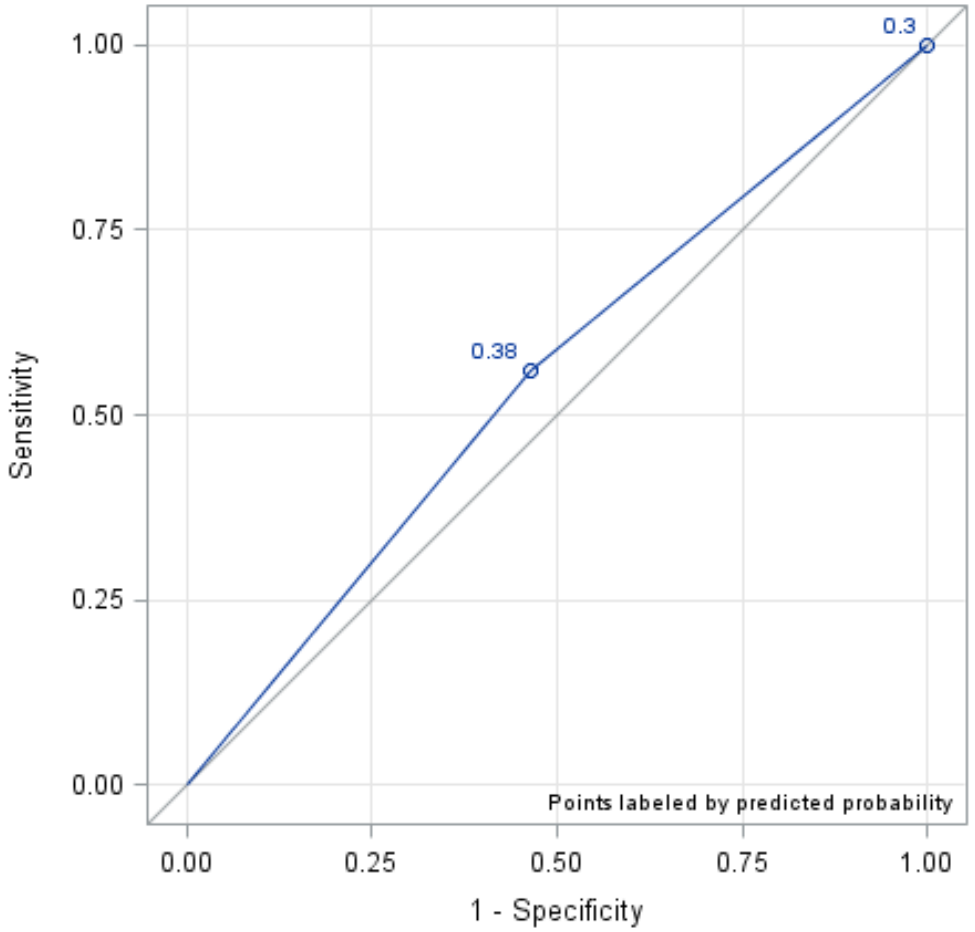
ROC Curve



- **ROC curve** plots *TPR* vs. *FPR* for a grid of thresholds.
- **Area under the curve** (AUC or AUROC) summarizes the overall quality of ROC curve – equivalent to c-statistic.
- Want high sensitivity and high specificity.

