Published in DataDrivenInvestor

Arif R
Mar 8, 2020 · 10 min read · Listen

Get unlimited access

Search

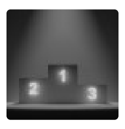
Arif R
179 Followers

Data Scientist and Artificial Intelligence Enthusiast

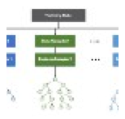
Follow

More from Medium

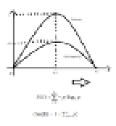
Eligij... in Toward...
Feature Importance in Decision Trees



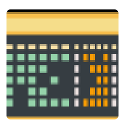
Ibtissam Makdo...
Random forest: Bagging Example



Sumiran
Understanding Decision Trees and Cost...



Jayant
Understanding Naive Bayes Algorithm



The Basics of Decision Trees

Decision Tree Algorithms - Part 1

1. Introduction

Decision Trees is the non-parametric supervised learning approach, and can be applied to both regression and classification problems. In keeping with the tree analogy, decision trees implement a sequential decision process. Starting from the root node, a feature is evaluated and one of the two nodes (branches) is selected, Each node in the tree is basically a decision rule. This procedure is repeated until a final leaf is reached, which normally represents the target. Decision trees are also attractive models if we care about interpretability.

2. Various Decision Tree Algorithms

There are algorithms for creating decision trees :

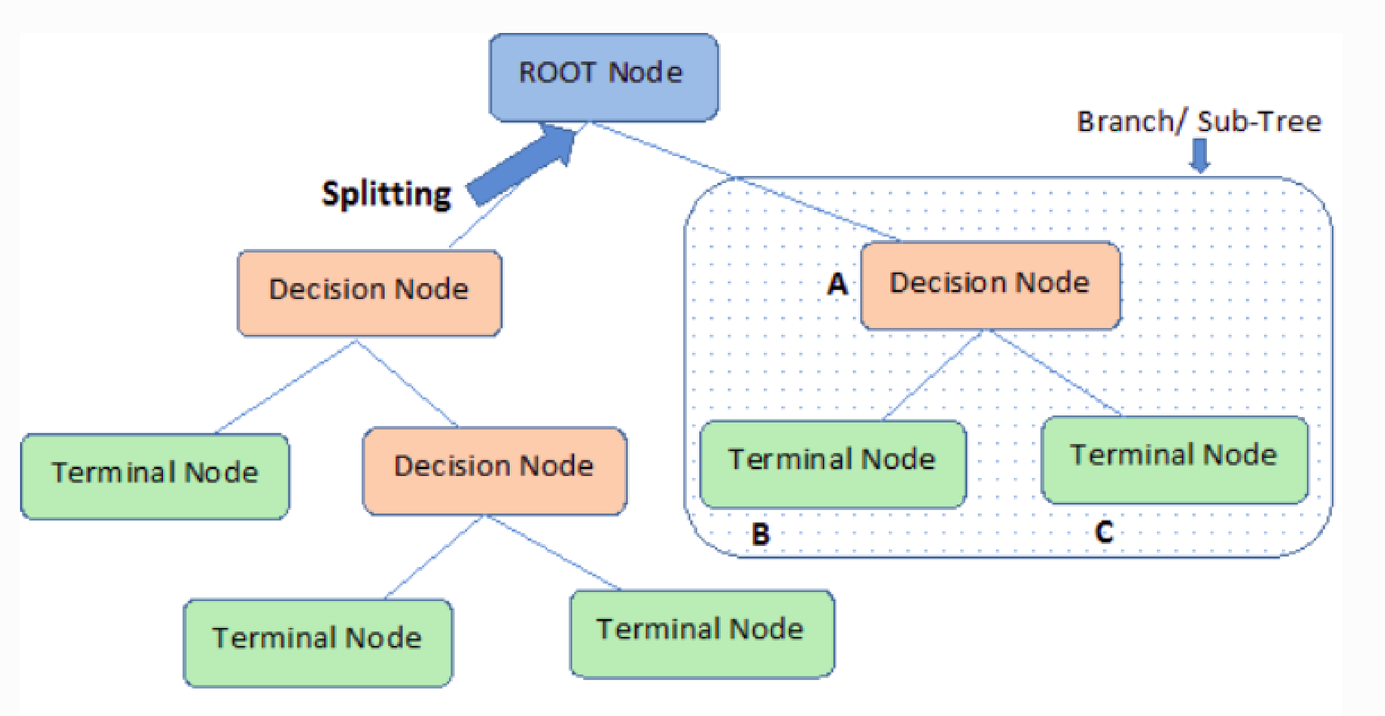
- **ID3 (Iterative Dichotomiser 3)** was developed in 1986 by Ross Quinlan. The algorithm creates a multiway tree, finding for each node (i.e. in a greedy manner) the categorical feature that will yield the largest information gain for categorical targets. Trees are grown to their maximum size and then a pruning step is usually applied to improve the ability of the tree to generalise to unseen data [1].
- **C4.5** was developed in 1993 by Ross Quinlan, is the successor to ID3 and removed the restriction that features must be categorical by dynamically defining a discrete attribute (based on numerical variables) that partitions the continuous attribute value into a discrete set of intervals. C4.5 converts the trained trees (i.e. the output of the ID3 algorithm) into sets of if-then rules. These accuracy of each rule is then evaluated to determine the order in which they should be applied. Pruning is done by removing a rule's precondition if the accuracy of the rule improves without it [1].
- **C5.0** is Quinlan's latest version release under a proprietary license. It uses less memory and builds smaller rulesets than C4.5 while being more accurate [1].
- **CART (Classification and Regression trees)** is very similar to C4.5, but it differs in that it supports numerical target variables (regression) and does not compute rule sets. CART constructs binary trees using the feature and threshold that yield the largest information gain at each node [1].

scikit-learn uses an optimised version of the **CART algorithm**

3. Decision Tree Terminology

In keeping with the tree analogy, the terminology was adopted from the terminology of the tree [2].

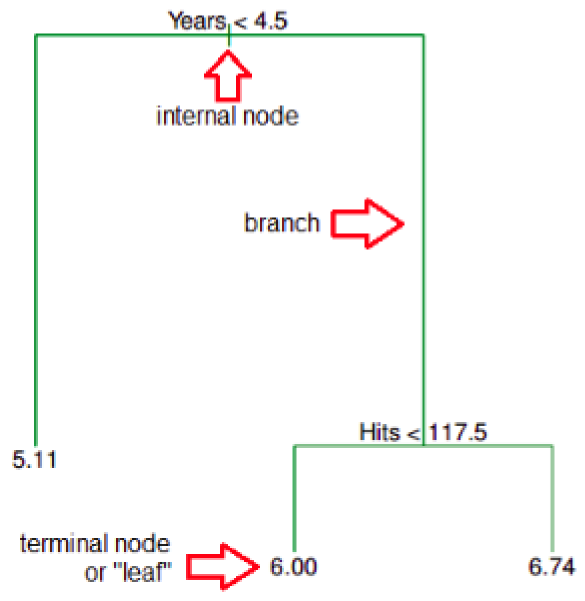
100 1



- **Root node** : is the first node in decision trees
- **Splitting** : is a process of dividing node into two or more sub-nodes, starting from the root node
- **Node** : splitting results from the root node into sub-nodes and splitting sub-nodes into further sub-nodes
- **Leaf or terminal node** : end of a node, since node cannot be split anymore
- **Pruning** : is a technique to reduce the size of the decision tree by removing sub-nodes of the decision tree. The aim is reducing complexity for improved predictive accuracy and to avoid overfitting
- **Branch / Sub-Tree** : A subsection of the entire tree is called branch or sub-tree.
- **Parent and Child Node:** A node, which is divided into sub-nodes is called parent node of sub-nodes whereas sub-nodes are the child of parent node.

4. Decision Tree Intuition

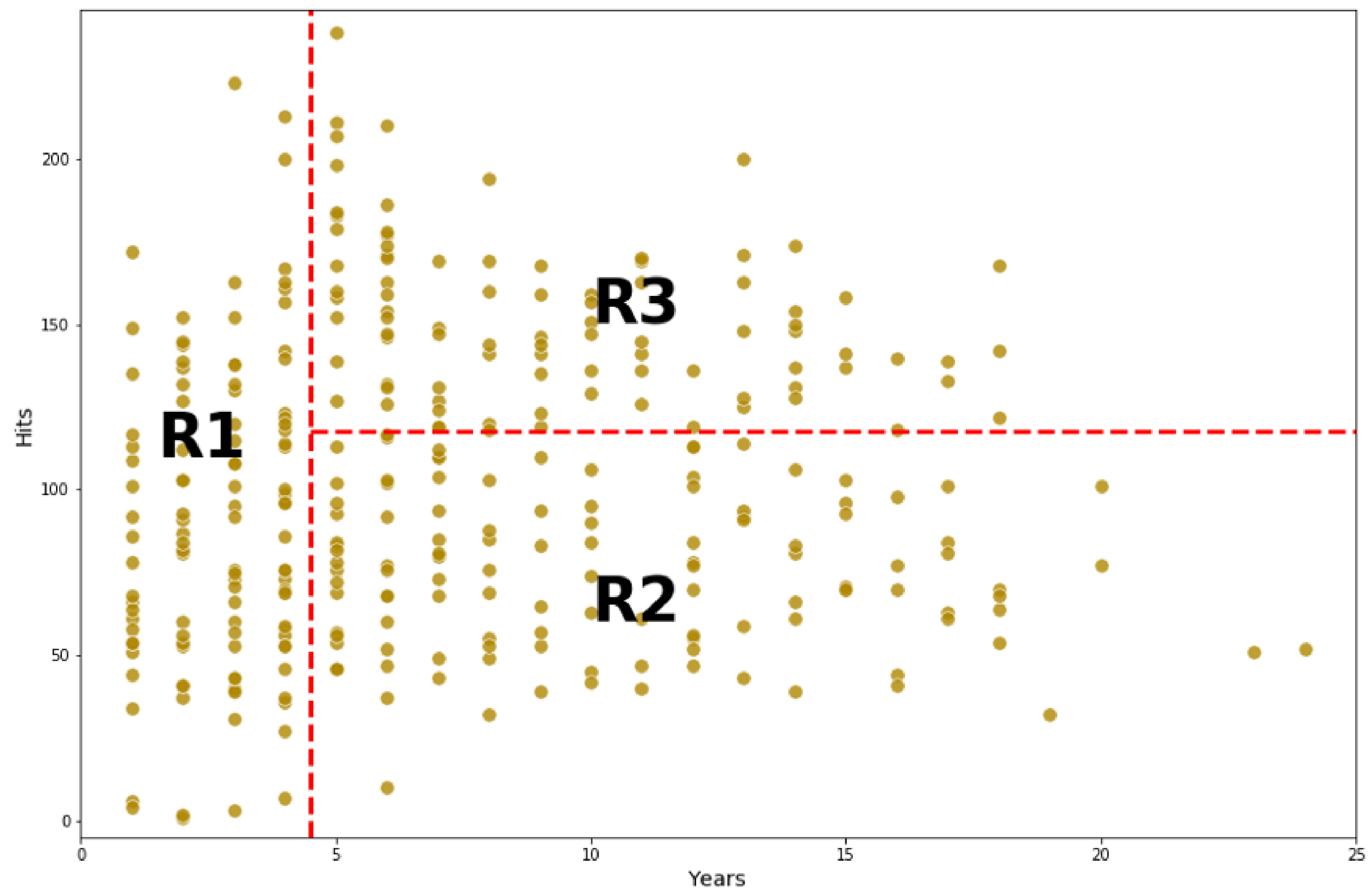
Let's consider the following example where a decision tree to decide tree to predict an salary on a baseball player case :



We use the Hitters data set to predict a baseball player's Salary (mean log salary) based on Years (the number of years that he has played in the major leagues) and Hits (the number of hits that he made in the previous year).

