PM592: Regression Analysis for Health Data Science
Lab 10 – Prediction Modeling
Data Needed: -

Outline

- Validating a Prediction Model

1. Model Validation

1.1. Your prediction model may have several variables, including polynomial and interaction terms.

1.2. One way to prevent overfitting is to split your entire sample into:

1.2.1. The **training** data set: where the model is created

1.2.2. The **testing** data set: where the model is tested

1.2.3. This will provide evidence for **external validity** – i.e., does your model generalize to different data sets?

1.3. What does this process look like?

1.3.1. Develop your prediction model on the training data set (70-90% of your full data set, depending on what the sample size allows

1.3.2. Use the final predictive model to calculate predicted probabilities of the final outcome for those in the testing data set.

1.3.3. Evaluate your classification indices in both data sets.

1.4. Make your data sets as "equal" as possible in terms of participant characteristics

1.4.1. You want the testing and training data sets to be representative of the same population.

1.4.2. You may use a **subsample** of individuals from the larger data set

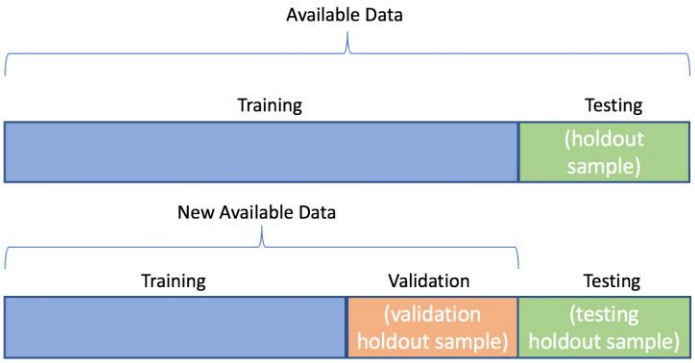
1.4.3. Alternately you may use an **independent sample** (e.g., if we are examining students from one school, we may test the model on a sample of individuals from a different school)

1.5. If your data set is large enough, you can make an additional split:

1.5.1. **Train** your model on a large training data set

1.5.2. **Validate** your model on a smaller validation set – refine the model to improve generalizability

1.5.3. **Test** your model on a smaller testing data set – this data set can be different from the original in order to see how the model generalizes to different conditions



1.6. Other approaches exist!

1.6.1. **Cross-Validation:** Split the data into several subsets. Each subset is used to train a different model which is then tested against the other subsets. The final model is an averaged version of the model from each subset.

1.6.2. The most common cross-validation approach is **k-folds validation**

Lab 10 Exercises

Objective(s):	Create a prediction model from start to finish, evaluate the model, provide predictive diagnostics
Datasets Required:	sos_dat.csv

In class we went through the model-building steps to predict suicide attempt in high school students.

Students who skip school days/classes are at increased risk of behavioral and academic problems. In this lab, we will use the same predictive variables as we did in-class, but with the variable "skip" as the outcome (did the student skip at least one class or day of school in the past year?).

1) Create a variable that indicates whether the observation belongs to the training or testing data set.

```
sos_dat$train <- sample(c(FALSE, TRUE), nrow(sos_dat), replace=TRUE, prob=c(0.2,0.8))

sos_train <- sos_dat[sos_dat$train,]
sos_test <- sos_dat[!sos_dat$train,]
```

