Prediction for Beginners

Sean McGrath

June 25, 2025

Yale SCHOOL OF PUBLIC HEALTH
Big Data Summer Immersion

Outline

- Estimation versus Prediction
- Assessing Prediction Models
- Sample Splitting
- Cross-Validation

Estimation vs Prediction: Linear Regression

- Consider the simple linear regression model

$$Y = \beta_0 + \beta_1 x + \epsilon, \qquad \epsilon \sim \text{Normal}(0, \sigma^2)$$

- Focus often lies on estimating the β_j or the conditional mean $E(Y|x)$

- However, we are sometimes interested in predicting the outcome based on new data (x_{new}), i.e.,

$$Y_{new} = \beta_0 + \beta_1 x_{new} + \epsilon_{new}$$

- **Example:** The model will be deployed for prediction purposes and would like to measure its performance for predicting new observations

- **Prediction Goal:** Predict Y_{new} and characterize the behavior of our estimator

Estimation vs Prediction: Linear Regression

- Suppose we have new data (x_{new}). Let's consider predicting the outcome and characterizing the variance of our prediction

- **Point estimate:**

$$\hat{Y}_{new} = \hat{\beta}_0 + \hat{\beta}_1 x_{new}$$

  – i.e., same as $\widehat{\mathrm{E}}(Y_{new}|x_{new})$

- **Variance:**

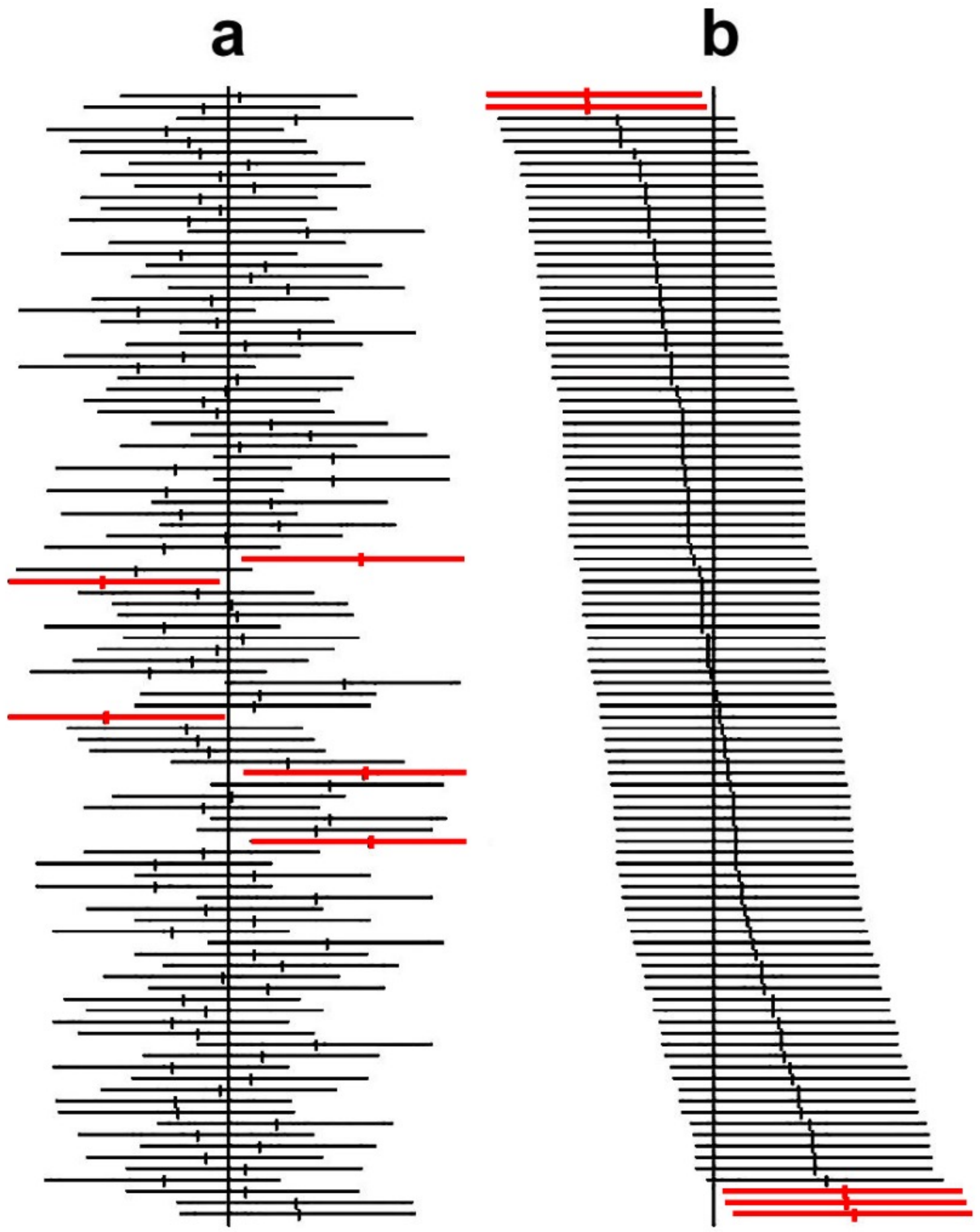
$$\begin{aligned}\text{Variance}\left(\hat{Y}_{new} + \epsilon_{new}\right) &= \text{Variance}\left(\hat{\beta}_0 + \hat{\beta}_1 x_{new} + \epsilon_{new}\right) \\ &= \text{Variance}\left(\hat{\beta}_0 + \hat{\beta}_1 x_{new}\right) + \text{Variance}(\epsilon_{new})\end{aligned}$$

