

Statistical Modelling III

HES 505 Fall 2023: Session 24

Matt Williamson

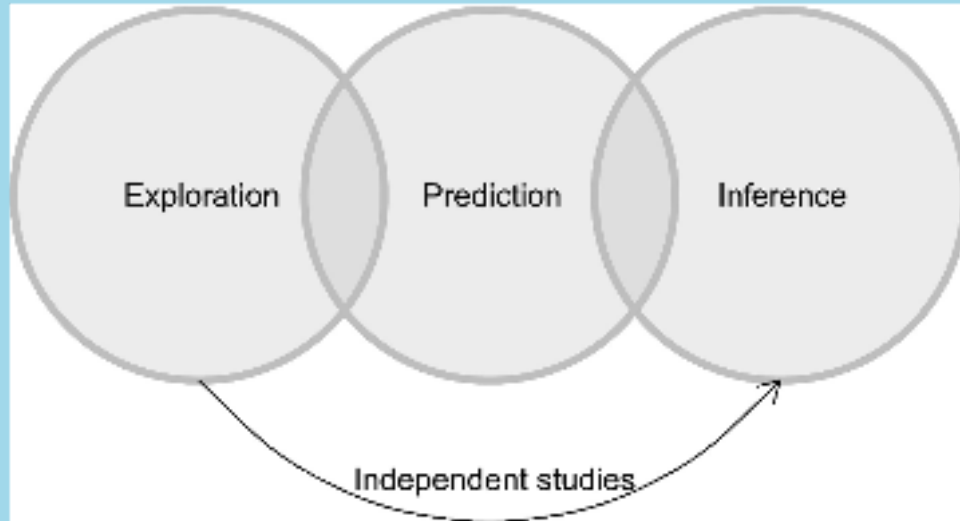
Objectives

By the end of today you should be able to:

- Articulate three different reasons for modeling and how they link to assessments of fit
- Describe and implement several test statistics for assessing model fit
- Describe and implement several assessments of classification
- Describe and implement resampling techniques to estimate predictive performance

The 3 Faces of Models

Best Model for What?



from Tradennick et al. 2021

- **Exploration:** describe patterns in the data and generate hypotheses
- **Inference:** evaluate the strength of evidence for some statement about the process
- **Prediction:** forecast outcomes at unsampled locations based on covariates

The Importance of Model Fit

- The general regression context:

$$\hat{y} = \mathbf{X}\hat{\beta}$$

- **Inference** is focused on robust estimates of $\hat{\beta}$ given the data we have
- **Prediction** is focused on accurate forecasts of \hat{y} at locations where we have yet to collect the data

Inference and Presence/Absence Data

- is conditional on variables in the model **and** those not in the model

```
1 nsamp <- 1000
2 df <- data.frame(x1 = rnorm(nsamp,0,1),
3                   x2 = rnorm(nsamp,0,1),
4                   x3 = rnorm(nsamp,0,1))
5
6 linpred <- 1 + 2*df$x1 -0.18*df$x2 -3.5*df$x3
7 y <- rbinom(nsamp, 1, plogis(linpred))
8 df <- cbind(df, y)
9
10 mod1 <- glm(y~x1 +x2, data=df, family="binomial")
11 mod2 <- glm(y~x1 +x2 + x3, data=df, family="binomial")
```