Based on the features, the decision tree model learns a series of splitting rules, starting at the top of the tree (root node).

1. The root node split into sub-node with observation rule having Years <4.5 to the left branch, which means the players in dataset with Years<4.5 having mean log salary is 5.107 and we make a prediction of e5.107 thousands of dollars, i.e. $165,174 for these players
2. Players with Years>=4.5 are assigned to the right branch and then that group is further subdivided by Hits < 177.5 having mean los salary of 6.
3. Players with Years>=4.5 are assigned to the right branch and then that group is further subdivided by Hits >= 177.5 having mean los salary of 6.74



In this case, it can be seen that the decision tree makes a segment into three regions where this region determines the salaries of baseball players and it can be said that the region is a decision boundary [3].

These three regions can be written as

- **R1** ={X | Years<4.5 }
- **R2** ={X | Years>=4.5 ,Hits<117.5 }
- **R3** ={X | Years>=4.5 , Hits>=117.5 }.

From this intuition there is a process, how a decision tree splitting features to form a region that can predict salary of baseball players. This process will be explained in more detail in the next article (Decision Tree Algorithms - Part 2).

5. Splitting in Decision Trees

In order to split the nodes at the most informative features using the decision algorithm, we start at the tree root and split the data on the feature that results in the largest information gain (IG). Here, the objective function is to maximize the information gain (IG) at each split, which we define as follows:

$$IG\,(D_p, f) \;=\; I\,(D_p) \;-\; \sum_{j=1}^{m} \frac{N_j}{N_p}\, I(D_j) \qquad (1)$$

f is the feature to perform the split, *Dp* and *Dj* are data set of the parent, *j*-th child node, *I* is our impurity measure, *Np* is the total number of samples at the parent node, and *Nj* is the number of samples in the *j*-th child node.

As we can see, the information gain is simply the difference between the impurity of the parent node and the sum of the child node impurities — the lower the impurity of the child nodes, the larger the information gain. however, for simplicity and to reduce the combinatorial search space, most libraries (including scikit-learn) implement binary decision trees. This means that each parent node is split into two child nodes, *D-left* and *D-right.*

$$IG\,(D_p, f) \;=\; I\,(D_p) \;-\; \frac{N_{left}}{N_p}\, I(D_{left}) \;-\; \frac{N_{right}}{N_p}\, I(D_{right}) \qquad (2)$$

impurity measure implements binary decisions trees and the three impurity measures or splitting criteria that are commonly used in binary decision trees are *Gini impurity (IG), entropy (IH), and misclassification error (IE)* [4]

5.1 Gini Impurity

According to Wikipedia [5],

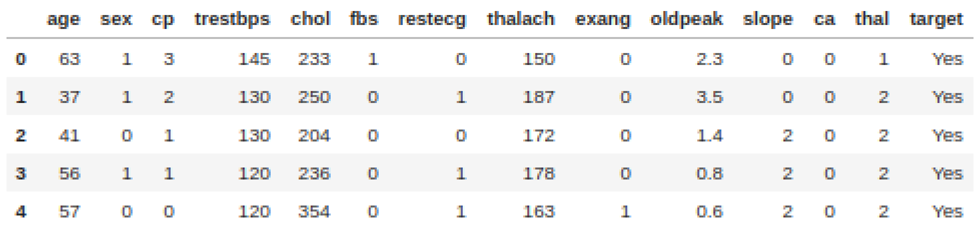
> *Used by the CART (classification and regression tree) algorithm for classification trees, Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset.*

Mathematically, we can write Gini Impurity as following

$$I_{Gini} = 1 - \sum_{i=1}^{j} p_i^2 \qquad (3)$$

where *j* is the number of classes present in the node and *p* is the distribution of the class in the node.

Simple simulation with Heart Disease Data set with 303 rows and has 13 attributes. Target consist 138 value 0 and 165 value 1



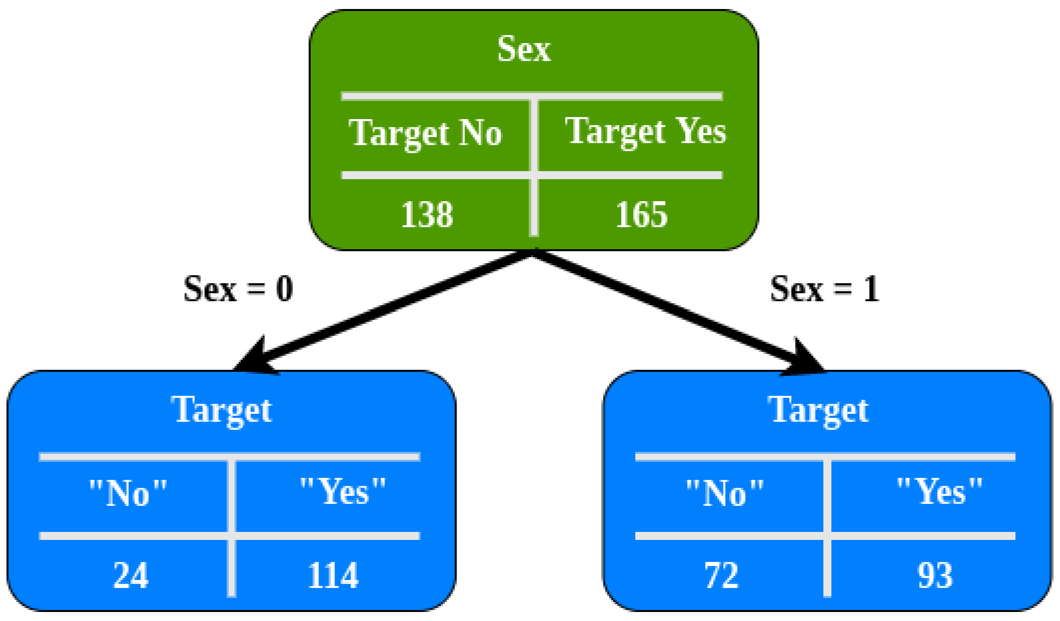
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	Yes
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	Yes
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	Yes
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	Yes
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	Yes

In order to build a decision tree from the dataset and to determine which separation is best, we need a way to measure and compare Gini Impurity in each attribute. The lowest Gini Impurity value on the first iteration will be the Root Node. we can write equation 3 as :

$$I_{Gini} = 1 - (the\ probability\ of\ target\ “No”)^2 - (the\ probability\ of\ target\ “Yes”)^2$$

In this simulation, only use the sex, fbs (fasting blood sugar), exang (exercise induced angina), and target attributes.

How to measure Gini Impurity in Sex attribute



Gini Impurity — Left Node

$$I_{Left-Sex} = 1 - \left(\frac{24}{24+114}\right)^2 - \left(\frac{114}{24+114}\right)^2$$

$$I_{Left-Sex} = 0.29$$

Gini Impurity — Right Node

$$I_{Right-Sex} = 1 - \left(\frac{72}{72+93}\right)^2 - \left(\frac{93}{72+93}\right)^2$$

$$I_{Right-Sex} = 0.49$$

Now that we have measured the Gini Impurity for both leaf nodes. We can calculate the total Gini Impurity with weight average. Left Node represented 138 patient while Right Node represented 165 patient

Total Gini Impurity — Leaf Node

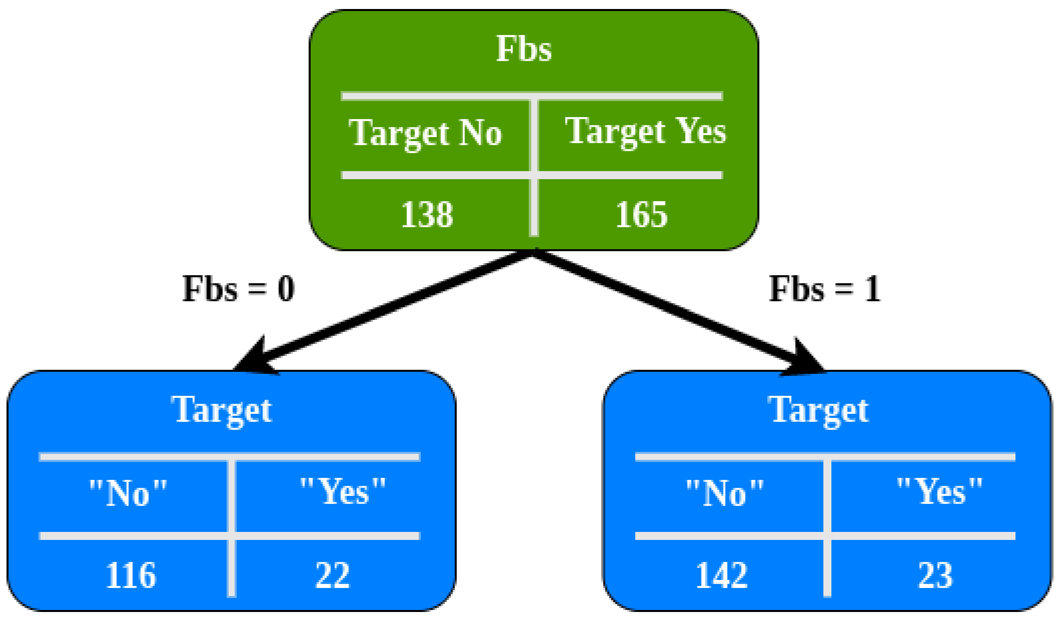
$$I_{Sex} = weight\ average\ of\ the\ leaf\ node\ impurities$$

$$I_{Sex} = \left(\frac{138}{138 + 165}\right) I_{Left-Sex} + \left(\frac{165}{138 + 165}\right) I_{Right-Sex}$$

$$I_{Sex} = \left(\frac{138}{138 + 165}\right) 0.29 + \left(\frac{165}{138 + 165}\right) 0.49$$

$$I_{Sex} = 0.399$$

How to measure Gini Impurity in Fbs (fasting blood sugar) attribute



Gini Impurity — Left Node

$$I_{Left-Fbs} = 1 - \left(\frac{116}{116 + 22}\right)^2 - \left(\frac{22}{116 + 22}\right)^2$$

$$I_{Left-Fbs} = 0.268$$

Gini Impurity — Right Node

$$I_{Right-Fbs} = 1 - \left(\frac{142}{142 + 23}\right)^2 - \left(\frac{23}{142 + 23}\right)^2$$

