DATA/STAT/BIOSTAT 558
SPRING QUARTER 2025

Homework # 2
Online Submission to Canvas: due Wednesday April 30th, 5pm PST

Instructions: You may discuss the homework problems in small groups, but you must write up the final solutions and code yourself. Please turn in your code used for the problems that involve coding. However, code without written answers will receive no credit. To receive credit, you must explain your answers and show your work. All plots should be appropriately labeled and legible, with axis labels, legends, etc., as needed.

1. Suppose we have a quantitative response Y, and two features X_1 and X_2. Let RSS_1 denote the residual sum of squares that results from fitting the model

$$Y = \beta_0 + \beta_1 X_1 + \epsilon$$

using least squares. Let RSS_{12} denote the residual sum of squares that results from fitting the model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$$

using least squares.

   (a) Argue that $\text{RSS}_{12} \leq \text{RSS}_1$.

   (b) For a fitted model $\hat{f}$ trained on data $(x_1, y_1), \ldots, (x_n, y_n)$, we define the R^2 (*proportion of variance explained*) of the fitted model to be

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{f}(x_i))^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$$

   where $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$. Using your result from 1(a), argue that the R^2 of the model containing just the feature X_1 is no greater than the R^2 of the model containing both X_1 and X_2.

   (c) If you used R^2 as the singular metric to evaluate the quality of your model, then the result in 1(b) suggests that you should add as many predictor variables as you possibly can to your model so that R^2 can be as high as possible. Why might this not be a good idea?

(d) Design a simulation to empirically verify your result in 1(b). For $i = 1, 2, \ldots, 200$, draw $X_i \overset{\text{i.i.d.}}{\sim} \text{Normal}(0, 3^2)$ and $Z_i \overset{\text{i.i.d.}}{\sim} \text{Exponential}(4)$, and then set

$$Y = 2 - 3X_i + \epsilon_i \tag{1}$$

where $\epsilon_i \overset{\text{i.i.d.}}{\sim} \text{Normal}(0, 2^2)$. Then, fit the following three models using least squares:

$$\begin{aligned} Y_i &= \beta_0 + \beta_1 X_i + \epsilon_i \\ Y_i &= \beta_0 + \beta_1 X_i + \beta_2 Z_i + \epsilon_i \\ Y_i &= \beta_0 + \beta_1 X_i + \beta_2 Z_i + \beta_3 \sin(X_i) + \epsilon_i \end{aligned}$$

Calculate the resulting R^2 for each of these three models, repeating the above simulation $b = 1, 2, \ldots, 100$ times. Using your own creativity, create a visual that demonstrates that the R^2 from the models with more predictors is never smaller than the models with less predictors. Importantly, note that this problem does not ever involve any "test data," you are only using training data for this problem.

(*For example, you could label the resulting R^2 values from the three models as $R^2_{1,b}$, $R^2_{2,b}$, and $R^2_{3,b}$, then you could calculate $R^2_{2,b} - R^2_{1,b}$ and $R^2_{3,b} - R^2_{2,b}$ for the $b = 1, \ldots, 100$ simulations and then collect these differences into a histogram or box plot and show that they never go below 0.*)

(e) In Q1(d), the R^2 that you computed was actually the *training set* R^2, in the sense that the observations you used to compute R^2 are the same as the observations that you used to fit the model $\hat{f}(\cdot)$ and to compute $\bar{y}$ (i.e., you used the training observations).

In this subproblem, compute *test set* R^2 instead. To do this, for each of the 100 simulated training sets in Q1(d), compute a test set consisting of 200 test observations drawn from the same model (1). Then, compute the R^2 over this test set, using $\hat{f}(\cdot)$ and $\bar{y}$ obtained using the *training* observations in the expression for R^2. Display the results.

(f) How does test set R^2 compare to training set R^2 in each of the three models considered? Explain your answer in terms of the bias-variance trade-off.

2. Find a data set (either online, or one of the datasets featured in the textbook) with $p = 2$ features X_1 and X_2, a qualitative response Y with $K = 2$ classes, and at least 20 observations per class. *(If you have a data set with more than two features or more than two classes, then feel free to just select a subset of the features and classes so that you can use the data for this problem.)* We are going to predict Y using X_1 and X_2.

First, divide your data into a training set and a test set by reserving 75% of your data for training and 25% for testing. Problems (a) through (e) below will involve only your training data, and then you will bring in your test data in part (g).

(a) Briefly describe the data. Where did you get it? Describe the $K = 2$ classes and the $p = 2$ features. Explain the classification task in words (e.g. a sentence along the lines of "I will use the expression levels of genes ABC and DEF to predict whether a patient belongs to class G or H.")

(b) Fit an LDA model to the data. Make a plot with X_1 and X_2 on the horizontal and vertical axes, and with the observations displayed and colored according to their true class labels. On the plot, indicate which observations are incorrectly classified.

(c) Fit a k-nearest neighbors classifier to the data using $k = 6$. Make a plot with X_1 and X_2 on the horizontal and vertical axes, and with the observations displayed and colored according to their true class labels. On the plot, indicate which observations are incorrectly classified.

(d) Fit a logistic regression model to the data. Make a plot with X_1 and X_2 on the horizontal and vertical axes, and with the observations displayed and colored according to their true class labels. On the plot, indicate which observations are incorrectly classified.

(e) Out of the three models, which one gave you the smallest training error?