Inference & Presence/Absence Data

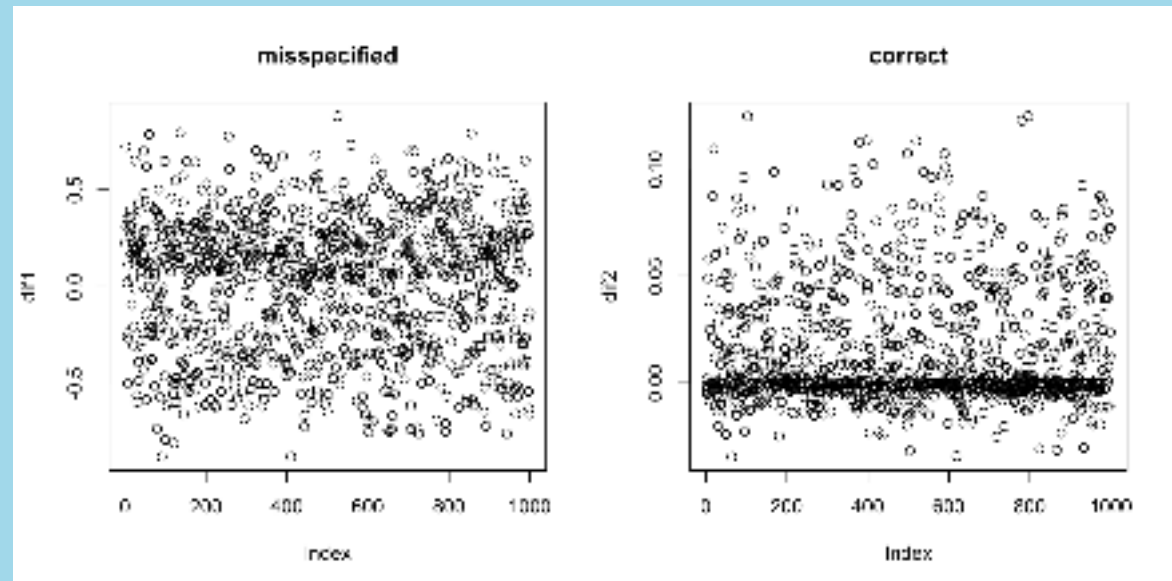
```
1 coef(mod1)
```

```
(Intercept)          x1  
x2  
  0.4438852    0.9572188  
-0.1308243
```

```
1 coef(mod2)
```

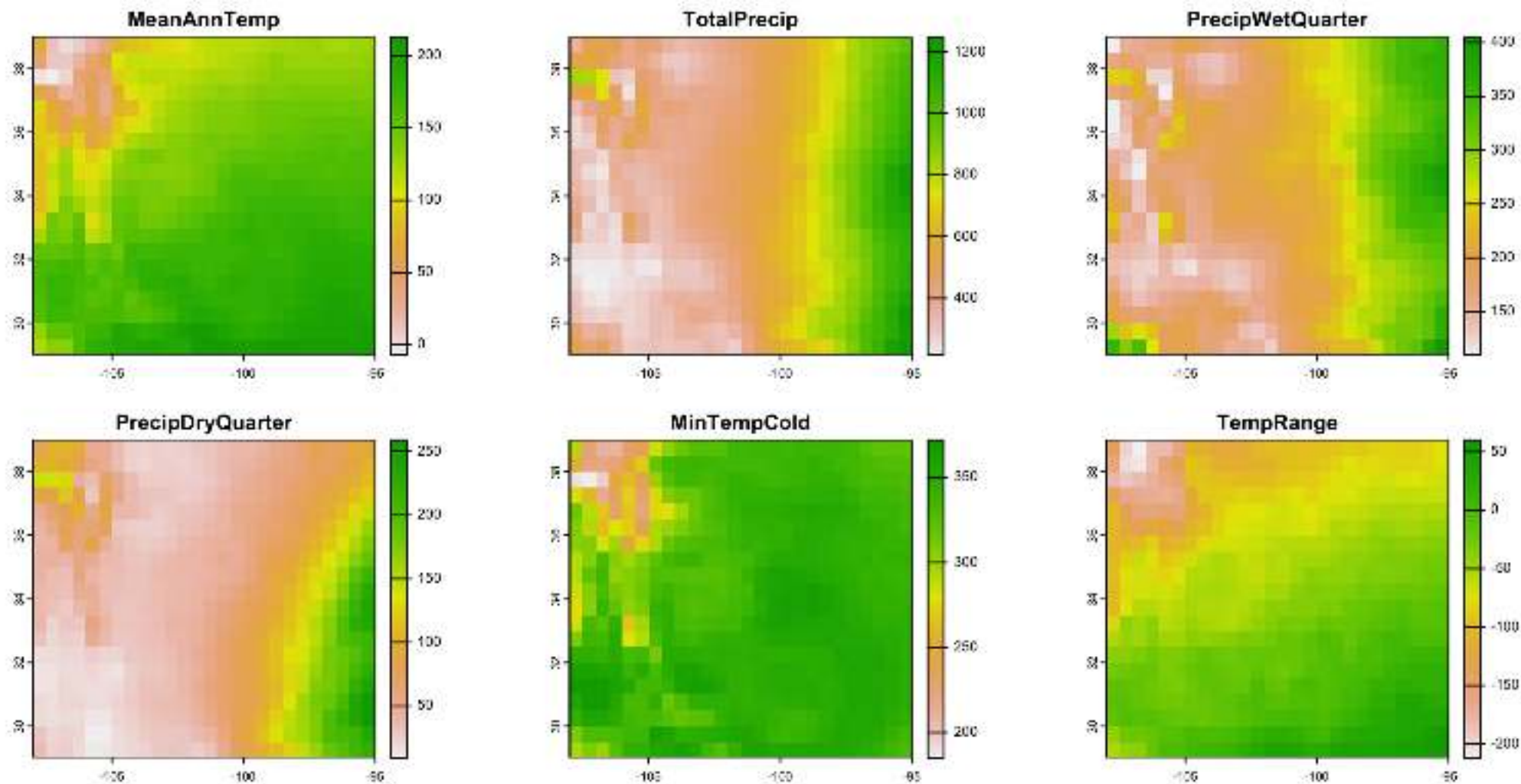
```
(Intercept)          x1  
x2          x3  
  0.9156104    2.3558197  
-0.2715651   -3.8891961
```

```
1 prd1 <- predict(mod1, df, "response")  
2 dif1 <- plogis(linpred) - prd1  
3 prd2 <- predict(mod2, df, "response")  
4 dif2 <- plogis(linpred) - prd2
```