  – Prediction involves accounting for the additional uncertainty due to ϵ_{new}

Prediction Interval

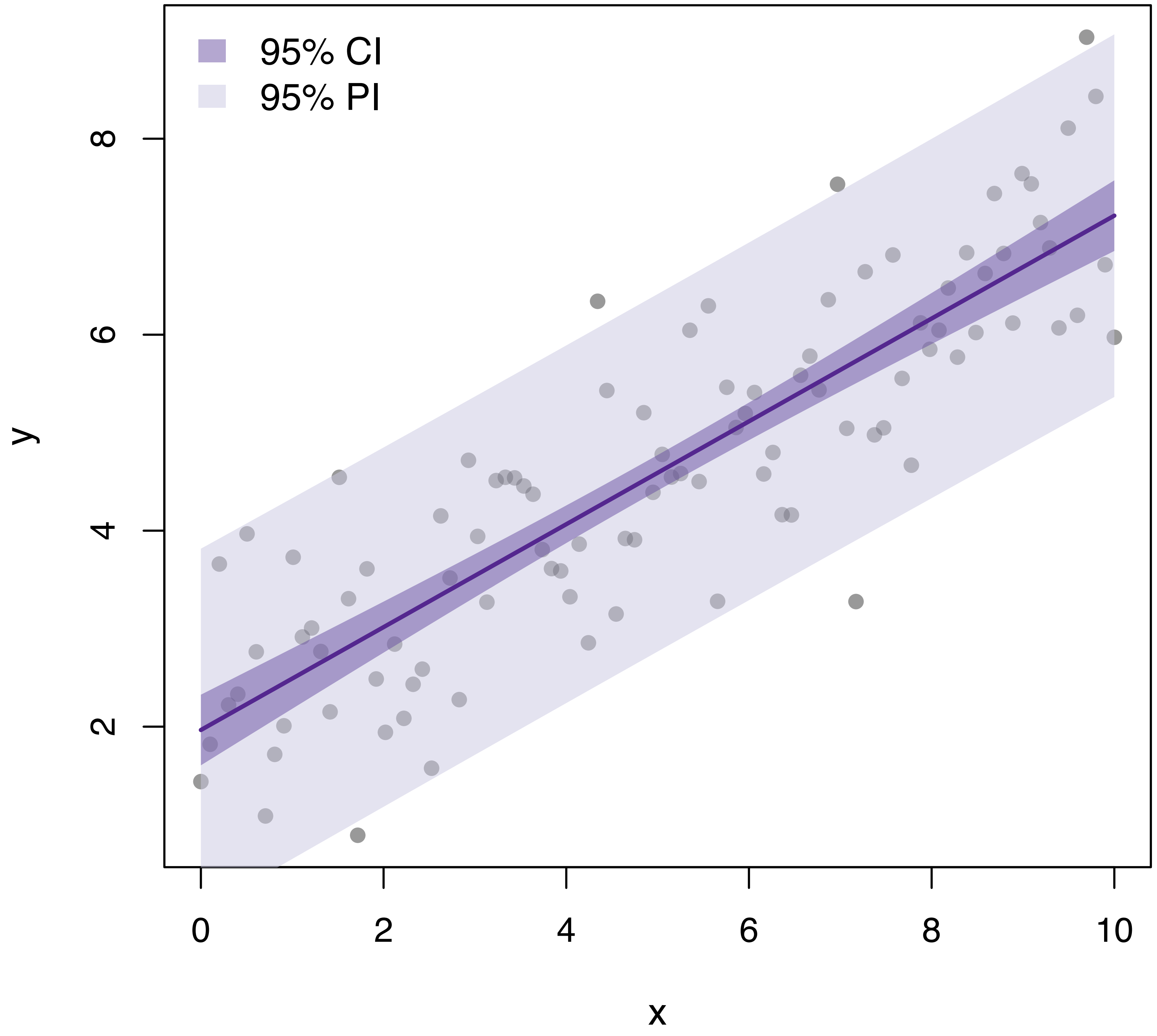
- **Prediction Interval (PI):** A $100(1-\alpha)\%$ prediction interval is an interval $[L, U]$ such that $P(Y_{new} \in [L, U]) = 100(1-\alpha)\%$

- **Interpretation:** If one were to repeatedly perform the experiment and construct PIs in this manner, $100\times(1-\alpha)\%$ of the intervals would contain Y_{new}



Christensen E. J Hepatol. 2007;46(5):947-54

Example: Prediction Interval (PI) vs Confidence Interval (CI)



Estimation vs Prediction: Logistic Regression

- **Model:** Consider the logistic regression model

$$\text{logit}(P(Y = 1|x)) = \beta_0 + \beta_1 x$$

- **Estimation Goal:** Estimate and characterize the behavior of our estimator the β_j or $P(Y = 1|x)$