(f) *Before making a formal evaluation using your test set,* which of the three models above do you intuitively suspect will do the best on your test set? Why do you think so? (Leave your guess as-is in your response to this problem before moving on to part (g).)

(g) Using your test set, evaluate the performance for each model. Because we have $K = 2$ classes, choose one of your classes to be "positive" and one of your classes to be "negative." (In some cases, this choice will be very easy as your classes may correspond to some notion of positivity and negativity. For other types of data sets, this choice may be totally arbitrary.)

For each model, report each of the following accuracy metrics:

- Proportion of correct predictions on the test set
- False negative rate (the proportion of times that your model predicts a "negative" class out of all the instances where the test point was actually "positive.")
- False positive rate (the proportion of times that your model predicts a "positive" class out of all the instances where the test point was actually "negative.")

(h) Is there a clear winner from your result in 2(g)? **Based upon the context of your chosen data set,** do you think that it would be *more* important to minimize the false negative rate, the false positive rate, or some combination of the two? Explain your answer.

3. Suppose we have a binary response $Y \in \{0, 1\}$, and a single predictor $X \in \mathbb{R}$. This question does not make any sort of modeling assumption (e.g. there is

no need to assume that $\Pr(Y = 1 \mid X = x) = \frac{\exp(\beta_0+\beta_1 X)}{(1+\exp(\beta_0+\beta_1 X))}$). Recall that the odds is defined as $\Pr(Y = 1 \mid X = x)/\Pr(Y = 0 \mid X = x)$.

(a) Suppose that for a given value x, the log-odds equals 0.7. What is $\Pr(Y = 1 \mid X = x)$? What is $\Pr(Y = 0 \mid X = x)$? (Your answer should be an actual number, like 0.2323 but not that number specifically.)

(b) Now for another value x_0, suppose that $\Pr(Y = 1 \mid X = x_0) = 0.3$. What is the log-odds? (Your answer should be an actual number, like -89.23211 but not that number specifically.)

4. Suppose we have a binary response $Y \in \{0, 1\}$, and a single predictor $X \in \mathbb{R}$, and that $\Pr(Y = 1 \mid X = x) = \exp(\beta_0 + \beta_1 x)/(1 + \exp(\beta_0 + \beta_1 x))$. A little birdie tells you that for any value of x, $P(Y = 1 \mid X = x) = 0.9$ and $P(Y = 1 \mid X = x + 2) = 0.5$.

   (a) What are the values of β_0 and β_1? (Your answer should be an actual number, like $\beta_0 = 0.232$ and $\beta_1 = -232444$, but not that number specifically.)

   (b) Make a plot with X on the horizontal axis and Y on the vertical axis. Using the values of β_0 and β_1 that you calculated in Q4(a), display $\Pr(Y = 1 \mid X = x)$ with a solid line and $\Pr(Y = 0 \mid X = x)$ using a dashed line.

5. We will now see that replacing $\beta_0 + \beta_1 X$ with a more flexible function in the expression for logistic regression leads to a more flexible shape for $\Pr(Y = 1 \mid X = x)$.

   Make a plot with X on the horizontal axis and Y on the vertical axis. With $\beta_0 = 0.2$, $\beta_1 = -0.7$, and $\beta_2 = 0.6$, display

   $$\Pr(Y = 1 \mid X = x) = \frac{\exp(\beta_0 + \beta_1 x + \beta_2 x^2)}{1 + \exp(\beta_0 + \beta_1 x + \beta_2 x^2)}.$$

   Make sure to choose a range for the horizontal axis that allows you to see the full shape of the function.

   Comment on the shape of this function. How does it compare to the shape of the logistic function in Q4(b)? Why did replacing $\beta_0+\beta_1 X$ with $\beta_0+\beta_1 X+\beta_2 X^2$ have this effect?

6. In this problem, you will simulate some classification data with 3 classes. Your simulated data should have $p = 2$ features (so that you can easily plot it), a training set of 200 observations, and a test set of 2,000 observations.

   You will compare the test error of two different classifiers: a K-nearest neighbors (KNN) classifier with 5 neighbors, and a classifier that uses a linear decision boundary (you can decide whether to use LDA for this, or multinomial logistic regression — I think the former is slightly easier in `R` and the latter is slightly easier in `Python`. The choice is yours; please just please clarify in your HW which one you used).

The goal of this problem is to figure out how to generate the data in two different ways: so that in (a) the KNN classifier will have lower test error, and in (b) the linear decision boundary classifier will have lower test error.

(a) First, simulate the data in such a way that KNN classifier with 5 nearest neighbors has lower test error than the linear decision boundary classifier.

  i. Describe the simulation setting: how did you generate the data, and why did this lead to KNN having a lower test error than the linear decision boundary classifier?

  ii. Make a plot that displays the *training* observations, as well as the decision boundary corresponding to KNN. Make another plot that displays the *test* observations, as well as the decision boundary from KNN that you obtained from the training data. Then, make these two plots again, but this time displaying the decision boundary corresponding to the linear decision boundary classifier. Within each plot, find a way to indicate which training and test observations are mislabeled by the corresponding classifier.
  Note: The horizontal and vertical axes of your plots should be X_1 and X_2. You might want to use different colors to represent the three classes. You might want to use different symbol types to represent the four types of observations: correctly versus incorrectly classified training observations, and correctly versus incorrectly classified test observations. Be sure to include a legend, and to fully label all aspects of your plot so that it is understandable!

(b) Repeat Q6(a), but in a simulation setting for which the linear decision boundary classifier has lower test error than KNN.