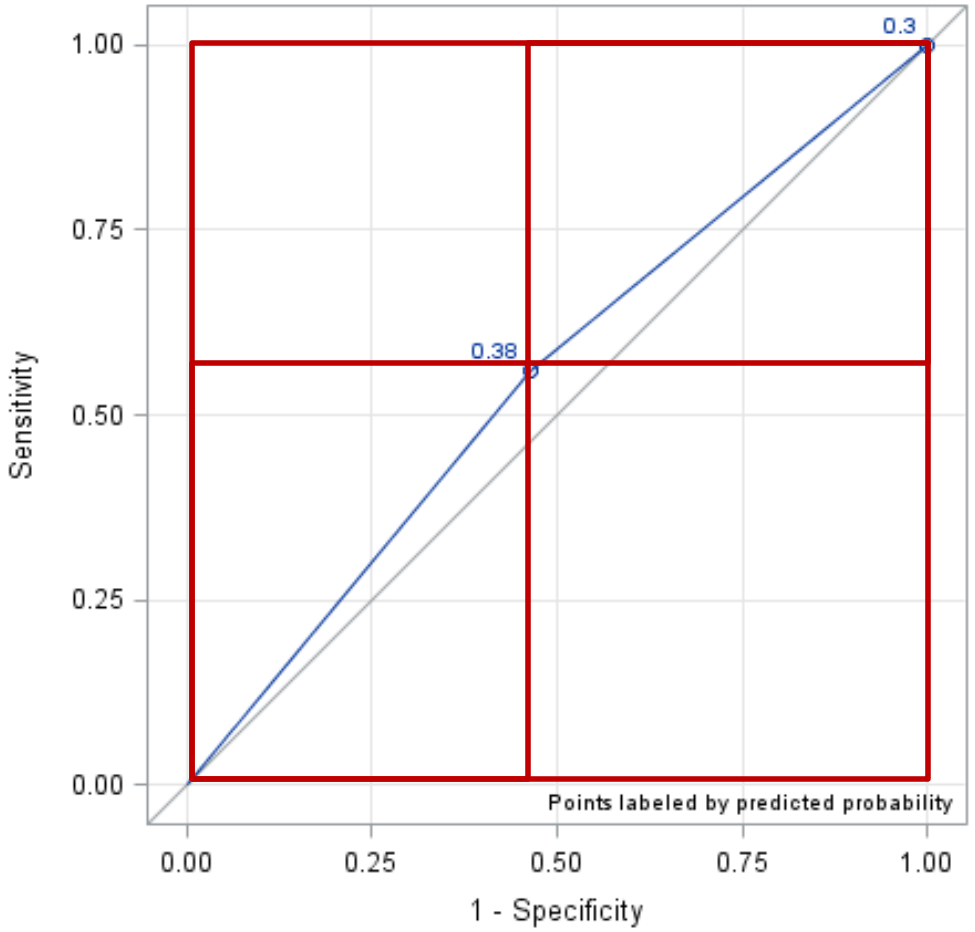
Area Under the ROC Curve

$$AUC = \%\, Concordant + \frac{1}{2}(\%\, Tied)$$



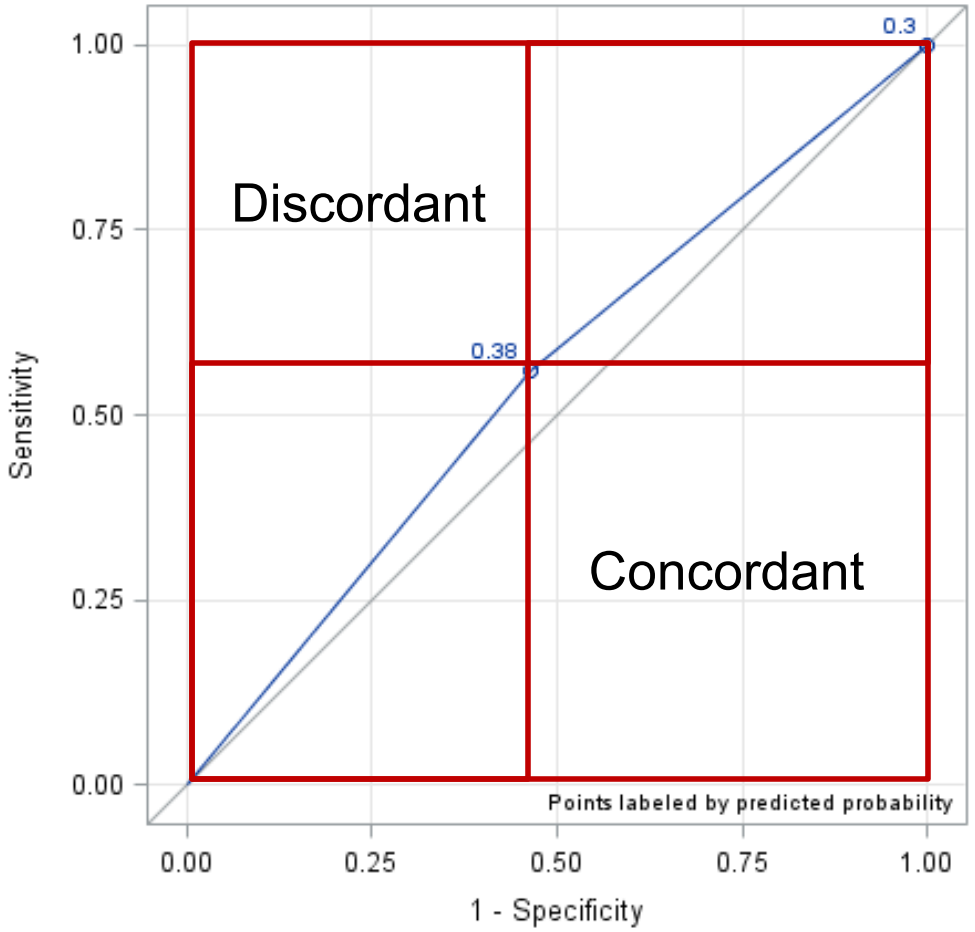
Area Under the ROC Curve

$$AUC = \%\, Concordant + \frac{1}{2}(\%\, Tied)$$



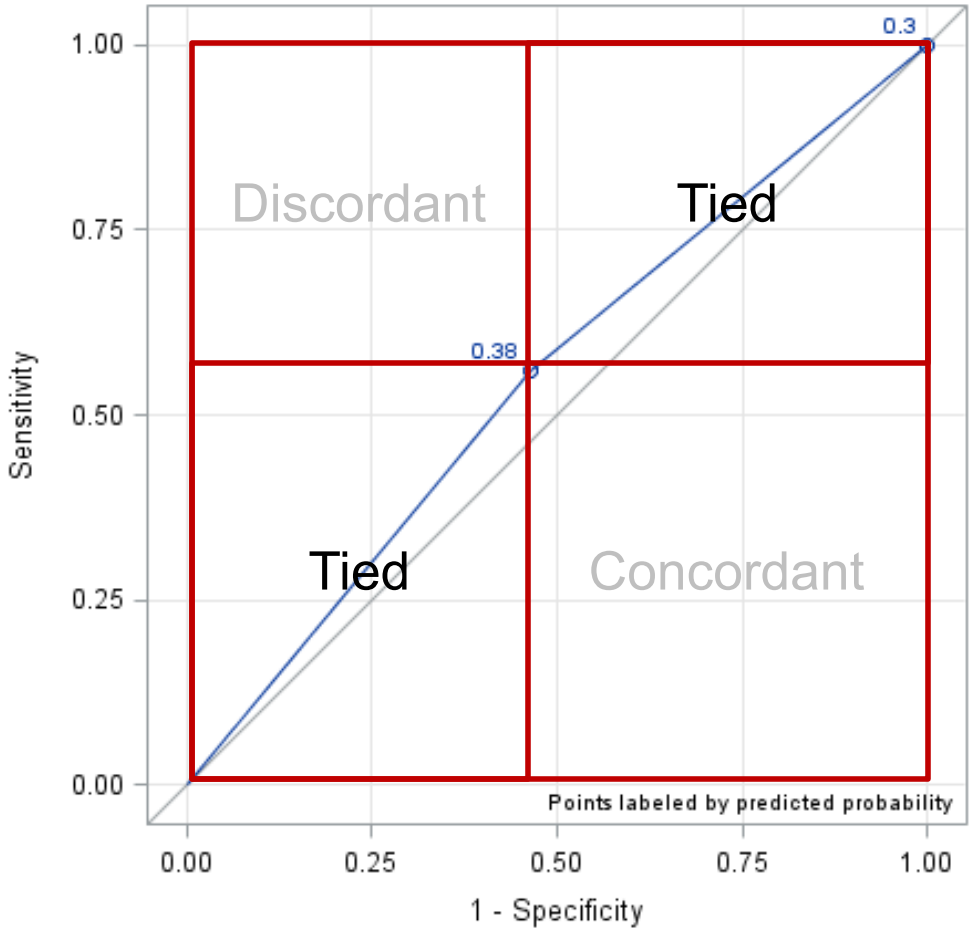
Area Under the ROC Curve

$$AUC = \%\, Concordant + \frac{1}{2}(\%\, Tied)$$



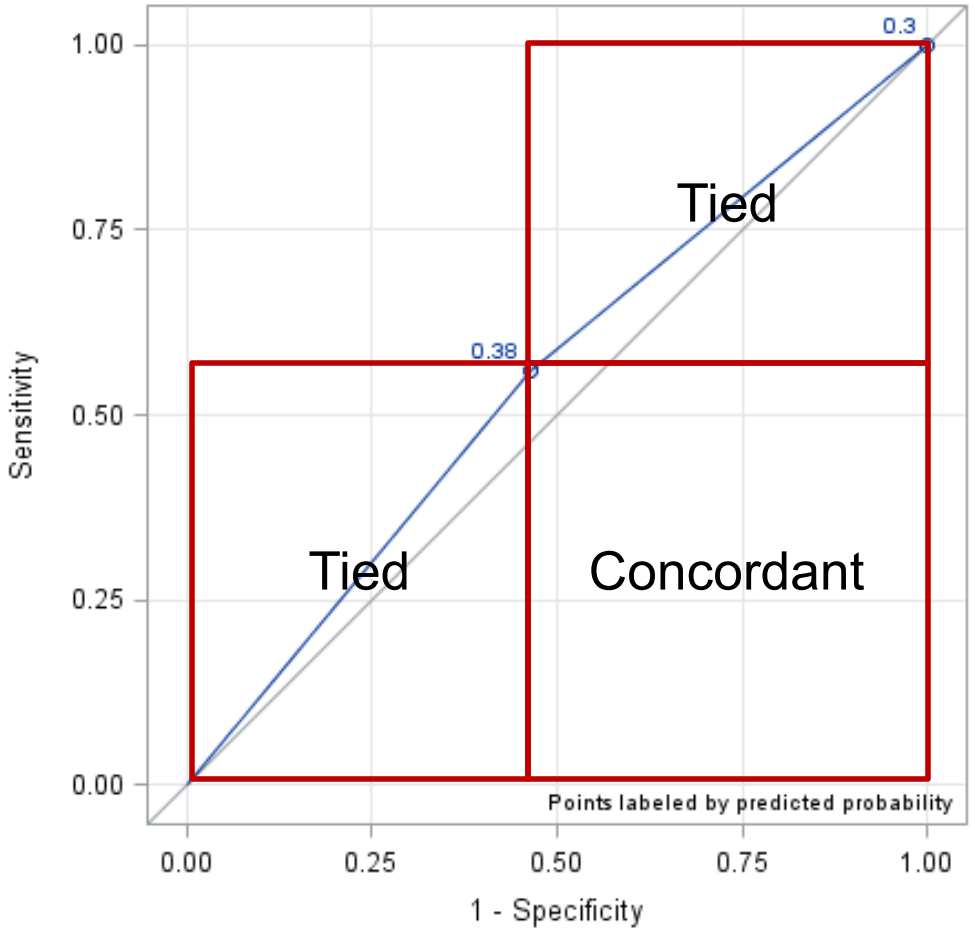
Area Under the ROC Curve

$$AUC = \%\, Concordant + \frac{1}{2}(\%\, Tied)$$



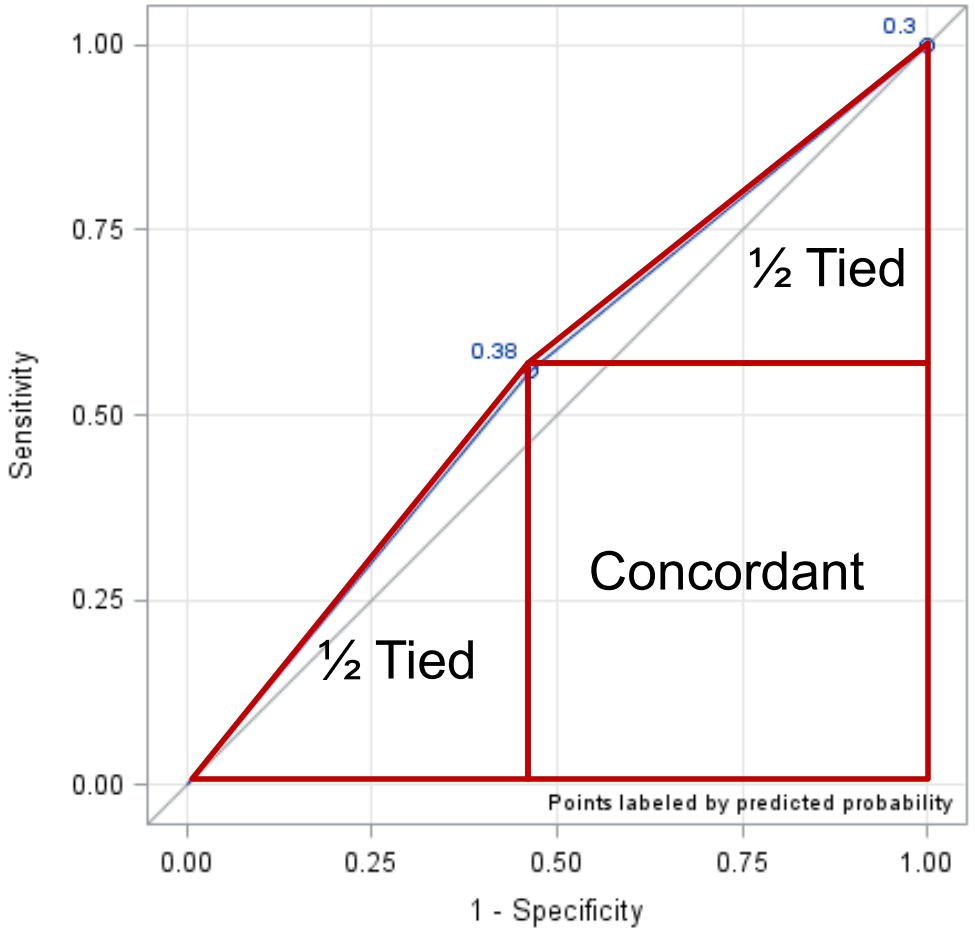
Area Under the ROC Curve

$$AUC = \%\, Concordant + \frac{1}{2}(\%\, Tied)$$



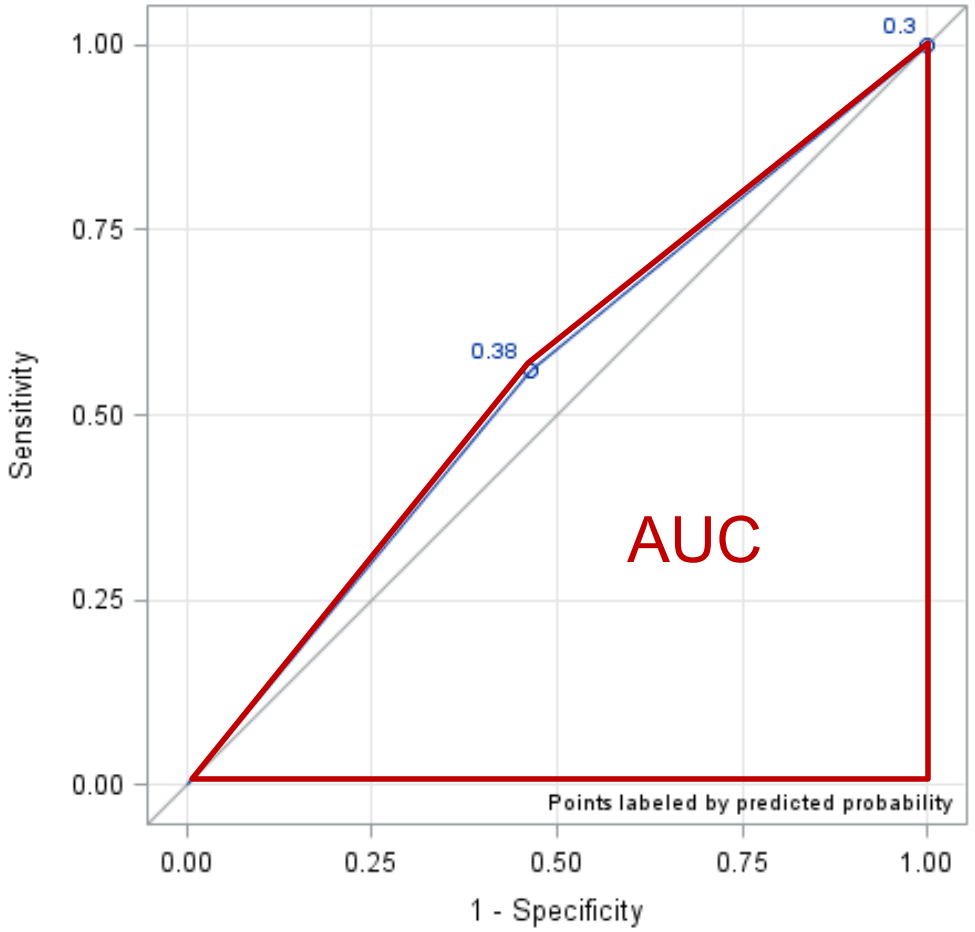
Area Under the ROC Curve

$$AUC = \%\ Concordant + \frac{1}{2}(\%\ Tied)$$



