Stats 315B: Homework 1

Joe Higgins, Austin Wang, Jessica Wetstone

Due 5/6/2018

Question 1

1. (15) Data Mining Marketing. The data set age_stats315B.csv represents an extract from a commercial marketing database. The goal is to fit a regression tree to predict the age of a person from 13 demographic attributes and interpret the results. Note that some of the variables are categorical: be sure to mark them as such using the R function as.factor, before running rpart. Use the RPART implementation of the decision tree algorithm to fulfill this task. Write a short report about the relation between the age and the other demographic predictors as obtained from the RPART output and answer the following questions:

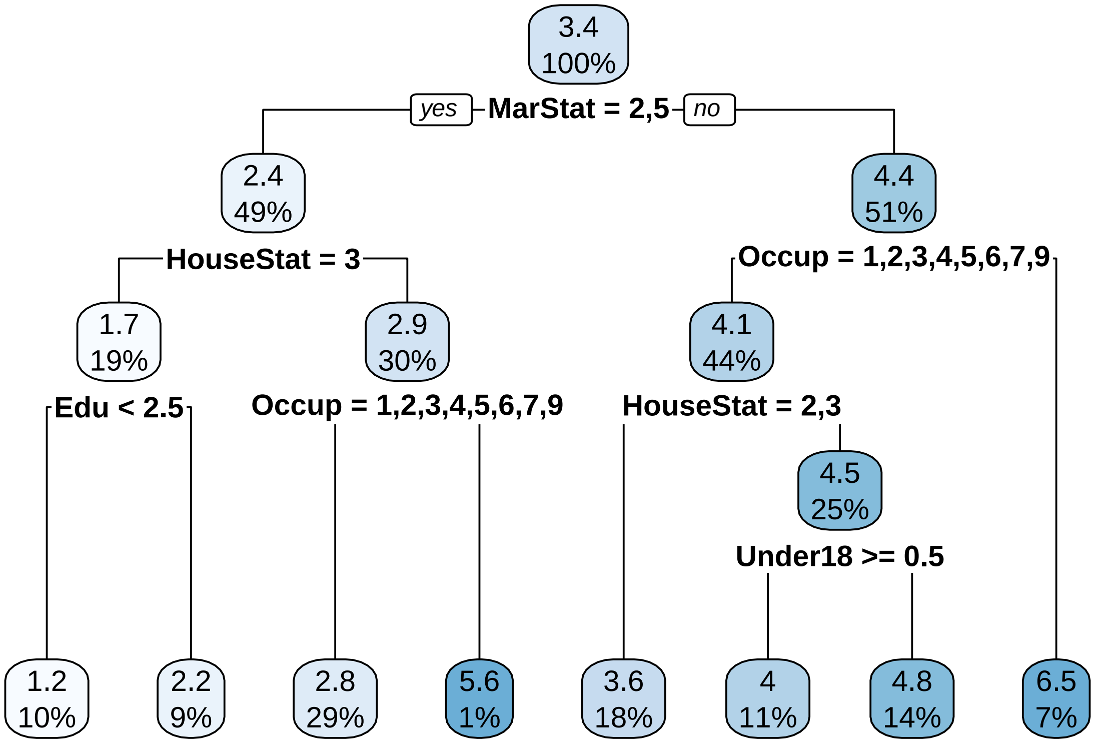
(a) Were surrogate splits used in the construction of the optimal tree you obtained? What does a surrogate split mean? Give an example of a surrogate split from your optimal decision tree. Which variable is the split on? Which variable(s) is the surrogate split on?

(b) Using your optimal decision tree, predict your age.

```r
rm(list = ls())
#Set working directory to data subdirectory
#current_path <- getActiveDocumentContext()$path
#current_directory <- dirname(current_path)
#data_path <- paste(current_directory,'/data',sep='')
#setwd(data_path)

data_path <- paste(getwd(),'/data',sep='')
setwd(data_path)


#Read and type data
age_data <- read.csv('age_stats315B.csv')
factor_columns <- c(
  'Occup',
  'TypeHome',
  'sex',
  'MarStat',
  'DualInc',
  'HouseStat',
  'Ethnic',
  'Lang'
)
age_data[factor_columns] <- lapply(age_data[factor_columns], as.factor)
fit <- rpart(age ~ ., data = age_data)
rpart.plot(fit)
```



```
#a)
colnames(age_data)[fit$frame$nsurrogate]
```

```
## [1] "MarStat" "MarStat" "MarStat" "age"     "MarStat" "sex"
```

```
#yes there were surrogate splits used. Surrogates are variables used to classify data points
#that have missing values for a given feature The surrogate variables used were:

#b)
ncols <- dim(age_data)[2]
joe_data <- data.frame(matrix(ncol = ncols, nrow = 0))
                              #1,2,3,4,5,6,7,8,9,10,11,12,13,14
joe_data <- rbind(joe_data, c(3,1,3,1,5,6,9,3,1, 4, 0, 2, 8, 1))
colnames(joe_data) <- colnames(age_data)
joe_data[factor_columns] <- lapply(joe_data[factor_columns], factor)
predict(fit, joe_data)
```

```
##        1
## 2.808754
```

```
#result: 2.8, on the upper end of 18 through 24. I'm 29. Gotcha tree!
```

