xiaoying zhao

Xiaoying.zhao1@baruchmail.cuny.edu; (917)445-7663



Classification tree

CART Application for Classification Problems

Table of Contents

[I. Chapter 1 Classification Tree 2](#_Toc533196879)

[A. Definition 2](#_Toc533196880)

[B. Building a Classification Tree 3](#_Toc533196881)

[C. Tree Pruning 3](#_Toc533196882)

[D. Selection Criteria 4](#_Toc533196883)

[1. Classification error rate 4](#_Toc533196884)

[2. Gini index 4](#_Toc533196885)

[3. Cross-entropy 4](#_Toc533196886)

[II. Chapter 2 Application of Classification Tree 5](#_Toc533196887)

[A. R package *tree* with *Carseats* example 5](#_Toc533196888)

[B. Stock Data 9](#_Toc533196889)

[C. Harbor Water Data 12](#_Toc533196890)

[III. Chapter 3 Comparison 14](#_Toc533196891)

Cover image is from

https://www.quartoknows.com/blog/quartocreates/beginning-watercolor-starting-with-trees

# Chapter 1 Classification Tree

Decision trees can be adopted for both regression and classification studies. This project focuses on presentation and application of classification tree. This project contains three components. In Chapter 1, it briefly introduces the definition of classification tree, building process, criteria or measurement, and tree pruning. In Chapter 2, it firstly presents the application of R package *tree* for textbook example, then it applies to the stock data and harbor water data. In Chapter 3, it compares the classification tree to other classification methods in the textbook.

## Definition

Classification and Regression Trees (CART) is firstly introduced by Leo Breiman and his co-authors (1984)[[1]](#footnote-1) in their book “Classification and regression trees (the wadsworth statistics/probability series)”. CART refers to decision tree algorithms that are for predictive methods. Regression tree is used to predict a quantitative response, and classification tree is designed to predict a qualitive response.[[2]](#footnote-2)

For example, figure 1 shows a decision tree for the mammal classification problem.[[3]](#footnote-3) It also shows three different kinds of node in the classification. The mammal classification starts with the first attribute, *body temperature,* which is also the root node. Root node has zero incoming edges and zero or more outgoing edges, usually the starting point of the tree. According to the body temperature, animals can be separated into two groups: cold-blooded creatures, which are non-mammals, and warm-blooded creatures. *Give birth*, the second attribute, which is an internal node, is adopted to distinguish mammals from other warm-blooded vertebrates, such as birds. Internal node has only on incoming edge and two or more outgoing edges. Both root nodes and internal nodes are non-terminal nodes, which contain attribute test conditions to separated records that have different characteristics. The final node labeled Non-mammals and Mammals are leaf nodes, which is also terminal nodes, having only one incoming edge and no outgoing edges.

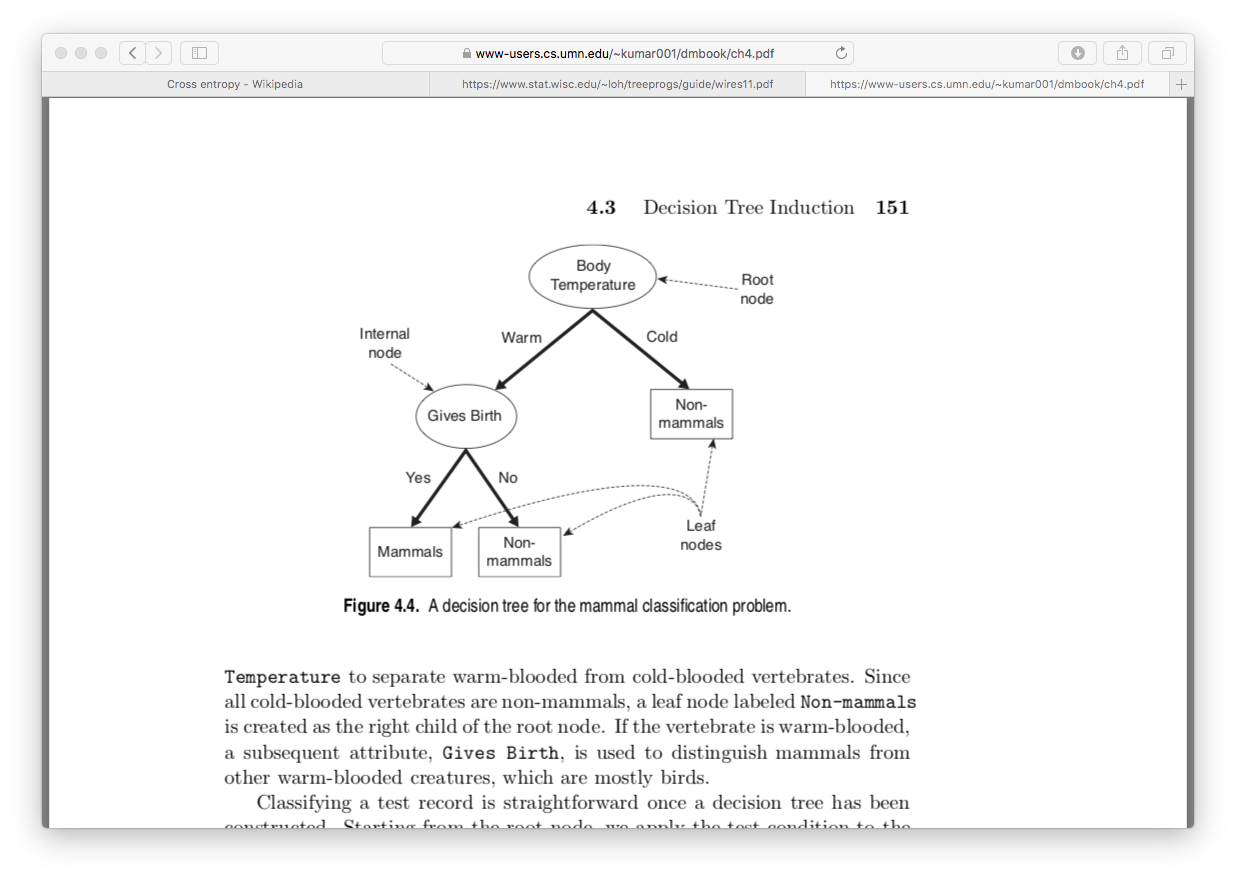


Figure A decision tree for the mammal classification problem

## Building a Classification Tree

The process of the mammal classification problem shows decision tree involves two important issues.[[4]](#footnote-4) First, every level of the tree-growing process needs test conditions to divide observations into subsets. It requires the system to have a method to specify the test conditions and a measurement to evaluate the test condition. Second, it is about the terminal of the tree-growing process. Although people may believe that the process is supposed to be stopped when all the observations belong to the same attribute, actually, under some criteria, the tree-growing process may be terminated earlier with obvious advantages.