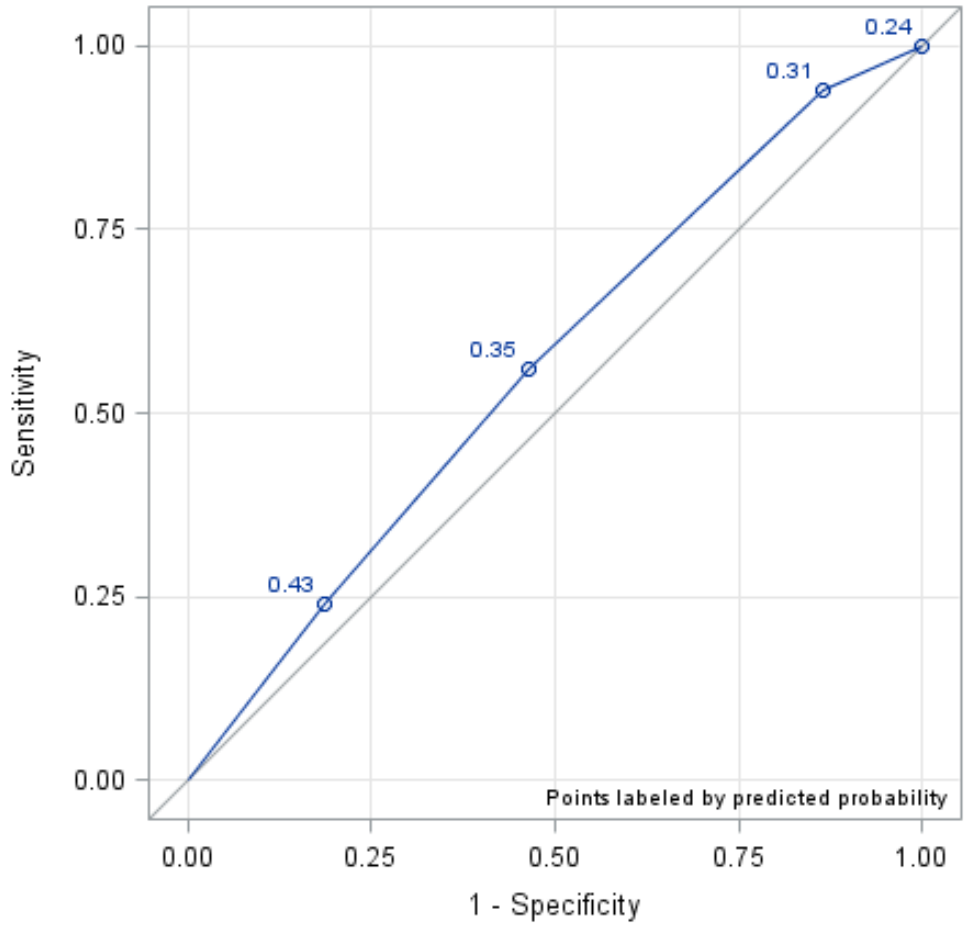
Area Under the ROC Curve

$$AUC = \%\, Concordant + \frac{1}{2}(\%\, Tied)$$



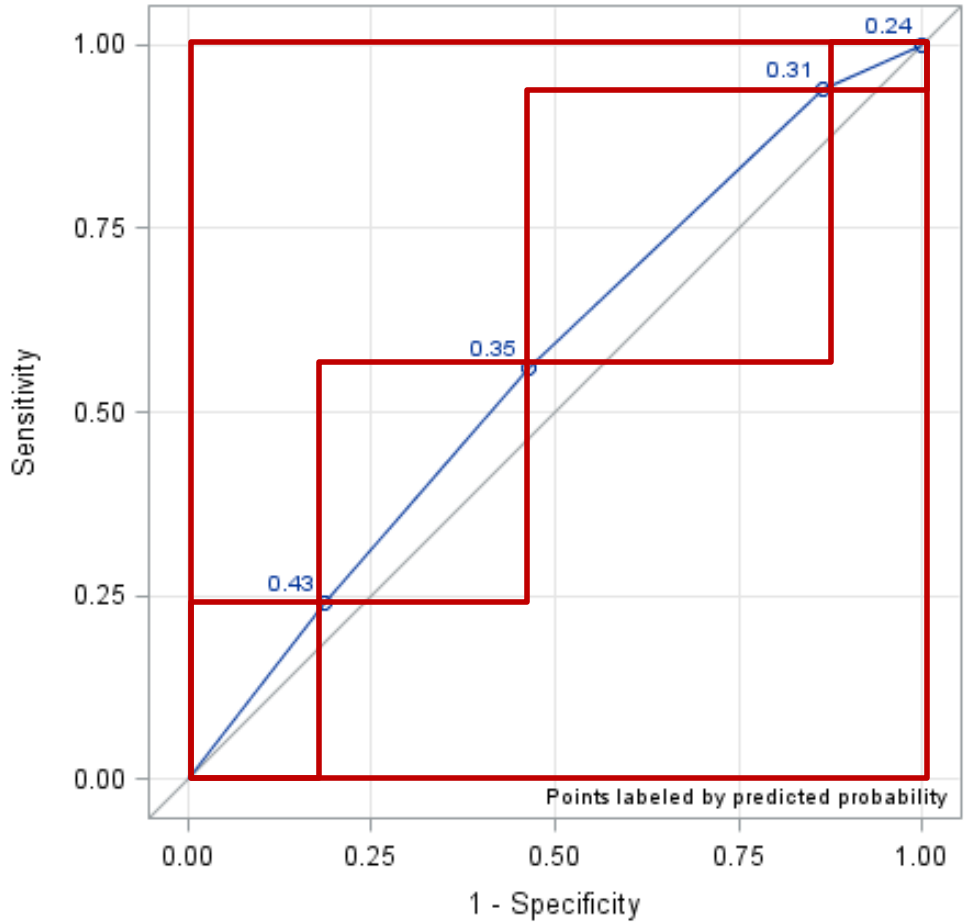
Area Under the ROC Curve

$$AUC = \%\ Concordant + \frac{1}{2}(\%\ Tied)$$



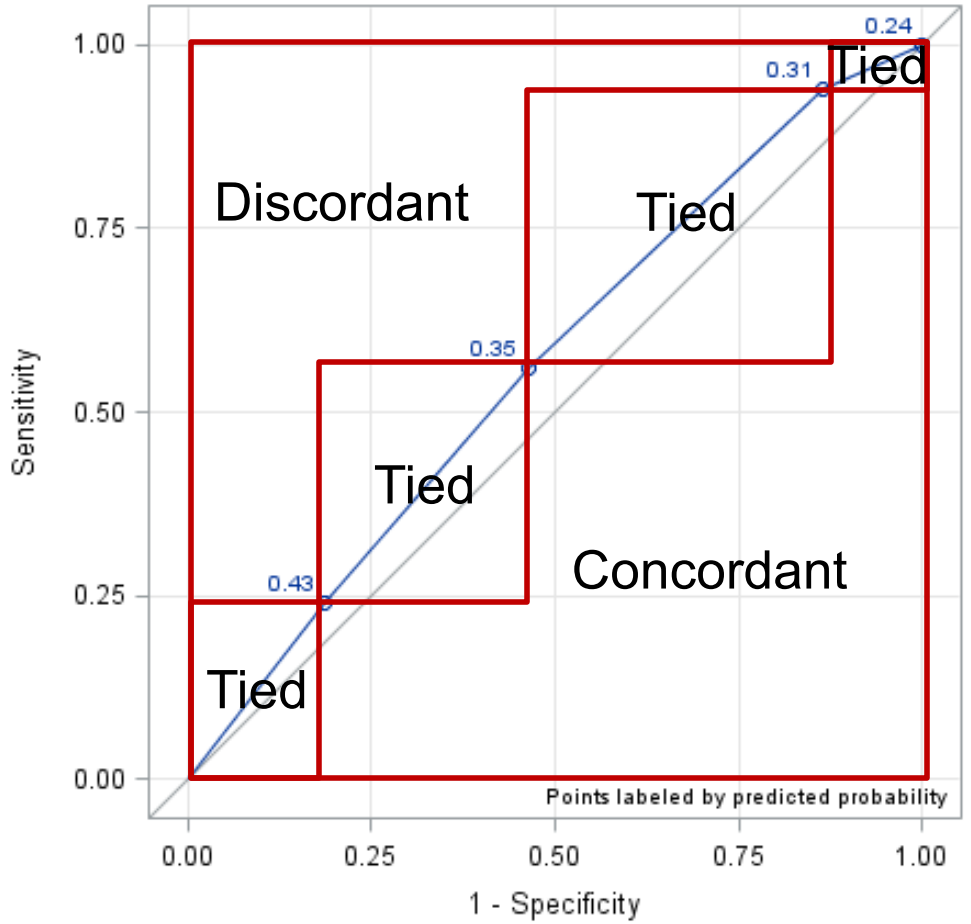
Area Under the ROC Curve

$$AUC = \%\, Concordant + \frac{1}{2}(\%\, Tied)$$



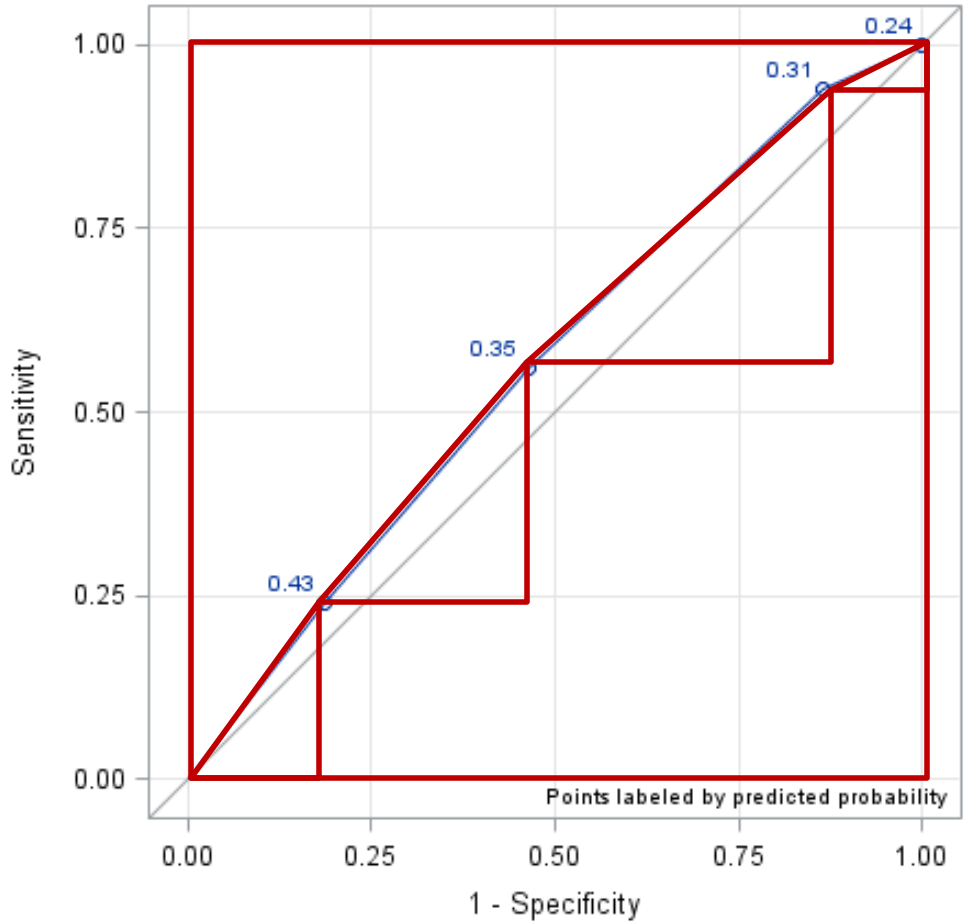
Area Under the ROC Curve

$$AUC = \%\, Concordant + \frac{1}{2}(\%\, Tied)$$



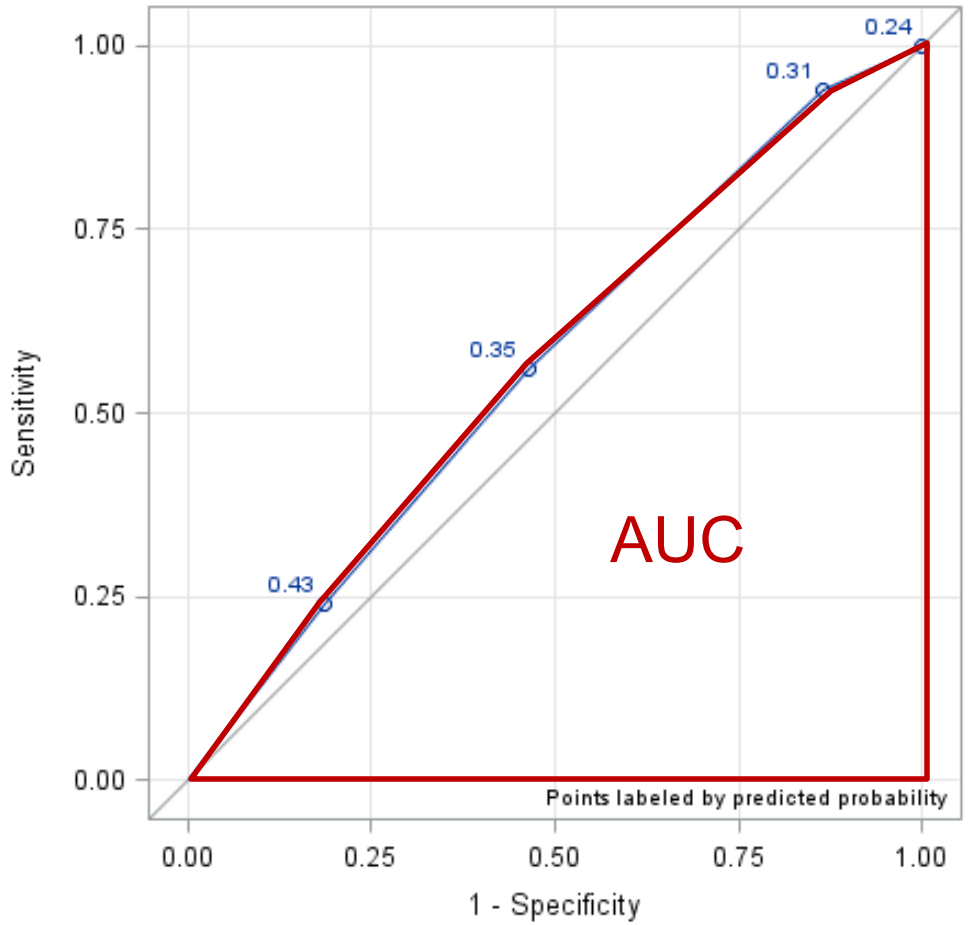
Area Under the ROC Curve

$$AUC = \%\, Concordant + \frac{1}{2}(\%\, Tied)$$



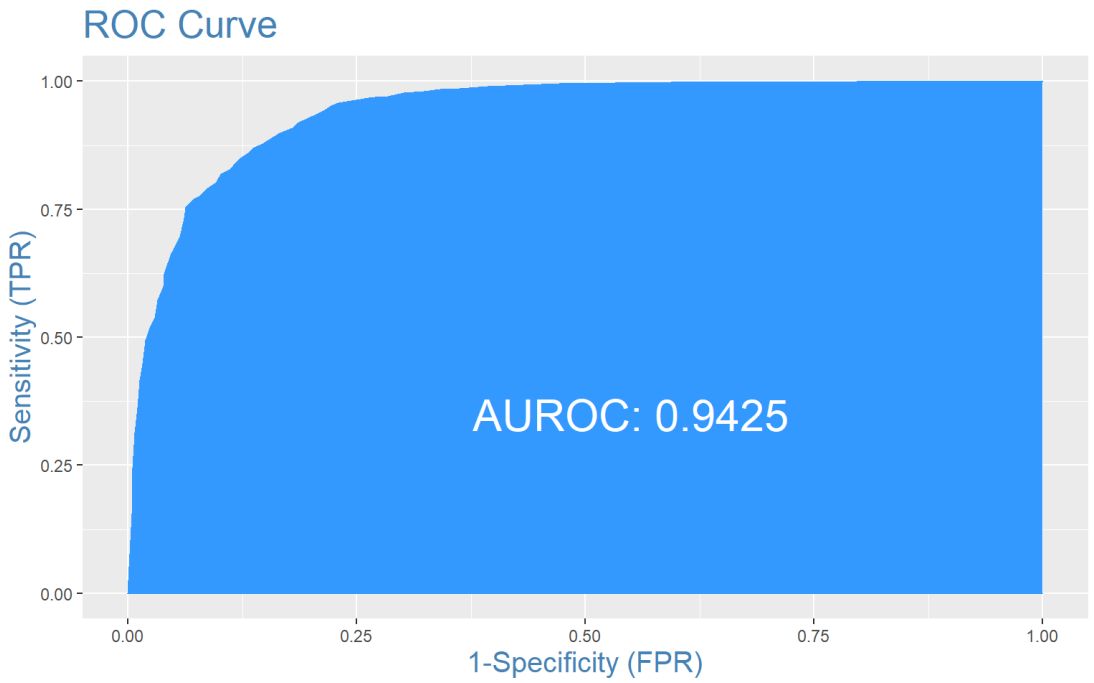
Area Under the ROC Curve

$$AUC = \%\, Concordant + \frac{1}{2}(\%\, Tied)$$

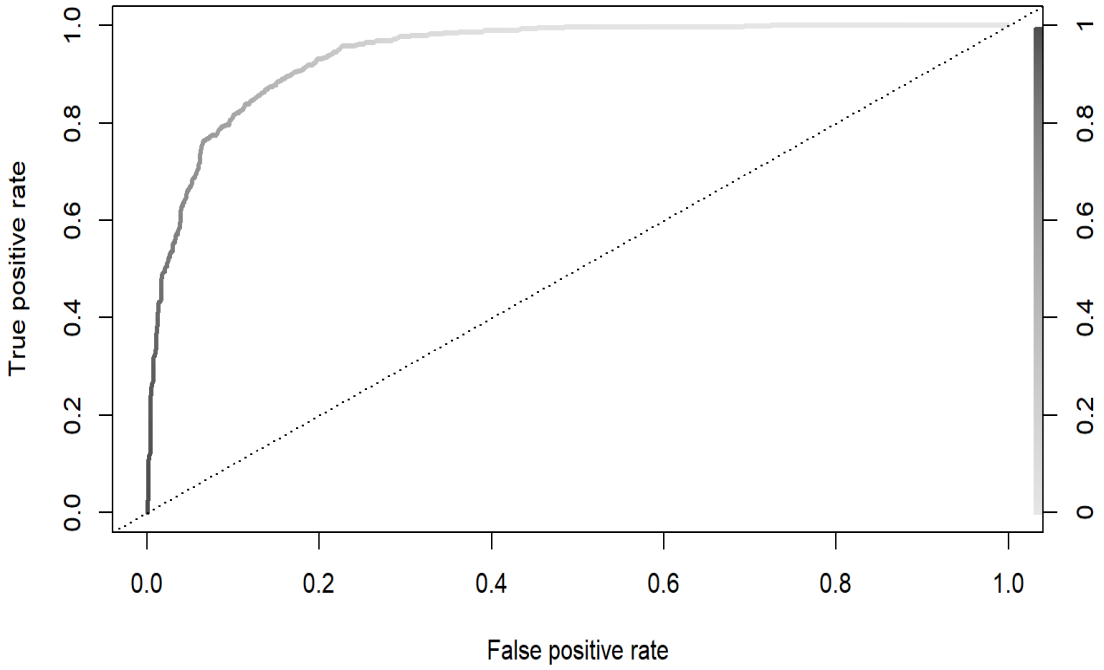