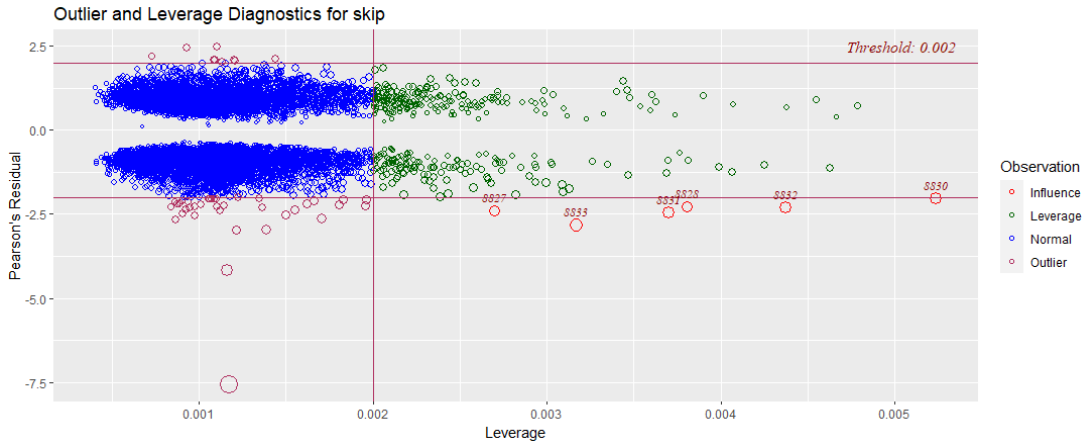
2) Using the training data set to develop the best prediction model for skipping school.
   a) What is the best model?
      - Grade was removed from the model, since it is correlated with age and no longer significant when added to the model with age

$$\hat{Y} = -6.3 + 0.65X_{bully} - 1.2\ln(X_{bullied}) - 0.07X_{odg} + 0.2X_{age} - 1.02X_{dens} - 0.31X_{recip}^{2} - 0.12X_{male} - 0.4\ln(X_{tatot}) + 2.31X_{tatot}^{\frac{1}{2}}$$

   b) Evaluate the goodness of fit and any influential points.

```
> ResourceSelection::hoslem.test(m2$y, fitted(m2), g=20)

        Hosmer and Lemeshow goodness of fit (GOF) test

data:  m2$y, fitted(m2)
X-squared = 50.071, df = 18, p-value = 7.364e-05
```

Hosmer-Lemeshow test is significant, indicating lack of good fit