Let us see the process of building a classification tree.

Step 1: use recursive binary splitting to grow a large tree on the training data.

Step 2: apply criteria pruning to the large tree in order to obtain a small tree with lowest misclassification rate.

Step 3: use cross-validation to evaluate the performance

1. Repeat step 1 and 2 on all the training data.
2. Evaluate the misclassification.

Step 4: return the subtree from step 2 the misclassification rate to achieve the best performance

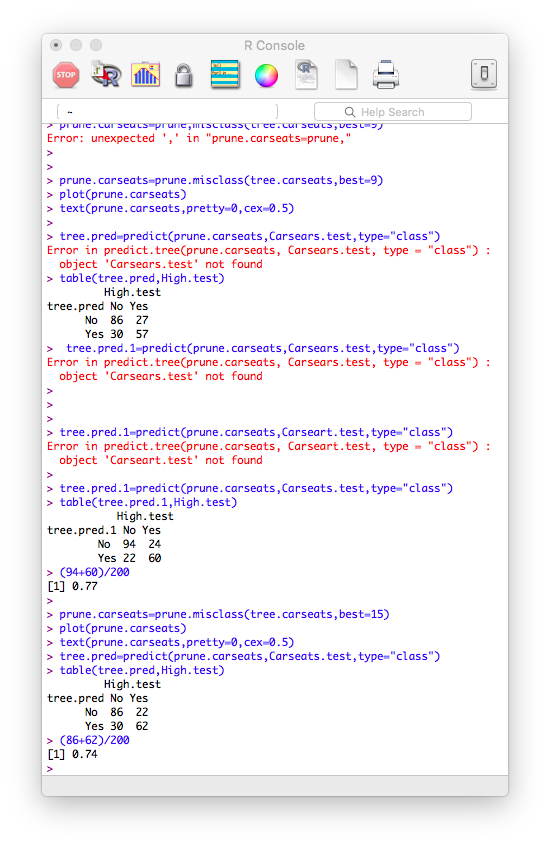
## Tree Pruning

The initial tree growing may lead to a huge tree with too many subsets, causing an undesired performance. A huge tree with many splits is also hard to interpret. Therefore, it is necessary to achieve a smaller tree with fewer splits but has better variance output. Usually, we would like to grow a large tree at the initial stage to check the status. Then we would like to prune the necessary splits to achieve the highest correct classification rate. How to make these splits involving the measurement and criteria.

## Selection Criteria

### Classification error rate

“Assign an observation in a given region to the most commonly occurring class of training observations in that region, the classification error rate is simply the fraction of the training observations in that region that do not belong to the most common class. represents the proportion of training observations in the mth region that are from the kth class. ”[[5]](#footnote-5)

The explanation and formula in the text book is hard to understand. But it can be easily understood by the example. In the case on the left, we can calculate the classification error rate, which is . It means the misclassified prediction is around 23% of the locations in the testing set.

### Gini index

The Gini index is also called Gini impurity. It measures the correct possibility if a randomly labeled element is randomly chosen from the set is incorrectly labeled.[[6]](#footnote-6)

### Cross-entropy

This approach is numerically similar to Gini index.

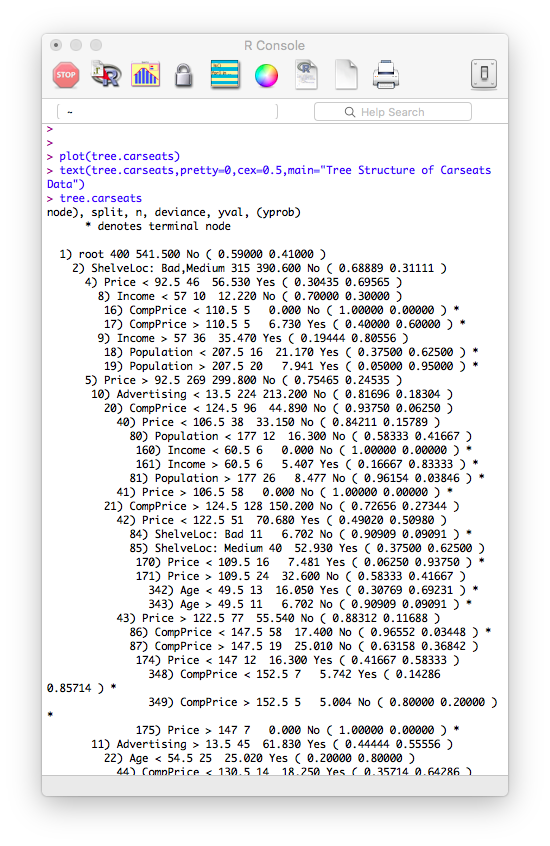
# Chapter 2 Application of Classification Tree

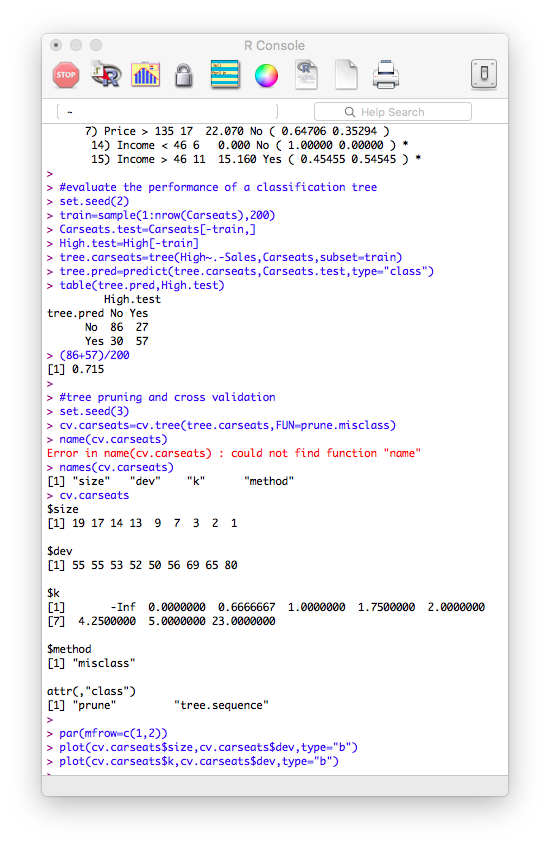
Chapter 2 contains the classification tree packages in R for three applications. It firstly goes through the *Carseats* example in the text book, then it uses classification tree method to analysis the stock data, and finally it applies to harbor water sampling data.