Question 2

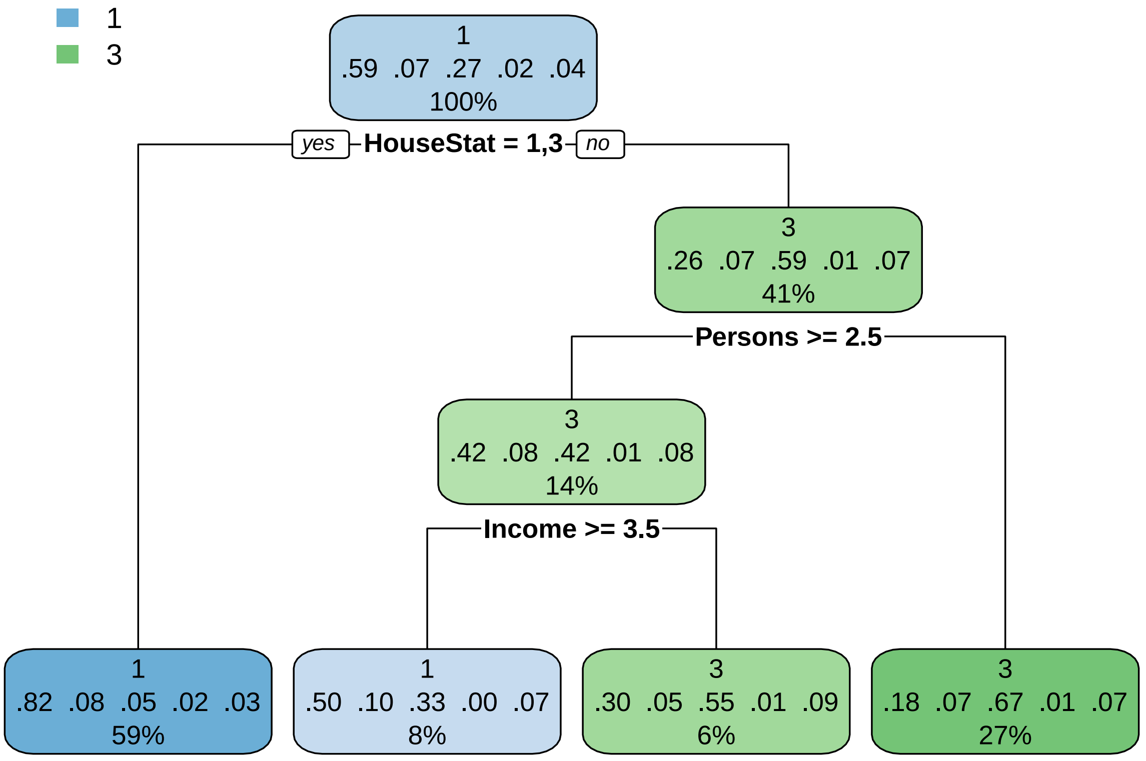
2. (15) Multi-Class Classification: Marketing Data. The data set housetype_stats315B.csv comes from the same marketing database that was used for problem 1. Refer to the documen- tation housetype_stats315B.txt for attributes names and order. From the original pool of 9409 questionnaires, those with non-missing answers to the question "What is your type of home?" were selected. There are 9013 such questionnaires. The goal in this problem is to construct a classification tree to predict the type of home from the other 13 demographics attributes. Give an estimate of the