$$I_{Right-Fbs} = 0.234$$

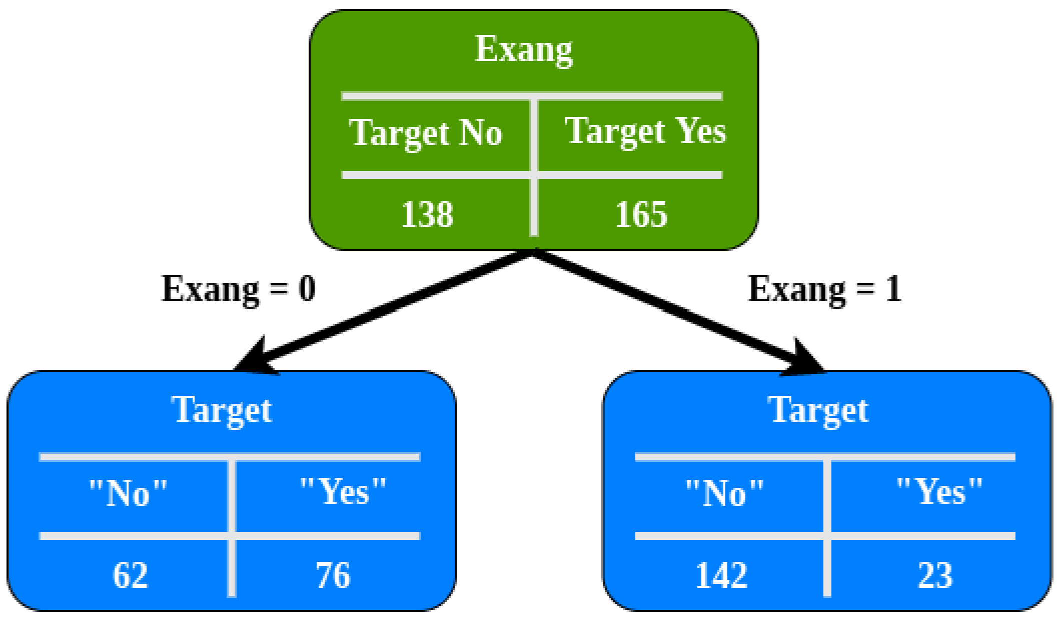
Total Gini Impurity — Leaf Node

$$I_{Fbs} = \left(\frac{138}{138 + 165}\right) I_{Left-Fbs} + \left(\frac{165}{138 + 165}\right) I_{Right-Fbs}$$

$$I_{Fbs} = \left(\frac{138}{138 + 165}\right) 0.268 + \left(\frac{165}{138 + 165}\right) 0.234$$

$$I_{Fbs} = 0.249$$

How to measure Gini Impurity in Exang (exercise induced angina) attribute



Gini Impurity — Left Node

$$I_{Left-Exang} = 1 - \left(\frac{62}{62+76}\right)^2 - \left(\frac{76}{62+76}\right)^2$$

$$I_{Left-Exang} = 0.596$$

Gini Impurity — Right Node

$$I_{Right-Exang} = 1 - \left(\frac{142}{142+23}\right)^2 - \left(\frac{23}{142+23}\right)^2$$

$$I_{Right-Exang} = 0.234$$

Total Gini Impurity — Leaf Node

$$I_{Exang} = \left(\frac{138}{138+165}\right) I_{Left-Exang} + \left(\frac{165}{138+165}\right) I_{-Right\ Exang}$$

$$I_{Exang} = \left(\frac{138}{138+165}\right) 0.596 + \left(\frac{165}{138+165}\right) 0.234$$

$$I_{Exang} = 0.399$$

Fbs (fasting blood sugar) has the lowest Gini Impurity, so well use it at the Root Node

5.2 Entropy

Used by the ID3, C4.5 and C5.0 tree-generation algorithms. Information gain is based on the concept of entropy, the entropy measure is defined as [5]:

$$I_{Entropy} = - \sum_{i=1}^{j} p_i \log_2 p_i$$

where *j* is the number of classes present in the node and *p* is the distribution of the class in the node.

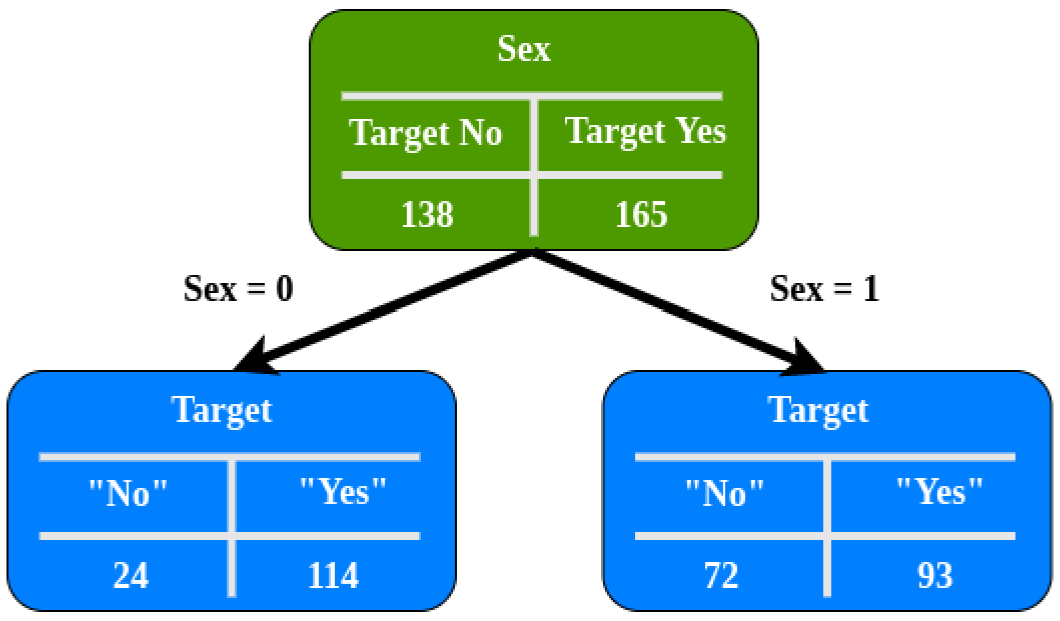
In the same case and same data set, we need a way to measure and compare Entropy in each attribute. The highest Entropy value on the first iteration will be the Root Node.

We need calculate entropy in Target attribute first

$$Entropy_{Target} = -\left(\frac{138}{138+165}\right) \log_2 \left(\frac{138}{138+165}\right) - \left(\frac{165}{138+165}\right) \log_2 \left(\frac{165}{138+165}\right)$$

$$Entropy_{Target} = 0.994$$

How to measure Entropy in Sex attribute



Entropy — Sex = 0

$$Entropy_{\ Sex0} \ = \ -\left(\frac{24}{24+144}\right)\ log_2\left(\frac{24}{24+144}\right)\ -\ \left(\frac{114}{24+144}\right)\ log_2\left(\frac{114}{24+144}\right)$$

$$Entropy_{Sex0} \ = \ 0.666$$

Entropy — Sex = 1

$$Entropy_{Sex1} \ = \ -\left(\frac{72}{72+93}\right)\ log_2\left(\frac{72}{72+93}\right)\ -\ \left(\frac{93}{72+93}\right)\ log_2\left(\frac{93}{24+93}\right)$$

$$Entropy_{Sex1} \ = \ 0.988$$

Now that we have measured the Entropy for both leaf nodes. We take the weight average again to calculate the total entropy value.

Entropy — Sex

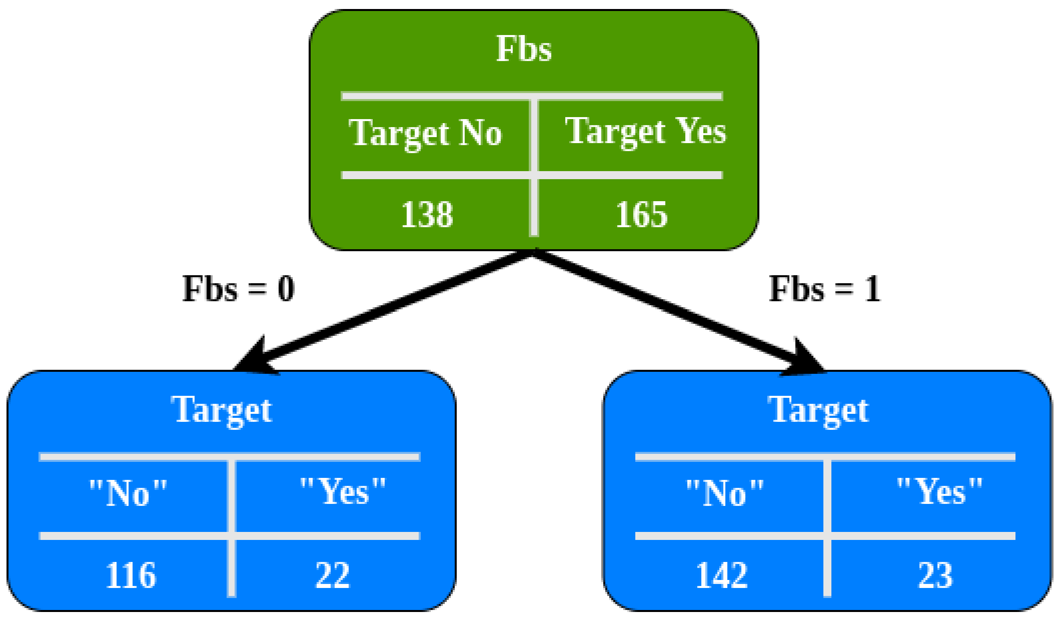
$$Entropy_{\ (Target,\ Sex)} \ = \ Entropy_{Target} \ - \ Entropy_{(Target,\ Sex)}$$

$$Entropy_{\ (Target,\ Sex)} \ = \ 0.994 \ - \ \left[\ \left(\frac{138}{138+165}\right) Entropy_{Sex0} \ + \ \left(\frac{165}{138+165}\right) Entropy_{Sex1}\ \right]$$

$$Entropy_{\ (Target,\ Sex)} \ = \ 0.994 \ - \ \left[\ \left(\frac{138}{138+165}\right) 0.666 \ + \ \left(\frac{165}{138+165}\right) 0.988\right]$$

$$Entropy_{\ (Target,\ Sex)} \ = \ 0.328$$

How to measure Entropy in Fbs attribute



Entropy — Fbs = 0

$$Entropy_{\ Fbs0} \ = \ -\left(\frac{116}{116+22}\right)\ log_2\left(\frac{116}{116+22}\right)\ -\ \left(\frac{22}{116+22}\right)\ log_2\left(\frac{22}{116+22}\right)$$

$$Entropy_{Fbs0} \ = \ 0.632$$

Entropy — Fbs = 1

$$Entropy_{Fbs1} \ = \ -\left(\frac{142}{142+23}\right)\ log_2\left(\frac{142}{142+23}\right)\ -\ \left(\frac{23}{142+23}\right)\ log_2\left(\frac{23}{142+23}\right)$$

$$Entropy_{Fbs1} \ = \ 0.582$$

Entropy — Fbs

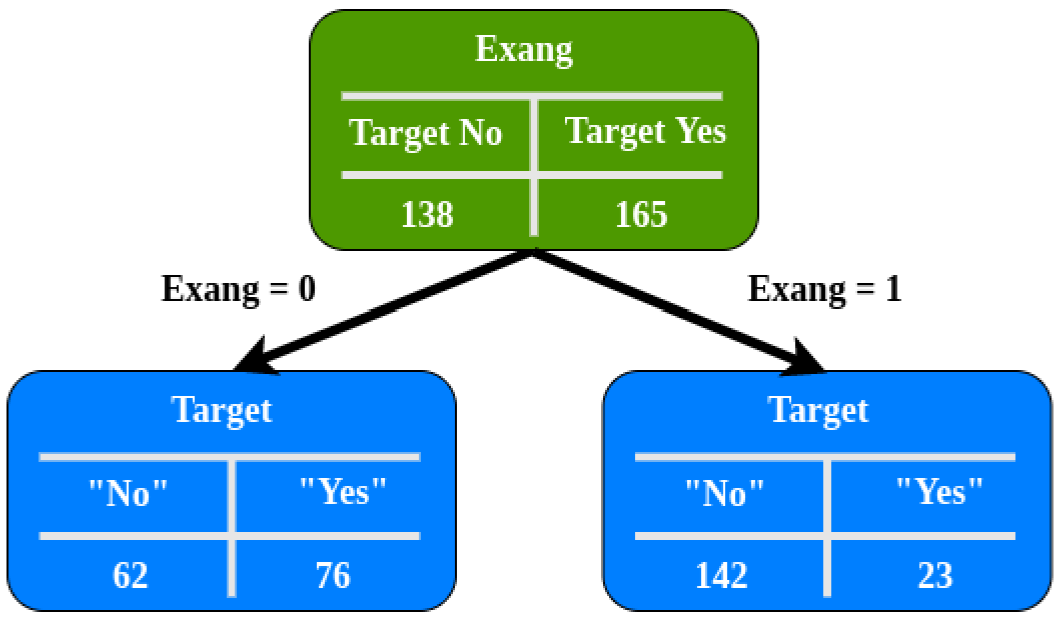
$$Entropy_{\ (Target,\ Fbs)} \ = \ Entropy_{Target} \ - \ Entropy_{(Target,\ Fbs)}$$