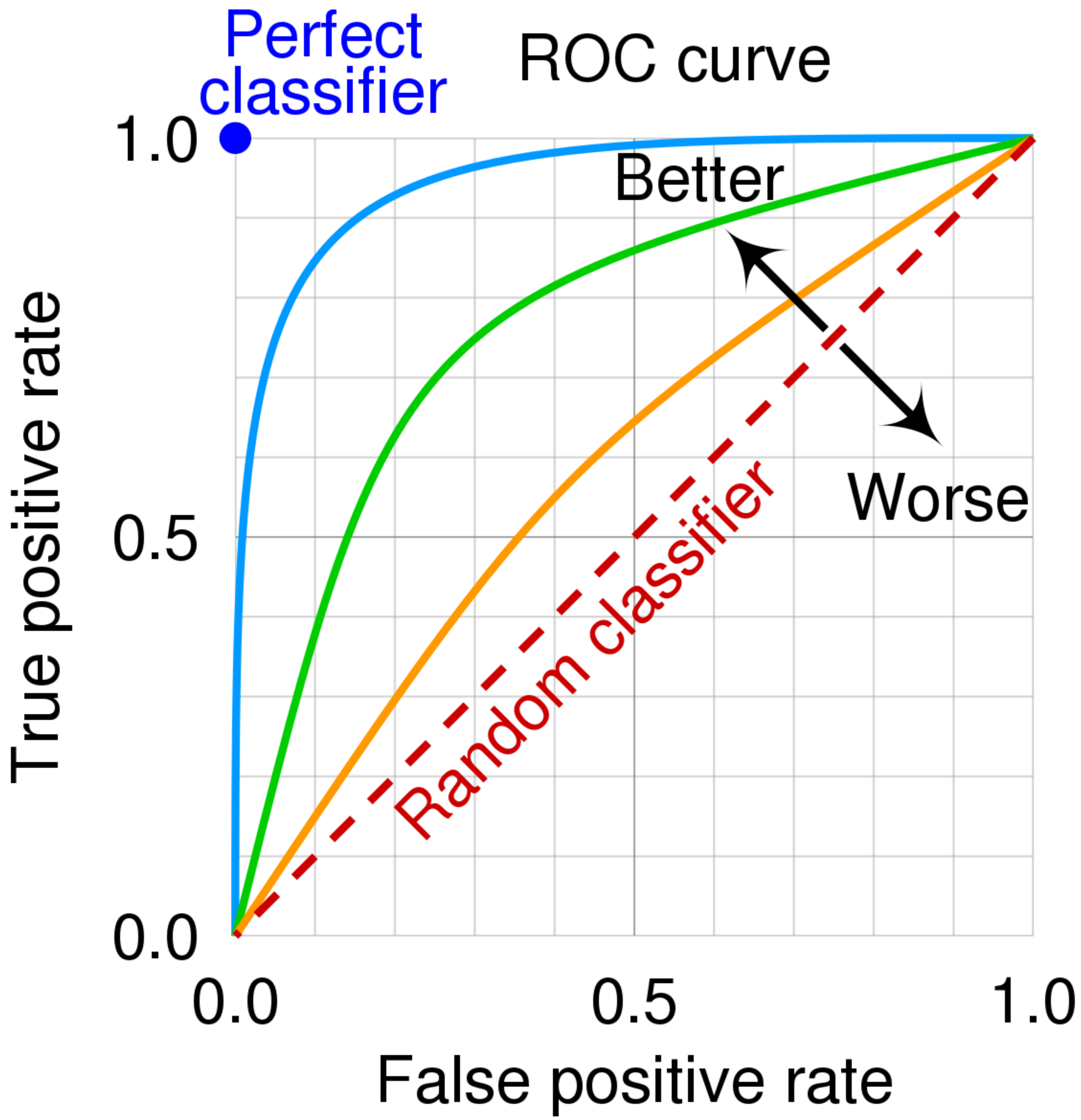
- **Prediction Goal:** Characterize the behavior of Y_{new}
  - Y_{new} follows a Bernoulli distribution with success probability $\text{logit}^{-1}(\beta_0 + \beta_1 x)$
  - i.e., Predictions will be 0 or 1

Measures of Predictive Performance: Confusion Matrix

		Predicted Class		
		Positive	**Negative**	
Actual Class	**Positive**	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN + FP)}$
		Precision $\frac{TP}{(TP + FP)}$	**Negative Predictive Value** $\frac{TN}{(TN + FN)}$	**Accuracy** $\frac{TP + TN}{(TP + TN + FP + FN)}$

https://encord.com/glossary/confusion-matrix/

Measures of Predictive Performance: ROC Curves



https://medium.com/@ilyurek/roc-curve-and-auc-evaluating-model-performance-c2178008b02

- **ROC Curve Axis Labels:**
  - **True positive rate:** Sensitivity
  - **False positive rate:** 1 – Specificity

- **Area Under Curve (AUC):** Measure of discriminative ability of a prediction model
  - AUC = 1: Perfect model
  - AUC = 0.5: Random guessing

- **Interpretation of AUC:** Probability that the model assigns a higher probability to an individual that has the outcome compared to an individual that does not have the outcome

Exercise: Part 1

Let's measure the predictive performance of logistic regression models and see what challenges arise.

