Chapter 7

Regression

Classification is a form of supervised learning in which the outcome is a category. **Regression** is another form of supervised learning in which the outcome is a numeric value. For example, it may be a lab value, physical characteristic (height, weight, etc.), or numeric measurement (e.g. oxygen saturation).

7.1 Visualizing the Regression Problem

Let's consider the same setup from Section 4.2 but this time with a quantitative outcome: a "recurrence biomarker" that indicates the likelihood of recurrence of disease.

Again, we have data on two predictors: a disease severity score (x_1), which characterizes the severity of the illness for which the patient was originally treated, and a social determinants score (x_2), which characterizes a patient's socioeconomic status. We have data on the same 200 patients that we examined in Section 4.2.

This is a plot of the data in a single plane. The color represents the value of the recurrence biomarker – the height of the point above the plane. The goal of regression is to predict the value of the biomarker (y) based on the values of the two predictors, x_1 and x_2.

Question 1

Just looking at the two predictors, which one appears to more highly influence the value of the recurrence biomarker? Why?

Question 2

Think about the three algorithms we discussed in Chapter 4. Now think about our new task, which is to predict the *numeric value* of the recurrence biomarker as a function of the two predictors, x_1 and x_2.

- How might you adapt KNN to deal with this problem?

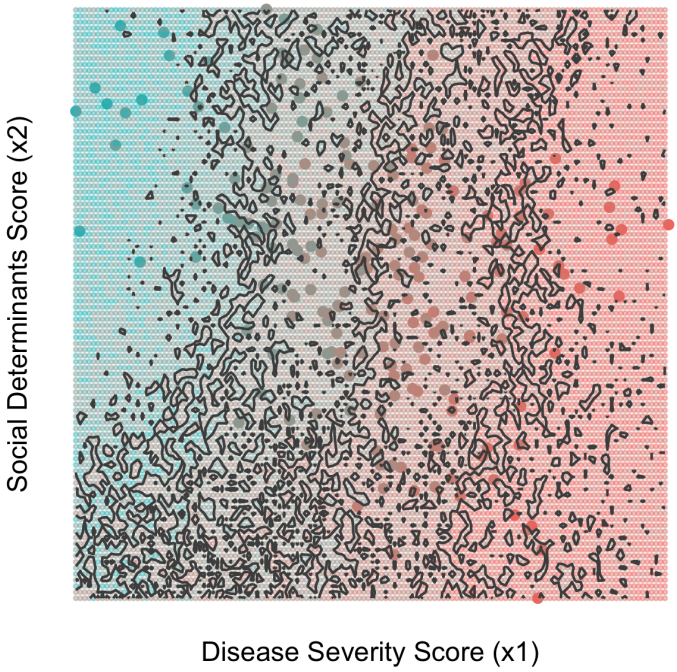
- How might you adapt a decision tree to deal with this problem?
- How might you adapt logistic regression to deal with this problem? You'll have to "break the algorithm" a bit more this time.

7.2 Three Regression Algorithms

7.2.1 K Nearest Neighbors (KNN)

Regression using KNN works very similarly to KNN for classification. In classification, we allow the nearest K points to vote on the label of a new test point. In regression, we interpolate between the values of the surrounding points to come up with the value of y for a test point. Typically this is done just by averaging the y values of the nearest K points, but you can also do something more sophisticated, like weight their contributions by distance to the test point.

Here is a **contour plot** of the regression surface produced by KNN ($K = 15$) for our example:



The contours are at the 25th, 50th, and 75th percentiles of height of the regression surface. Basically this is a topographical map. Its construction is identical to the classification boundaries I drew in Chapter 4; it's just that instead of a single boundary, I've drawn several contours at different heights. This looks like a bit of a mess, but at the same time, the algorithm is able to capture arbitrarily complex relationships between x_1, x_2, and y that can be missed by other regression algorithms.

7.2.2 Decision Tree

In Chapter 12, we built a decision tree for an example in which the outcome in question was binary: yes/no. But what if the outcome is numeric? In that case, we can use **standard deviation reduction** instead of information gain to decide which variables to split on. The sample standard deviation of an outcome, y, is defined as:

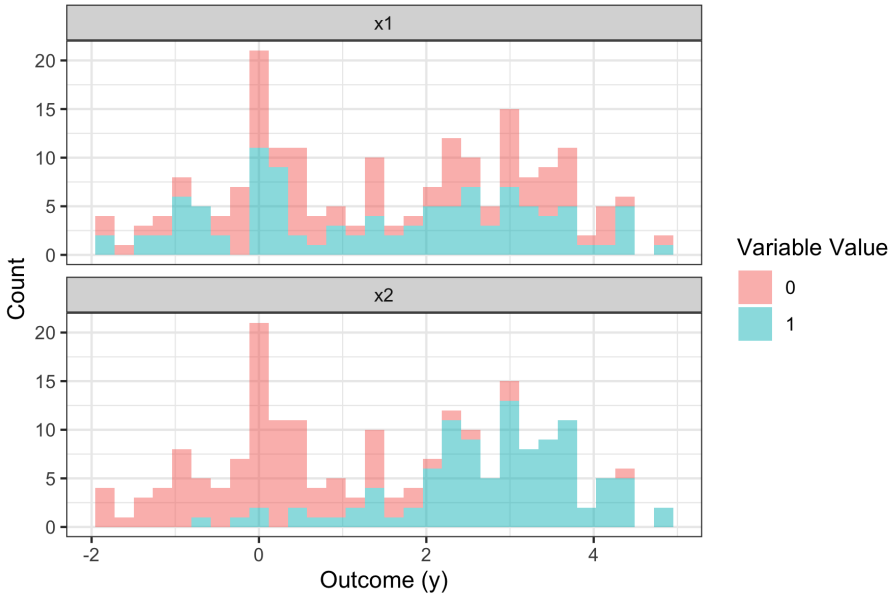
$$S(Y) = \sqrt{\frac{\sum_i (y^{(i)} - \overline{y})^2}{n-1}}$$