Residual plot shows 6 influential points, but none of them are strong outliers

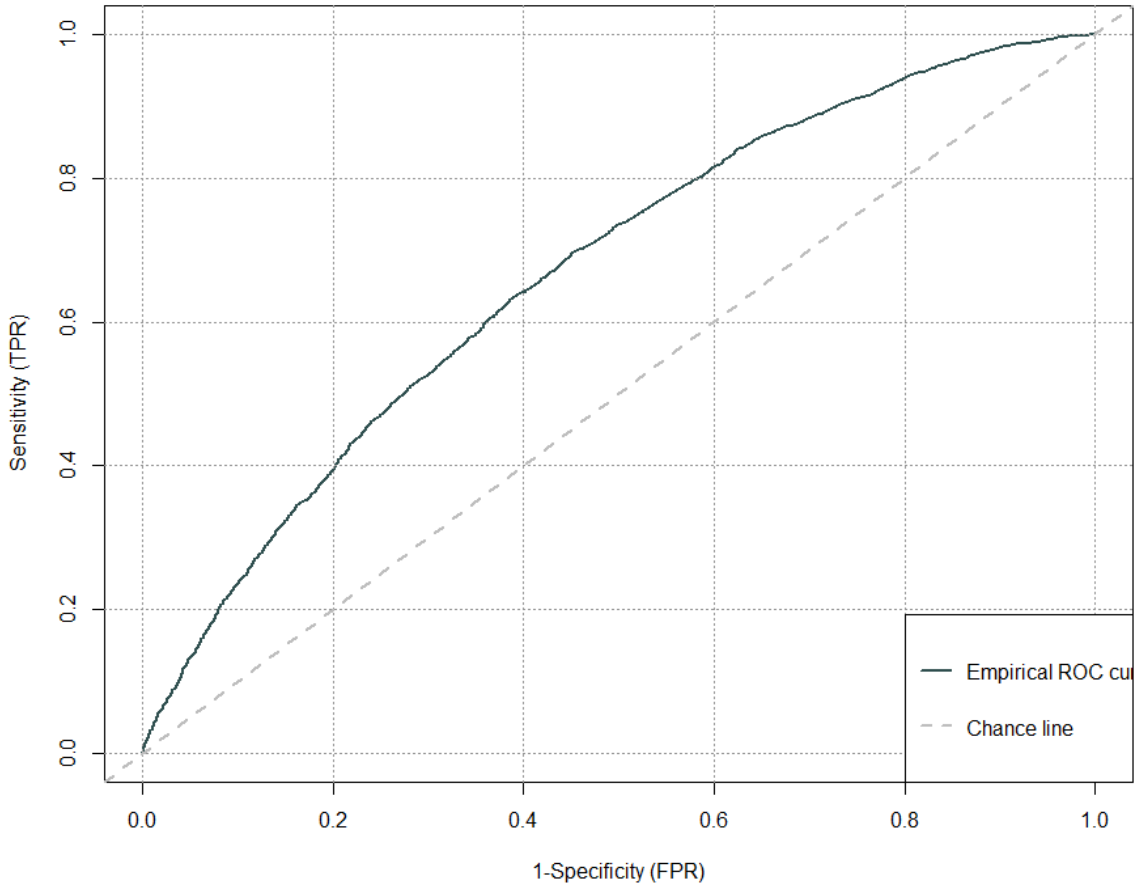
```
> DescTools::Conf(m2, pos = 1)

Confusion Matrix and Statistics

          Reference
Prediction    1    0
         1 3005 1786
         0 1558 2485

               Total n : 8'834
              Accuracy : 0.6215
                95% CI : (0.6113, 0.6315)
    No Information Rate : 0.5165
    P-Value [Acc > NIR] : < 2.2e-16

                 Kappa : 0.2408
 Mcnemar's Test P-Value : 8.66e-05

            Sensitivity : 0.6586
            Specificity : 0.5818
         Pos Pred Value : 0.6272
         Neg Pred Value : 0.6146
             Prevalence : 0.5165
         Detection Rate : 0.5423
   Detection Prevalence : 0.3402
      Balanced Accuracy : 0.6202
         F-val Accuracy : 0.6425
     Matthews Cor.-Coef : 0.2411

       'Positive' Class : 1
```