Inferring coefficient effects requires that your model fit the data well

Assessing Model Fit



Using Test Statistics

- for linear regression:
 - Perfect prediction ($R^2 = 1$); ; and
 - Null prediction (Intercept only) ($R^2 = 0$); ; and
 - No direct way of implementing for logistic regression

Pseudo-

- Cohen's Likelihood Ratio
- Deviance (D), the difference between the model and some hypothetical perfect model (lower is better)
- Challenge: Not monotonically related to
- Challenge: How high is too high?

Cohen's Likelihood Ratio

```
1 logistic.rich <- glm(y ~ MeanAnnTemp + PrecipWetQuarter + PrecipDryQuarter,  
2                     family=binomial(link="logit"),  
3                     data=pts.df[,2:8])  
4  
5 with(logistic.rich,  
6       null.deviance - deviance)/with(logistic.rich,  
7                                       null.deviance)
```

```
[1] 0.4495966
```

Pseudo-

- Cox and Snell
- Likelihood (), the probability of observing the sample given an assumed distribution
- Challenge: Maximum value is less than 1 and changes with
- Correction by Nagelkerke so that maximum is 1

Cox and Snell

```
1 logistic.null <- glm(y ~ 1,  
2                       family=binomial(link="logit"),  
3                       data=pts.df[,2:8])  
4  
5 1 - exp(2*(logLik(logistic.null)[1] - logLik(logistic.rich)[1])/nobs(logist  
[1] 0.4308873
```

Using Test Statistics

- Based on the data used in the model (i.e., not prediction)
- Likelihood Ratio behaves most similarly to
- Cox and Snell (and Nagelkerke) increases with more presences
- Ongoing debate over which is “best”
- **Don't defer to a single statistic**

Assessing Predictive Ability

Predictive Performance and Fit

- Predictive performance can be an estimate of fit
- Comparisons are often relative (better good)
- Theoretical and subsampling methods

Theoretical Assessment of Predictive Performance



- Information Criterion Methods
- Minimize the amount of information lost by using model to approximate true process
- Trade-off between fit and overfitting
- Can't know the true process (so comparisons are relative)