ROC Curve

```
plotROC(train$Bonus, train$p_hat)
```




```
pred <- prediction(fitted(logit.model), factor(train$Bonus))
perf <- performance(pred, measure = "tpr",
                          x.measure = "fpr")

plot(perf, lwd = 3, colorize = TRUE, colorkey = TRUE,
     colorize.palette = rev(gray.colors(256)))

abline(a = 0, b = 1, lty = 3)
```

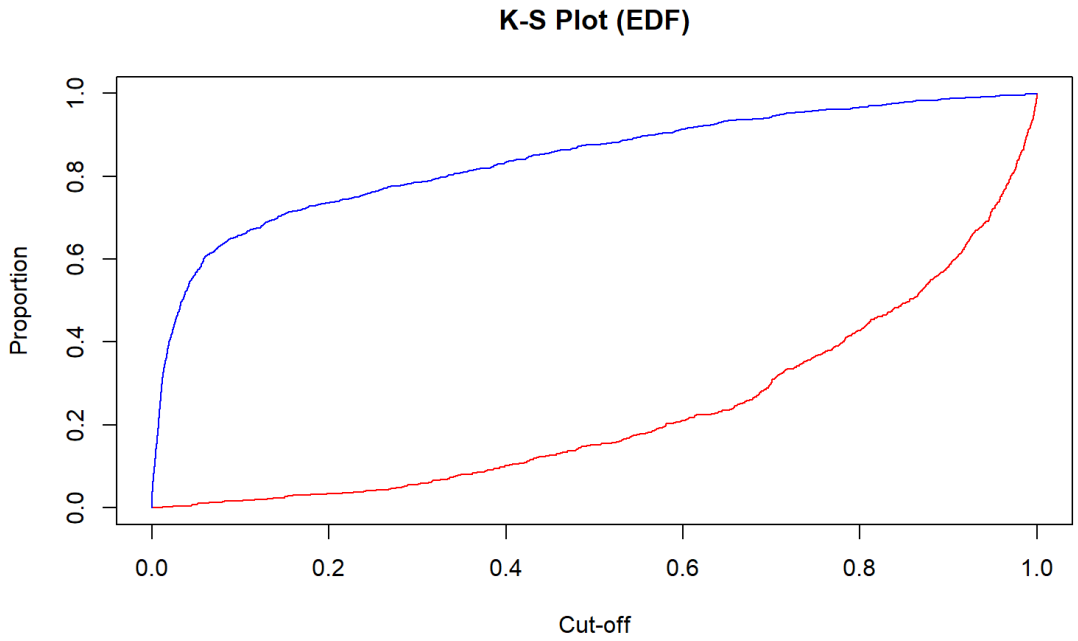


ASSESSING PREDICTIVE POWER

KS Statistic

K-S Statistic

- Very popular measure in banking and finance industries.
- The Two-Sample K-S statistic can determine if there is a difference between two cumulative distribution functions.
- Has a corresponding hypothesis test, with **D test statistic** (used for model comparison), and p-value.



