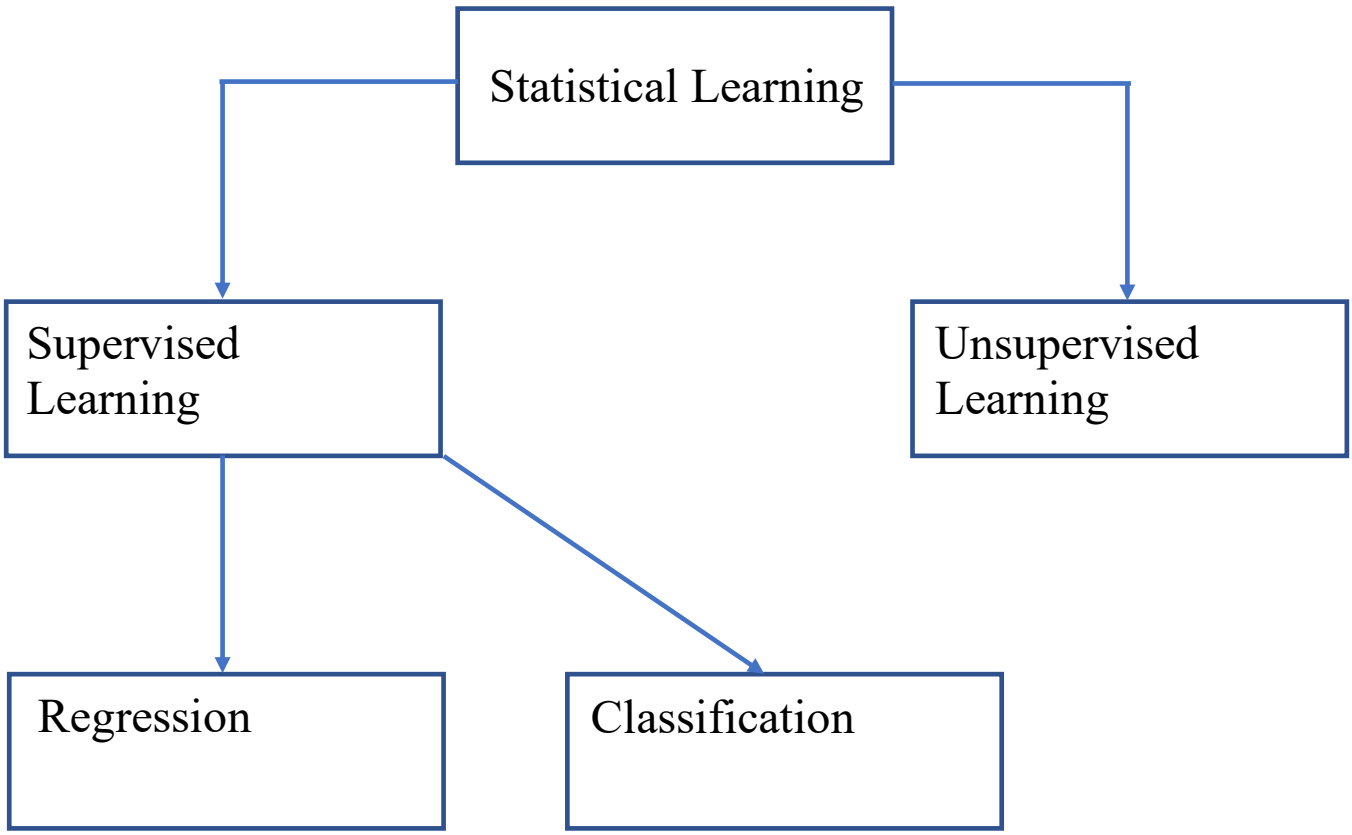
STAT 380 – Decision Trees Part 1 (Lecture 15)

RECALL: Consider the following:



Regression Techniques:

- Multiple Linear Regression
- k-nearest Neighbor

Classification Techniques:

