Multivariate Analysis III

HES 505 Fall 2022: Session 21

Matt Williamson

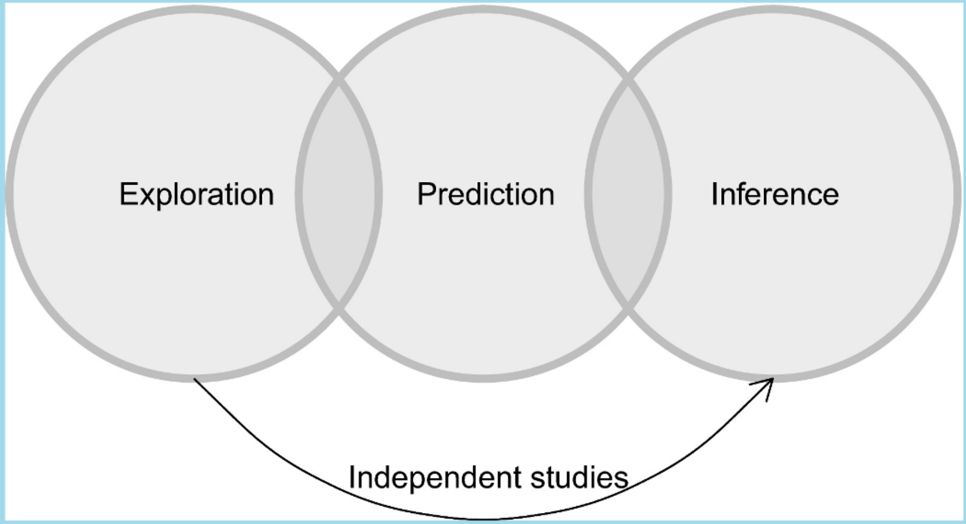
Objectives

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

- $\hat{\beta}$ is conditional on variables in the model **and** those not in the model

```
nsamp <- 1000
df <- data.frame(x1 = rnorm(nsamp,0,1),
                 x2 = rnorm(nsamp,0,1),
                 x3 = rnorm(nsamp,0,1))

linpred <- 1 + 2*df$x1 -0.18*df$x2 -3.5*df$x3
y <- rbinom(nsamp, 1, plogis(linpred))
df <- cbind(df, y)

mod1 <- glm(y~x1 +x2, data=df, family="binomial")
mod2 <- glm(y~x1 +x2 + x3, data=df, family="binomial")
```

Inference & Presence/Absence Data

```r
coef(mod1)
coef(mod2)
```

```r
prd1 <- predict(mod1, df, "response")
dif1 <- plogis(linpred) - prd1
prd2 <- predict(mod2, df, "response")
dif2 <- plogis(linpred) - prd2
```

> Inferring coefficient effects requires that your model fit the data well

Assessing Model Fit

Using Test Statistics

- R^2 for linear regression:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

$$SS_{res} = \sum_i (y_i - f_i)^2$$

$$SS_{tot} = \sum_i (y_i - y)^2$$

- Perfect prediction ($f_i = y_i$); $SS_{res} = 0$; and $R^2 = 1$
- Null prediction (Intercept only) ($f_i = y$); $SS_{res} = SS_{tot}$; and $R^2 = 0$
- No direct way of implementing for logistic regression

Pseudo- R^2

$$R_L^2 = \frac{D_{null} - D_{fitted}}{D_{null}}$$

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

Cohen's Likelihood Ratio

```
logistic.rich <- glm(y ~ MeanAnnTemp + PrecipWetQuarter + PrecipDryQuarter,
                     family=binomial(link="logit"),
                     data=pts.df[,2:8])

with(logistic.rich,
     null.deviance - deviance)/with(logistic.rich,
                                    null.deviance)
```

Pseudo- R^2

$$
\begin{aligned}
R^2_{CS} &= 1 - \left(\frac{L_0}{L_M}\right)^{(2/n)} \\
&= 1 - \exp^{2(\ln(L_0) - \ln(L_M))/n}
\end{aligned}
$$

- Cox and Snell R^2
- Likelihood (L), the probability of observing the sample given an assumed distribution
- Challenge: Maximum value is less than 1 and changes with n
- Correction by Nagelkerke so that maximum is 1

Cox and Snell R^2

```
logistic.null <- glm(y ~ 1,
                     family=binomial(link="logit"),
                     data=pts.df[,2:8])

1 - exp(2*(logLik(logistic.null)[1] - logLik(logistic.rich)[1])/nobs(logist
```

Using Test Statistics

- Based on the data used in the model (i.e., not prediction)
- Likelihood Ratio behaves most similarly to R^2
- 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 $\neq$ good)
- Theoretical and subsampling methods