K-S Statistic or Youden?

- D test statistic is used for model comparison.

$$D = \max(TPR - FPR)$$
$$= \max(Sensitivity + Specificity - 1)$$
$$= \max(Youden\ J)$$

- Mathematically **equivalent** to Youden's J statistic.

Best Cut-off?

- **Always** consider the cost of false positives and false negatives when doing classification.
- When **NOT** considering costs, many different techniques to "optimal" cut-off.
- $\boldsymbol{KS}$ **statistic D** (maximum difference between TPR and FPR):

$$D = \max_{depth}(TPR - FPR)$$

- "Optimal" – select cut-off that produces highest D statistic (same as Youden's).

K-S Statistic

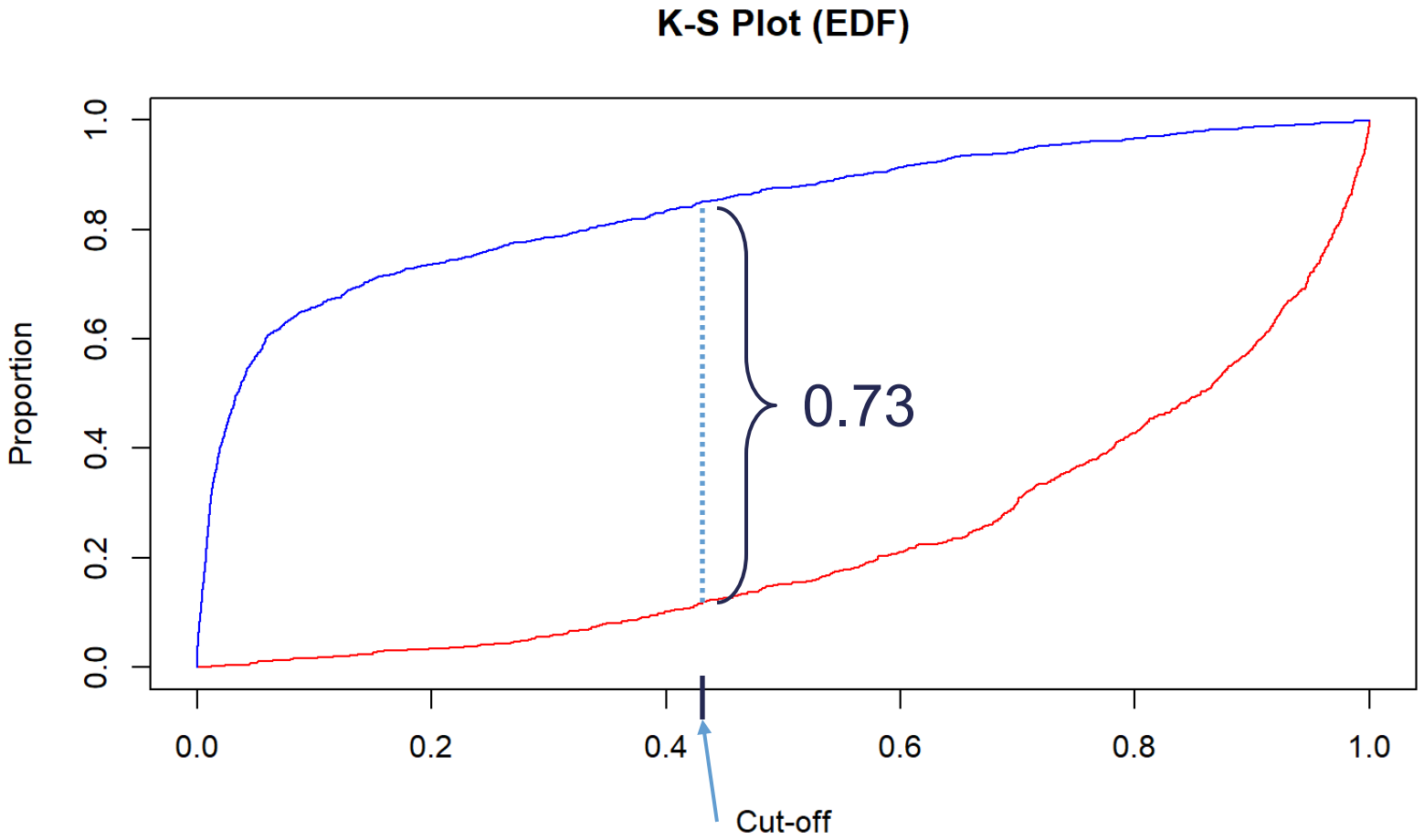
```
ks_stat(train$Bonus, train$p_hat)
```

```
[1] 0.7323
```

```
perf <- performance(pred, measure = "tpr", x.measure = "fpr")
KS <- max(perf@y.values[[1]] - perf@x.values[[1]])
cutoffAtKS <- unlist(perf@alpha.values)[which.max(perf@y.values[[1]]
                      - perf@x.values[[1]])]

print(c(KS, cutoffAtKS))
```

```
[1] 0.7352326 0.4229724
```