## R package *tree* with *Carseats* example

Example in the textbook uses data set *Carseats*, which is attached to the *ISLR* package. Firstly, it sets condition that if sales are greater than 8, then its sales volume is high, then it is converted to “Yes”. If sales are less than or equal to 8, then it is converted to “No”. This new yes or no column is combined to the original *Carseats* data set, and the old sales numeric column is deleted from the data set. It uses *tree()* function directly create a classification tree and uses *summary()* function to see the output. It can be seen from output that there total 27 nodes, and misclassification error rate is 0.09.

Let’s look into the details of the tree output. The content in every row follows the same order: node, split, n, deviance, y-val, (yprob). Row 1 which is root includes all the observations of the data set and there is no split. Take node 8 income < 57 as an example. n=10, which means that there are 10 rows where income <57. “No” means in the majority of those 10 rows that the value of High = No. The percentage of “No” values in High is 0.7, and the misclassification rate is 0.3. It can be also seen that deviance is 12.220. Figure 2 shows the plot of this classification tree.



 Then Let’s evaluate the performance of the tree with 27 possible nodes. Use *predict()* function to check the classification rate. The correct prediction rate for the tree with 27 nodes is about 71.5%.

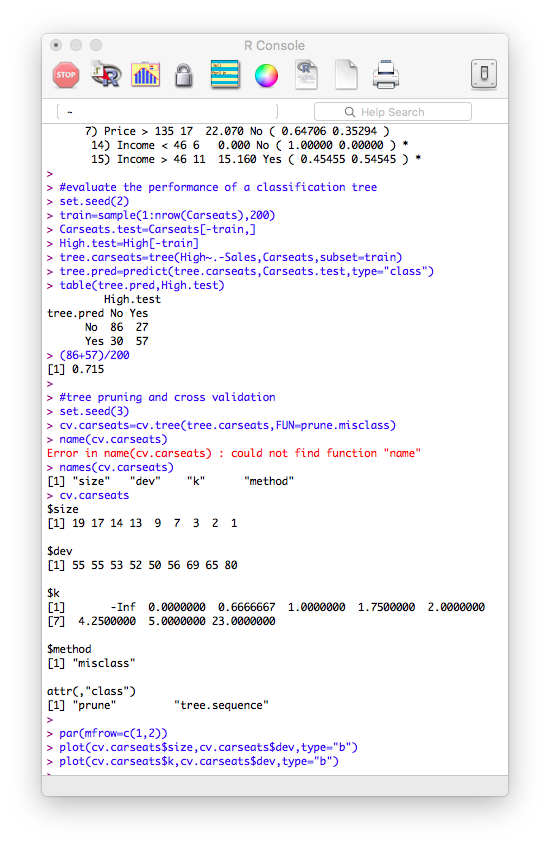
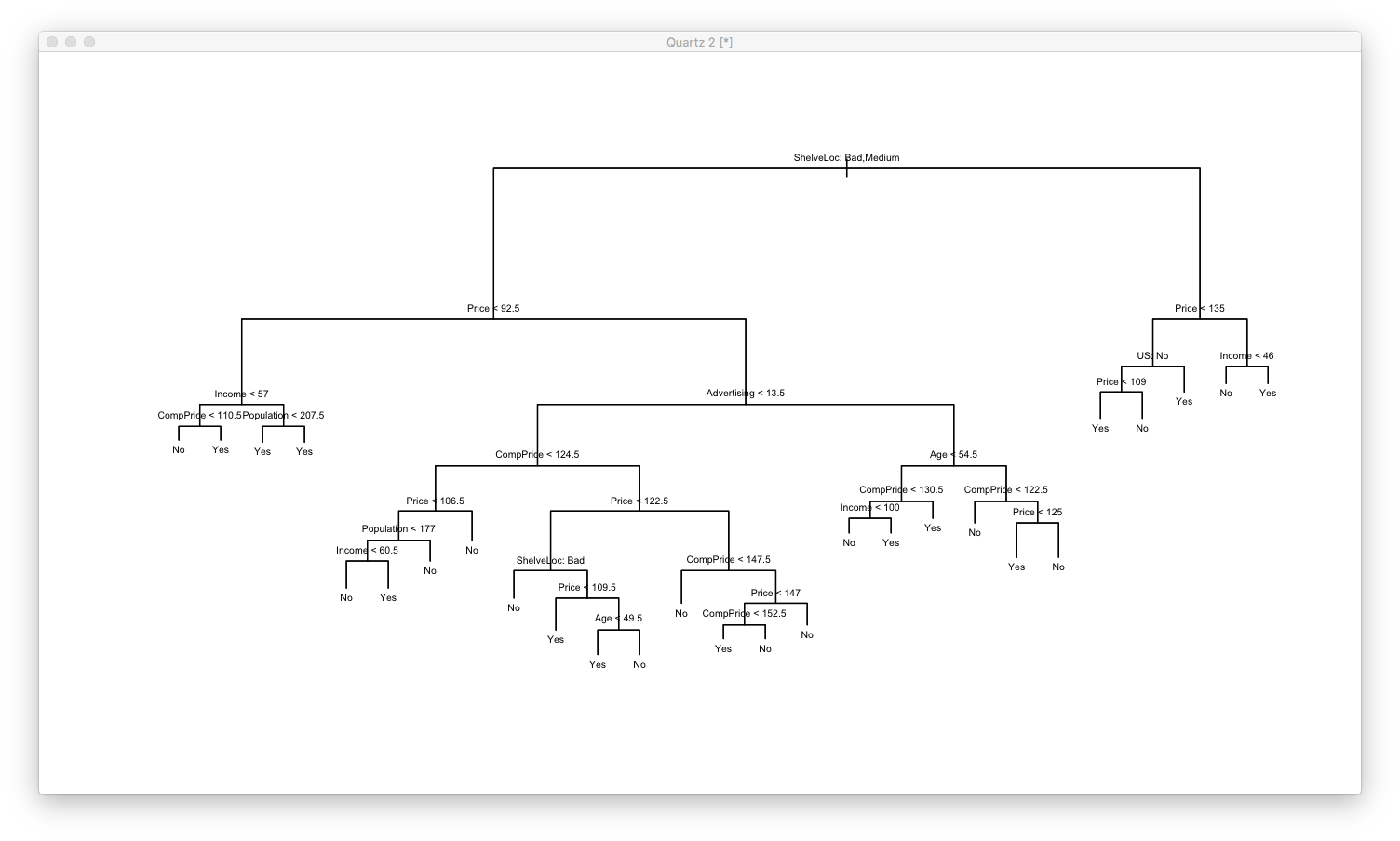


Figure Tree with all 27 possible nodes



Let’s try to prune the tree by cross-validation to achieve performance improvement. Check size and corresponding deviance in the cross-validation output. When the size is 9, meaning 9 nodes, the deviance is lowest, which is 50 cross-validation errors. “k” means the value of the cost-complexity parameter. We can use size and k to plot the misclassification rate.