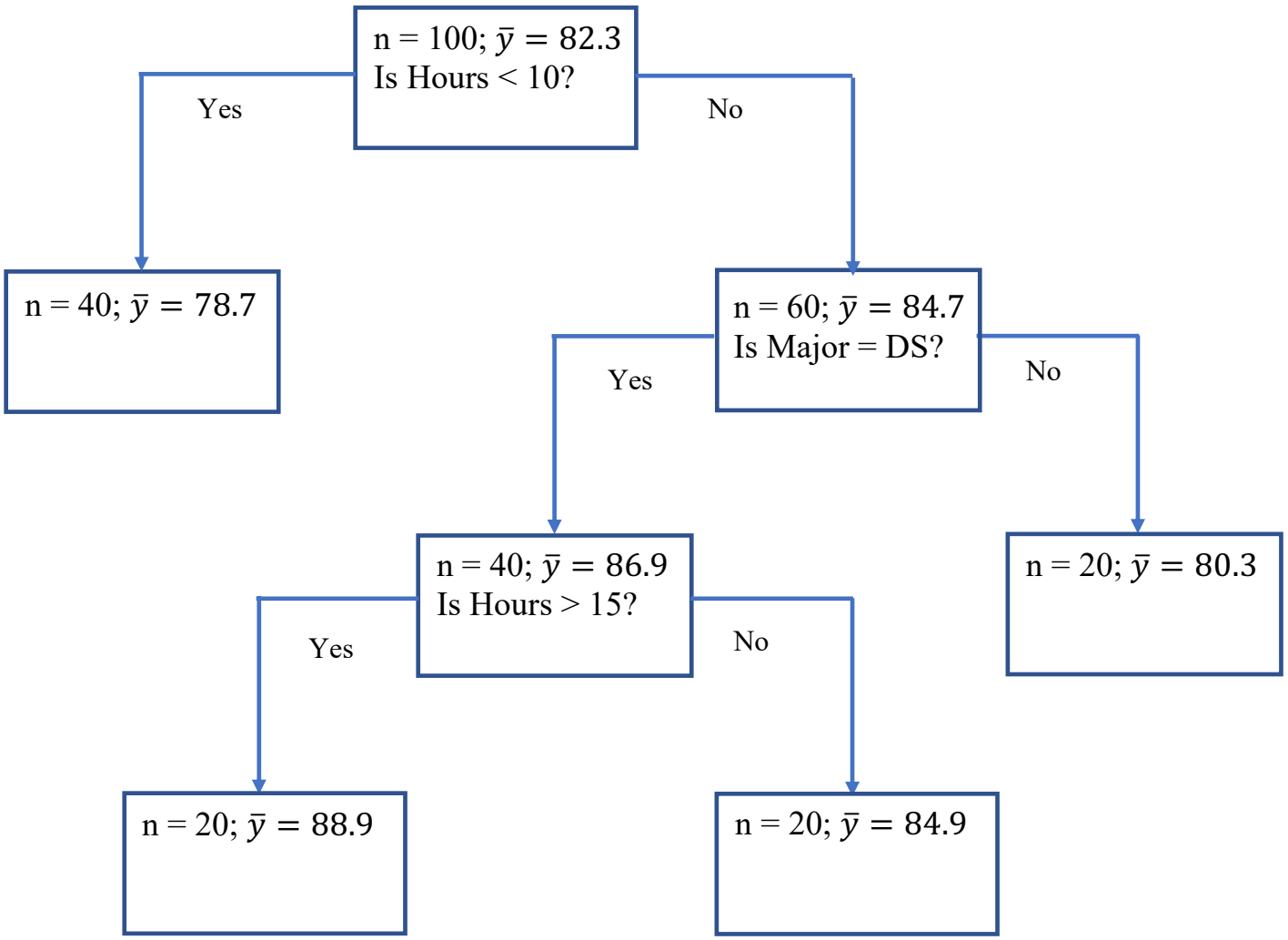
- Logistic Regression
- k-nearest Neighbor

Cross Validation is used for assessing quality of prediction in supervised learning

IDEA: We introduce tree-based methods for regression and classification. Tree-based methods involve stratifying or segmenting the predictor space (X's) into a number of simple regions. In order to make a prediction for a given observation, we typically use the mean or the mode response value for the training observations in the region to which it belongs. Since the set of splitting rules used to segment the predictor space can be summarized in a tree, these approaches are known as decision tree methods.