$$Entropy_{\ (Target,\ Fbs)} \ = \ 0.994 \ - \ \left[\ \left(\frac{138}{138+165}\right) Entropy_{Fbs0} \ + \ \left(\frac{165}{138+165}\right) Entropy_{Fbs1}\ \right]$$

$$Entropy_{\ (Target,\ Fbs)} \ = \ 0.994 \ - \ \left[\ \left(\frac{138}{138+165}\right) 0.632 \ + \ \left(\frac{165}{138+165}\right) 0.582\right]$$

$$Entropy_{\ (Target,\ Fbs)} \ = \ 0.389$$

How to measure Entropy in Exang attribute



Entropy — Exang = 0

$$Entropy_{Exang0} = -\left(\frac{62}{62+76}\right) \ log_2\left(\frac{62}{62+76}\right) - \left(\frac{76}{62+76}\right) \ log_2\left(\frac{76}{62+76}\right)$$

$$Entropy_{Exang0} = 0.992$$

Entropy — Exang = 1

$$Entropy_{Exang1} = -\left(\frac{142}{142+23}\right) \ log_2\left(\frac{142}{142+23}\right) - \left(\frac{23}{142+23}\right) \ log_2\left(\frac{23}{142+23}\right)$$

$$Entropy_{Exang1} = 0.582$$

Entropy — Exang

$$Entropy_{\ (Target,\ Exang)} = Entropy_{Target} - Entropy_{(Target,\ Exang)}$$

$$Entropy_{\ (Target,\ Exang)} = 0.994 - \left[\left(\frac{138}{138+165}\right) Entropy_{Exang0} + \left(\frac{165}{138+165}\right) Entropy_{Exang1}\right]$$

$$Entropy_{\ (Target,\ Exang)} = 0.994 - \left[\left(\frac{138}{138+165}\right) 0.992 + \left(\frac{165}{138+165}\right) 0.582\right]$$

$$Entropy_{\ (Target,\ Fbs)} = 0.224$$

Fbs (fasting blood sugar) has the highest gini impurity, so we will use it at the Root Node, Precisely the same results we got from Gini Impurity.

5.3 Misclassification Impurity

Another impurity measure is the misclassification impurity , Mathematically, we can write misclassification impurity as following

$$I_{Misclassification} = 1 - \ max \ p(i|j)$$

In terms of quality performance, this index is not the best choice because it's not particularly sensitive to different probability distributions (which can easily drive the selection to a subdivision using Gini or entropy)[6].

6. Advantages and Disadvantages of Trees

Advantages

1. Trees are very easy to explain to people. In fact, they are even easier to explain than linear regression!
2. Some people believe that decision trees more closely mirror human decision-making than do the regression and classification approaches seen in previous chapters.
3. Trees can be displayed graphically, and are easily interpreted even by a non-expert (especially if they are small).
4. Trees can easily handle qualitative predictors without the need to create dummy variables[3].

Disadvantages

1. Unfortunately, trees generally do not have the same level of predictive accuracy as some of the other regression and classification approaches [3].

Continue Learning — Splitting Process in Decision Trees

1. Classification in Decision Tree — A Step by Step CART (Classification And Regression Tree) — Part 2)
2. Regression in Decision Tree — A Step by Step CART (Classification And Regression Tree) — Part 3)

*The scikit-learn package implements the CART as its default decision tree.

About Me

I'm a Data Scientist, Focus on Machine Learning and Deep Learning. You can reach me from Medium and Linkedin

My Website : https://komuternak.com/

Reference

1. Sklearn - Decision Trees
2. https://gdcoder.com/decision-tree-regressor-explained-in-depth/
3. Introduction to Statistical Learning
4. Raschka, Sebastian. Python Machine Learning
5. https://en.wikipedia.org/wiki/Decision_tree_learning
6. Bonaccorso, Giuseppe. Machine Learning Algorithm

Published in Analytics Vidhya

Arif R

Apr 5, 2020 · 5 min read · Listen

Classification in Decision Tree — A Step by Step CART (Classification And Regression Tree)

Decision Tree Algorithms — Part 2

1. Introduction

CART (Classification And Regression Tree) is a decision tree algorithm variation, in the previous article — The Basics of Decision Trees. Decision Trees is the non-parametric supervised learning approach. CART can be applied to both regression and classification problems[1].

As we know, data scientists often use decision trees to solve regression and classification problems and most of them use scikit-learn in decision tree implementation. Based on documentation, scikit-learn uses an optimised version of the CART algorithm

2. How Does CART work in Classification

in the previous article it was explained that CART uses Gini Impurity in the process of splitting the dataset into a decision tree.

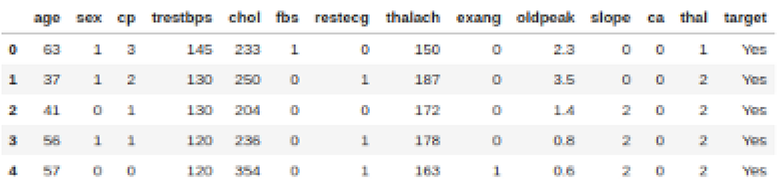
Mathematically, we can write Gini Impurity as following

$$I_{Gini} = 1 - \sum_{i=1}^{j} p_i^2$$

$$I_{Gini} = 1 - (the\ probability\ of\ target\ \text{“}No\text{”})^2 - (the\ probability\ of\ target\ \text{“}Yes\text{”})^2$$

How does CART process the splitting of the dataset

This simulation uses a Heart Disease Data set with 303 rows and has 13 attributes. Target consist 138 value 0 and 165 value 1

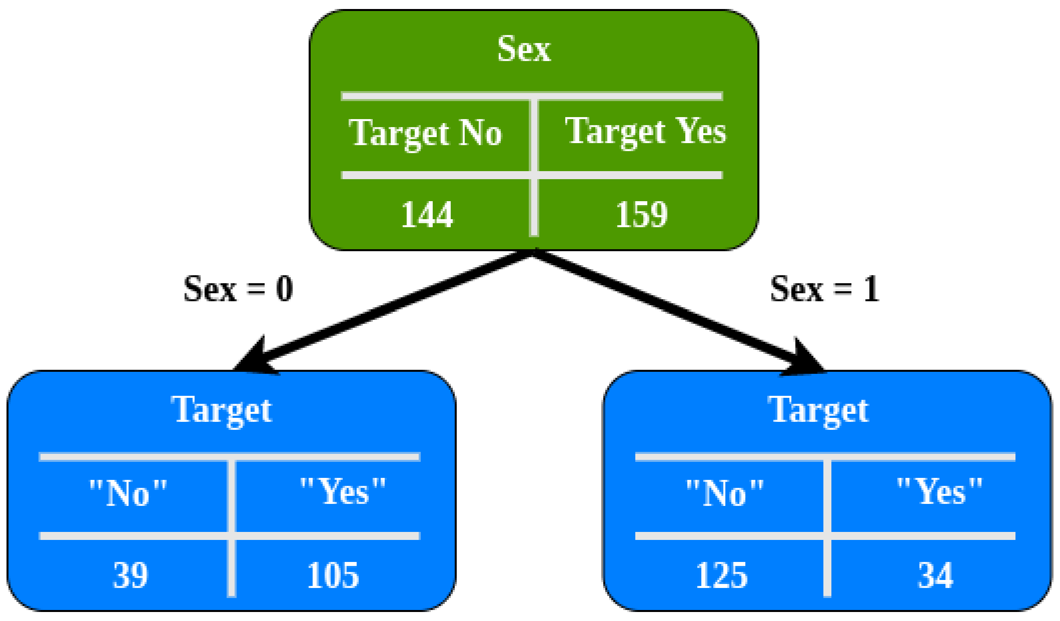


	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	Yes
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	Yes
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	Yes
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	Yes
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	Yes

In this simulation, only use the Sex, Fbs (fasting blood sugar), Exang (exercise induced angina), and target attributes.

Classification

Measure Gini Impurity in Sex



Gini Impurity - Left Node

$$I_{Left} = 1 - \left(\frac{105}{105+39}\right)^2 - \left(\frac{39}{105+39}\right)^2 = 0.395$$

Gini Impurity - Right Node

$$I_{Right} = 1 - \left(\frac{34}{34+125}\right)^2 - \left(\frac{125}{34+125}\right)^2 = 0.336$$

Total Gini Impurity - Leaf Node

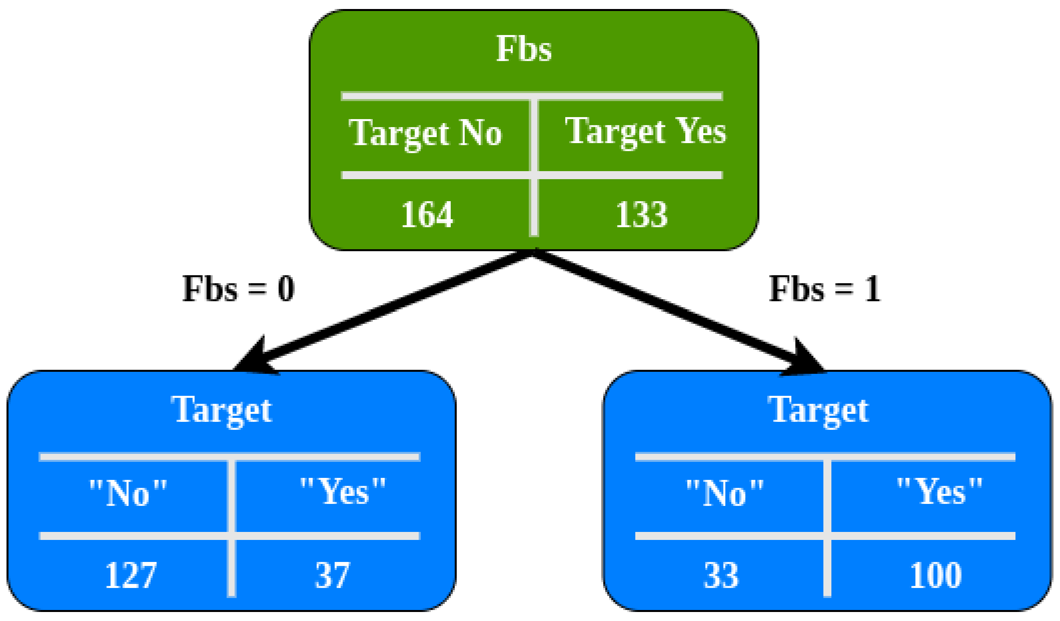
$$I_{Sex} = \textit{weight average of the leaf node impurities}$$

$$I_{Sex} = \left(\frac{144}{144+159}\right) I_{Left} + \left(\frac{159}{144+159}\right) I_{Right}$$

$$I_{Sex} = \left(\frac{144}{144+159}\right) 0.29 + \left(\frac{159}{144+159}\right) 0.49$$

$$I_{Sex} = 0.364$$

Measure Gini Impurity in Fbs (fasting blood sugar)