We can see from the plot that when the size=9, the deviance is the smallest. Therefore, I would like to use *prune.misclass()* function to prune the tree into that with 9 nodes.



Figure Cross-Validation: dev vs. size and dev vs. k

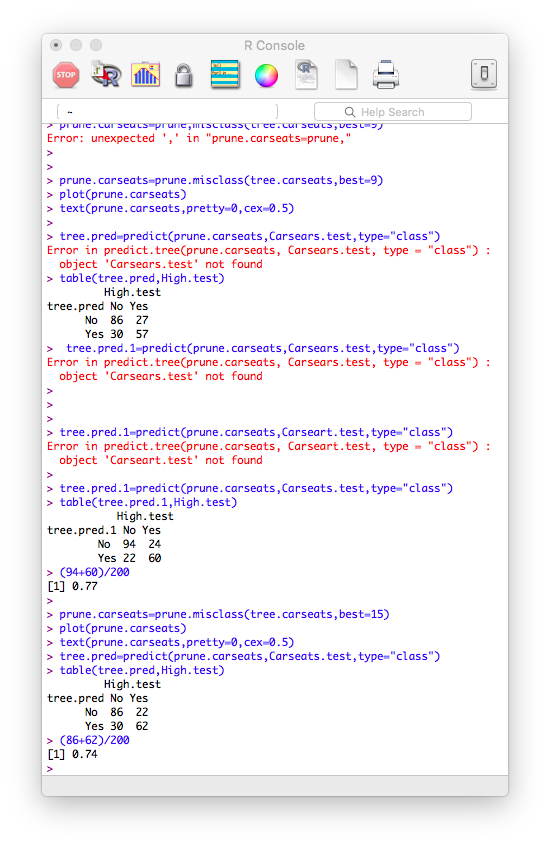
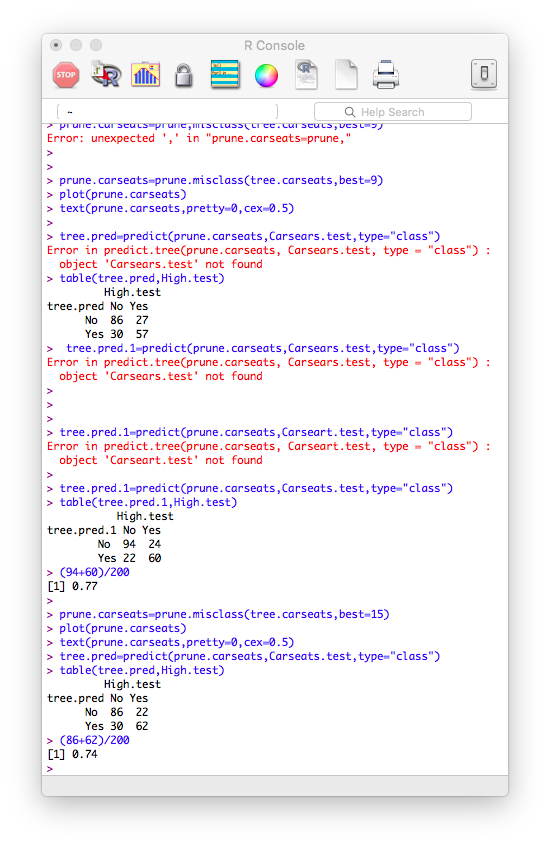
After being cut 22 nodes, the tree with 9 nodes is much clearer. To check the correct classification rate to make sure the performance is indeed improved. The output is in the next page. The correct classification rate is improved from 71.5% to 77%.



Figure Tree with 9 nodes CV

 You may not believe that this is the best performance we can have for this data set. We can randomly try a node size to see the result. If we choose the best 15 nodes, then the correct classification rate decreased from 77% to 74%.

## Stock Data

I would like to apply classification tree to stock data in this section. The company I choose for this application is Diamond Offshore Drilling, Inc., with stock symbol of DO. Diamond Offshore Drilling is a leading company in offshore drilling with a total of 17 offshore drilling rigs.[[7]](#footnote-7) Also, the stock of the company has a sound performance. “The stock that is trading at $19.99 went higher by 46.55% from its 52-week low of $13.64 that it attained back on 2018-02-15. The stock recorded a 52-week high of $21.92 nearly 90 days ago on 2018-06-28” (The GV Times,2018)[[8]](#footnote-8). The most interesting part is the company is the first one to apply the innovation service of industrial blockchain technology. It also provides this service to companies in the drilling industry to help with improvement of efficiencies and reduction of wastes.

There are totally 2770 observations in the data file. There are total 8 variables (opening price, highest price, lowest price, closing price, adjusted closing price, volume, daily return, and risk level) in the data. The 8th variable is the classification variable, also the y variable in the model. It indicates the risk level of stock at that day, either high risk or low risk. As for the initial tree output, there are 9 terminal nodes in this case. The misclassification rate is 14.55%.