1. Download the dataset data-prediction-exercise.csv. It contains 1000 rows with the following variables:
   - Binary Outcome: Y
   - Continuous Predictors: X1, …, X15
2. Fit a logistic regression model for Y with each of the 15 predictors
3. Predict the outcome for all 1000 rows and construct the confusion matrix (i.e., no sample splitting / cross-fitting for now). Compute your favorite predictive performance measures.
4. (Time Permitting) Construct an ROC curve and compute the AUC
   - Suggestion: See the pROC R package

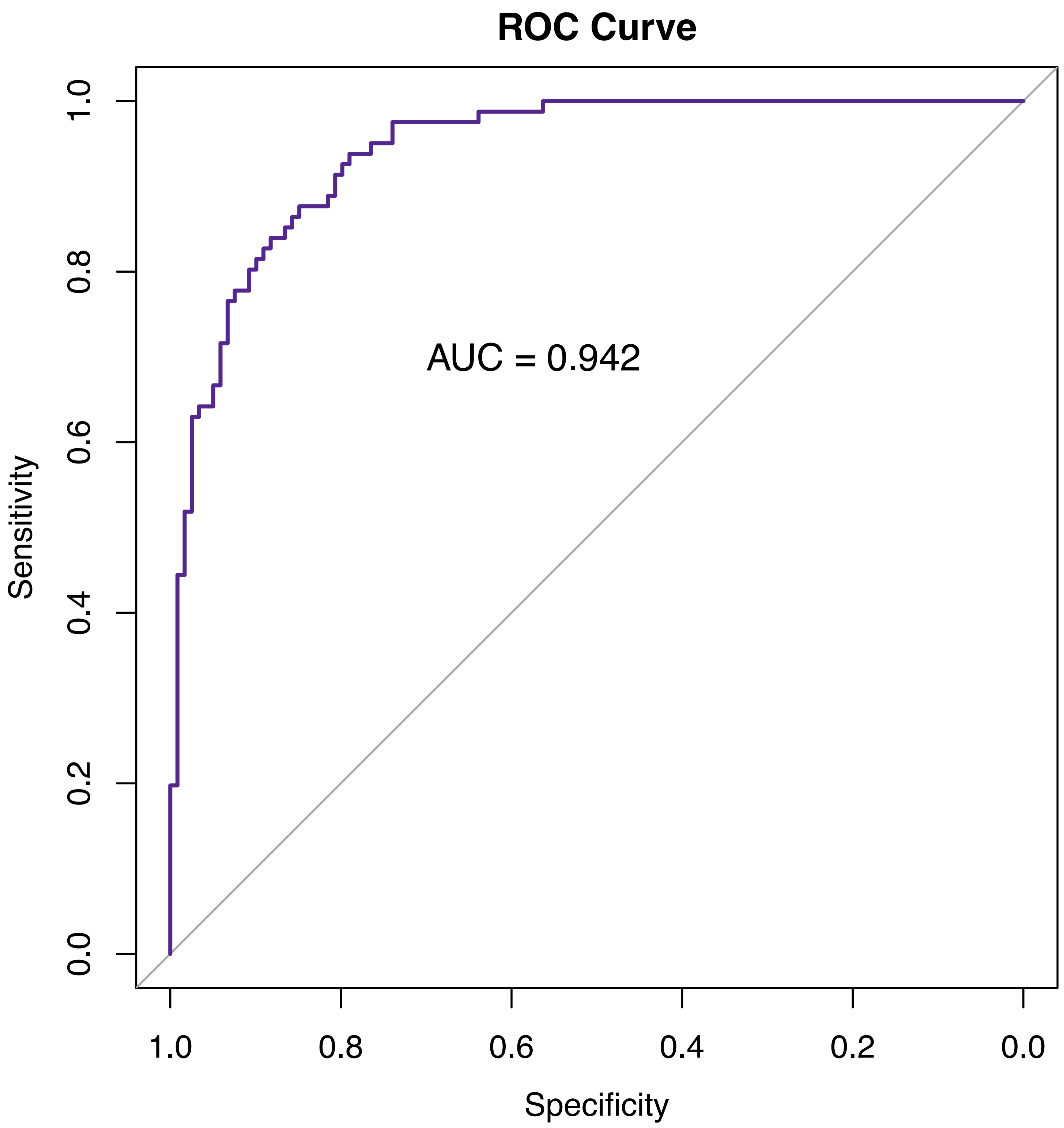
If you finish early: Repeat Steps 2-4 with your favorite machine learning algorithm. Does it perform better?

Solution: Part 1

```
# Step 1: Read data
dat <- read.csv('data-prediction-exercise.csv')

# Step 2: Fit logistic regression model
fit <- glm(Y ~ ., family = binomial, data = dat)

# Step 3: Form confusion matrix
predicted_probability <- predict(fit, type = "response")
predicted_class <- ifelse(predicted_probability >= 0.5, 1, 0)
confusion_matrix <- table(Actual = dat$Y, Predicted = predicted_class)

print(confusion_matrix)
      Predicted
Actual   0   1
     0 106  13
     1  15  66
```

Solution: Part 1 (Continued)

```
# Step 4: ROC curve analysis
library('pROC')
roc_obj <- roc(dat$Y, predicted_probability)
plot(roc_obj, col = "#54278f",
     lwd = 2, main = "ROC Curve")
text(x = 0.7, y = 0.7,
     labels = paste0("AUC = ", round(auc(roc_obj), 3)),
     adj = 0, cex = 1.2)
```



Our Estimates of Predictive Performance Are Too Optimistic

- **We estimated:**
  - Sensitivity: 0.89
  - Specificity: 0.81
  - AUC: 0.94

- **The truth is actually:**
  - Sensitivity: 0.80
  - Specificity: 0.69
  - AUC: 0.83

- **Question:** Our estimates are overly optimistic by about 10 percentage points! Why?
  - We used the same test to fit the model and evaluate it

Exercise: Part 2

5. Now, consider that we have access to a second data set (which follows the same data generating mechanism). Download this dataset, “data-prediction-exercise-external.csv”.