NOTE: We are going to learn about a method called Classification and Regression Trees (CART).

EXAMPLE 1: Suppose we wish to predict a student's final exam grade based on their major and the hours spent studying for the final exam. Use the following decision tree to predict the final exam grade for a Data Science (DS) student who studied for 12 hours.



QUESTION: What does R^2 represent in regression?

ANSWER: R^2 is the proportion of the total variation in the response that is explained by the model

RECALL: The sample variance is given by:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2$$

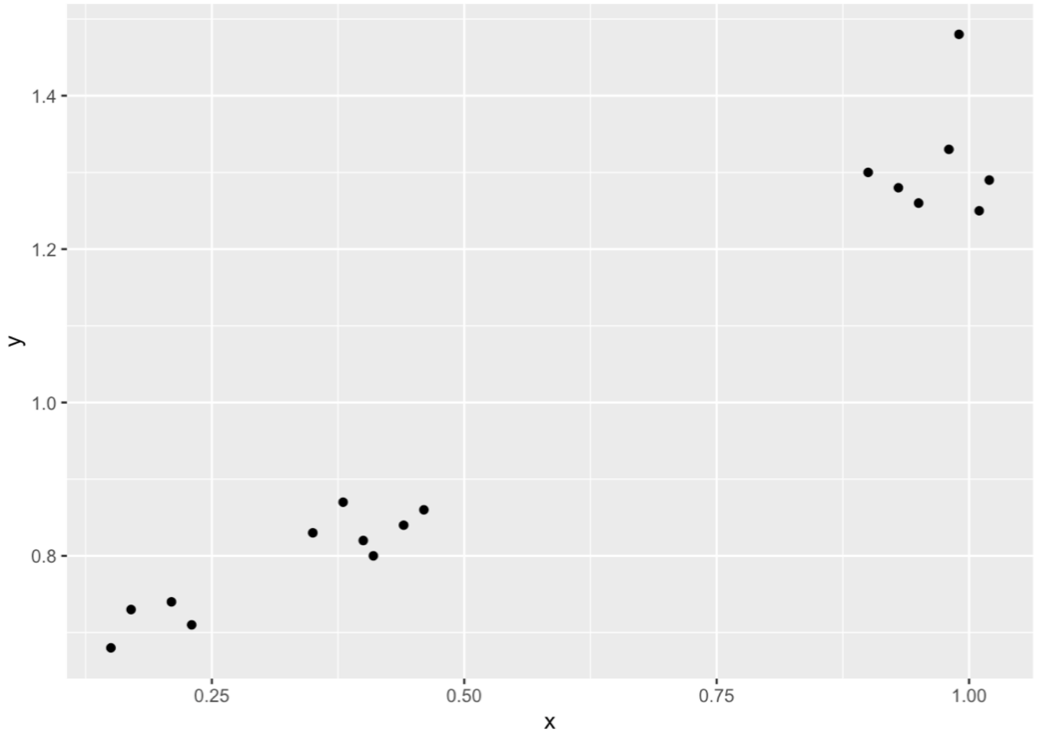
NOTE: Some important definitions:

$$Total\ Sum\ of\ Squares = SSTotal = \sum_{i=1}^{n}(y_i - \bar{y})^2 \qquad Residual\ Sum\ of\ Squares = RSS = \sum_{i=1}^{n}(y_i - \hat{y})^2$$

IDEA: In a regression tree, we get $\hat{y}$ by finding the mean of the y's for the training data in a given node (bin).