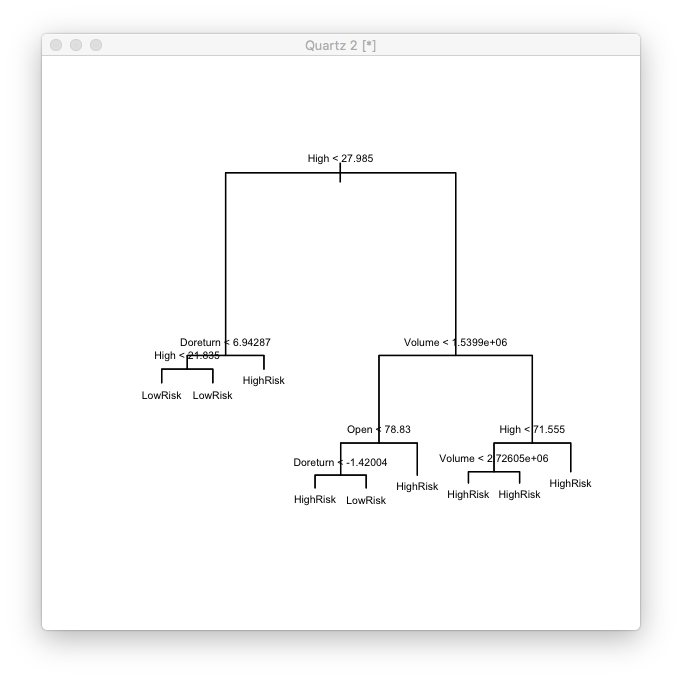
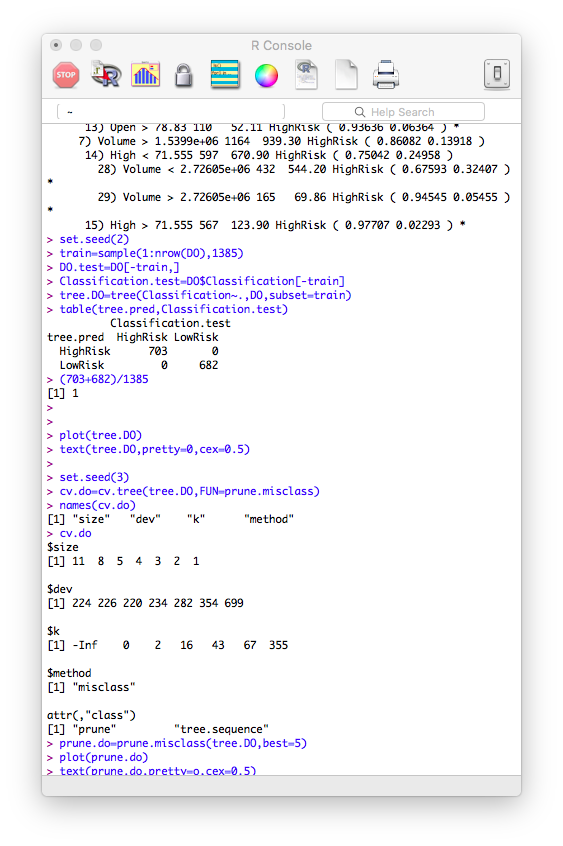
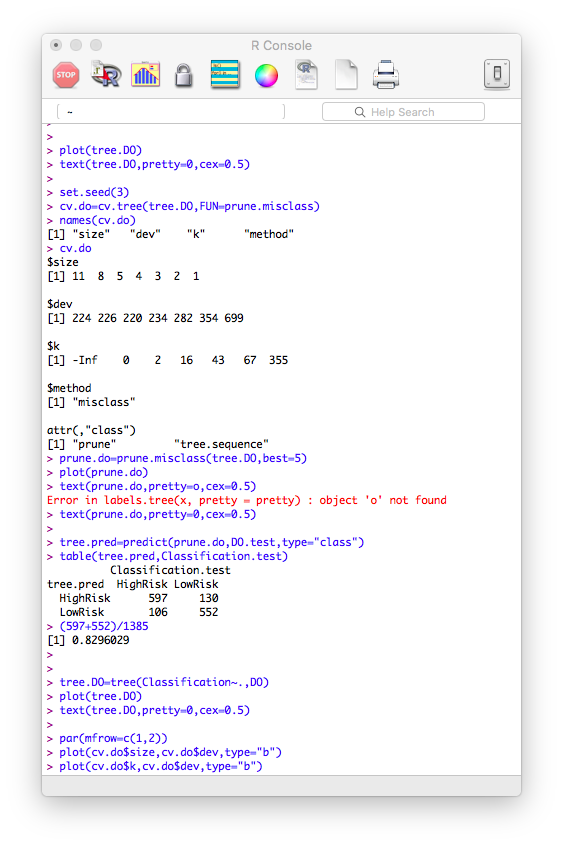


Figure Stock tree with 9 terminal nodes



It is surprising that the correct classification rate for the initial tree is 100%! It means it can correctly predict 100% of observation in the test set. When I got this result first time, I believe I made some mistakes about coding or importing. After several checks, I started to think about the stock data itself. It is the nature of stock market. It is naturally dangerous trading in the stock market. The company is oil drilling company, which is also naturally high-risk choice. Although under terminal node size=9, the correct classification rate 100%, it does not reach the lowest deviance. The tree with terminal nodes = 5 reaches the lowest deviance 220. It turns out the correct classification rate is 82.96%.

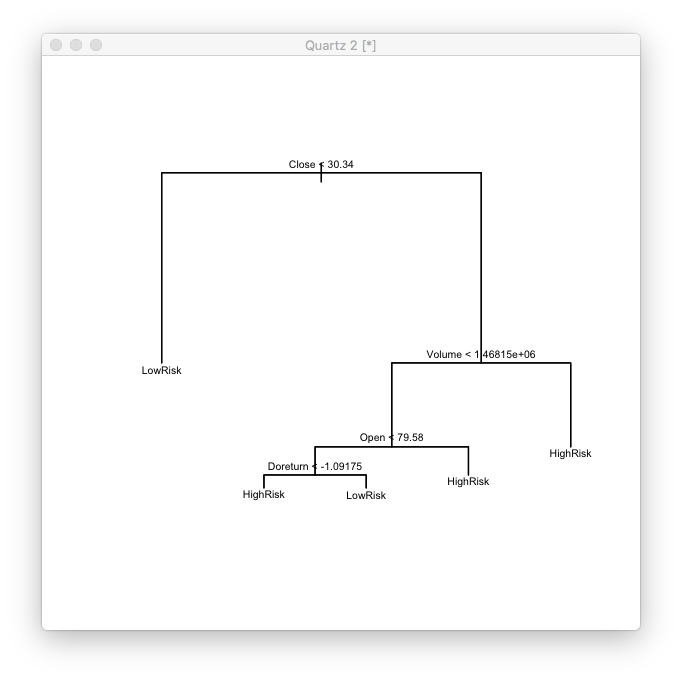
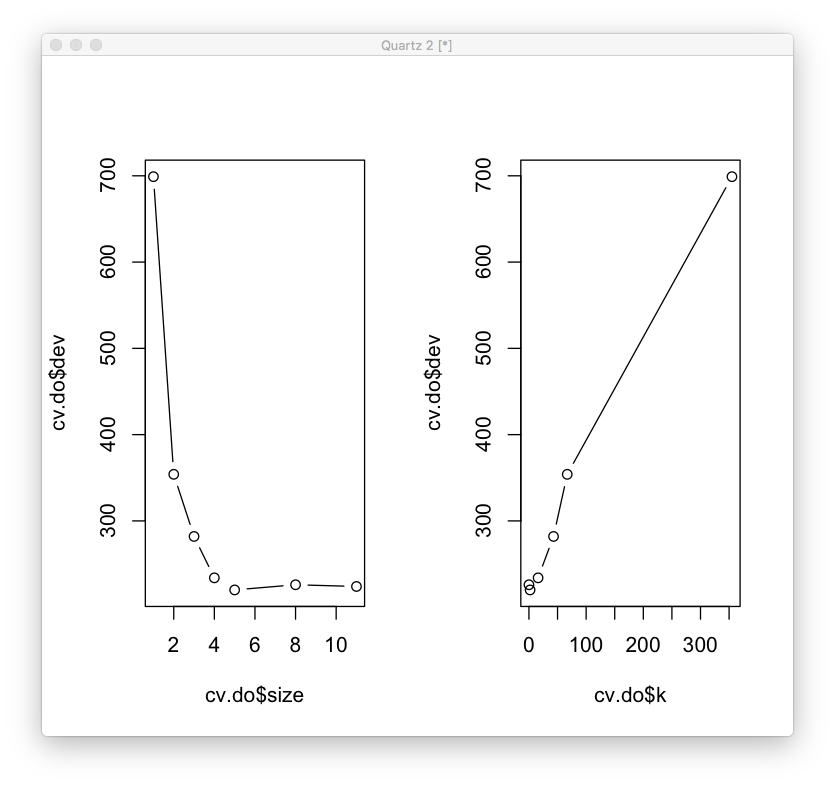