6. Using the **model fit in step 2**, predict the outcome **in the dataset in step 5.** Construct the confusion matrix by comparing the predicted outcome to the actual outcome. Compute your favorite predictive performance measures again. How have they changed?

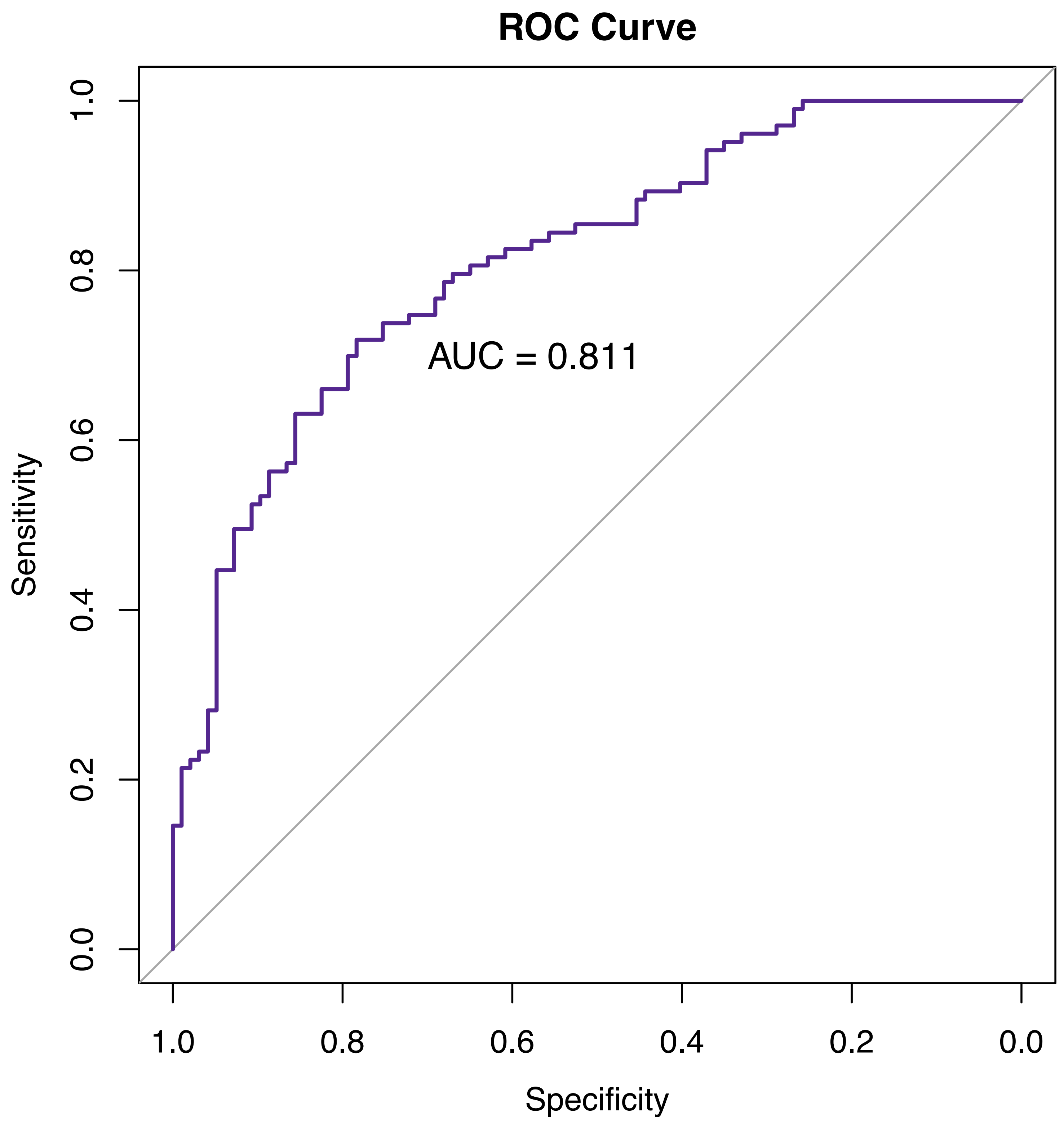
7. (Time Permitting) Construct an ROC curve and compute the AUC in the same manner as the previous step (i.e., using the model fit in step 2 and obtaining predictions in the dataset in step 5). How have these changed?

If you finish early: Repeat these steps with your favorite machine learning algorithm. Does using an external data set make a bigger impact for such a method?