EXAMPLE 2: Using L15_CART_Toy.csv, plot the data, find the mean value of y, annotate a few of the distances that go into the total variation, calculate the total variation, and calculate the variance by writing R code. HINT: SSTotal should be 1.212012.

```
ggplot(data = Toy, mapping = aes(x=x, y=y)) +
  geom_point()
```



```r
#Find/store mean
Toy <-
  Toy %>%
  mutate(overallMean = mean(y))

#Use mean to find SSTotal
SSTotal <- sum((Toy$y - Toy$overallMean)^2)

#Calculate Variance Manually
SSTotal/(nrow(Toy)-1)
```

```
## [1] 0.07575074
```

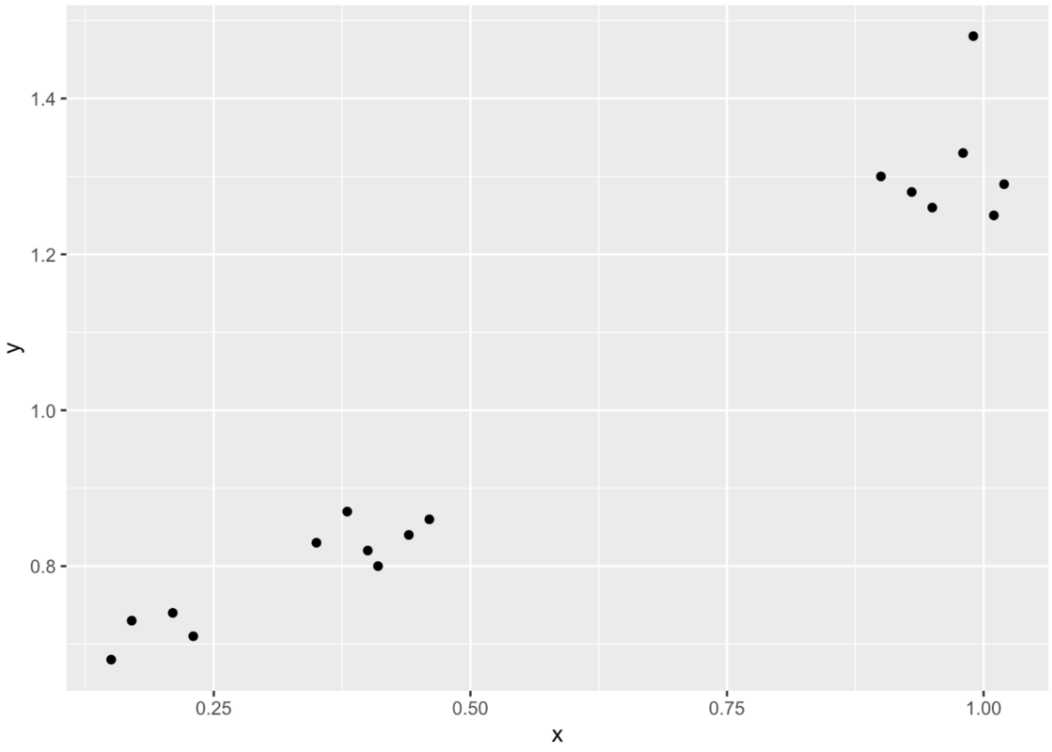
```r
#Calculate variance using the var() function
var(Toy$y)
```

```
## [1] 0.07575074
```

IDEA: In a regression tree, we are going to divide up the x-space. Instead of building a regression model for each subset, we will predict y based on the mean of the observations in each subset.

