STAT 380 – Assessing Model Accuracy Part 1 (Lecture 6)

NOTE: When analyzing data, it is important to consider our goals. Two of the most common goals are:

1. Inference -
   - Drawing conclusions about the population based on the sample;
   - understanding relationships between variables

2. Prediction -
   - Predict as accurately as possible without concern for the ability to explain relationships

Our goals will typically be a combination of inference and prediction.

NOTE: One of the key aims of this class is to introduce you to a wide range of statistical/machine learning methods that extend far beyond the standard linear regression approach. The reason for learning a wide range of methods is because no one method dominates the others over all possible data sets.

IDEA: If we are comparing a wide range of methods, an important task is to figure out which method produces the "best" results for a given dataset.

NOTE: In order to evaluate the performance of a statistical/machine learning method on a given data set, we need some way to measure how well its predictions actually match the observed data.

RECALL: In the regression setting (i.e., the situation in which the response is quantitative), the most commonly used measure for how closely predictions match the observed truth is the mean squared error (MSE) which is given by:

General Regression Problem: $y_i = f(x_i) + \epsilon_i$

Goal: Estimate the unknown $f(x_i)$ with $\hat{f}(x_i)$ to estimate y_i

IDEA: We want to see how well our predictions match the observed data. In regression, we use:

$$\text{Mean Squared Error} = MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}(x_i)\right)^2$$

where

$$y_i = \text{observed response for } i^{th} \text{ observation}$$
$$\hat{f}(x_i) = \text{predicted response for } i^{th} \text{ observation}$$

NOTE: In simple/multiple linear regression, we denote $\hat{f}(x_i)$ as $\hat{y}_i$.

NOTE: Since the MSE is in terms of squared errors, it is often difficult to interpret. For this reason, it is also common to report the square Root of the Mean Square Error (RMSE).

$$RMSE = \sqrt{MSE}$$

EXAMPLE 1: Calculate the MSE and RMSE for the regression model for predicting price of a car using miles and model.

$$\text{Mean Squared Error} = MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

The order of operations for finding MSE:

NOTE: In most cases, we are primarily concerned with how well our model will generalize. In other words, we want to see how accurate the model is when making predictions for new data that was not used in the construction of the model.

Holdout (or Validation) Set Approach

IDEA: We'll use the Holdout Set method (also known as the Validation Set method) of cross validation. Figure 5.1 from Introduction to Statistical Learning with Applications in R (James et al.) depicts the idea:

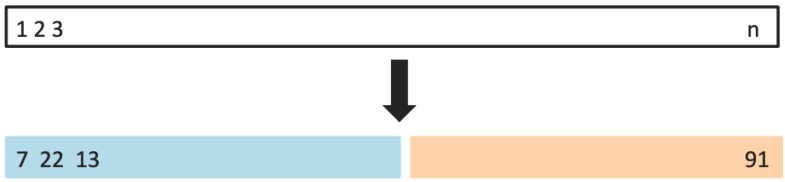
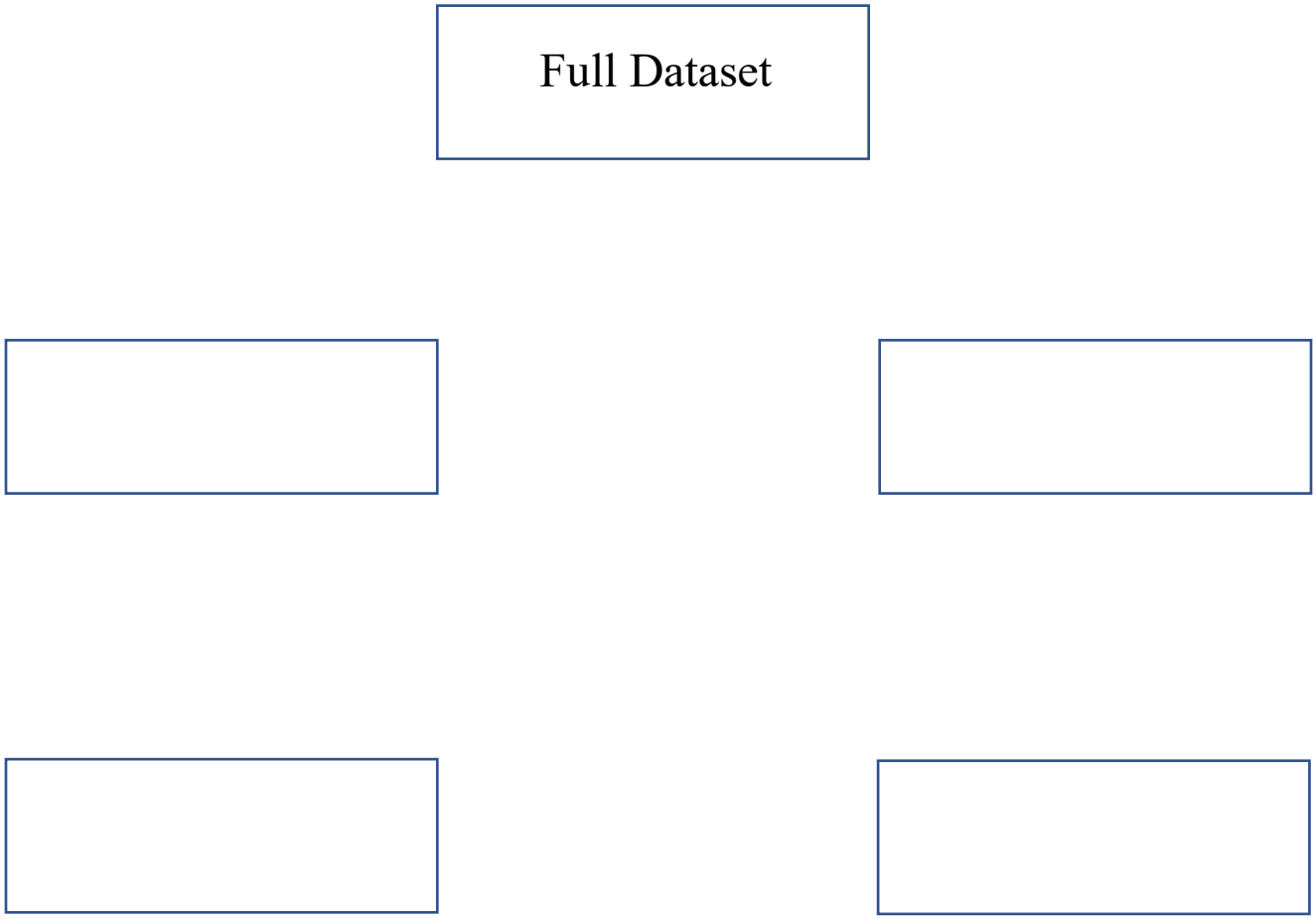