Solution: Part 2

```
# Step 5: Read data
dat_external <- read.csv('data-prediction-exercise-external.csv')

# Step 6: Form confusion matrix
predicted_probability <- predict(fit, type = "response", newdata = dat_external)
predicted_class <- ifelse(predicted_probability >= 0.5, 1, 0)
confusion_matrix <- table(Actual = dat_external$Y, Predicted = predicted_class)

print(confusion_matrix)
      Predicted
Actual  0  1
     0 76 21
     1 31 72

# Step 7: ROC curve analysis
roc_obj <- roc(dat_external$Y, predicted_probability)
plot(roc_obj, col = "#54278f", lwd = 2, main = "ROC Curve")
text(x = 0.7, y = 0.7,
     labels = paste0("AUC = ", round(auc(roc_obj), 3)),
     adj = 0, cex = 1.2)
```

Solution: Part 2 (Continued)



Our Estimates Are Now Reasonable

- We originally estimated:
  - **Sensitivity:** 0.89
  - **Specificity:** 0.81
  - **AUC:** 0.94

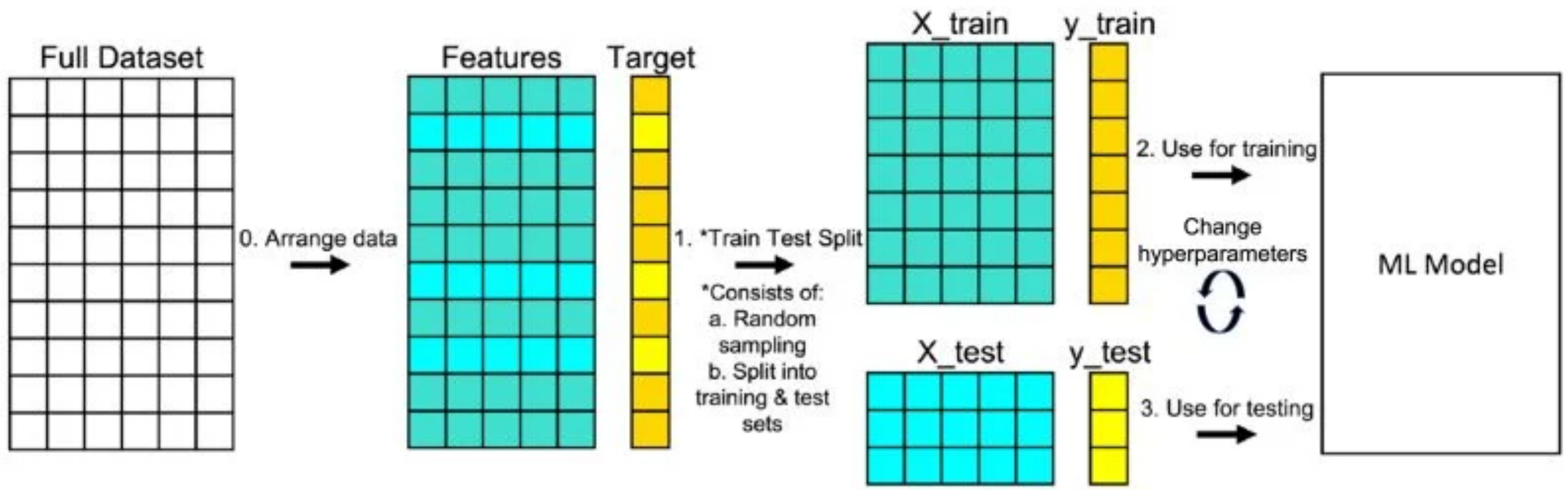
- The truth is actually:
  - **Sensitivity:** 0.80
  - **Specificity:** 0.69
  - **AUC:** 0.83

- When using the second dataset:
  - **Sensitivity:** 0.78
  - **Specificity:** 0.70
  - **AUC:** 0.81

Sample Splitting

- In practice, we usually don't have a second dataset available like this

- However, we can create one by performing **sample splitting**
  - Randomly divide the rows into two data sets
    - **Training dataset:** Fit the model
    - **Testing dataset:** Evaluate the performance of the model



Exercise: Part 3

8. Let's return to the case where we only have a single dataset (i.e., data-prediction-exercise.csv). Form a training dataset by taking a random sample of 70% of rows of the data. Form a testing dataset by the remaining rows.

