Homework 01: Model Building and Model Selection/Fitting

MACS 30100: Perspectives on Computational Modeling
University of Chicago

```r
library(tidyverse)
library(broom)
library(rcfss)

options(digits = 3)
set.seed(1234)
theme_set(theme_minimal())
```

Overview

Due before class January 14th.

Fork the `hw01` repository

Go here to fork the repo for homework 01.

Building models (5 points)

For each of the following prompts, produce 800-1200 word written responses. Only PDF or Markdown (`.md`) submissions will be graded. Using LaTeXis a great option - see the recommended materials to get started.

Deviant aggressive behavior

Consider four well-known and widely believed theories of socially deviant aggressive behavior (e.g. criminal behavior, revolutionary behavior, rude behavior):

- **Theory I**: Deviant aggressive behavior is learned from experience. Individuals in a society learn to do those things for which they receive rewards and to avoid those things for which they receive punishment.
- **Theory II**: Deviant aggressive behavior is a symbolic expression of hostility toward personal authority figures. When an individual is frustrated in his personal life, he becomes angry toward parents, bosses, or public officials. He will express this anger by deviant aggressive behavior.
- **Theory III**: Deviant aggressive behavior is the rational action of oppressed individuals. Social rules systematically discriminate among people. People who are most hurt by the rules are least likely to profit from conforming to them and thus do conform less.
- **Theory IV**: Deviant aggressive behavior is a social role. Individuals are socialized into the role through contact with a deviant subculture.

Answer the following questions:

1. What social policy would be appropriate to reduce deviant aggressive behavior if Theory I were correct? Theory II? Theory III? Theory IV?

2. During the past ten years, American society has been running a series of "experiments" with deviant aggressive behavior. Take any one of these experiments (e.g. #MeToo, mass shootings, political rhetoric) and discuss what we have learned about the four theories from this series of experiments.

Waiting until the last minute

People often do things at the last minute (students turning in papers, professors grading exams, and so on).

a. Ask yourself **why** the observation might be true and write down your explanations.
b. Generalize the explanatory model – that is, induce the most general, abstract model you can produce that still has the original observation as a consequence.
c. Induce an alternative model that also has the original observation as a consequence.
d. For each of the two general models produced in (b) and (c), derive two interesting predictions (four predictions in total). Be sure the logical connection between your model and your predictions is explicitly stated and that any assumed facts concerning the world are made explicit.

Selecting and fitting a model (5 points)

1. For each part, indicate whether we would generally expect the performance of a flexible statistical learning method to be better or worse than an inflexible method. Justify your answer.

   a. The sample size n is extremely large, and the number of predictors p is small.

   Flexible learning method would perform **better** because sample size is large enough to fit more parameters and small number of predictors limits model variance.

   b. The number of predictors p is extremely large, and the number of observations n is small.

   Flexible learning method would perform **worse** because it would be more likely to overfit.

   c. The relationship between the predictors and response is highly non-linear.

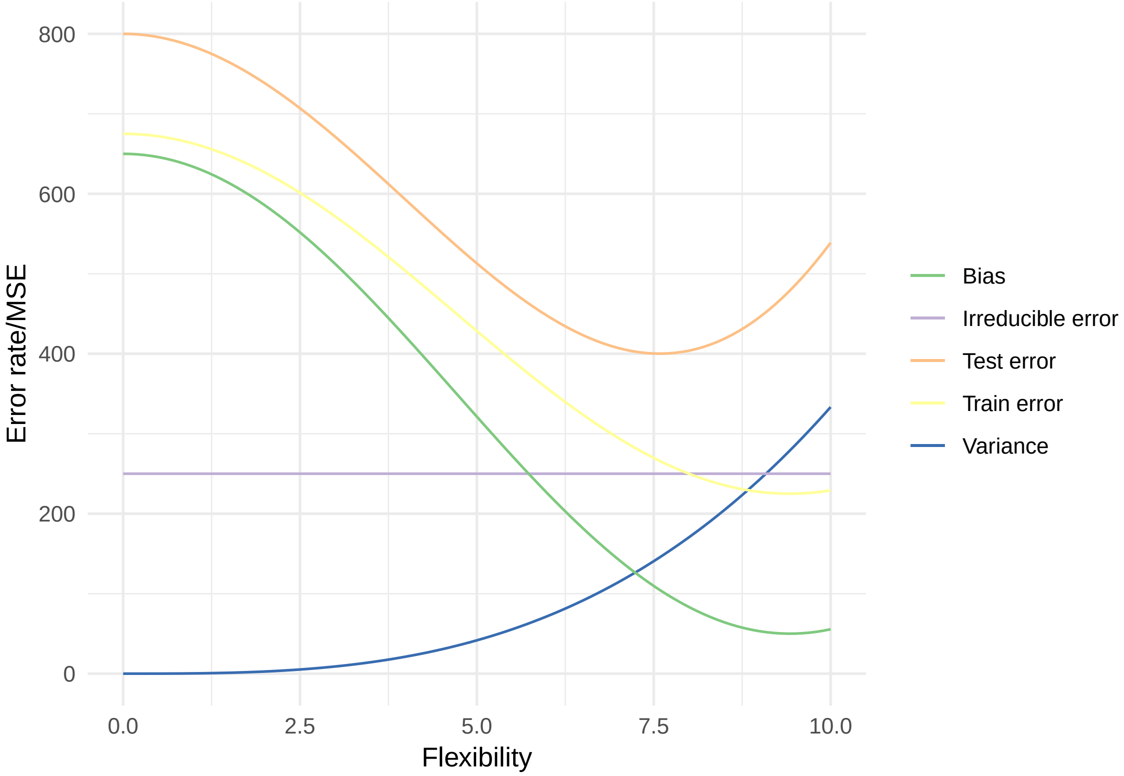
   Flexible learning method would perform **better** because it is less restrictive on the shape of fit.