misclassification error of an 1 optimal tree. Plot the optimal tree if possible (otherwise plot a smaller tree) and interpret the results.

```r
rm(list = ls())
#Set working directory to data subdirectory
#current_path <- getActiveDocumentContext()$path
#current_directory <- dirname(current_path)
#data_path <- paste(current_directory,'/data',sep='')
#setwd(data_path)

data_path <- paste(getwd(),'/data',sep='')
setwd(data_path)

#Read and type data
house_data <- read.csv('housetype_stats315B.csv')
factor_columns <- c(
  'TypeHome',
  'sex',
  'MarStat',
  'Occup',
  'LiveBA',
  'DualInc',
  'HouseStat',
  'Ethnic',
  'Lang'
)
house_data[factor_columns] <- lapply(house_data[factor_columns], as.factor)
fit <- rpart(TypeHome ~ ., data = house_data, method = 'class')
rpart.plot(fit)
```



```r
#say something here

#misclassification
y_hat <- predict(fit, house_data)
y_hat_max_p <- apply(y_hat,1,which.max)
y <- house_data$TypeHome
correct <- y == y_hat_max_p
pct_correct <- sum(correct)/length(correct)
1-pct_correct
```

```
## [1] 0.2614002
```

3. (5) What are the two main reasons why a model that accurately describes the data used to build it, may not do a good job describing future data?

- Overfitting to training data
- Future data comes from different distribution than training data

4. (5) Why can't the prediction function be chosen from the class of all possible functions?

Because then we would have a problem: there are many (infinite) solutions that minimize empirical risk and we can overfit. If our function class was all functions, we would get many bad and overfitting functions if we don't have enough data or we don't regularize our score with certain penalties.

5. (5) What is the definition of the target function for a given problem? Is it always an accurate function for prediction? Why/why not?

The target function is the function that minimizes prediction risk (expected value of loss). No it is not always an accurate function for prediction because… if you have a bad class of functions

6. (5) Is the empirical risk evaluated on the training data always the best surrogate for the actual (population) prediction risk? Why/why not? In what settings would it be expected to be good?

Often empirical risk evaluated on training data works well. However, this is not always the case.

7. (10) Suppose the loss for an incorrect classification prediction is the same regardless of either the predicted calue ck of the true value cl of the outcome y. Show that in this case misclassification risk reduces the classification error rate. What is the Bayes rule for this case in terms of the probabilities of y realizing each of its values PR(y=ck)Kk=1? Derive this rule from the gneral (unequal loss) Bayes rule, for this particular loss structure Lkl = 1(k != l).

8. (5) Does a low error rate using a classification rule derived by substituting probability KK estimates {Pr(y = ck)}k=1 in place of the true probabilities{Pr(y = ck)}k=1 in the Bayes rule imply accurate estimates of those probabilities? Why?

To look at.

9. (5) Explain the bias-variance trade-off.

The bias-variance trade-off states that in general if we increase model complexity, we tend to reduce squared bias but increase variance.

An example of a high-bias low-variance estimating function is a constant (for a target function that is not a constant): there is a consistent and predictable bias, and there is no variance.

However we can reduce bias by introducing more model complexity. Specifically, we can change the estimating function to rely on the data. As we increase model complexity by having it rely on the data, we inherently introduce variance since the it is assumed there is a random component to the data and thus our loss is random.

10. Why not choose surrogate splits to best predict the outcome variable y rather than the primary split.

jess has coverage

Question 11

Show that the values of c_m that minimize the squared-error risk score criterion are given by:

$$\hat{c}_m = \frac{\sum_{i=1}^{N} y_i I(x_i \in R_m)}{\sum_{i=1}^{N} I(x_i \in R_m)}$$

We will show this by taking the partial derivative of the score function $S(\mathbf{c}) = \sum_{i=1}^{N}[y_i \; - \sum_{k=1}^{M} c_k I(x_i \in R_k)]^2$ with respect to an arbitrary c_m:

$$\begin{aligned}
S(\mathbf{c}) &= \sum_{i=1}^{N}[y_i \; - \sum_{k=1}^{M} c_k I(x_i \in R_k)]^2 \\
\implies \frac{\partial S}{\partial c_m} &= 2\sum_{i=1}^{N}[y_i \; - \sum_{k=1}^{M} c_k I(x_i \in R_k)](-I(x_i \in R_m)) \\
&= -2\sum_{i=1}^{N} y_i I(x_i \in R_m) \; + \; 2\sum_{i=1}^{N}\sum_{k=1}^{M} c_k I(x_i \in R_k)(I(x_i \in R_m)) \\
&= -2\sum_{i=1}^{N} y_i I(x_i \in R_m) \; + \; 2\sum_{i=1}^{N} c_m I(x_i \in R_m)
\end{aligned}$$