9. Fit the logistic regression model in the training data. Compute the predictive performance measures in the test data.

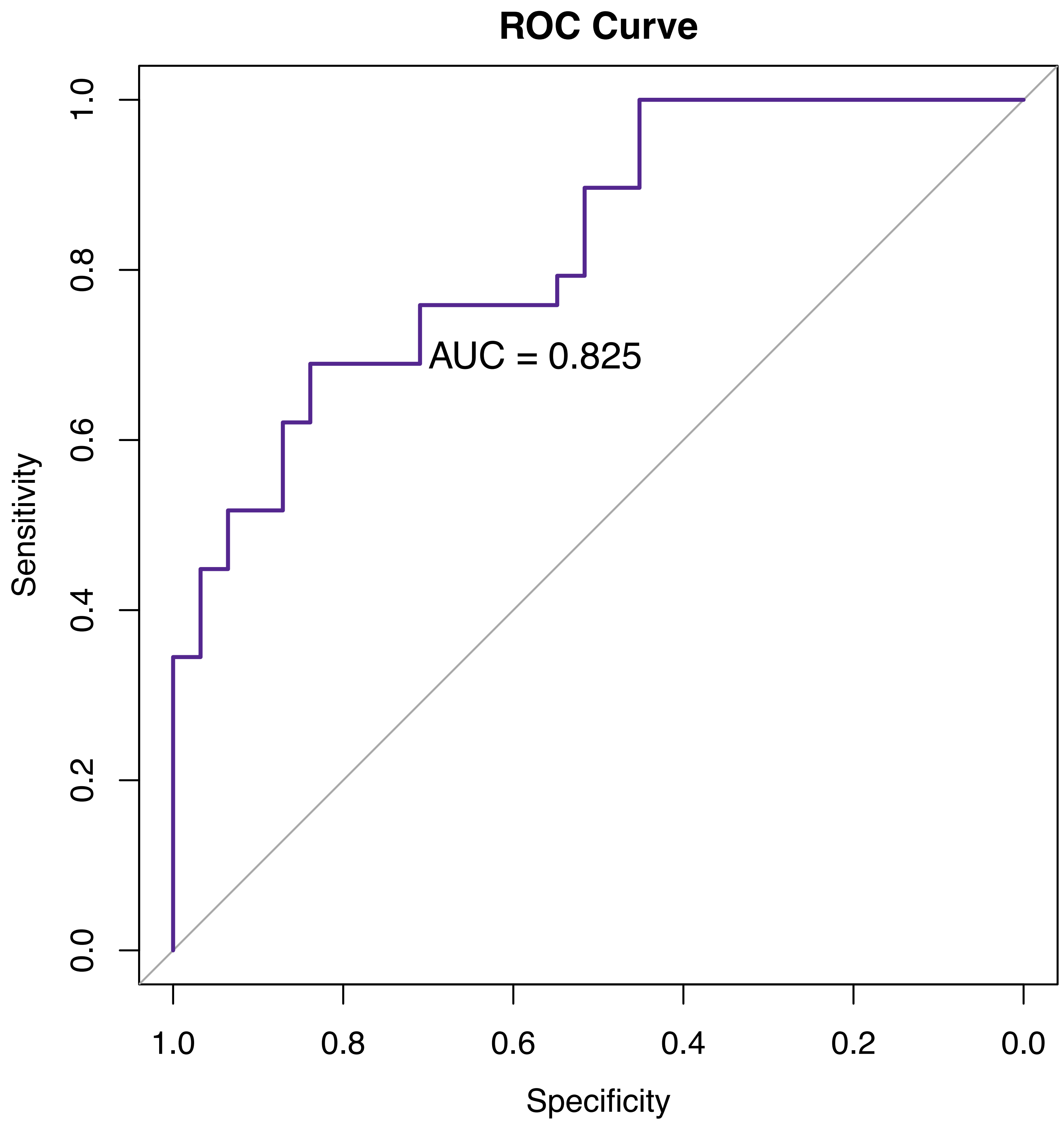
If you finish early: Repeat these steps with your favorite machine learning algorithm. Does using sample splitting have a bigger impact for such a method?

Solution: Part 3

```
# Step 8: Form test and train datasets
set.seed(1234)
n <- nrow(dat)
train_ind <- sample(1:n, size = round(n * 0.70))
dat_train <- dat[train_ind, ]
dat_test <- dat[-train_ind, ]

# Step 9: Fit model on training data and evaluate model on testing data
fit <- glm(Y ~ ., family = binomial, data = dat_train)
predicted_probability <- predict(fit, type = "response", newdata = dat_test)
predicted_class <- ifelse(predicted_probability >= 0.5, 1, 0)
confusion_matrix <- table(Actual = dat_test$Y, Predicted = predicted_class)
print(confusion_matrix)
      Predicted
Actual  0  1
     0 25  6
     1  9 20
```

Solution: Part 3 (Continued)

```
roc_obj <- roc(dat_test$Y, predicted_probability)
plot(roc_obj, col = "#54278f",
     lwd = 2, main = "ROC Curve")
text(x = 0.7, y = 0.7,
     labels = paste0("AUC = ", round(auc(roc_obj), 3)),
     adj = 0, cex = 1.2)
```