Hirotugu Akaike of AIC

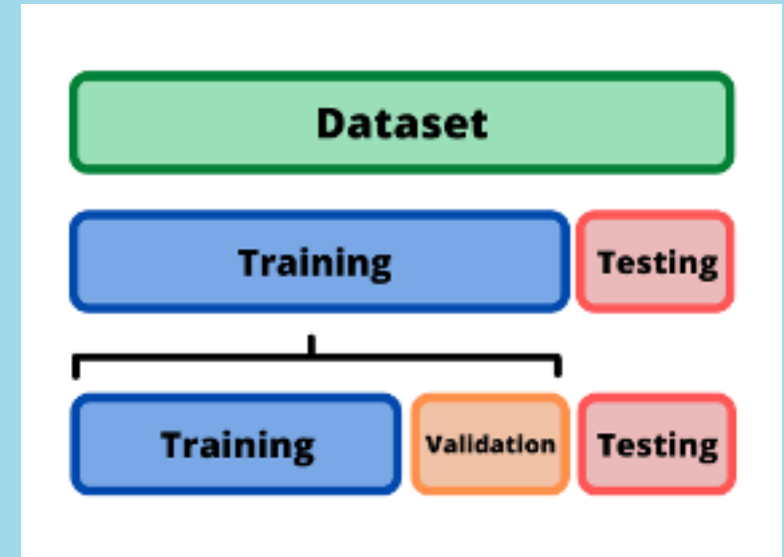
AIC Comparison

```
1 logistic.null <- glm(y ~ 1,  
2                       family=binomial(link="logit"),  
3                       data=pts.df[,2:8])  
4  
5 logistic.rich <- glm(y ~ MeanAnnTemp + PrecipWetQuarter + PrecipDryQuarter,  
6                       family=binomial(link="logit"),  
7                       data=pts.df[,2:8])  
8  
9 AIC(logistic.null, logistic.rich)
```

	df	AIC
logistic.null	1	127.37389
logistic.rich	4	77.00622

Sub-sampling Methods

- Split data into *training* and *testing*
- Testing set needs to be large enough for results to be statistically meaningful
- Test set should be representative of the data as a whole
- Validation data used to tune parameters (not always)



Subsampling your data with **caret**

```
1 pts.df$y <- as.factor(ifelse(pts.df$y == 1, "Yes", "No"))
2 library(caret)
3 Train <- createDataPartition(pts.df$y, p=0.6, list=FALSE)
4
5 training <- pts.df[ Train, ]
6 testing <- pts.df[ -Train, ]
```

Misclassification

- Confusion matrices compare actual values to predictions
- True Positive (TN) - This is correctly classified as the class of interest / target.
- True Negative (TN) - This is correctly classified as not a class of interest / target.
- False Positive (FP) - This is wrongly classified as the class of interest / target.
- False Negative (FN) - This is wrongly classified as not a class of interest / target.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Confusion Matrices in R

```
1 train.log <- glm(y ~ .,  
2                 family="binomial"  
3                 data=training[,2:  
4  
5 predicted.log <- predict(train.log  
6                         newdata=t  
7                         type="res  
8  
9 pred <- as.factor(  
10   ifelse(predicted.log > 0.5,  
11           "Yes",  
12           "No" ))
```

```
1 confusionMatrix(testing$y, pred)
```

Confusion Matrix and Statistics

	Reference	
Prediction	No	Yes
No	24	3
Yes	2	10

Accuracy : 0.8718
95% CI : (0.7257, 0.957)
No Information Rate : 0.6667
P-Value [Acc > NIR] : 0.003321

Kappa : 0.7059

Mcnemar's Test P-Value : 1.000000

Sensitivity : 0.9231
Specificity : 0.7692
Pos Pred Value : 0.8889
Neg Pred Value : 0.8333
Prevalence : 0.6667
Detection Rate : 0.6154

Confusion Matrices

**Depends upon
threshold!!**

Confusion Matrices in R

```
1 library(tree)
2 tree.model <- tree(y ~ . , training)
3 predict.tree <- predict(tree.model,
```

```
1 confusionMatrix(testing$y, predict.tree)
```

Confusion Matrix and Statistics

	Reference	
Prediction	No	Yes
No	22	5
Yes	2	10

Accuracy : 0.8205
95% CI : (0.6647, 0.9246)
No Information Rate : 0.6154
P-Value [Acc > NIR] : 0.005008

Kappa : 0.6061

Mcnemar's Test P-Value : 0.449692

Sensitivity : 0.9167
Specificity : 0.6667
Pos Pred Value : 0.8148
Neg Pred Value : 0.8333
Prevalence : 0.6154
Detection Rate : 0.5641