   d. The variance of the error terms $\sigma^2 = \text{Var}(\epsilon)$ is extremely high. Flexible learning method would perform **worse** because it would be more likely to overfit

2. Bias-variance decomposition

   a. Generate a graph of typical (squared) bias, variance, training error, test error, and Bayes (irreducible) error curves, on a single plot, as we go from less flexible statistical learning methods towards more flexible approaches. The x-axis should represent the amount of flexibility in the method, and the y-axis should represent the values for each curve. There should be five curves. Make sure to label each one.

```r
tibble(x = c(0, 10),
       y = c(0, 800)) %>%
  ggplot(aes(x, y)) +
  stat_function(
    aes(color = "Test error"),
    fun = function(x)
      300 * cos(x / 3) + 500 + x ^ 3 / 3
  ) +
  stat_function(
    aes(color = "Variance"),
    fun = function(x)
      x ^ 3 / 3
  ) +
  stat_function(
    aes(color = "Irreducible error"),
    fun = function(x)
      0 * x + 250
  ) +
  stat_function(
    aes(color = "Bias"),
    fun = function(x)
      300 * cos(x / 3) + 350
  ) +
  stat_function(
    aes(color = "Train error"),
    fun = function(x)
      225 * cos(x / 3) + 450
  ) +
  scale_color_brewer(type = "qual") +
  labs(x = "Flexibility",
       y = "Error rate/MSE",
       color = NULL)
```



b. Explain why each of the five curves has the shape displayed in part (a).

- Bias will decrease with higher flexibility because there are fewer assumptions made about the shape of the fit
- Variance will increase with higher flexibility because changing data points will have more effect on the parameter estimates
- Training error will always decrease with more model flexibility because an overfit model will

produce lower error/MSE on the training data
- Test error will have a U-shaped curve because it reflects the interaction between variance and bias
- Irreducible error is the same regardless of model fit

3. For classification problems, the test error rate is minimized by a simple classifier that assigns each observation to the most likely class given its predictor values:

$$\Pr(Y = j | X = x_0)$$

where x_0 is the test observation and each possible class is represented by J. This is a **conditional probability** that $Y = j$, given the observed predictor vector x_0. This classifier is known as the **Bayes classifier**. If the response variable is binary (i.e. two classes), the Bayes classifier corresponds to predicting class one if $\Pr(Y = 1 | X = x_0) > 0.5$, and class two otherwise.

Figure 2.13 in [ISL] illustrates a simulated example defining the decision boundary for a hypothetical data set. Produce a graph illustrating this concept. Specifically, implement the following elements in your program:

   a. Set your random number generator seed.
   b. Simulate a dataset of $N = 200$ with X_1, X_2 where X_1, X_2 are random uniform variables between $[-1, 1]$.
   c. Calculate $Y = X_1 + X_1^2 + X_2 + X_2^2 + \epsilon$, where $\epsilon \sim N(\mu = 0, \sigma^2 = 0.25)$.
   d. Y is defined in terms of the log-odds of success on the domain $[-\infty, +\infty]$. Calculate the probability of success bounded between $[0, 1]$.
   e. Plot each of the data points on a graph and use color to indicate if the observation was a success or a failure.
   f. Overlay the plot with Bayes decision boundary, calculated using X_1, X_2.
   g. Give your plot a meaningful title and axis labels.
   h. The colored background grid is optional.