Gini Impurity - Left Node

$$I_{Left} = 1 - \left(\frac{127}{127+37}\right)^2 - \left(\frac{37}{127+37}\right)^2 = 0.349$$

Gini Impurity - Right Node

$$I_{Right} = 1 - \left(\frac{100}{100+33}\right)^2 - \left(\frac{33}{100+33}\right)^2 = 0.373$$

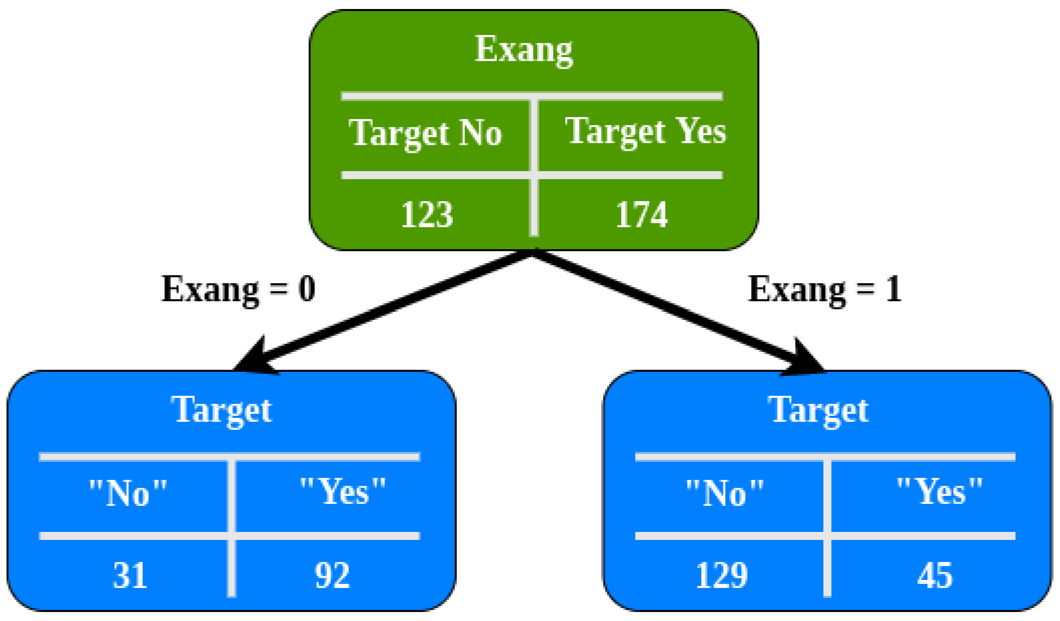
Total Gini Impurity - Leaf Node

$$I_{Fbs} = \left(\frac{164}{164+133}\right) I_{Left} + \left(\frac{133}{164+133}\right) I_{Right}$$

$$I_{Fbs} = \left(\frac{164}{164+133}\right) 0.349 + \left(\frac{133}{164+133}\right) 0.373$$

$$I_{Fbs} = 0.360$$

Measure Gini Impurity in Exang (exercise induced angina)



Gini Impurity - Left Node

$$I_{Left} = 1 - \left(\frac{31}{31 + 92}\right)^2 - \left(\frac{92}{31 + 92}\right)^2 = 0.377$$

Gini Impurity - Right Node

$$I_{Right} = 1 - \left(\frac{129}{129 + 45}\right)^2 - \left(\frac{45}{129 + 45}\right)^2 = 0.383$$

Total Gini Impurity - Leaf Node

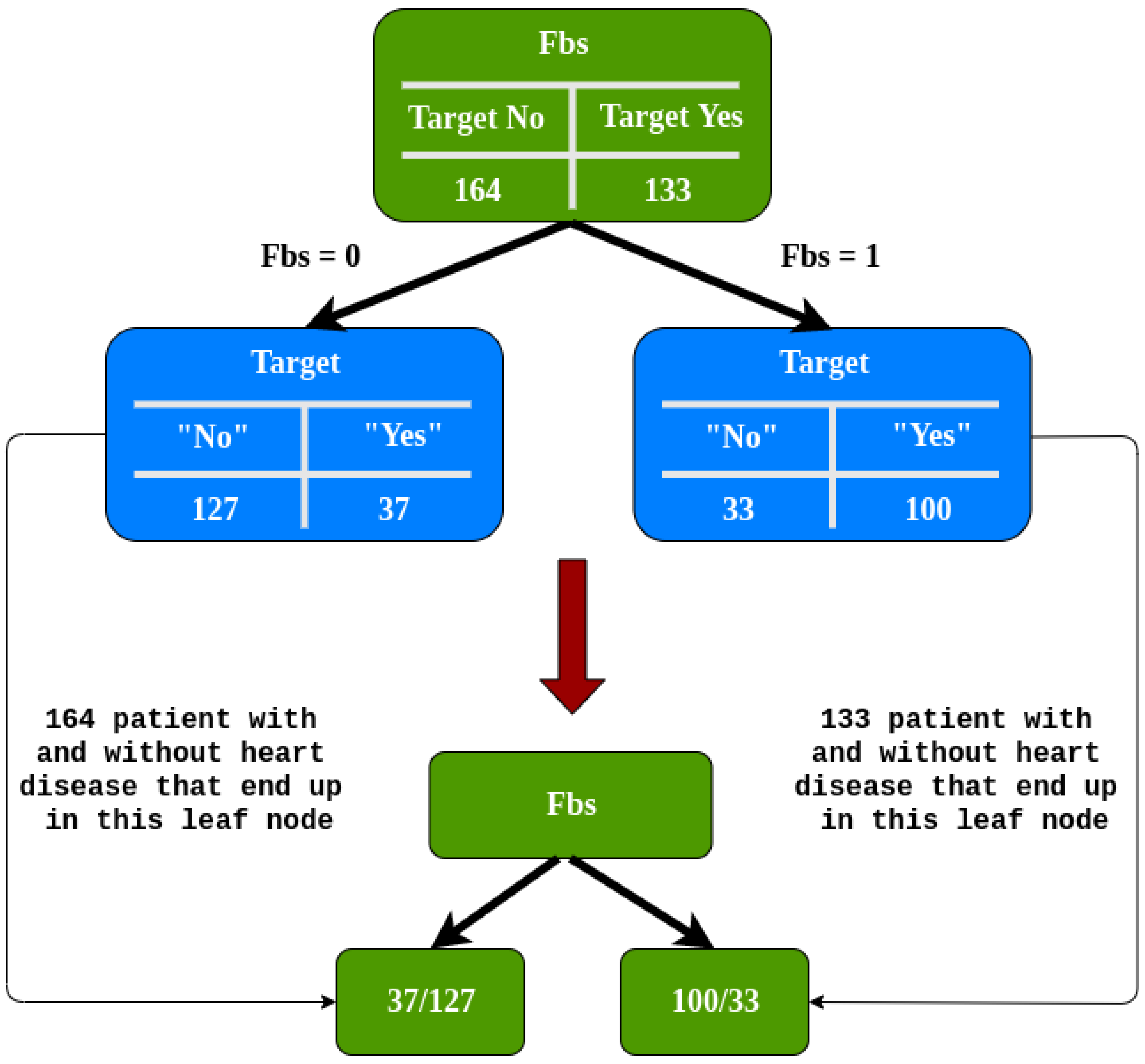
$$I_{Exang} = \left(\frac{123}{123 + 174}\right) I_{Left} + \left(\frac{174}{123 + 174}\right) I_{Right}$$

$$I_{Exang} = \left(\frac{123}{123 + 174}\right) 0.377 + \left(\frac{174}{123 + 174}\right) 0.383$$

$$I_{Exang} = 0.381$$

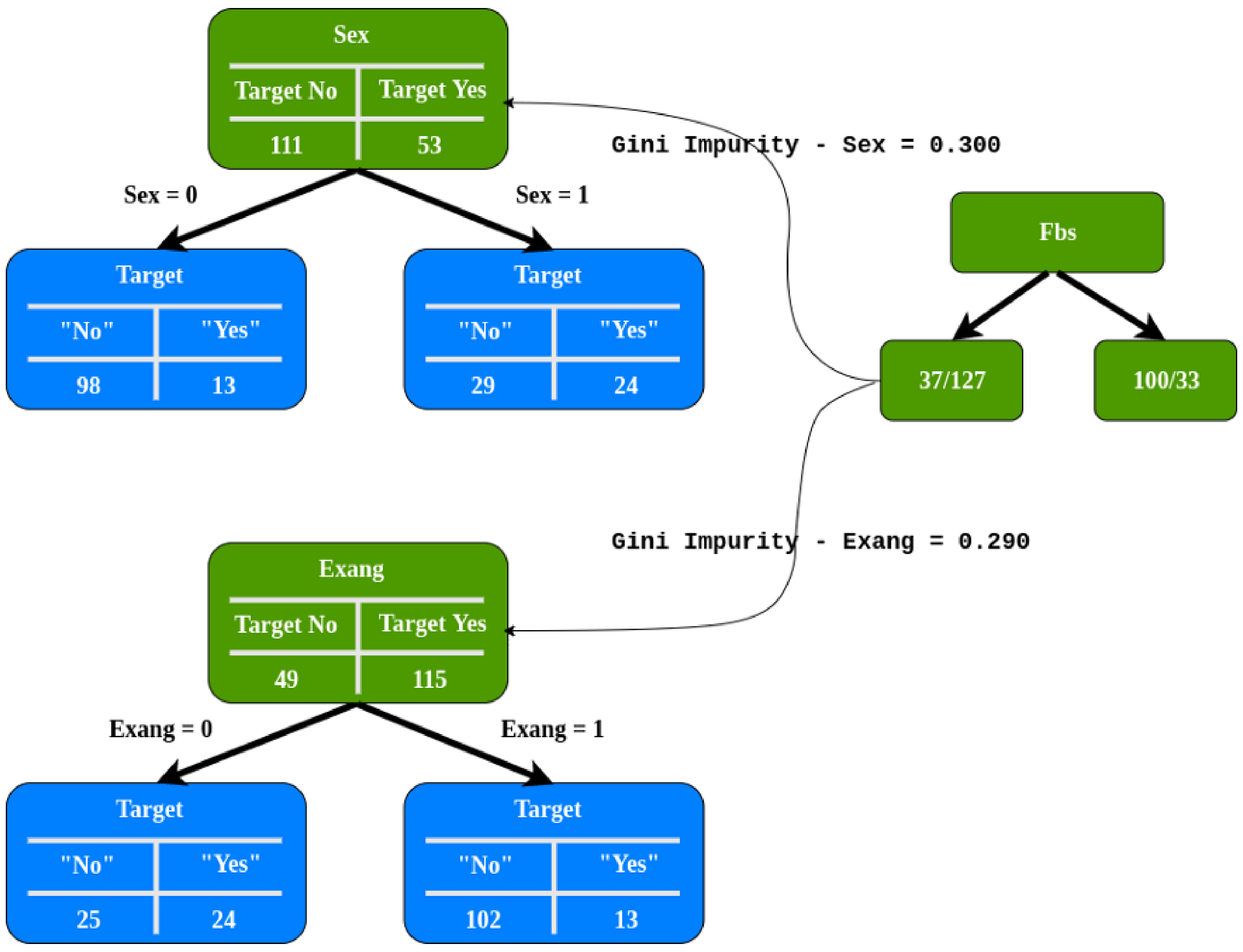
Fbs (fasting blood sugar) has the lowest Gini Impurity, so well use it at the Root Node

As we know, we have Fbs as Root Node, when we divide all of the patients using Fbs (fasting blood sugar), we end up with “Impure” leaf nodes. Each leaf contained with and without heart disease.