Let $\hat{c}_m$ be the value that sets $\frac{\partial S}{\partial c_m} = 0$. Then:

$$\begin{aligned}
-2\sum_{i=1}^{N} y_i I(x_i \in R_m) \; + \; 2\sum_{i=1}^{N} \hat{c}_m I(x_i \in R_m) = 0& \\
\implies \sum_{i=1}^{N} \hat{c}_m I(x_i \in R_m) &= \sum_{i=1}^{N} y_i I(x_i \in R_m) \\
\implies \hat{c}_m \sum_{i=1}^{N} I(x_i \in R_m) &= \sum_{i=1}^{N} y_i I(x_i \in R_m) \\
\implies \hat{c}_m &= \frac{\sum_{i=1}^{N} y_i I(x_i \in R_m)}{\sum_{i=1}^{N} I(x_i \in R_m)}
\end{aligned}$$

Finally, we show that this value is in fact the unique minimizer by examining the second derivative with respect to c_m:

$$\begin{aligned}
\frac{\partial S}{\partial c_m} &= -2\sum_{i=1}^{N} y_i I(x_i \in R_m) \; + \; 2\sum_{i=1}^{N} c_m I(x_i \in R_m) \\
\implies \frac{\partial^2 S}{\partial c_m^2} &= 2\sum_{i=1}^{N} I(x_i \in R_m) \\
&> 0
\end{aligned}$$

where the final inequality comes from the fact that at least one training example must exist in R_m for it to be defined.

Therefore, the second partial derivative with respect to c_m is strictly positive, so the score function is strictly convex in c_m and admits a unique minimum given by $\hat{c}_m = \frac{\sum_{i=1}^{N} y_i I(x_i \in R_m)}{\sum_{i=1}^{N} I(x_i \in R_m)} \qquad \blacksquare$.

Question 12

Show that the improvement in squared-error risk (1) when one of the regions R_m is split into two daughter regions, where n is the number of observations in the parent R_m, n_l, n_r the numbers respectively in the left and right daughters, and $\bar{y_l}$ and $\bar{y_r}$ are the means of the outcome variable y for observations in the respective daughter regions.

To show this, we first define some notation. Let:

$$\bar{m} = \{k \mid x_k \in R_m\}$$
$$\bar{l} = \{k \mid x_k \in R_l\}$$
$$\bar{r} = \{k \mid x_k \in R_r\}$$

and note that this implies:

$$\bar{l} \, \cup \, \bar{r} = \bar{m}$$

Now, we can define the squared-error risk from the full region R_m and the squared-error risk from the daughter regions R_l and R_r as:

$$S_m = \sum_{i \in \bar{m}} (y_i - \bar{y}_m)^2$$
$$S_{lr} = \sum_{i \in \bar{l}} (y_i - \bar{y}_l)^2 + \sum_{i \in \bar{r}} (y_i - \bar{y}_r)^2$$

where

$$\bar{y}_m = \frac{1}{n} \sum_{i \in \bar{m}} y_i$$
$$\bar{y}_l = \frac{1}{n_l} \sum_{i \in \bar{l}} y_i$$
$$\bar{y}_r = \frac{1}{n_r} \sum_{i \in \bar{r}} y_i$$

Note that from the above equations, we can derive the following result:

$$\begin{aligned} \bar{y}_m &= \frac{1}{n} \sum_{i \in \bar{m}} y_i \\ \implies \bar{y}_m &= \frac{1}{n} (\sum_{i \in \bar{l}} y_i + \sum_{i \in \bar{r}} y_i) \\ \implies \bar{y}_m &= \frac{n_l}{n} \bar{y}_l + \frac{n_r}{n} \bar{y}_r \end{aligned}$$

We are interested in the difference between S_m and S_{lr}. Note that this will give the total improvement in the squared-error risk because the risk associated with other regions $\neq m$ is not affected by the splitting of region m into two daughter regions.

We begin by rewriting the squared-error risks as follows:

$$\begin{aligned} S_m &= \sum_{i \in \bar{m}} (y_i - \bar{y}_m)^2 \\ &= \sum_{i \in \bar{m}} y_i^2 - 2\bar{y}_m \sum_{i \in \bar{m}} y_i \; + \sum_{i \in \bar{m}} \bar{y}_m^2 \\ &= \sum_{i \in \bar{m}} y_i^2 \; - 2n\bar{y}_m^2 \; + n\bar{y}_m^2 \\ &= \sum_{i \in \bar{m}} y_i^2 \; - n\bar{y}_m^2 \end{aligned}$$