Accuracy: 0.62 (95% CI: (0.61, 0.63))

Sensitivity: 0.66

Specificity: 0.58

c) Present a figure of the ROC curve and the value of the AUC.

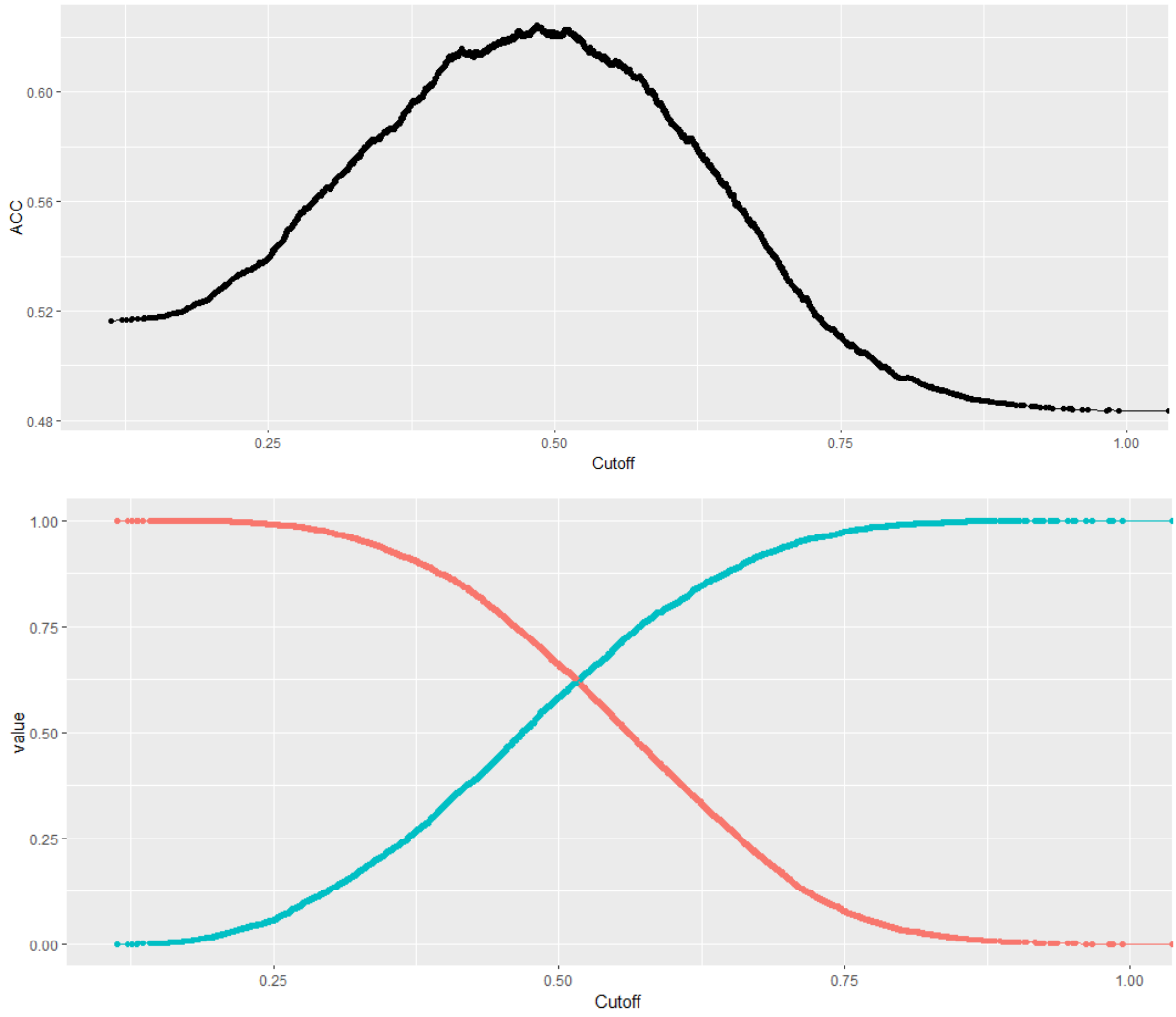



```
> plot(roc_empirical, YIndex = F)
> summary(roc_empirical)

 Method used: empirical
 Number of positive(s): 4563
 Number of negative(s): 4271
 Area under curve: 0.6683
> ciAUC(roc_empirical)

   estimated AUC : 0.668348267469352
   AUC estimation method : empirical

   CI of AUC
   confidence level = 95%
   lower = 0.657193707514334     upper = 0.67950282742437
>
```

AUC: 0.6683 (CI: (0.6572, 0.6796))

d) Determine the best classification cut point and the values of sensitivity and specificity for this cut

point.


ACC
0.60
0.56
0.52
0.48
0.25
0.50
0.75
1.00
Cutoff
value
1.00
0.75
0.50
0.25
0.00
0.25
0.50
0.75
1.00
Cutoff
metric
SENS
SPEC

```
> tibble(
+   Cutoff = roc$Cutoff,
+   SENS = roc$SENS,
+   SPEC = roc$SPEC,
+   SUM = SENS + SPEC
+ ) %>%
+   arrange(-SUM, -SENS, -SPEC) # 0.494 cutoff
# A tibble: 8,835 × 4
   Cutoff  SENS  SPEC   SUM
    <dbl> <dbl> <dbl> <dbl>
 1  0.511 0.636 0.609  1.24
 2  0.485 0.698 0.547  1.24
 3  0.512 0.631 0.614  1.24
 4  0.512 0.631 0.613  1.24
 5  0.485 0.697 0.548  1.24
 6  0.511 0.635 0.610  1.24
 7  0.485 0.697 0.548  1.24
 8  0.511 0.635 0.609  1.24
 9  0.511 0.636 0.609  1.24
10  0.485 0.697 0.547  1.24
# i 8,825 more rows
# i Use `print(n = ...)` to see more rows
```

I determine the optimal cutoff to be 0.511

3) When you are happy with your model, proceed to the testing data set:

   a) Apply the model from step (2) to your testing data to get predicted probabilities for each individual in the testing data set.