EXAMPLE 3: Consider the following.

a. Where would you make a cut in the x values to reduce the variation in the y-space as much as possible?



b. Find the value of RSS based on the cut selected in a. NOTE: As part of this, you will have to find 2 values of $\bar{y}$, one for the points to the left of the cut and one for the values to the right of the cut. HINT: RSS = 0.07770286

EXAMPLE 3:

b. (Continued)

```
threshold1 <- 0.6
Toy <-
  Toy %>%
  mutate(group = ifelse(x > threshold1, 1, 2)) %>%
  group_by(group) %>%
  mutate(yMean = mean(y))

RSS <- sum((Toy$y - Toy$yMean)^2)
```

c. What proportion of the total variation in y is explained by this model?

$$R^2 = 1 - \frac{RSS}{SSTotal}$$

```
1 - RSS/SSTotal
```

```
## [1] 0.9358894
```

d. Where would the next cut be?

Growing the Tree

EXAMPLE 4: Use L15_CART_Toy.csv.

a. Grow the regression tree using the rpart function from the rpart package. Suppose that we want the minimum number of observations in any terminal node to be at least 3. What splitting rules does R use?

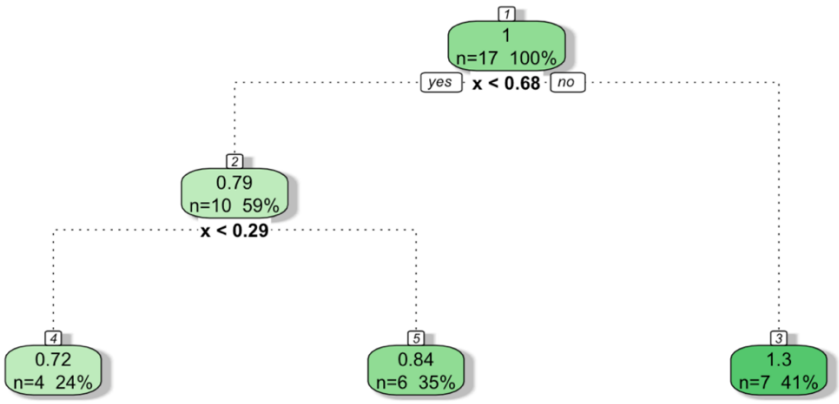
```
res = rpart(y ~ x, method="anova", data=Toy,
            minbucket=3)
```

```
summary(res)
```

```
## Call:
## rpart(formula = y ~ x, data = Toy, method = "anova", minbucket = 3)
##   n= 17
##
##           CP nsplit  rel error     xerror       xstd
## 1 0.93588935      0 1.00000000 1.11061571 0.17775426
## 2 0.02931215      1 0.06411065 0.08304435 0.02977970
## 3 0.01000000      2 0.03479850 0.06776194 0.02954112
##
## Variable importance
##   x
## 100
##
## Node number 1: 17 observations,    complexity param=0.9358894
##   mean=1.004118, MSE=0.07129481
##   left son=2 (10 obs) right son=3 (7 obs)
##   Primary splits:
##       x < 0.68 to the left,  improve=0.9358894, (0 missing)
##
## Node number 2: 10 observations,    complexity param=0.02931215
##   mean=0.788, MSE=0.004096
##   left son=4 (4 obs) right son=5 (6 obs)
##   Primary splits:
##       x < 0.29 to the left,  improve=0.8673503, (0 missing)
##
## Node number 3: 7 observations
##   mean=1.312857, MSE=0.00524898
##
## Node number 4: 4 observations
##   mean=0.715, MSE=0.000525
##
## Node number 5: 6 observations
##   mean=0.8366667, MSE=0.0005555556
```

b. Use R to create a nice tree using the fancyRpartPlot function from the rattle library. Then, predict y for a new observation with x = 0.53.

```
#Create plot using fancyRpartPlot for rattle library
fancyRpartPlot(res, cex=0.8)
```



c. Based on the results in a. and b., how many nodes are created? How many terminal nodes?

d. Putting each observation into its own node would produce an R^2 of 1. Explain why this is a bad idea.