Similarly:

$$\begin{aligned} S_{lr} &= \sum_{i \in \bar{l}} (y_i - \bar{y}_l)^2 \; + \sum_{i \in \bar{r}} (y_i - \bar{y}_r)^2 \\ &= \sum_{i \in \bar{l}} y_i^2 \; - n_l \bar{y}_l^2 \; + \sum_{i \in \bar{r}} y_i^2 \; - n_r \bar{y}_r^2 \\ &= \sum_{i \in \bar{m}} y_i^2 \; - n_l \bar{y}_l^2 \; - n_r \bar{y}_r^2 \end{aligned}$$

Then, the improvement in the squared-error risk from splitting R_m into the two daughter regions R_l and R_r is given by:

$$\begin{aligned}
S_m - S_{lr} &= (\sum_{i\in\bar{m}} y_i^2 - n\bar{y}_m^2) - (\sum_{i\in\bar{m}} y_i^2 - n_l\bar{y}_l^2 - n_r\bar{y}_r^2) \\
&= -n\bar{y}_m^2 + n_l\bar{y}_l^2 + n_r\bar{y}_r^2 \\
&= -n(\frac{n_l}{n}\bar{y}_l + \frac{n_r}{n}\bar{y}_r)^2 + n_l\bar{y}_l^2 + n_r\bar{y}_r^2 \\
&= -n(\frac{n_l^2}{n^2}\bar{y}_l^2 + \frac{2n_ln_r}{n^2}\bar{y}_l\bar{y}_r + \frac{n_r^2}{n^2}\bar{y}_r^2) + n_l\bar{y}_l^2 + n_r\bar{y}_r^2 \\
&= -\frac{n_l^2}{n}\bar{y}_l^2 - \frac{2n_ln_r}{n}\bar{y}_l\bar{y}_r - \frac{n_r^2}{n}\bar{y}_r^2 + n_l\bar{y}_l^2 + n_r\bar{y}_r^2 \\
&= (\frac{nn_l - n_l^2}{n}\bar{y}_l^2) + (\frac{nn_r - n_r^2}{n}\bar{y}_r^2) - \frac{2n_ln_r}{n}\bar{y}_l\bar{y}_r \\
&= [\frac{(n_l + n_r)n_l - n_l^2}{n}\bar{y}_l^2] + [\frac{(n_l + n_r)n_r - n_r^2}{n}\bar{y}_r^2] - \frac{2n_ln_r}{n}\bar{y}_l\bar{y}_r \\
&= \frac{n_ln_r}{n}\bar{y}_l^2 + \frac{n_ln_r}{n}\bar{y}_r^2 - \frac{2n_ln_r}{n}\bar{y}_l\bar{y}_r \\
&= \frac{n_ln_r}{n}(\bar{y}_l^2 - 2\bar{y}_l\bar{y}_r + \bar{y}_r^2) \\
&= \frac{n_ln_r}{n}(\bar{y}_l - \bar{y}_r)^2 \qquad \blacksquare
\end{aligned}$$

13. (10) Derive an updating formula for calculating the change in the improvement in prediction risk as the result of a split when the split is modified by one observation changing sides.

should follow from 12.

14. Is this always a good idea? Will it necessarily lead to better expected mse on future data? Why or why not? Conversely, is it always better to reduce the size of F (increasing the restriction on g(x)), thereby fitting the training data less well? Why or why not?

No it is not always a good idea. This can often lead to over-fitting. One example is polynomial fits. We can reduce MSE by allowing higher and higher order polynomials in F, but the result on new data may be worse than even a simple first order polynomial fit (linear model). The higher order polynomial will have many curves and pass through all the data points, but it will not generalize well. Therefor, is not always a good idea to allow larger and larger function classes for scripty F.

15. (5) The recursive partitioning strategy described in class for building decision trees uses two-way (binary) splits at each step. This is not fundamental, and one could envision multi-way splits of each non-terminal node creating several (rather than two) daughter regions with each split. What would be the relative advantages and disadvantages of a such a multi-way splitting strategy?

it exists in lectures... more restrictive, likely better to do first split then look and do that same subsequent split if its still the best. maybe slower if greedy algo over all multi-splits.

16. (5) As described in class, the recursive binary splitting strategy considers splits that only involve a single input variable. One could generalize the procedure by including "linear combination" splits of the form

where the sum is over the numeric input variables only. The values of the coefficients {aj} and the split points are (jointly) chosen to optimize the splitting criterion, which is same as that used for single variable splits. What would be the advantages and disadvantages of including such linear combination splits in the tree building strategy?

talks about in lecture.