K-S Statistic

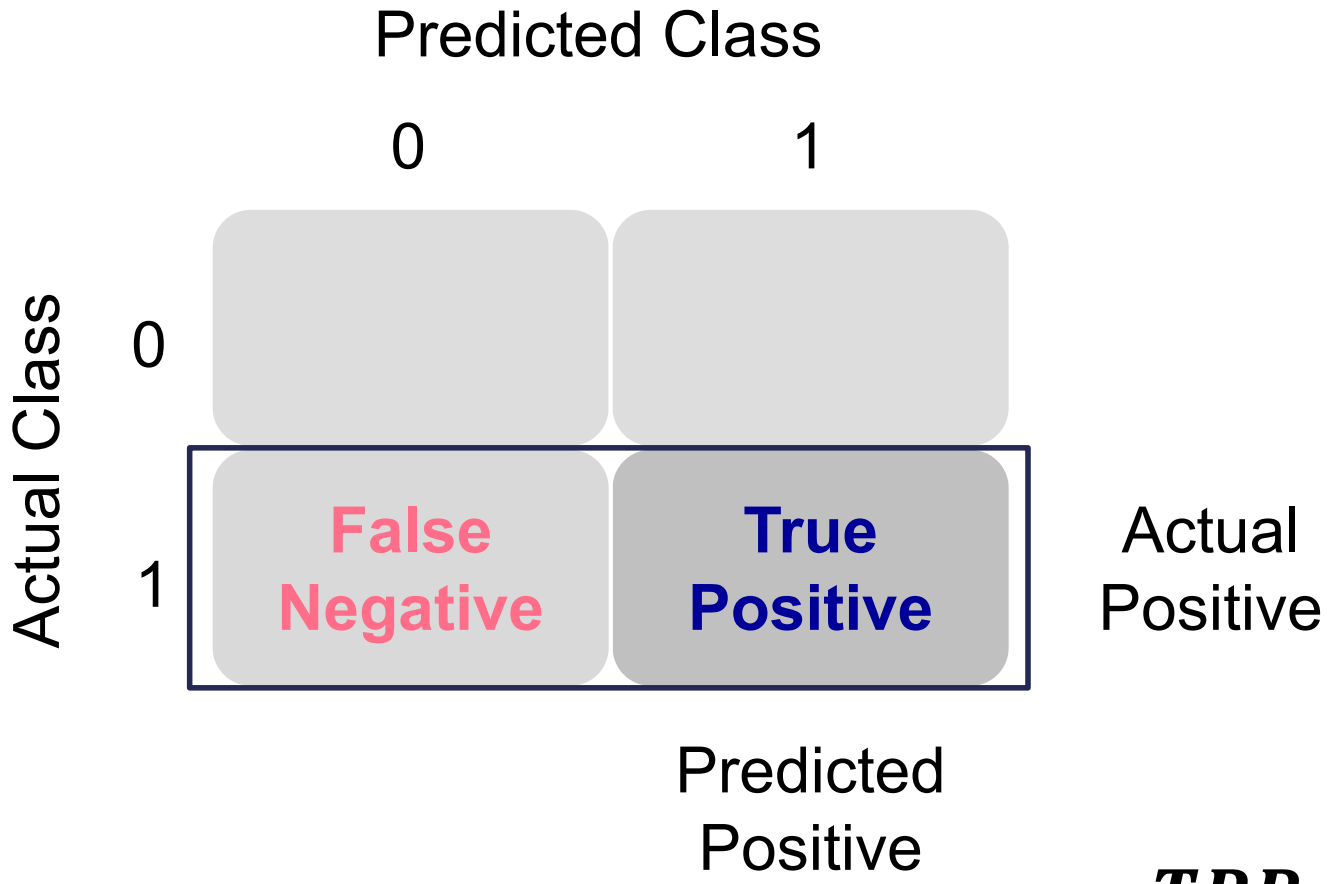


0.423 – “optimal” cut-off

ASSESSING PREDICTIVE POWER

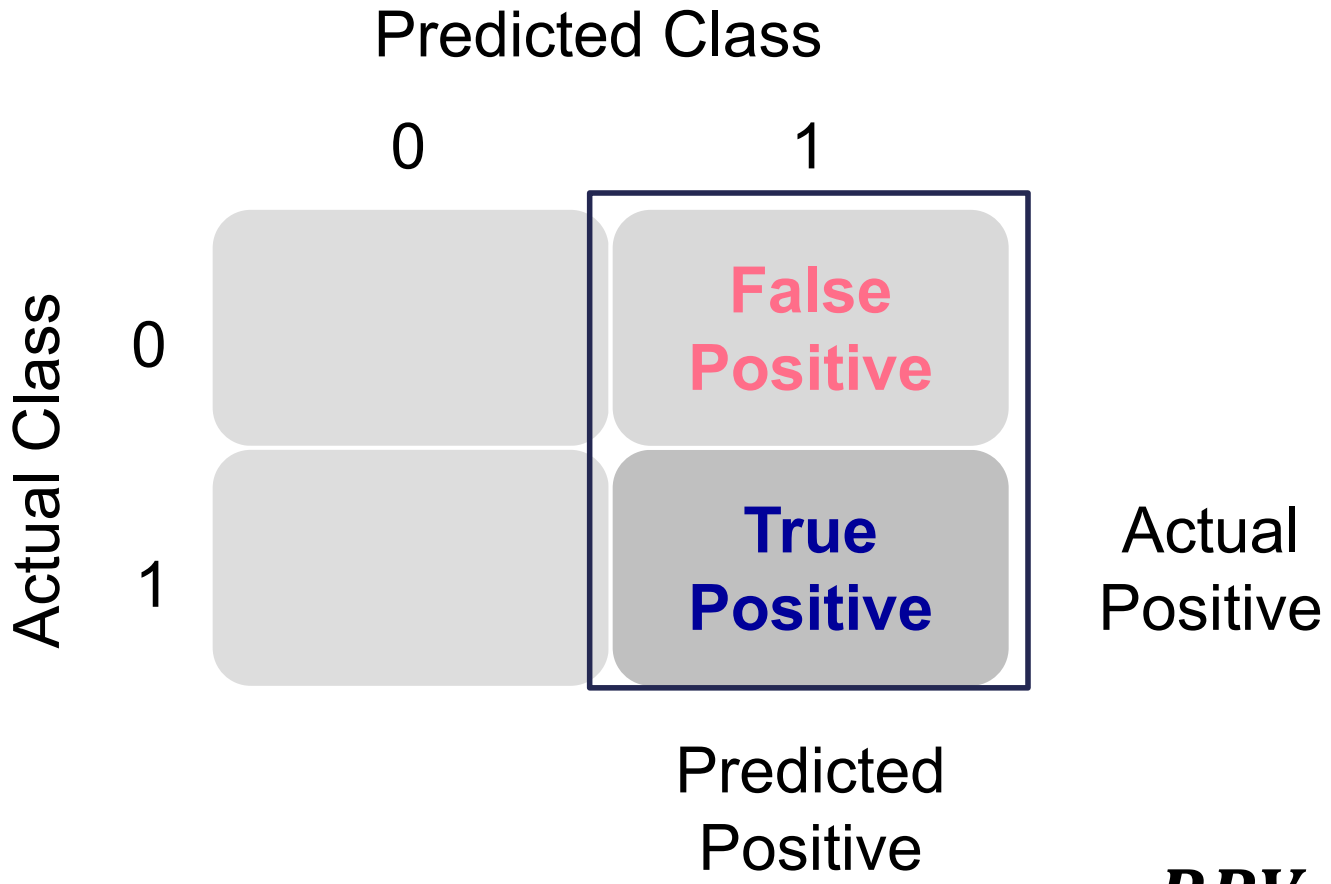
Precision vs. Recall

Sensitivity / Recall



$$TPR = \frac{TP}{TP + FN}$$

Precision



$$PPV = \frac{TP}{TP + FP}$$

Best Cut-off?

- **Always** consider the cost of false positives and false negatives when doing classification.
- When **NOT** considering costs, many different techniques to "optimal" cut-off.
- $\boldsymbol{F_1}$ **score** (precision-recall version of Youden's Index):

$$F_1 = 2\left(\frac{\text{precision}\times\text{recall}}{\text{precision} + \text{recall}}\right)$$

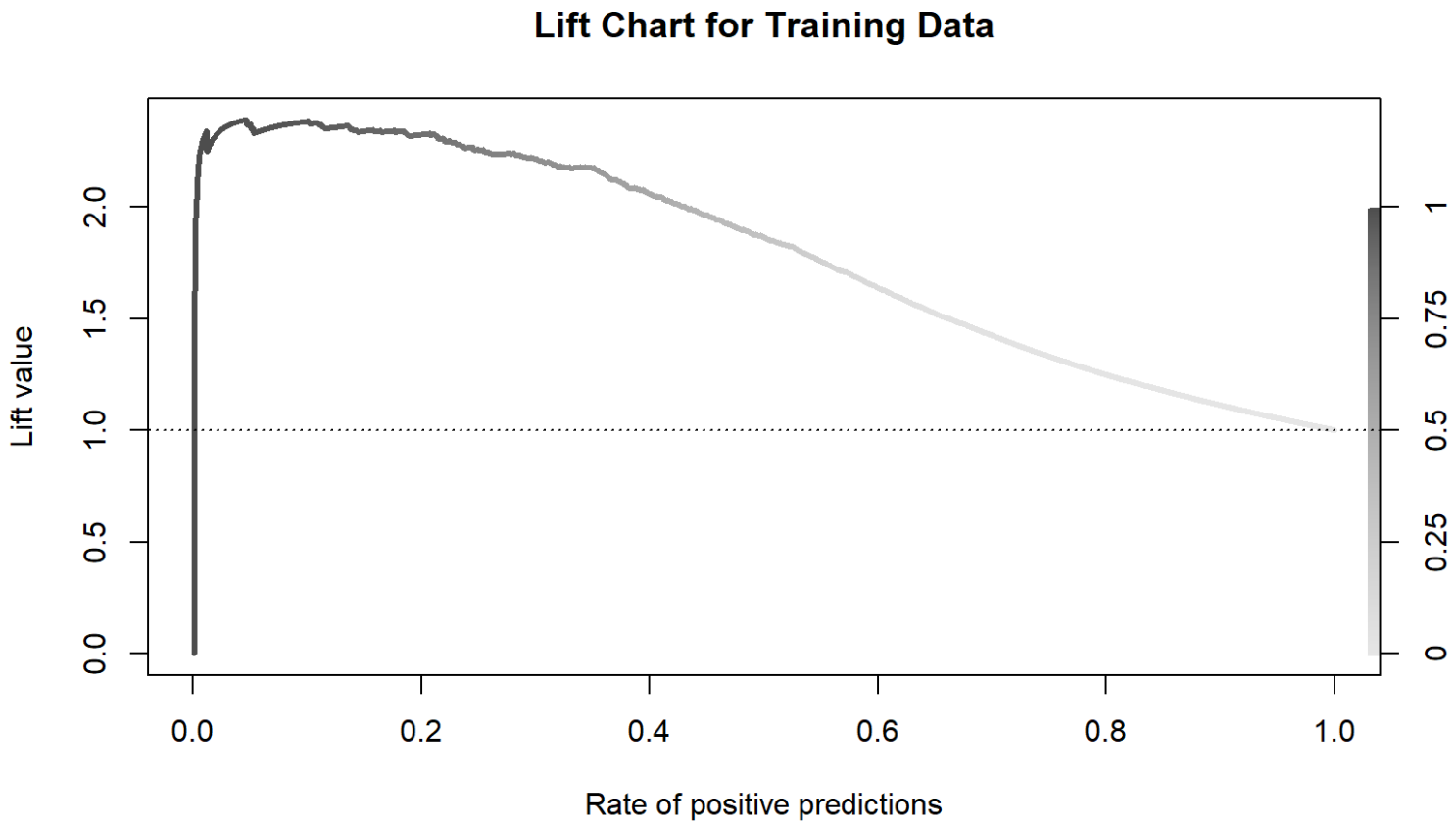
- "Optimal" – precision and recall are weighed equally, so select cut-off that produces highest F_1 score.

Precision & Lift

$$PPV = \frac{TP}{TP + FP}$$

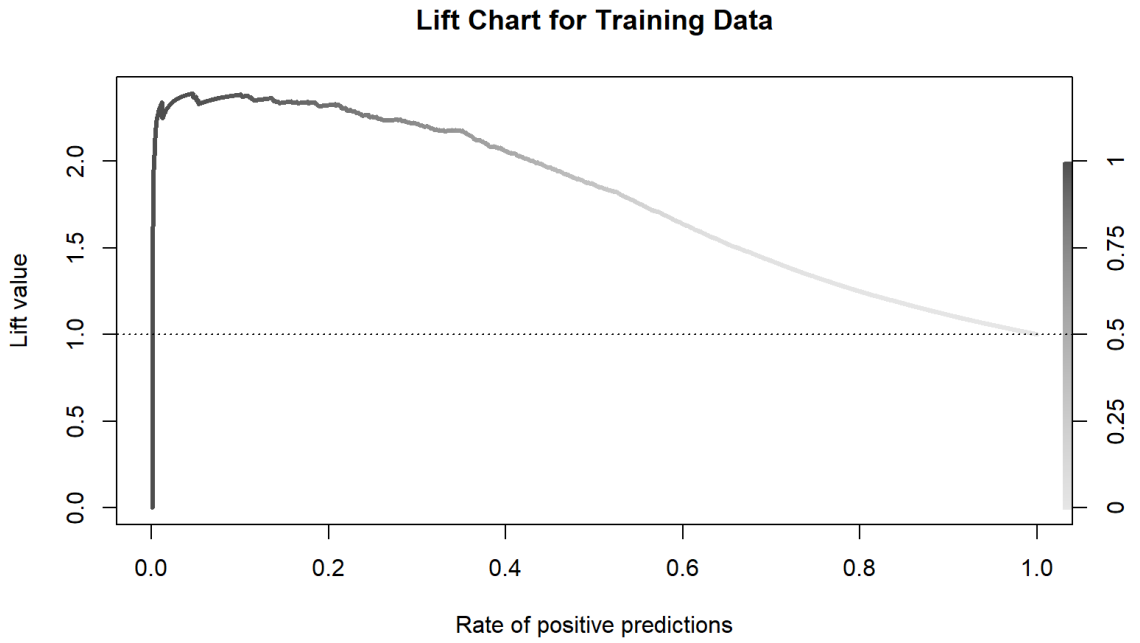
$$PPV = \frac{TP}{Depth}$$

$$Lift = \frac{PPV}{\pi_1}$$



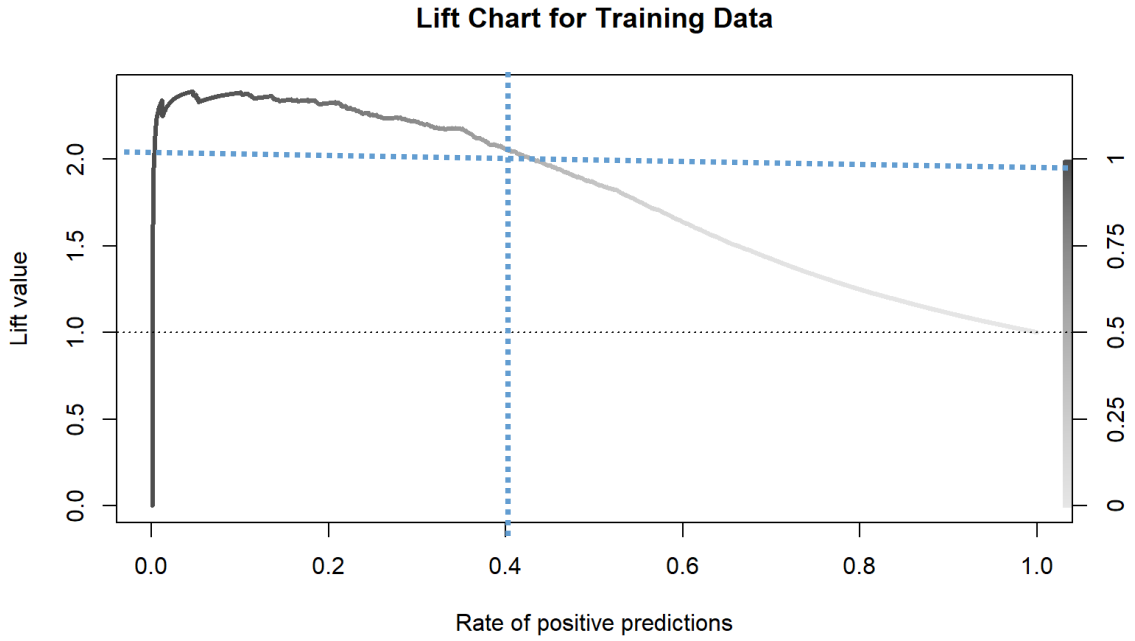
Lift Interpretation

- The top **depth**% of your customers, based on predicted probability, you get **lift** times as many responses compared to targeting a random sample of **depth**% of your customers.