where $\overline{y}$ is the overall mean of the outcome. The procedure is identical to the ID3 algorithm (see Chapter 12) except you use conditional standard deviation instead of information gain to decide on features. We define

$$S(Y, X) = \sum_{x \in \text{Values(X)}} \frac{|Y(X = x)|}{|Y|} S(Y(X = x))$$

and at each current leaf node, we split on the variable where the reduction in standard deviation, $S(Y) - S(Y, X)$, is the highest.

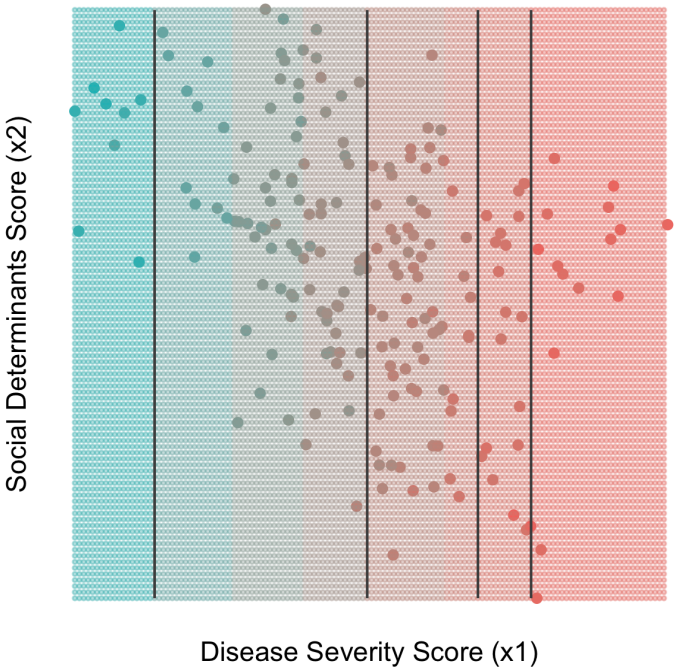
For example, imagine you had a dataset similar in structure to our example, but instead of two real-valued predictors, you have two binary predictors, x_1 and x_2. The decision tree algorithm could choose either one of them to split on first. Here are the distributions of outcome values associated with x_1 and x_2.



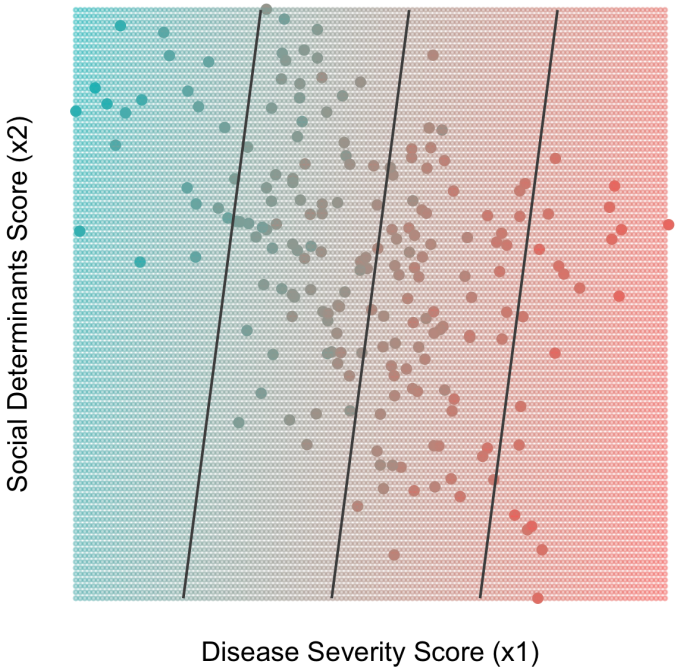
Question 3

Which of these two variables, x_1 or x_2, would make the most sense for a decision tree to split on? What would such a split look like and what would the output value of the tree (the predicted value of y) be for each side of the split?

The predicted biomarker values for a decision tree trained on this dataset (created using the `rpart` package in R with default parameters) are shown here:


Social Determinants Score (x2)
Disease Severity Score (x1)

7.2.3 Linear Regression



Question 4

What are the advantages and disadvantages of each regression algorithm?

Index