Confusion Matrices in R

```
1 library(randomForest)
2 class.model <- y ~ .
3 rf <- randomForest(class.model, da
4 predict.rf <- predict(rf, newdata=
```

```
1 confusionMatrix(testing$y, predict.rf)
```

Confusion Matrix and Statistics

	Reference	
Prediction	No	Yes
No	24	3
Yes	3	9

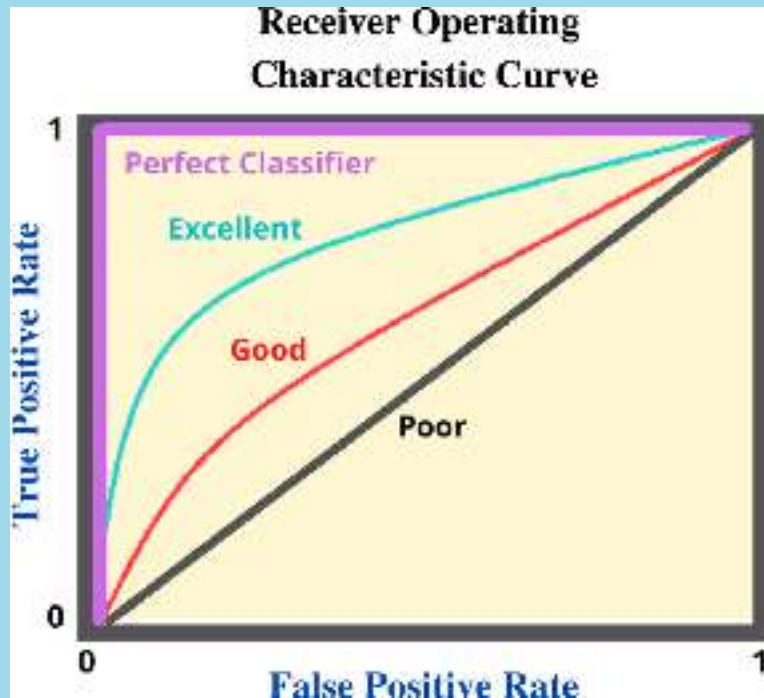
Accuracy : 0.8462
95% CI : (0.6947, 0.9414)
No Information Rate : 0.6923
P-Value [Acc > NIR] : 0.02323

Kappa : 0.6389

Mcnemar's Test P-Value : 1.00000

Sensitivity : 0.8889
Specificity : 0.7500
Pos Pred Value : 0.8889
Neg Pred Value : 0.7500
Prevalence : 0.6923
Detection Rate : 0.6154

Threshold-Free Methods



- Receiver Operating Characteristic Curves
- Illustrates discrimination of binary classifier as the threshold is varied
- Area Under the Curve (AUC) provides an estimate of classification ability

Criticisms of ROC/AUC

- Treats false positives and false negatives equally
- Undervalues models that predict across smaller geographies
- Focus on *discrimination* and not *calibration*
- New methods for presence-only data