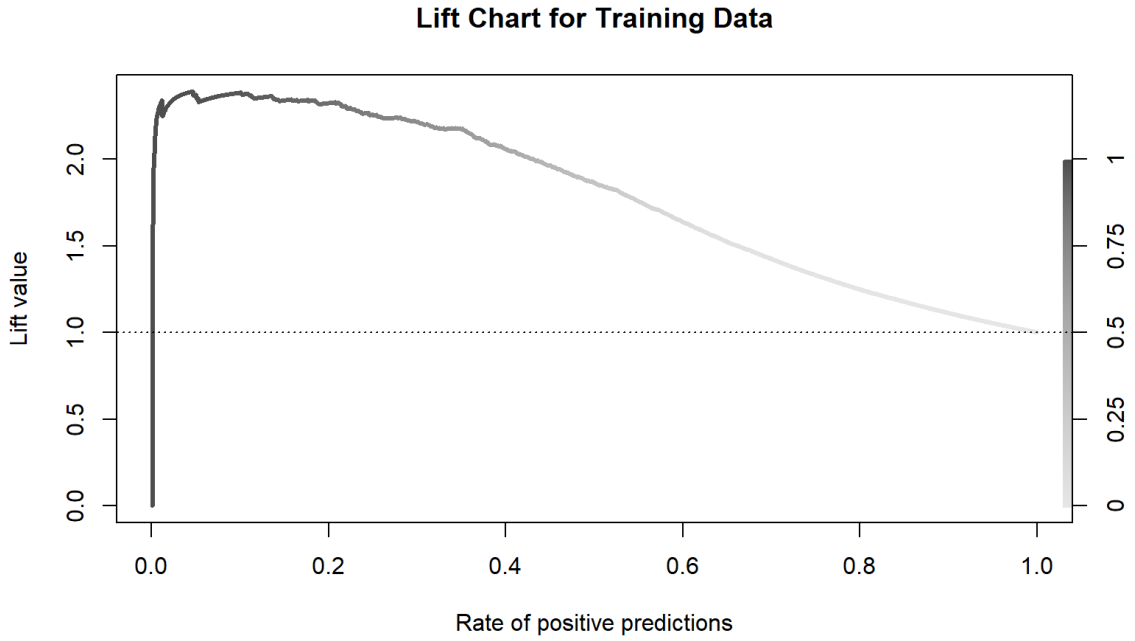
Lift Interpretation

- The top **40**% of your customers, based on predicted probability, you get **2** times as many responses compared to targeting a random sample of **40**% of your customers.



Lift Interpretation

- The top **depth**% of your customers, based on predicted probability, you get **lift** times as many responses compared to targeting a random sample of **depth**% of your customers.
- Careful, in oversampled data, you need to readjust your predicted probabilities!



Precision, Recall, F_1

```
sens <- NULL
spec <- NULL
youden <- NULL
cutoff <- NULL

for(i in 1:49){
  cutoff = c(cutoff, i/50)
  reca <- c(reca, sensitivity(train$Bonus, train$p_hat, threshold = i/50))
  prec <- c(prec, precision(train$Bonus, train$p_hat, threshold = i/50))
  f1 <- c(f1, 2*((prec[i]*reca[i])/(prec[i] + reca[i])))
}

ctable <- data.frame(cutoff, sens, prec, f1)

print(ctable[order(-f1),])
```

Precision, Recall, F_1

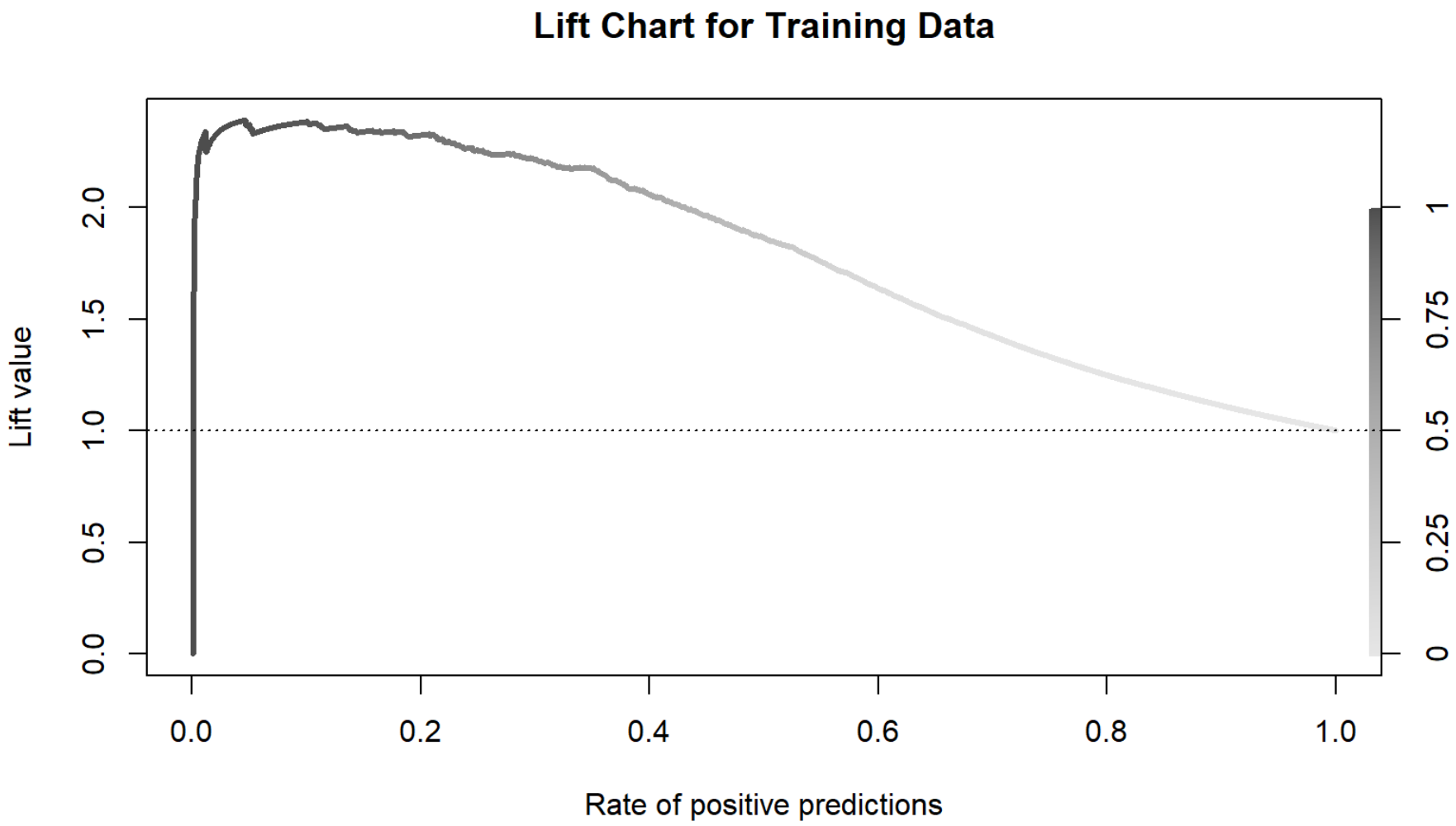
	cutoff	reca	prec	f1
23	0.46	0.8690476	0.8147321	0.8410138
21	0.42	0.8916667	0.7951168	0.8406285
20	0.40	0.8976190	0.7895288	0.8401114
22	0.44	0.8773810	0.8054645	0.8398860
18	0.36	0.9178571	0.7740964	0.8398693
17	0.34	0.9261905	0.7665025	0.8388140
24	0.48	0.8583333	0.8193182	0.8383721
19	0.38	0.9083333	0.7777778	0.8380011
25	0.50	0.8488095	0.8271462	0.8378378
16	0.32	0.9345238	0.7591876	0.8377801
14	0.28	0.9511905	0.7481273	0.8375262
15	0.30	0.9428571	0.7521368	0.8367670
13	0.26	0.9571429	0.7423823	0.8361934

⋮

Lift Chart

```
perf <- performance(pred, measure = "lift", x.measure = "rpp")

plot(perf, lwd = 3, colorize = TRUE, colorkey = TRUE,
     colorize.palette = rev(gray.colors(256)),
     main = "Lift Chart for Training Data")
abline(h = 1, lty = 3)
```