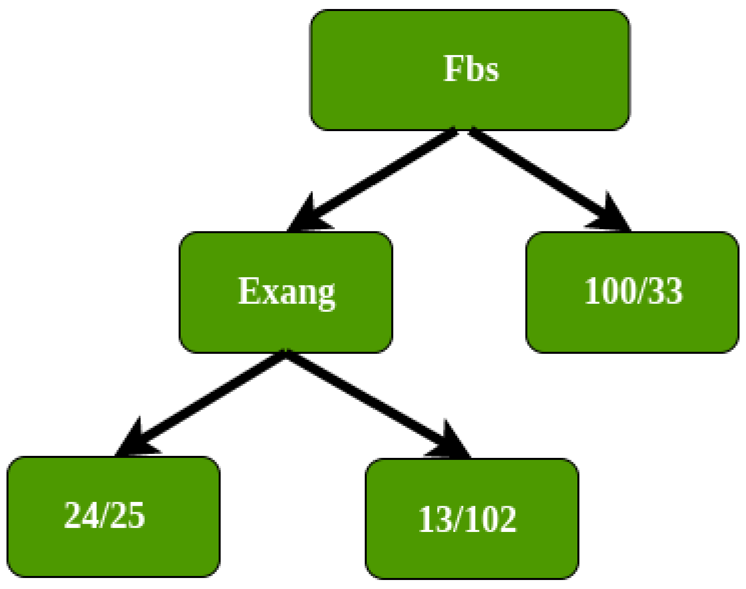


Fbs
Target No
Target Yes
164
133
Fbs = 0
Fbs = 1
Target
"No"
"Yes"
127
37
Target
"No"
"Yes"
33
100
164 patient with and without heart disease that end up in this leaf node
133 patient with and without heart disease that end up in this leaf node
Fbs
37/127
100/33

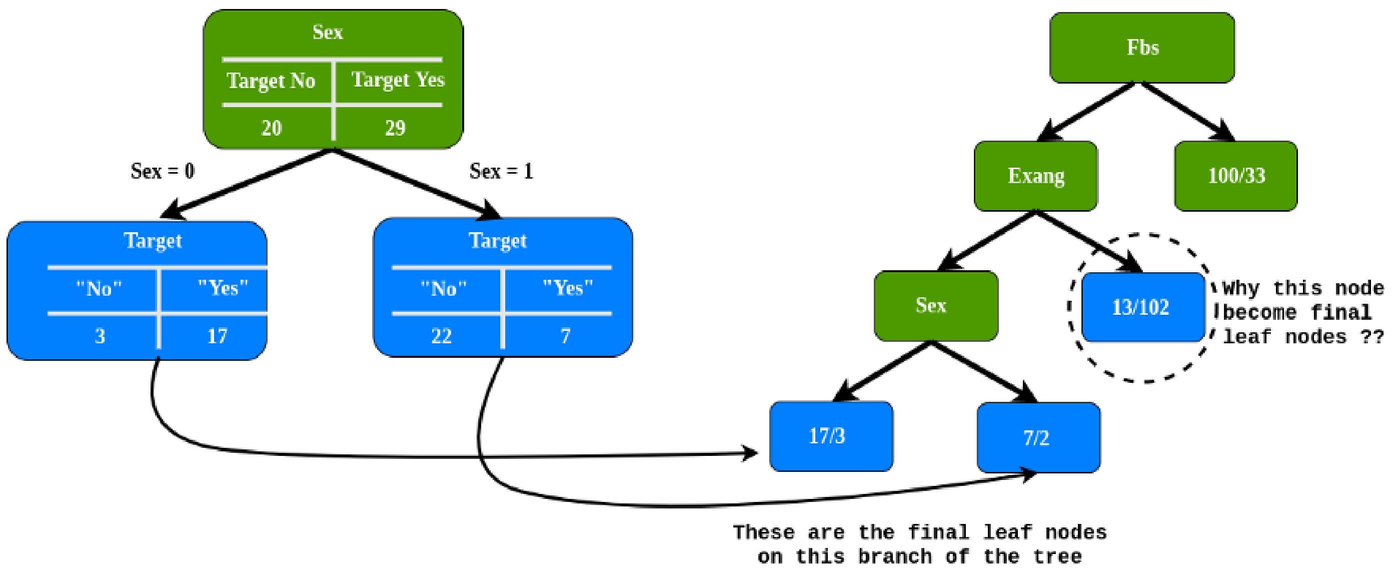
we need to figure how well Sex and Exang separate these patient in left node of Fbs



Exang (exercise induced angina) has the lowest Gini Impurity, we will use it at this node to separate patients.



In the left node of Exang (exercise induced angina), how well it separate these 49 patients (24 with heart disease and 25 without heart disease. Since only the attribute sex is left, we put sex attribute in the left node of Exang

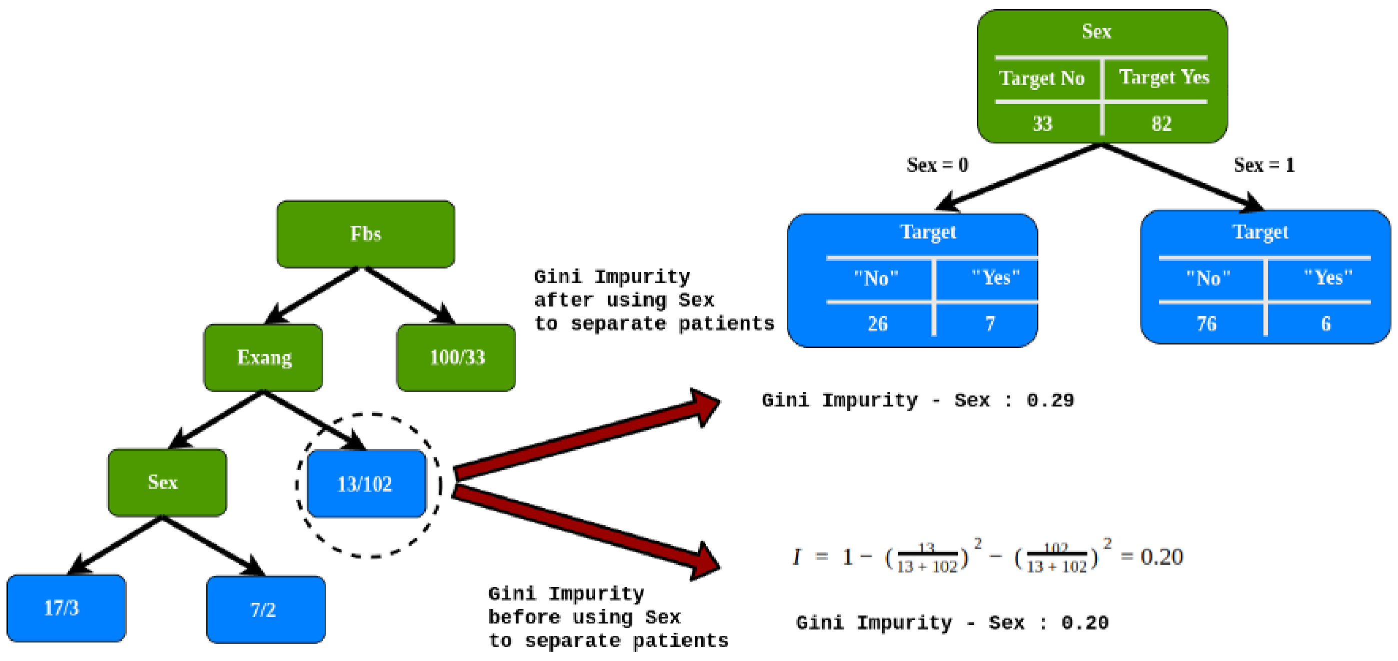


As we can see, we have final leaf nodes on this branch, but why is the leaf node circled including the final node?

Note : the leaf node circled, 89% don't have heart diseases

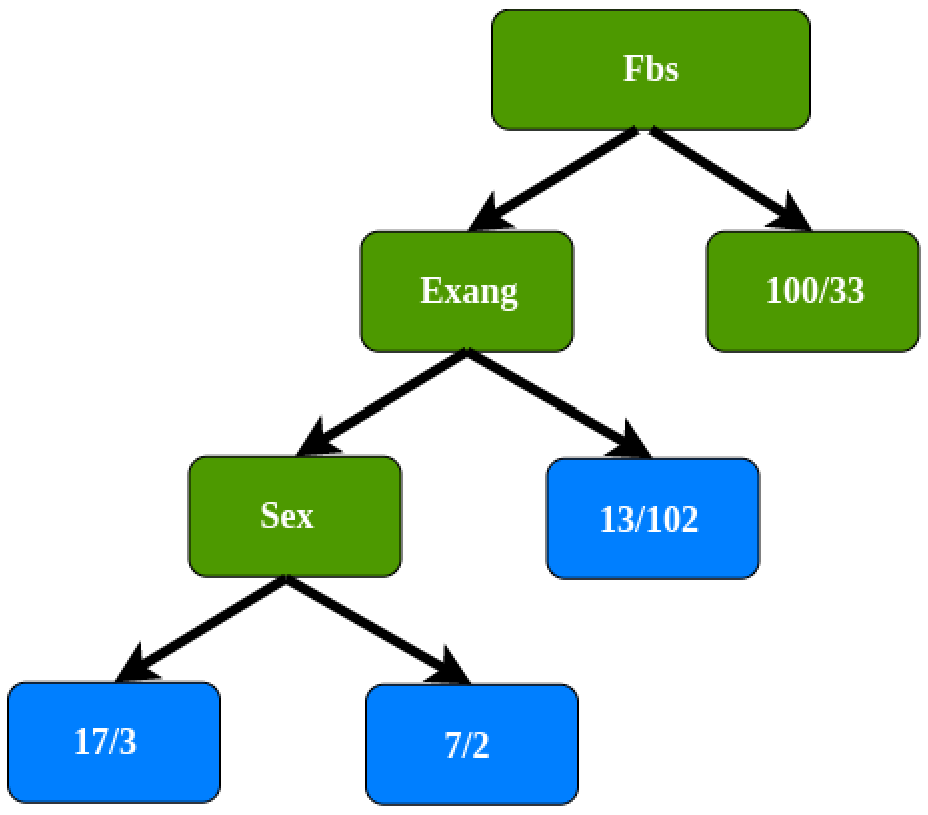
Do these new leaves separate patients better than what we had before ?

In order to answer those question, we must compare Gini Impurity using attribute sex and Gini Impurity before using attribute sex to separate patients.