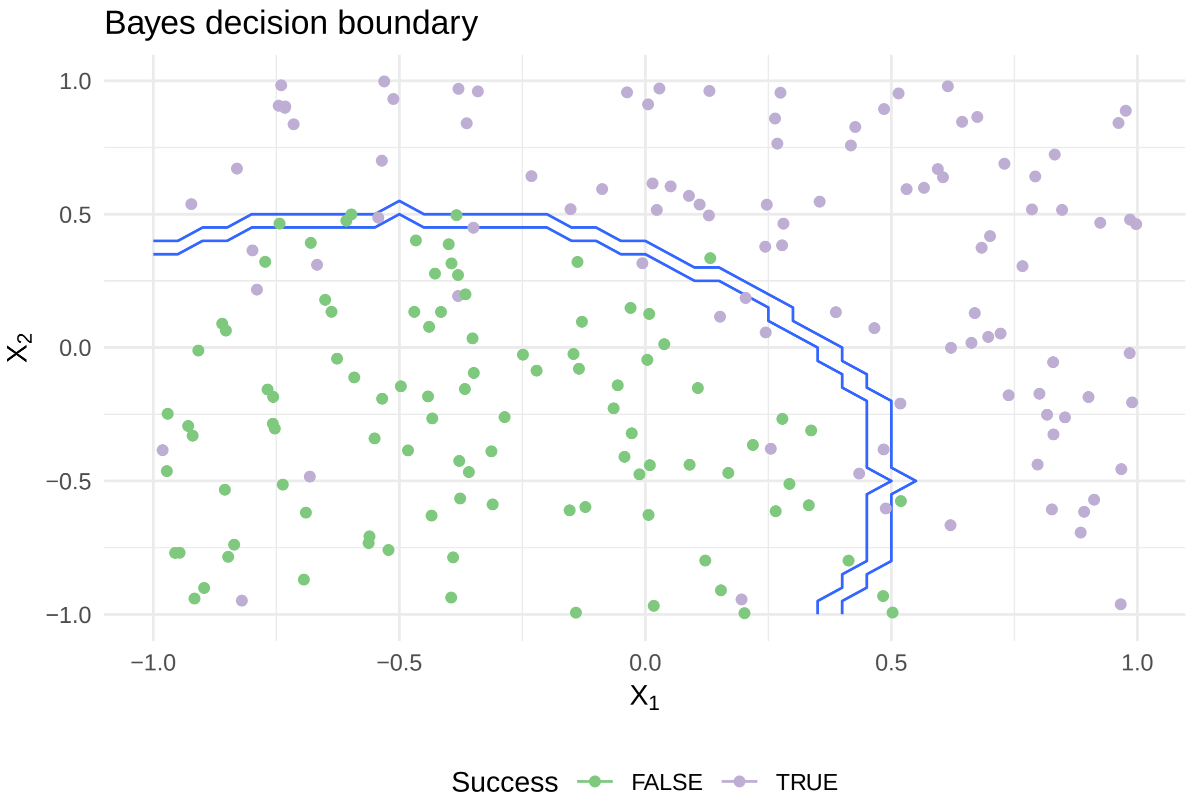
```r
# function to calculate Bayes decision rule
bayes_rule <- function(x1, x2) {
  x1 + x1^2 + x2 + x2^2
}

# generate data for grid background
bayes_grid <- expand.grid(x1 = seq(-1, 1, by = .05),
            x2 = seq(-1, 1, by = .05)) %>%
  as_tibble %>%
  mutate(logodds = bayes_rule(x1, x2),
         y = logodds > .5,
         prob = logit2prob(logodds))

## provide proper color coding
bayes_bound <- bind_rows(
  bayes_grid %>%
    mutate(cls = TRUE,
           prob_cls = ifelse(y == cls, 1, 0)),
  bayes_grid %>%
    mutate(cls = FALSE,
           prob_cls = ifelse(y == cls, 1, 0))
)

# generate simulated data
sim_bayes <- tibble(
  x1 = runif(200, -1, 1),
  x2 = runif(200, -1, 1),
  logodds = bayes_rule(x1, x2) + rnorm(200, 0, .5),
  y = logodds > .5,
  y_actual = bayes_rule(x1, x2) > .5
)
sim_bayes_err <- mean(sim_bayes$y != sim_bayes$y_actual)

ggplot(bayes_bound, aes(x1, x2, color = y)) +
  # geom_point(size = .5, alpha = .5) +
  geom_contour(aes(z = prob_cls, group = cls), bins = 1) +
  geom_point(data = sim_bayes) +
  scale_color_brewer(type = "qual") +
  labs(title = "Bayes decision boundary",
       x = expression(X[1]),
       y = expression(X[2]),
       color = "Success") +
  theme(legend.position = "bottom")
```


Bayes decision boundary
X_2
X_1
Success
FALSE
TRUE