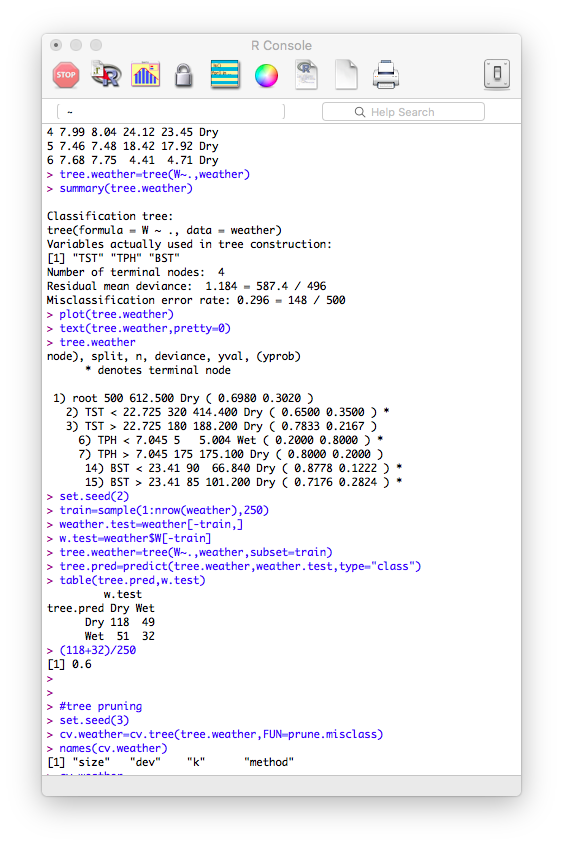
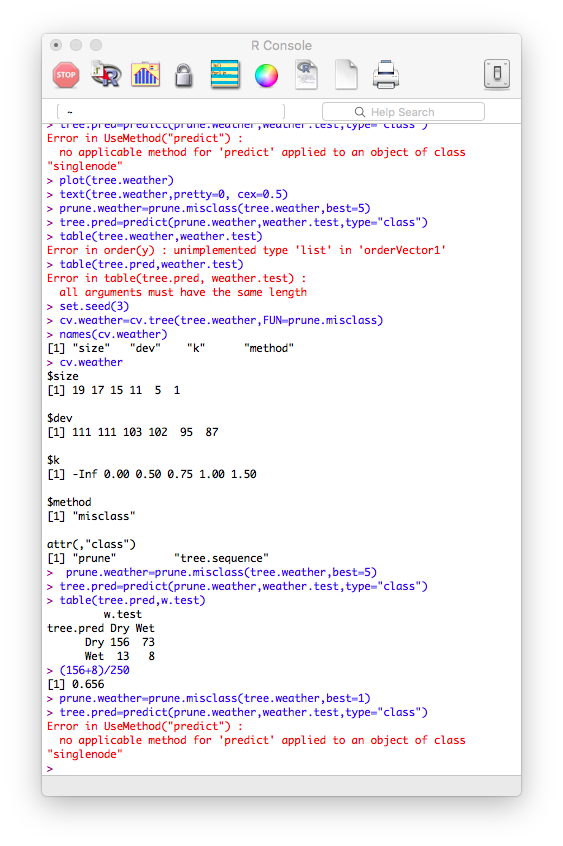
Figure 6 Stock tree with 5 terminal nodes

## Harbor Water Data

I would like to apply classification tree to harbor water sampling data in this section. The original data file “Harbor Water Quality”[[9]](#footnote-9) is acquired from NYC Open Data and is generated by The New York City Department of Environmental Protection (DEP). DEP collects data from totally 87 sampling stations harbor wide to monitor and analysis water conditions. The Harbor Survey is the one of the longest continuous reginal water quality monitoring programs in the US. According to the most recent report, the Harbor is cleaner now than at any time in the last 100 years[[10]](#footnote-10).

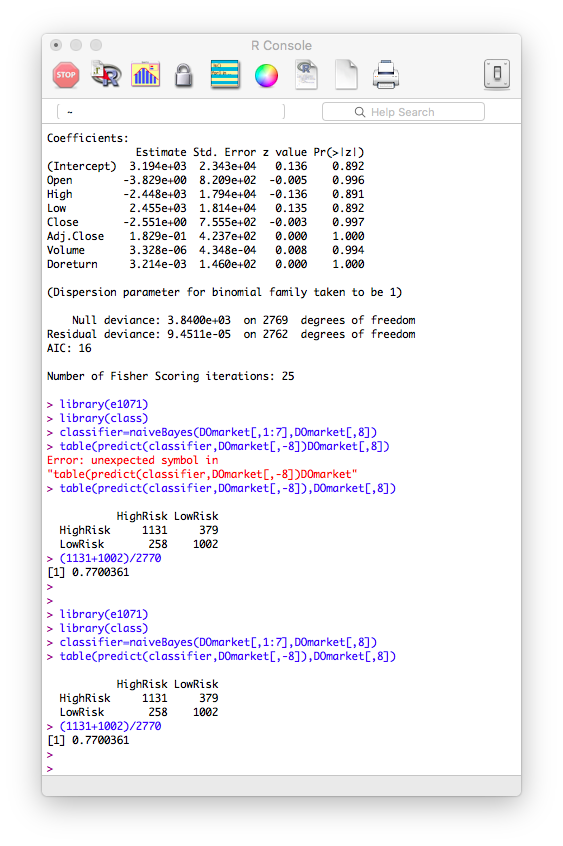
I have four x-variables and one classification variable. There are 500 observations.