FIGURE 5.1. *A schematic display of the validation set approach. A set of n observations are randomly split into a training set (shown in blue, containing observations 7, 22, and 13, among others) and a validation set (shown in beige, and containing observation 91, among others). The statistical learning method is fit on the training set, and its performance is evaluated on the validation set.*

NOTE: Consider the following:



IDEA: For the sake of assessing the accuracy of competing models, we are primarily interested in the mean square error of the testing set. The method that produces the lowest MSE (or square Root of MSE, called RMSE) on the testing set is often selected as the "best" method for that particular data set.

NOTE: The method that we will implement involves splitting the dataset into two datasets: 1) a training set and 2) a testing (validation) set. Other methods include:

- Splitting the data into a training, testing, and validation set (3 sets)
- k-fold cross validation (split the data into k sets/folds, treat one set as the holdout, use remaining k-1 sets to train; repeat until each set is used as the holdout one time)
- Leave One Out Cross Validation (LOOCV)

NOTE: There are many ways to implement a random splitting of the dataset in R (or anything other language); however, since the split involves randomness, we must settle on an agreed approach so that we can arrive at the same answers.

EXAMPLE 2: (Sample data dictionary) L06_Insurance_m.csv contains information about a number of health insurance policies. In particular, the data set contains some attributes of the policy holder (such as age, sex, etc.) and the total charges billed by the health care provider. (While similar, this dataset is different than the Insurance.txt dataset used earlier in the course.) Here are the variables included:

- age: age of primary beneficiary
- sex: sex of primary beneficiary
- bmi: Body mass index, providing an understanding of body, weights that are relatively high or low relative to height, objective index of body weight (kg / m ^ 2) using the ratio of height to weight, ideally 18.5 to 24.9
- children: Number of children covered by the health insurance policy (i.e., the number of dependents)
- smoker: Status indicating whether the person is a smoker (options include 'yes' and 'no')
- region: the beneficiary's residential area in the US (options include northeast, southeast, southwest, northwest).
- charges: Individual medical costs as billed by health insurance

a. Read L06_Insurance_m.csv into R and name the result Ins. Then, run the code shown below to create Ins2 (results shown below code). The purpose of the next series of questions is to understand the impact of [156:160,] in the code.

```
Ins2 <- Ins[156:160,]
```

	age	sex	bmi	children	smoker	region	charges
156	44	male	39.520	0	no	northwest	6948.701
157	48	male	24.420	0	yes	southeast	21223.676
158	18	male	25.175	0	yes	northeast	15518.180
159	30	male	35.530	0	yes	southeast	*NA*
160	50	female	27.830	3	no	southeast	19749.383

b. What value is returned when you call: Ins2[3, 1]? Where is this value located?

Value:

Location:

NOTE: Unlike other programming languages that begin indexing with 0, R begins indexing with 1.

c. What value is returned when you call: Ins2[1, 3]?

d. What values are returned when you call: Ins2[2,]?

e. What values are returned when you call: Ins2[, 2]?

f. What values are returned when you call: Ins2$sex?

g. What values are returned when you call: Ins2[c(1, 3, 5) , 3]?

h. What do you get when you call: Ins2$Age[3:5 , 1]?

i. What error do you get when you call: Ins2$age[3:5 , 1].

j. Correct the error in the previous question to get the ages of persons 3, 4, and 5.