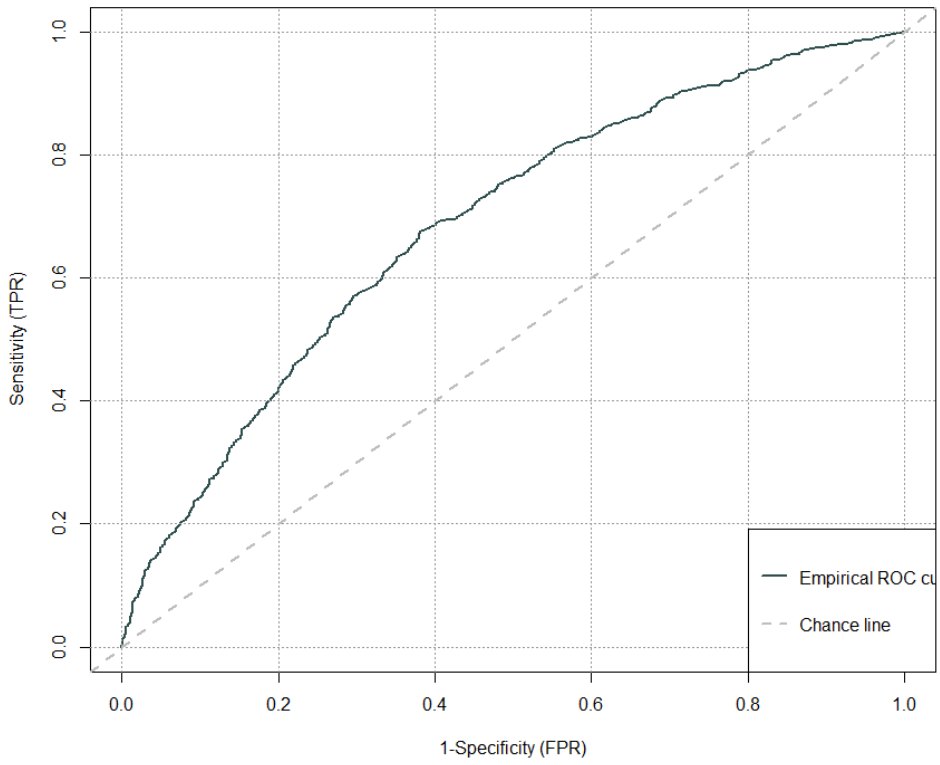
```
> m2_test.p <-
+   tibble(
+     pred_p = predict(m2, newdata = sos_test, type = "response"),
+     y = sos_test$skip
+   )
> head(m2_test.p)
# A tibble: 6 × 2
  pred_p     y
   <dbl> <dbl>
1  0.374     0
2  0.519     1
3  0.636     1
4  0.235     0
5  0.316     1
6  0.630     1
```

   b) Examine the model's discriminant ability by calculating the value of the AUC for the testing data set only.

```
> summary(test_roc_empirical)

 Method used: empirical
 Number of positive(s): 1141
 Number of negative(s): 1065
 Area under curve: 0.6855
> ciAUC(test_roc_empirical)

   estimated AUC : 0.68554311554398
   AUC estimation method : empirical

   CI of AUC
   confidence level = 95%
   lower = 0.663601145959164     upper = 0.707485085128795
```

AUC for testing set is 0.6855 (CI 0.6636, 0.7075))

c) Provide the ROC curve for the testing data set only.



d) Provide the values of sensitivity and specificity.

```
> library(caret)
> binary_outcome <- ifelse(m2_test.p$pred_p > 0.511, 1, 0)
> confusionMatrix(as.factor(binary_outcome), as.factor(m2_test.p$y))
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 668 392
         1 397 749

               Accuracy : 0.6423
                 95% CI : (0.6219, 0.6624)
    No Information Rate : 0.5172
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.2837

 Mcnemar's Test P-Value : 0.8868

            Sensitivity : 0.6272
            Specificity : 0.6564
         Pos Pred Value : 0.6302
         Neg Pred Value : 0.6536
             Prevalence : 0.4828
         Detection Rate : 0.3028
   Detection Prevalence : 0.4805
      Balanced Accuracy : 0.6418

       'Positive' Class : 0
```

Sensitivity: 0.6272

Specificity: 0.6564