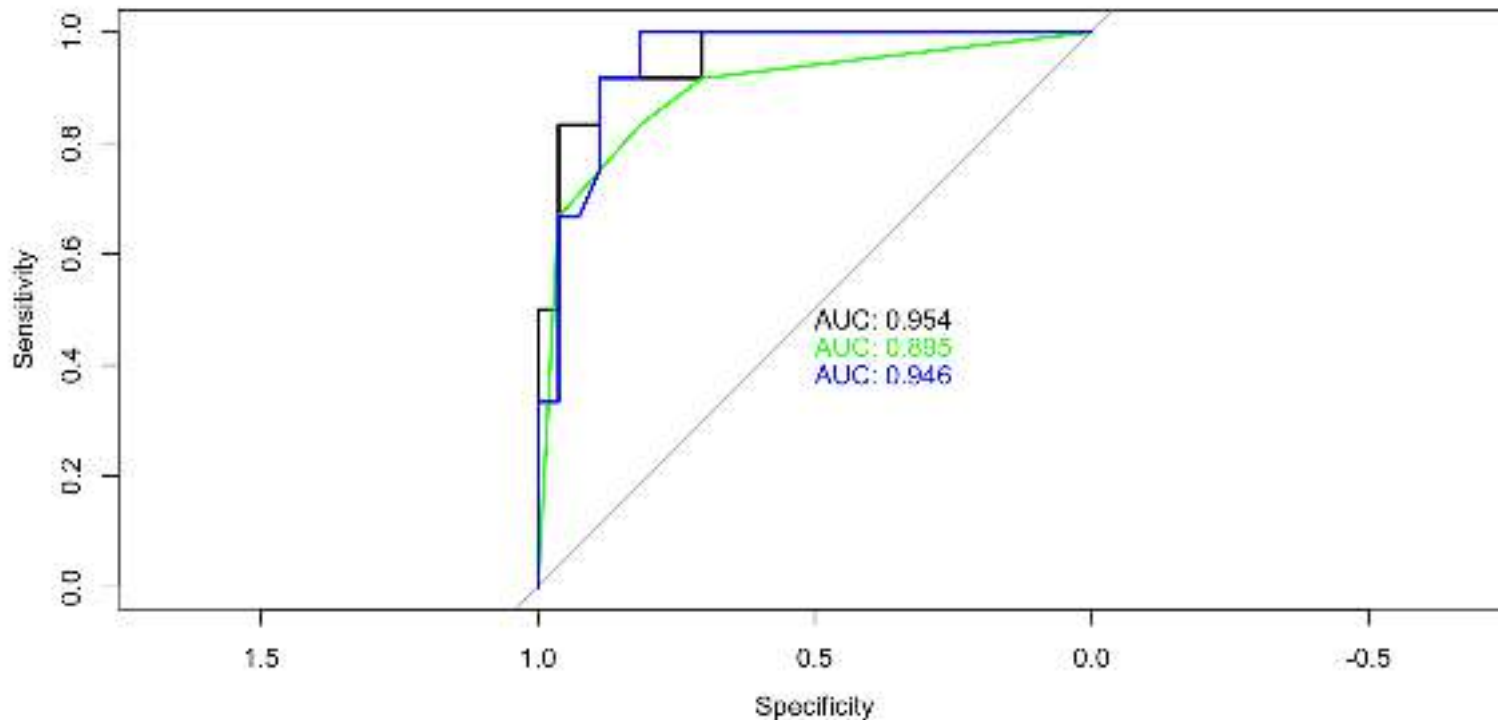
ROC in R (using pROC)

- Generate predictions (note the difference for tree and rf)

```
1 library(pROC)
2 train.log <- glm(y ~ .,
3                 family="binomial",
4                 data=training[,2:8])
5
6 predicted.log <- predict(train.log,
7                          newdata=testing[,2:8],
8                          type="response")
9
10 predict.tree <- predict(tree.model, newdata=testing[,2:8], type="vector")[,
11
12 predict.rf <- predict(rf, newdata=testing[,2:8], type="prob")[,2]
```

ROC in R (using pROC)

```
1 plot(roc(testing$y, predicted.log), print.auc=TRUE)
2
3 plot(roc(testing$y, predict.tree), print.auc=TRUE, print.auc.y = 0.45, col=
4
5 plot(roc(testing$y, predict.rf), print.auc=TRUE, print.auc.y = 0.4, col="bl
```