|  |  |  |  |
| --- | --- | --- | --- |
|  | variable | symbol | Definition |
| X-variables | PH Top Sample | TPH | Top sample pH measure using a portable meter- Method 4500- H B. |
| PH Bottom Sample | BPH | Bottom sample pH measure using a portable meter- Method 4500- H B. |
| Temperature Top Sample(ºC) | TST | Top sample temperature in Celsius |
| Temperature Bottom Sample (ºC) | BST | Bottom sample Temperature in Celsius |
| Y-variable | Weather Condition (Dry or Wet) | W | Weather condition in the last 48 hours (Rain or Dry) |

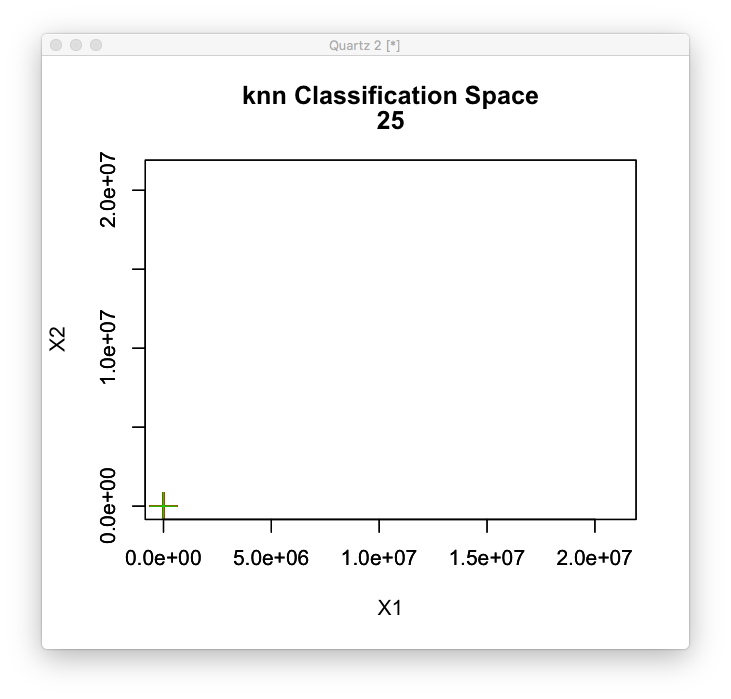
The initial tree output is shown on the left. It presents 4 terminal nodes. The misclassification error rate is 29.6%. To evaluate this result, I divide all the data into 2 groups: a training group and a testing group. Use *predict()* function to check the classification rate. The correct classification rate is 60% in this case, which means it there are around 60% of observations in the testing group can be predicted correctly. Use cross validation to find out the specific terminal nodes for this case. The interesting thing is that the when the size=1, it reaches the best performance, which is the lowest deviance 87. But there is no need to practise on single node, even the program does not apply the method. Then I would like to try on the second-best choice, size = 5, which has corresponding deviance is 95. Then I prune the tree into size = 5. It shows the classification rate increases from 60% to 65.5%. In the real word data analysis, I guess it may happen a lot. Classification tree is not suitable for every classification data set.

# Chapter 3 Comparison

In this chapter, I would like to try different classification method on my stock data to see the difference on the performance output. I use exactly same 7 x-variables and

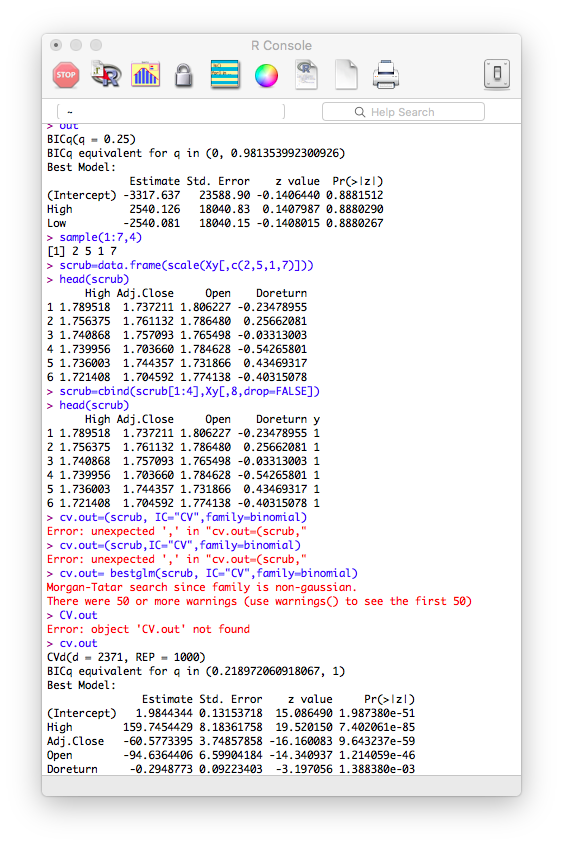
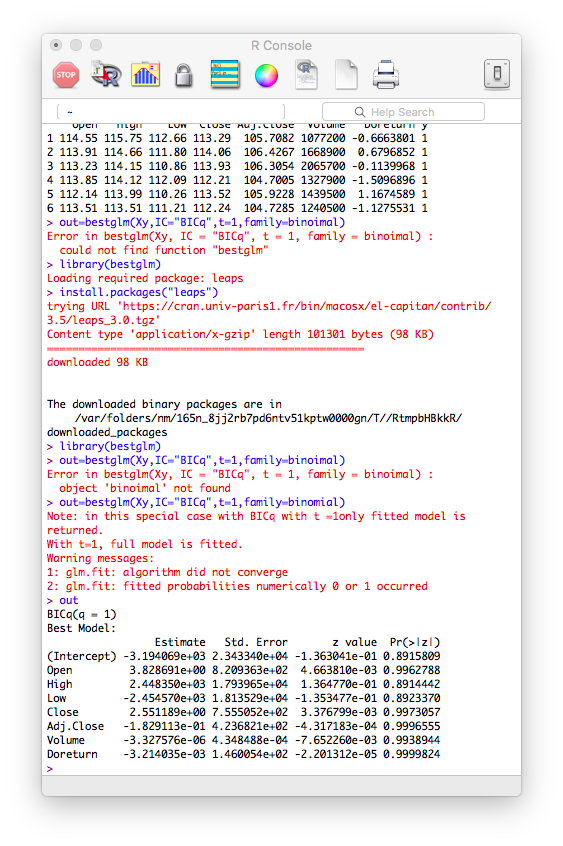