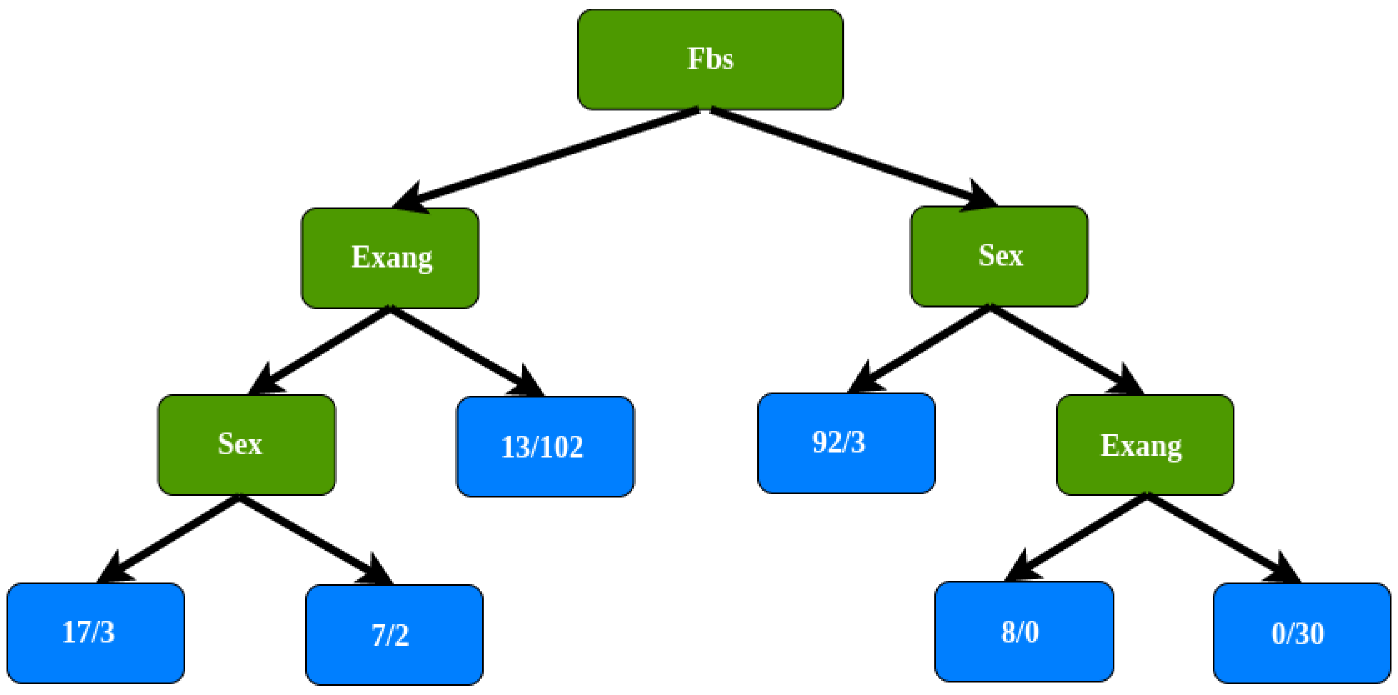

Sex
Target No
Target Yes
33
82
Sex = 0
Sex = 1
Target
"No"
"Yes"
26
7
Target
"No"
"Yes"
76
6
Fbs
Exang
100/33
Sex
13/102
17/3
7/2
Gini Impurity after using Sex to separate patients
Gini Impurity - Sex : 0.29
Gini Impurity before using Sex to separate patients
Gini Impurity - Sex : 0.20

$$I = 1 - \left(\frac{13}{13+102}\right)^2 - \left(\frac{102}{13+102}\right)^2 = 0.20$$

The Gini Impurity before using sex to separate patients is lowest, so we don't separate this node using Sex. The final leaf node on this branch of tree



Do the same thing on the right branch, so the end result of a tree in this case is



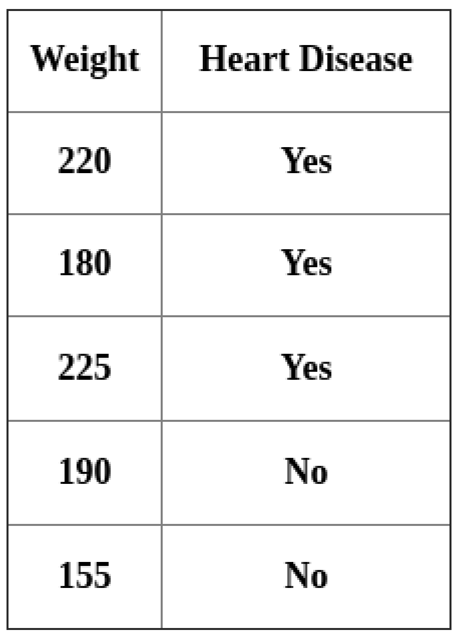
Main point when process the splitting of the dataset

> *1. calculate all of the Gini impurity score*
>
> *2. compare the Gini impurity score, after n before using new attribute to separate data. If the node itself has the lowest score, than there is no point in separating the data*
>
> *3. If separating the data result in an improvement, than pick the separation with the lowest impurity score*

Bonus

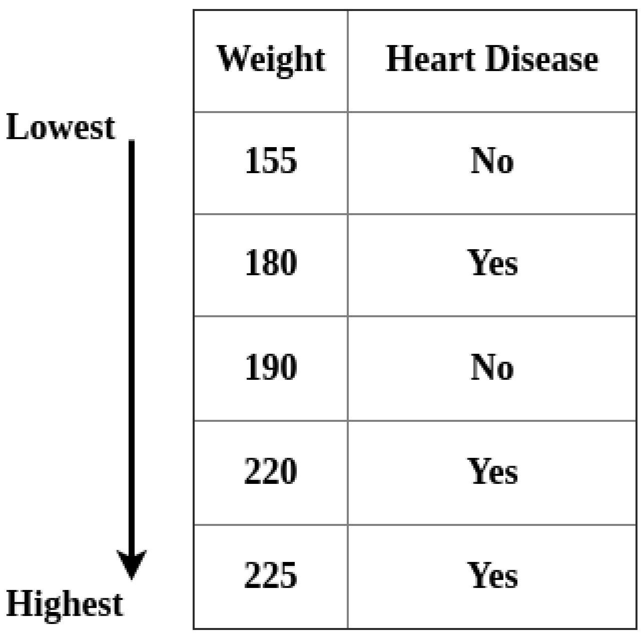
How to calculate Gini Impurity in continuous data?

such as weight which is one of the attributes to determine heart disease, for example we have weight attribute



Weight	Heart Disease
220	Yes
180	Yes
225	Yes
190	No
155	No

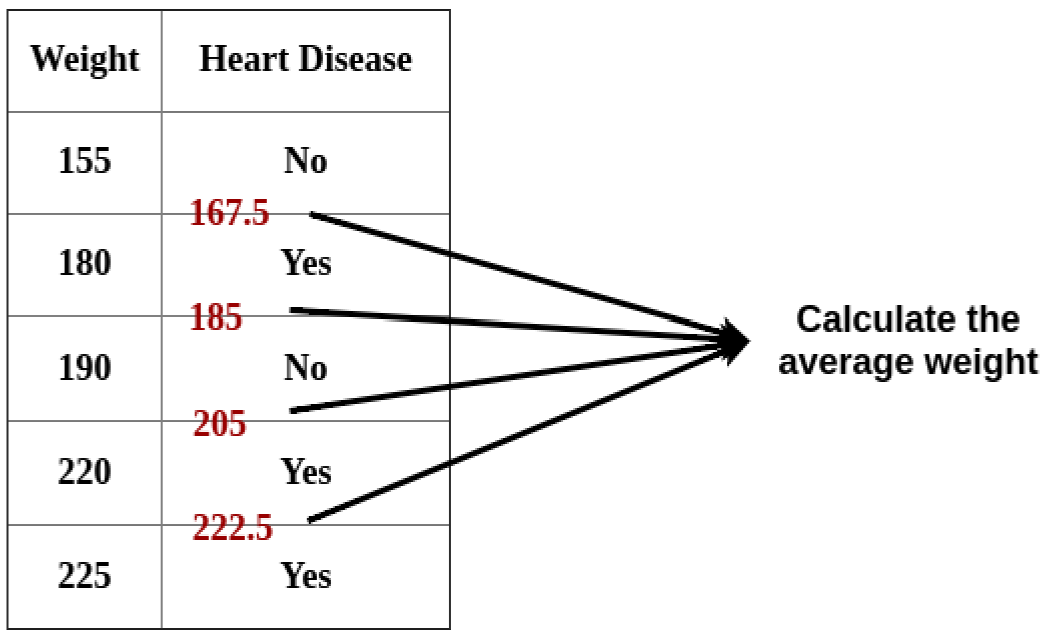
Step 1 : Order data by ascending



Weight	Heart Disease
155	No
180	Yes
190	No
220	Yes
225	Yes

Lowest
Highest

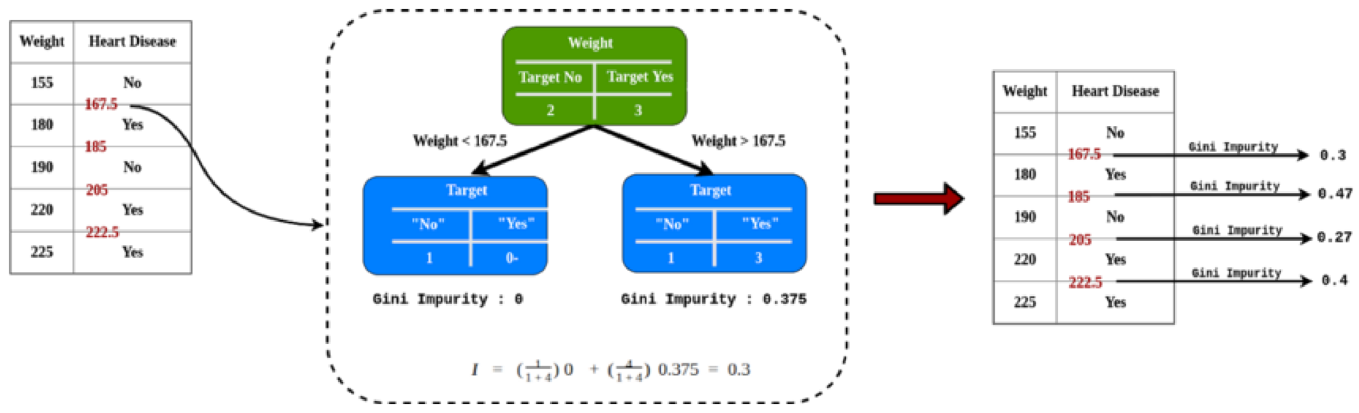
Step 2 : Calculate the average weight



Weight	Heart Disease
155	No
180	Yes
190	No
220	Yes
225	Yes

167.5
185
205
222.5
Calculate the average weight

Step 3 : Calculate Gini Impurity values for each average weight

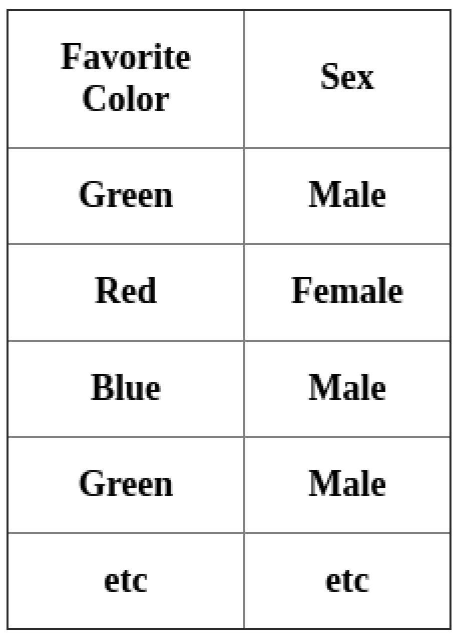


Weight	Heart Disease
155	No
180	Yes
190	No
220	Yes
225	Yes

The lowest Gini Impurity is **Weight < 205,** this is the cutoff and impurity value if used when we compare with another attribute

How to calculate Gini Impurity in categorical data?

we have a favorite color attribute to determine a person's gender



Favorite Color	Sex
Green	Male
Red	Female
Blue	Male
Green	Male
etc	etc

In order to know Gini Impurity this attribute, calculate an impurity score for each one as well as each possible combination

Continue Learning — How Does CART work in Regression

[Regression in Decision Tree — A Step by Step CART (Classification And Regression Tree) — Part 3)]

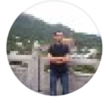
About Me

I'm a Data Scientist, Focus on Machine Learning and Deep Learning. You can reach me from Medium and Linkedin

My Website : https://komuternak.com/

Reference

1. https://medium.com/@arifromadhan19/the-basics-of-decision-trees-e5837cc2aba7
2. Introduction to Statistical Learning
3. Raschka, Sebastian. Pyth
4. https://en.wikipedia.org/wiki/Decision_tree_learning
5. Bonaccorso, Giuseppe. Machine Learning Algorithm
6. *Adapted from YouTube Channel of "StatQuest with Josh Stamer"*



Arif R

May 2, 2020 · 5 min read · Listen

Regression in Decision Tree — A Step by Step CART (Classification And Regression Tree)

Decision Tree Algorithms — Part 3

1. Introduction

In previous learning has been explained about The Basics of Decision Trees and A Step by Step Classification in CART, This section will explain A Step by Step Regression in CART.