Cross-validation

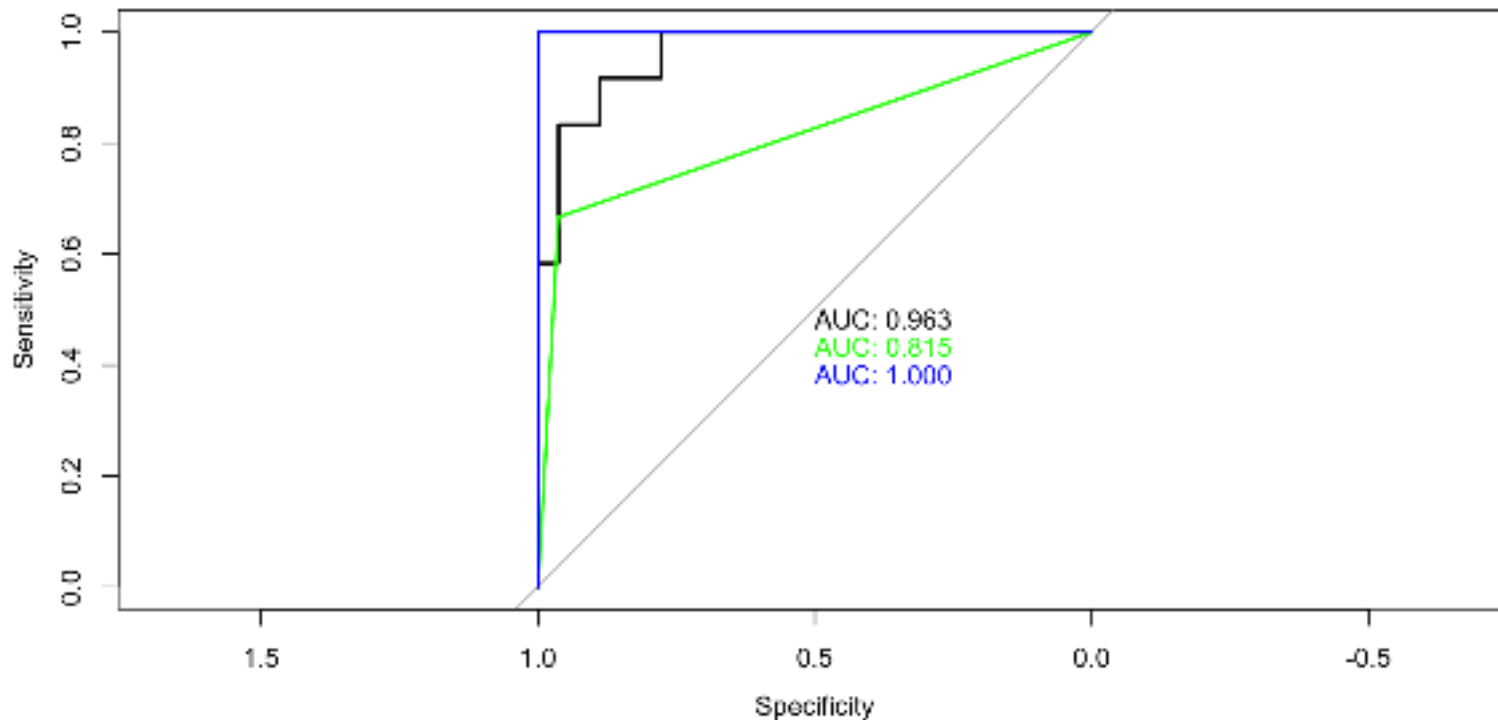
- Often want to make sure that fit/accuracy not a function of partition choice
- Cross-validation allows resampling of data (multiple times)
- K-fold - Data are split into K datasets of \sim equal size, model fit to observations to predict heldout set
- Leave One Out (LOO) - Model fit to $n-1$ observations to predict the held out observation