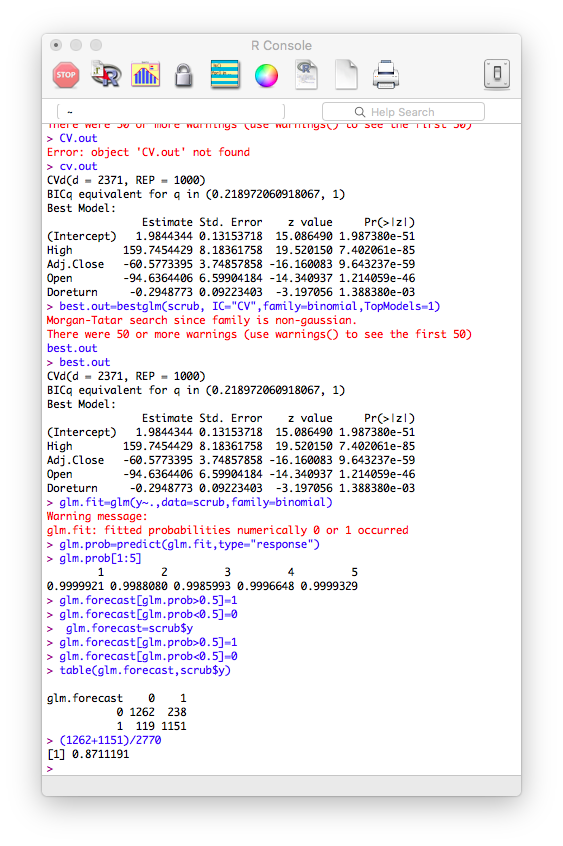
Firstly, let’s see the output of Naïve Bayes method. The correct classification rate is 77%.



It seems KNN does not suit for this stock application. It requires some necessary transformations on the original data, then we can apply knn on it.

Then I would like to use logistic regression to apply the classification. The *bestglm()* function is trying to fit all x-variables into a model. The initial output is not desired. The p-values of every variable is close to 1. So, I need to clean the data. I randomly select 4 column of sock data to group a new data set. Then I use to the new



*bestglm()* function to do cross validation to get the output. We can see that, for this time, we can keep all four variables in the model. Then we compute the classification rate, which is 87%, which is really close to the output of classification tree method 82.96%

We can see from the outputs of these methods and find that every method has its unique advantages and disadvantages. A data set may be qualified to adopt several methods. A method may have strong superiority on some areas. Therefore, I would like to summarize all the classification methods we have studied.

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Method | Definition | Assumptions |
| Tree-Based Methods | Classification tree | One field of the classification and regression tree (CART), the decision tree algorithm. When the predicted outcome is the classification field to which the data belongs to.[[11]](#footnote-11) | Classification features can be both discrete and continuous; logical decisions summarized by tree can predict the class label; tree pruning to achieve a small tree with low error. [[12]](#footnote-12) |
|  | KNN Classification | A non-parametric methodology adopted for both classification and regression problem.[[13]](#footnote-13) | No assumption made about the decision boundary |
| Linear Discriminant Analysis | Naïve bays Classification | A classification method based on Bayes’ Theorem which assumes every variable is independent with each other. [[14]](#footnote-14) | Observations follows a Gaussian distribution; Linear decision boundary |
| Logistic Regression | Binary Classification | Classify the observation into two subsets based on the classification rule. (only two outcomes: 0 or 1) | Binomial distribution; Linear decision boundary; require conditional distribution of Y given X |
| Logistic Regression | Best subset selection | An exploratory model building regression and classification analysis | Normal distribution; sample estimators are unbiased to estimate the population parameters |

|  |  |  |
| --- | --- | --- |
| Method | Advantages | Disadvantages |
| Classification tree | Easy to interpret; graphical display; suitable for qualitative response | Lack of predictive accuracy. |
| KNN Classification | No assumptions; effective for large training data | Determination of parameter K |
| Naïve bays Classification | Simple and fast; less training data | Require conditional distribution of Y given X |
| Logistic Regression for Classification | Easy to fit; simple interpretation of coefficients; test of statistical significance; probabilistic interpretation | Strong assumption about x-variables |
| Best subset selection | Easy to fit; simple interpretation of coefficients; test of statistical significance; probabilistic interpretation | Strong assumption about x-variables |

1. https://www.amazon.com/Classification-Regression-Wadsworth-Statistics-Probability/dp/0412048418 [↑](#footnote-ref-1)
2. ISLR, P311. [↑](#footnote-ref-2)
3. https://www-users.cs.umn.edu/~kumar001/dmbook/ch4.pdf [↑](#footnote-ref-3)
4. https://www-users.cs.umn.edu/~kumar001/dmbook/ch4.pdf [↑](#footnote-ref-4)
5. ISLR, p312 [↑](#footnote-ref-5)
6. https://en.wikipedia.org/wiki/Decision\_tree\_learning#Gini\_impurity [↑](#footnote-ref-6)
7. <http://www.diamondoffshore.com/diamond-offshore-profile> [↑](#footnote-ref-7)
8. <https://gvtimes.com/2018/09/26/why-would-anyone-buy-zendesk-inc-zen-and-diamond-offshore-drilling-inc-do/> [↑](#footnote-ref-8)
9. https://data.cityofnewyork.us/Environment/Harbor-Water-Quality/5uug-f49n [↑](#footnote-ref-9)
10. http://www.nyc.gov/html/dep/html/harborwater/index.shtml [↑](#footnote-ref-10)
11. <https://en.wikipedia.org/wiki/Decision_tree_learning#Decision_tree_types> [↑](#footnote-ref-11)
12. <http://www.cs.princeton.edu/courses/archive/spr07/cos424/scribe_notes/0220.pdf> [↑](#footnote-ref-12)
13. <https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm> [↑](#footnote-ref-13)
14. <https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/> [↑](#footnote-ref-14)