Our Estimates Are Reasonable

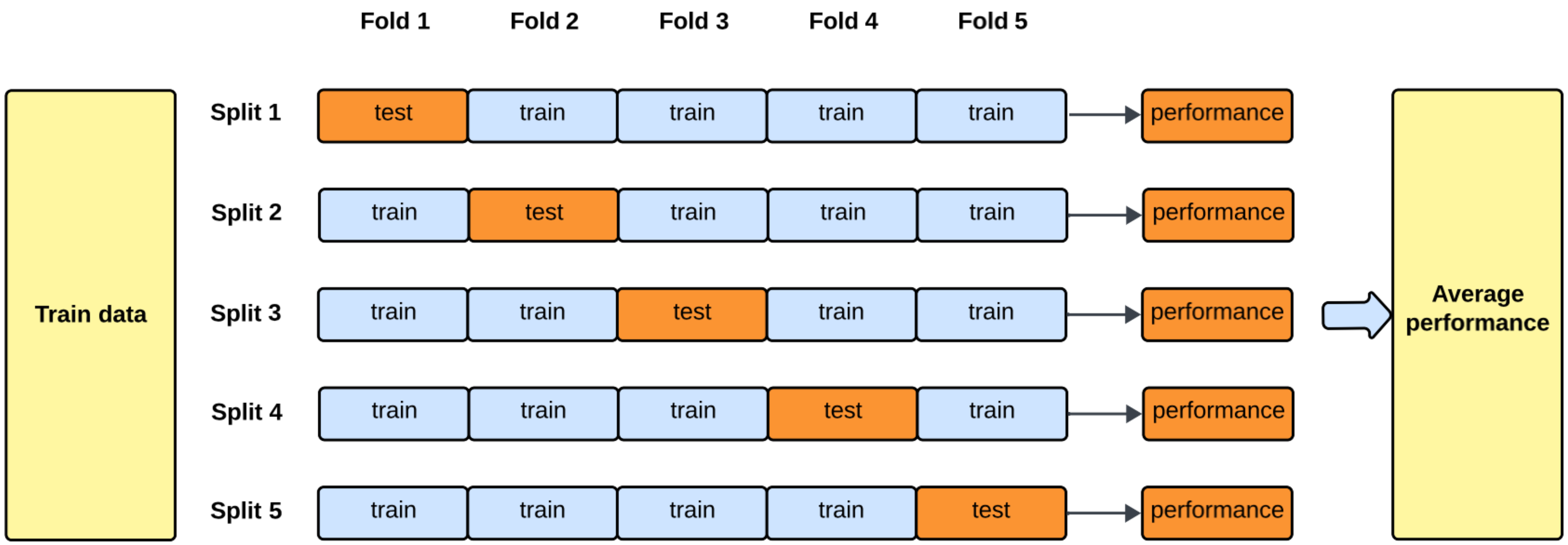
- We originally estimated:
  - **Sensitivity:** 0.89
  - **Specificity:** 0.81
  - **AUC:** 0.94
- The truth is actually:
  - **Sensitivity:** 0.80
  - **Specificity:** 0.69
  - **AUC:** 0.83
- When using an external dataset:
  - **Sensitivity:** 0.78
  - **Specificity:** 0.70
  - **AUC:** 0.81
- When using sample splitting:
  - **Sensitivity:** 0.81
  - **Specificity:** 0.69
  - **AUC:** 0.83

Limitation of Sample Splitting

- **Limitation:** Our test data set only includes 30% of our observations
  - Only 60 observations were used to assess the predictive performance of the model
  - May get highly variable estimates of the predictive performance

- I got very lucky in my illustration
  - If we re-run the code with a different seed, results widely differ
  - E.g. Using a random number seed of 1, we get
    - **Sensitivity:** 0.88 (truth is 0.80)
    - **Specificity:** 0.54 (truth is 0.60)
    - **AUC: 0.89** (truth is 0.83)

A Refinement: Cross-Validation



https://towardsdatascience.com/how-to-cross-validation-with-time-series-data-9802a06272c6/

Exercise: Part 4 (not covered in lecture)

10. Perform 5-fold cross-validation to assess your favorite performance measures of the logistic regression model. Are your results closer to the truth now?

Solution: Part 4

```r
# Creating training/testing datasets
set.seed(1234)
indices <- sample(1:n, size = n)

test_ind1 <- indices[1:(n/5)]
dat_train1 <- dat[-test_ind1, ]
dat_test1 <- dat[test_ind1, ]

test_ind2 <- indices[(n/5+1):(n*2/5)]
dat_train2 <- dat[-test_ind2, ]
dat_test2 <- dat[test_ind2, ]

test_ind3 <- indices[(n*2/5+1):(n*3/5)]
dat_train3 <- dat[-test_ind3, ]
dat_test3 <- dat[test_ind3, ]

test_ind4 <- indices[(n*3/5+1):(n*4/5)]
dat_train4 <- dat[-test_ind4, ]
dat_test4 <- dat[test_ind4, ]

test_ind5 <- indices[(n*4/5+1):n]
dat_train5 <- dat[-test_ind5, ]
dat_test5 <- dat[test_ind5, ]
```

Solution: Part 4 (Continued)

```
# Function for computing sensitivity
get_sensitivity <- function(dat_train, dat_test){
  fit <- glm(Y ~ ., family = binomial, data = dat_train)
  predicted_probability <- predict(fit, type = "response", newdata = dat_test)
  predicted_class <- ifelse(predicted_probability >= 0.5, 1, 0)
  confusion_matrix <- table(Actual = dat_test$Y, Predicted = predicted_class)
  return(confusion_matrix[1, 1] / (confusion_matrix[1, 1] + confusion_matrix[1, 2]))
}

# Computing sensitivity in each fold
sensitivity1 <- get_sensitivity(dat_train = dat_train1, dat_test = dat_test1)
sensitivity2 <- get_sensitivity(dat_train = dat_train2, dat_test = dat_test2)
sensitivity3 <- get_sensitivity(dat_train = dat_train3, dat_test = dat_test3)
sensitivity4 <- get_sensitivity(dat_train = dat_train4, dat_test = dat_test4)
sensitivity5 <- get_sensitivity(dat_train = dat_train5, dat_test = dat_test5)

# Taking average sensitivity across folds
mean(c(sensitivity1, sensitivity2, sensitivity3, sensitivity4, sensitivity5))
0.8470123
```

Cross-Validation in Other Contexts

- Sample splitting and cross-validation are powerful tools used in contexts beyond assessing predictive performance of models
  - **Key feature:** Create independencies when estimating different objects

- Active areas of research involving sample splitting / cross-validation:
  - Model selection and model averaging
  - Hyperparameter tuning
  - Post selection inference
  - Double/debiased machine learning for causal inference