Crossvalidation in R using caret

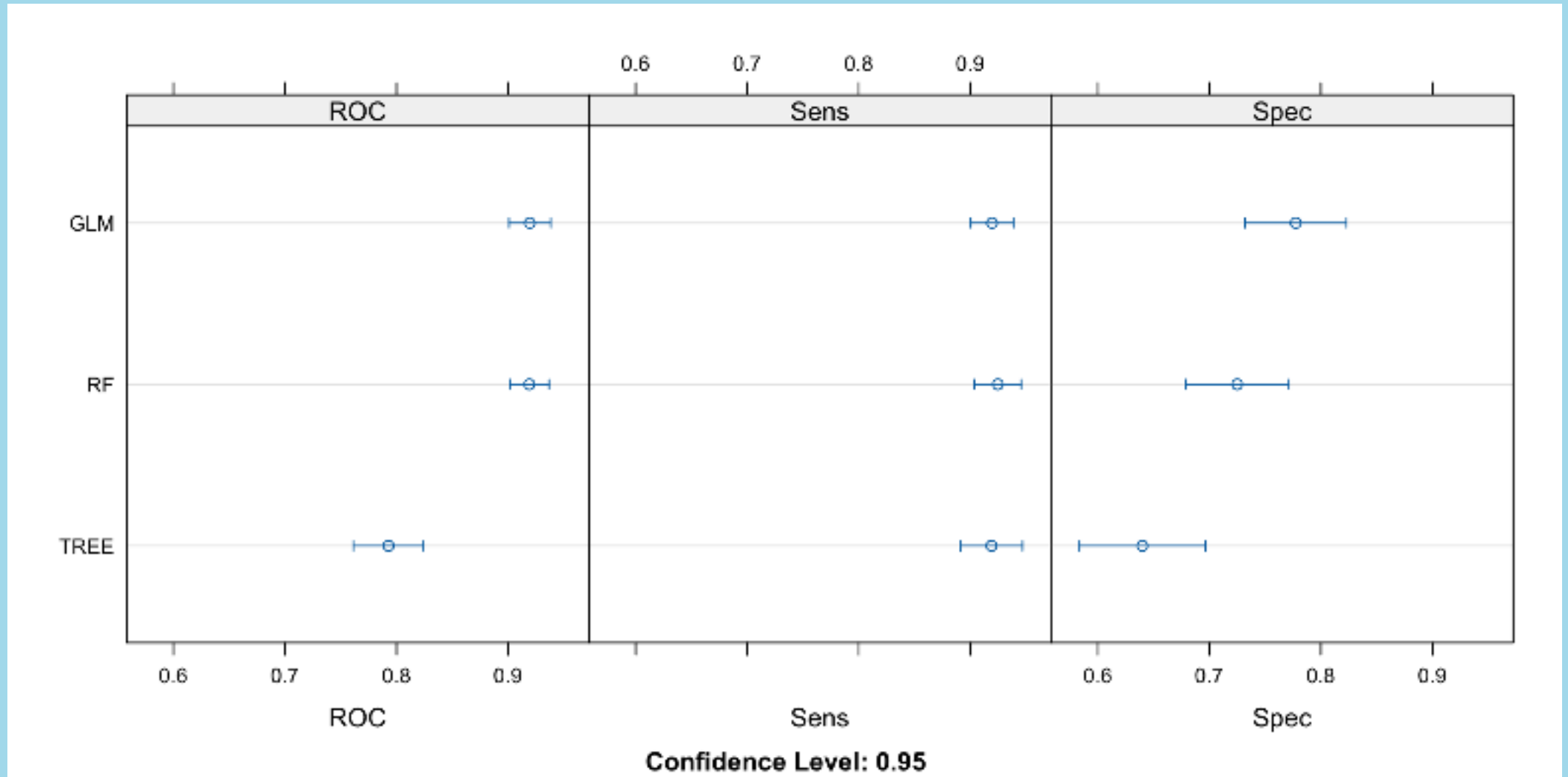
```
1 fitControl <- trainControl(method = "repeatedcv",
2                             number = 10,
3                             repeats = 10,
4                             classProbs = TRUE,
5                             summaryFunction = twoClassSummary)
6
7 log.model <- train(y ~., data = pts.df[,2:8],
8                   method = "glm",
9                   trControl = fitControl)
10 pred.log <- predict(log.model, newdata = testing[,2:8], type="prob")[,2]
11
12 tree.model <- train(y ~., data = pts.df[,2:8],
13                   method = "rpart",
14                   trControl = fitControl)
15
16 pred.tree <- predict(tree.model, newdata=testing[,2:8], type="prob")[,2]
17
18 rf.model <- train(v ~., data = pts.df[,2:8],
```

Crossvalidation in **R** using **caret**

```
1 plot(roc(testing$y, pred.log), print.auc=TRUE)
2
3 plot(roc(testing$y, pred.tree), print.auc=TRUE, print.auc.y = 0.45, col="gr
4
5 plot(roc(testing$y, pred.rf), print.auc=TRUE, print.auc.y = 0.4, col="blue"
```



Crossvalidation in **R** using **caret**



Spatial predictions

```
1 best.rf <- rf.model$finalModel
2 best.glm <- log.model$finalModel
3
4 rf.spatial <- terra::predict(pred.stack.scl, best.rf, type="prob")
5
6
7 glm.spatial <- terra::predict(pred.stack.scl, best.glm, type="response" )
```

Spatial predictions