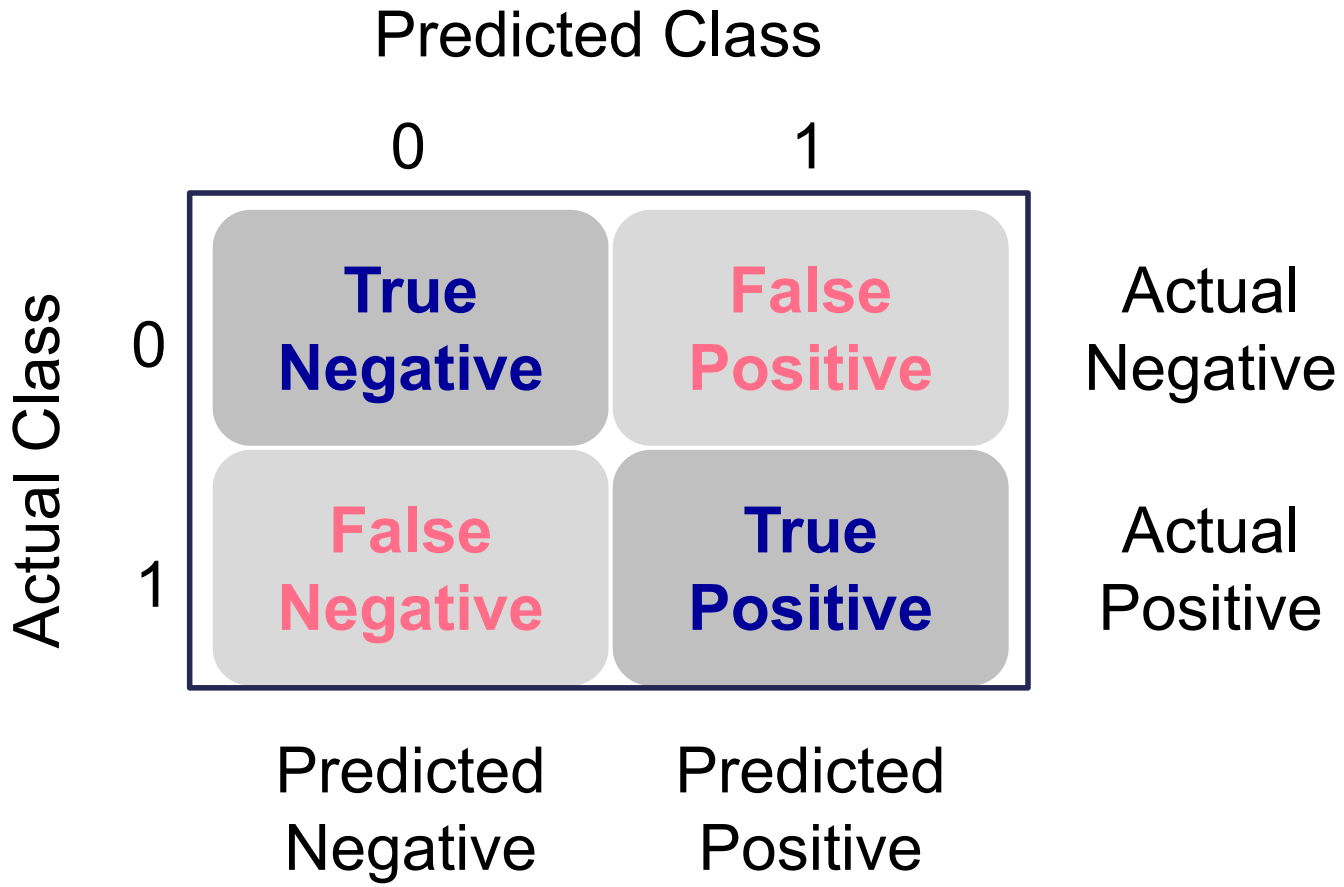
Lift Chart


Lift Chart for Training Data
Lift value
0.0
0.5
1.0
1.5
2.0
0.0
0.2
0.4
0.6
0.8
1.0
Rate of positive predictions
0
0.25
0.5
0.75
1

ASSESSING PREDICTIVE POWER

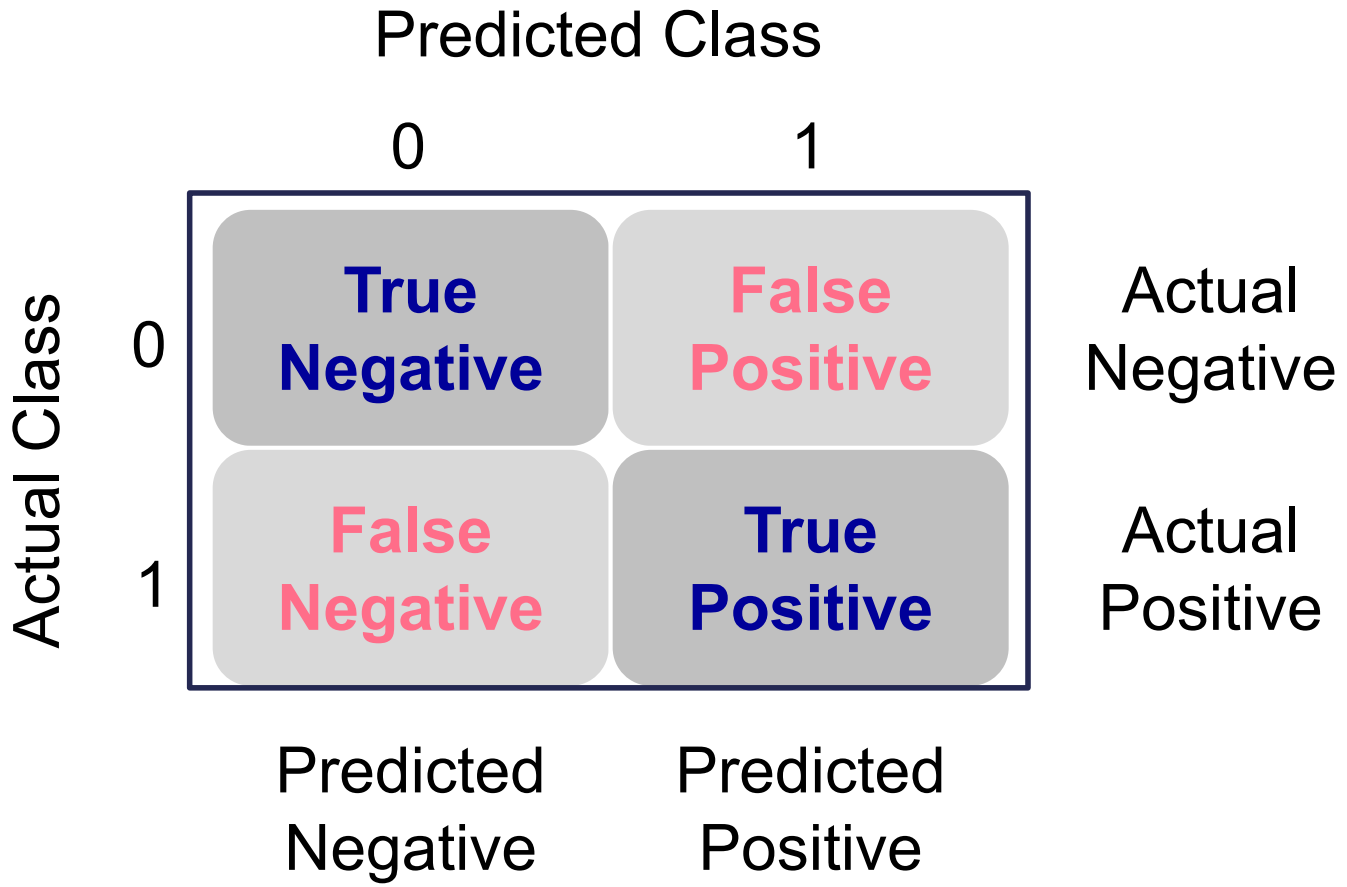
Accuracy vs. Error

Accuracy



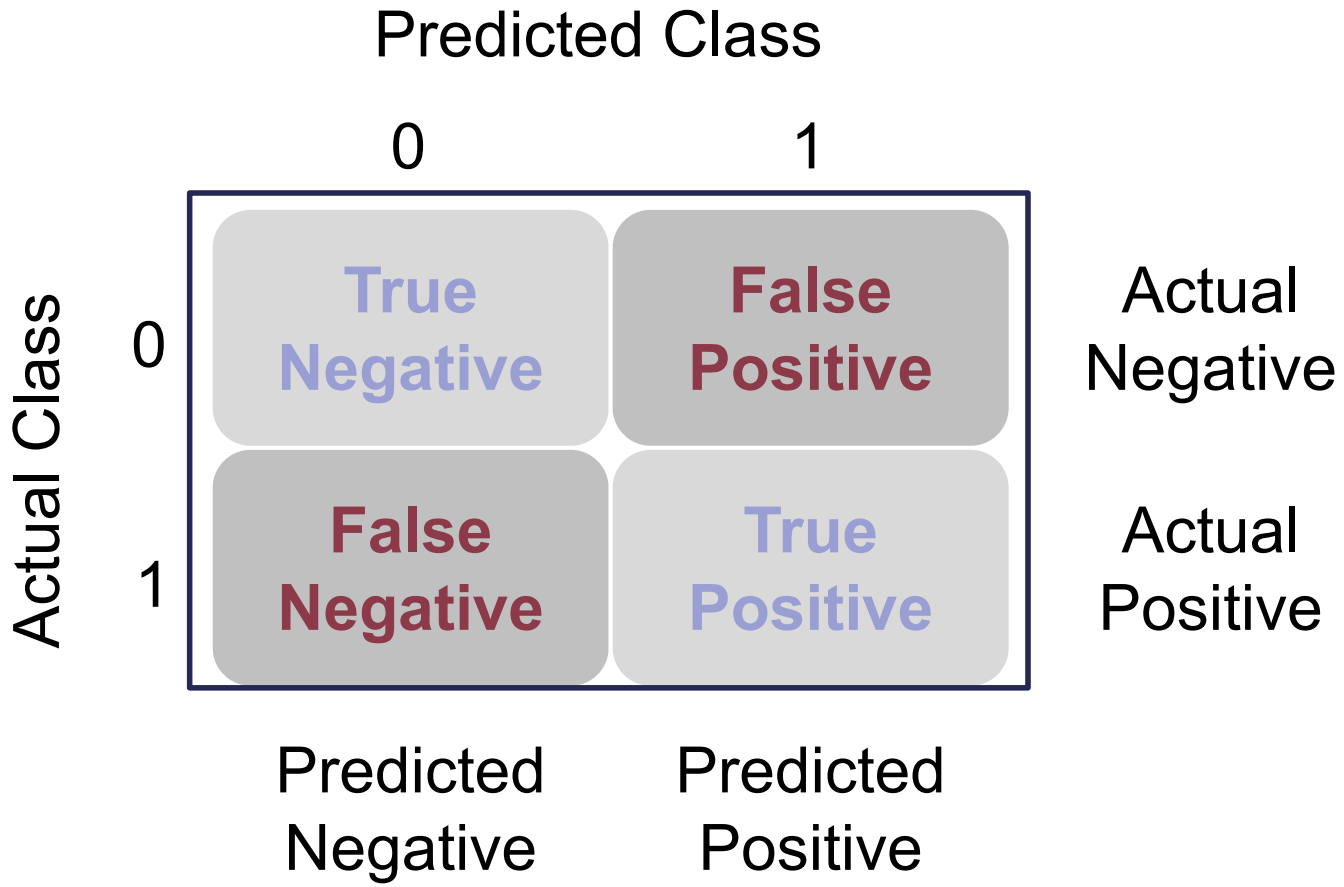
$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Accuracy



$$Accuracy = \frac{TP + TN}{n}$$

Misclassification (Error) Rate



$$Error = \frac{FP + FN}{n}$$

Accuracy and Error

- Accuracy and error can be easily fooled so careful focusing only on them.
- If your data has 10% events and 90% non-events, you can have a 90% accurate model by guessing non-events for **every** observation.
- There is more to model building than simply maximizing overall classification accuracy.
- Good numbers to report, but not necessarily to choose models on.

Closing Thoughts on Classification

- Classification is a **decision** that is extraneous to statistical modeling.
- Although logistic regression tends to work well in classification, it is a **probability model** and does not output 1's and 0's.
- Classification assumes cost for each individual is the same.
  - Useful for groups.
  - Careful about single observation decisions.