Theoretical Assessment of Predictive Performance

Hirotugu Akaike of AIC

- 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)

$$\mathrm{AIC} = -2\ln(\hat{L}) + 2k$$

AIC Comparison

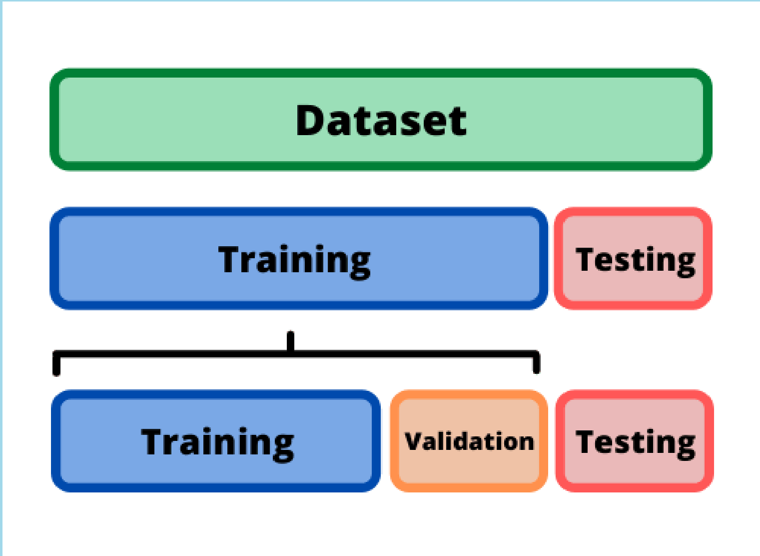
```
logistic.null <- glm(y ~ 1,
                     family=binomial(link="logit"),
                     data=pts.df[,2:8])

logistic.rich <- glm(y ~ MeanAnnTemp + PrecipWetQuarter + PrecipDryQuarter,
                     family=binomial(link="logit"),
                     data=pts.df[,2:8])

AIC(logistic.null, logistic.rich)
```

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

```
pts.df$y <- as.factor(ifelse(pts.df$y == 1, "Yes", "No"))
library(caret)
Train <- createDataPartition(pts.df$y, p=0.6, list=FALSE)

training <- pts.df[ Train, ]
testing <- pts.df[ -Train, ]
```

Misclassification

- Confusion matrices compare actual values to predictions

- True Positive (TN) - This is correctly classified as the class if 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

```
train.log <- glm(y ~ .,
                 family="binomial"
                 data=training[,2:
                 
predicted.log <- predict(train.log
                         newdata=t
                         type="res

pred <- as.factor(
  ifelse(predicted.log > 0.5,
                         "Yes",
                         "No"))
```

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

Confusion Matrices

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

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{FP} + \text{TN}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Depends upon threshold!!

Confusion Matrices in R

```
library(tree)
tree.model <- tree(y ~ . , trainin
predict.tree <- predict(tree.model
```

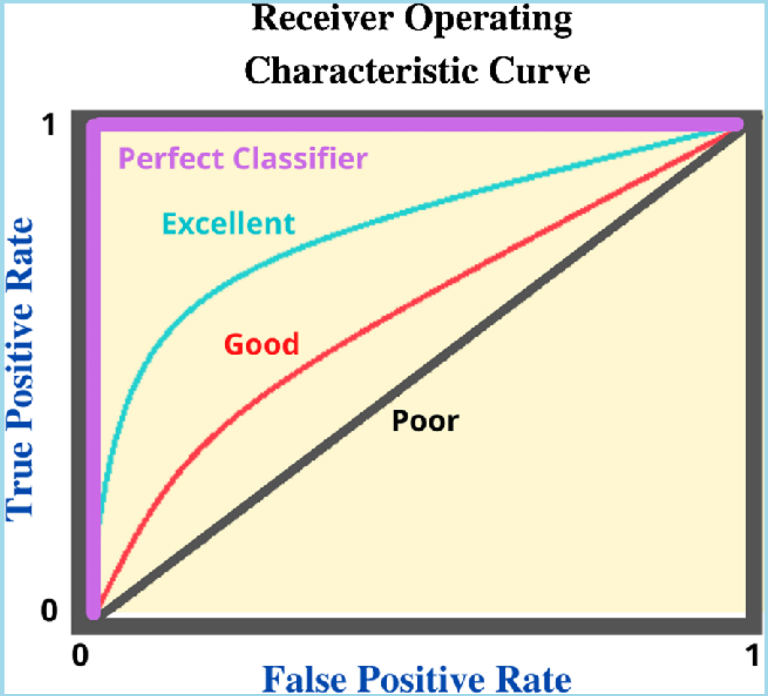
```
confusionMatrix(testing$y, predict.tree)
```

Confusion Matrices in R

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

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

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

Cross-validation

Spatial predictions…