As has been explained, Decision Trees is the non-parametric supervised learning approach. In addition to classification with continuous data on the target, we also often find cases with discrete data on the target called regression. In the regression, the simple way can be to use Linear Regression to solve this case. This time the way to solve the regression case will use a decision tree.

For regression trees, two common impurity measures are:

- Least squares. This method is similar to minimizing least squares in a linear model. Splits are chosen to minimize the residual sum of squares between the observation and the mean in each node.
- Least absolute deviations. This method minimizes the mean absolute deviation from the median within a node. The advantage of this over least squares is that it is not as sensitive to outliers and provides a more robust model. The disadvantage is in insensitivity when dealing with data sets containing a large proportion of zeros [1].

Note : mostly people implement regression case with scikit-learn library, Based on documentation, scikit-learn uses an optimised version of the CART algorithm

2. How Does CART Work in Regression with one predictor?

CART in classification cases uses Gini Impurity in the process of splitting the dataset into a decision tree. On the other hand CART in regression cases uses least squares, intuitively splits are chosen to minimize the **residual sum of squares** between the observation and the mean in each node. Mathematically, we can write residual as follow

$$\varepsilon_i = y_i - \hat{y}_i \ (1.0)$$

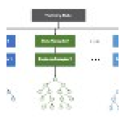
Mathematically, we can write **RSS (residual sum of squares)** as follow

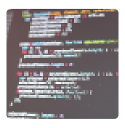
Arif R

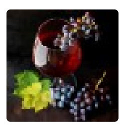
179 Followers

Data Scientist and Artificial Intelligence Enthusiast

Follow







51 | 3 | ...

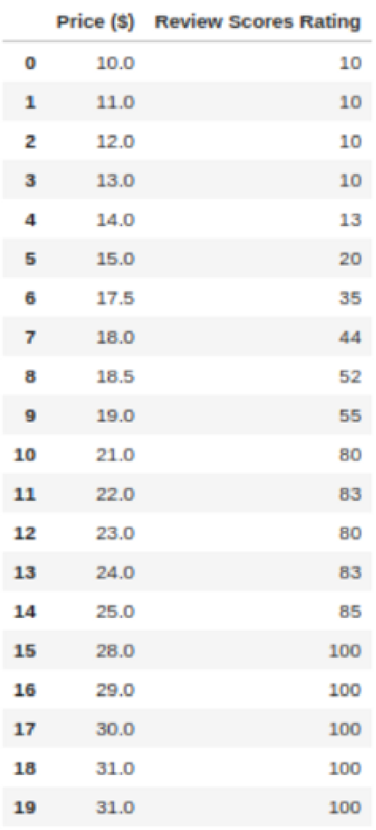
$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad (2.0)$$

$$RSS = \varepsilon_1^2 + \varepsilon_2^2 + .. + \varepsilon_n^2 \quad (2.1)$$

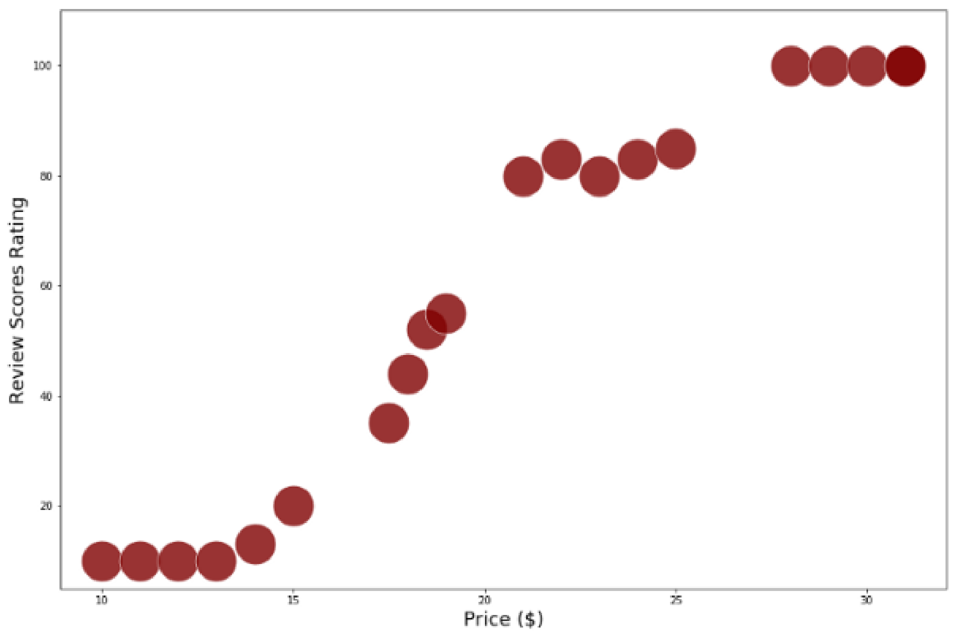
In order to find out the "best" split, we must minimize the RSS

2.1 Intuition

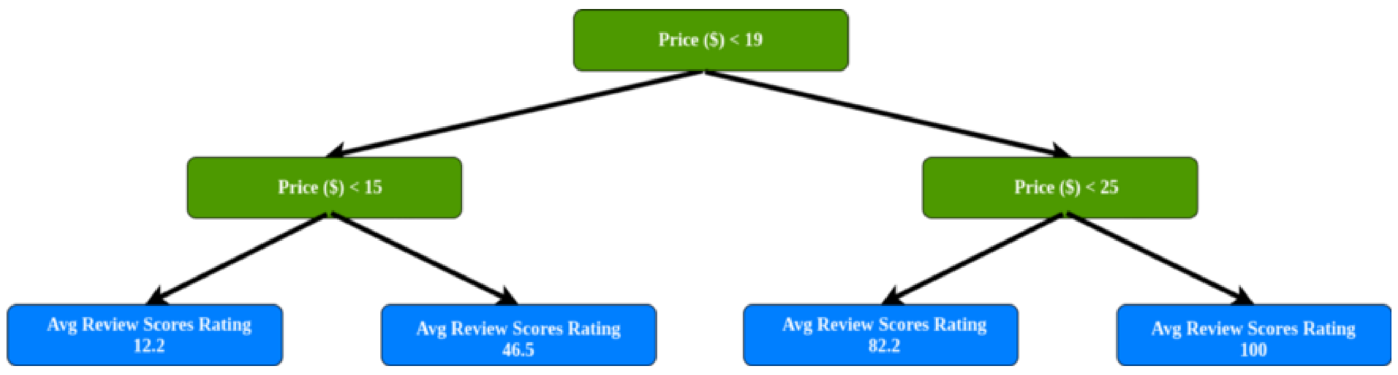
This simulation uses a "dummy" dataset as follow



	Price ($)	Review Scores Rating
0	10.0	10
1	11.0	10
2	12.0	10
3	13.0	10
4	14.0	13
5	15.0	20
6	17.5	35
7	18.0	44
8	18.5	52
9	19.0	55
10	21.0	80
11	22.0	83
12	23.0	80
13	24.0	83
14	25.0	85
15	28.0	100
16	29.0	100
17	30.0	100
18	31.0	100
19	31.0	100



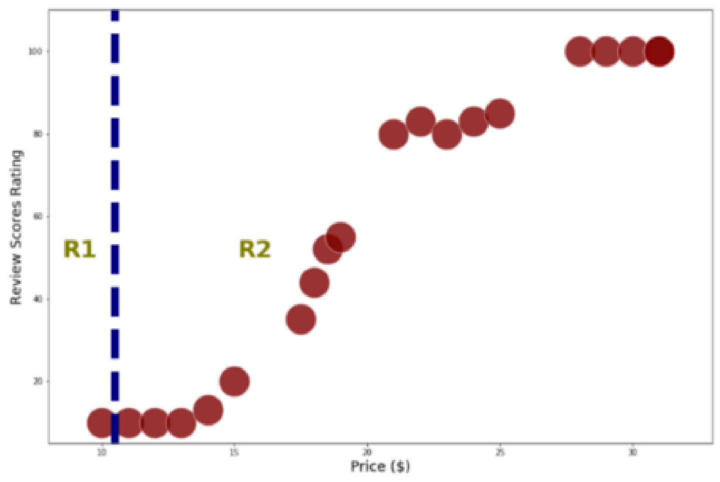
The decision tree as follow

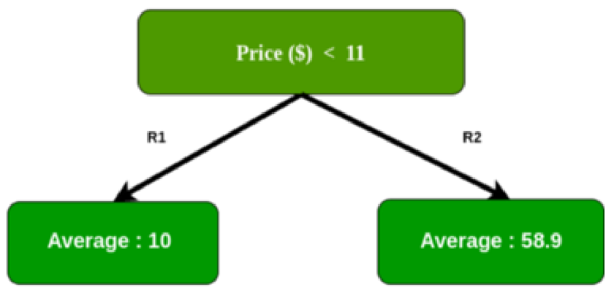

Price ($) < 19
Price ($) < 15
Price ($) < 25
Avg Review Scores Rating 12.2
Avg Review Scores Rating 46.5
Avg Review Scores Rating 82.2
Avg Review Scores Rating 100

2.2 How does CART process the splitting of the dataset (predictor =1)

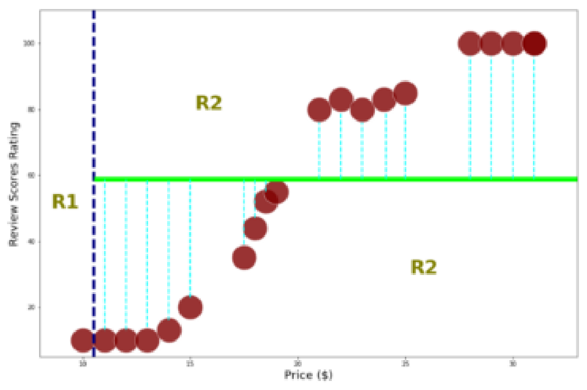
As mentioned before, **In order to find out the "best" split, we must minimize the RSS.** first, we calculate **RSS** by split into two regions, start with index 0

Start within index 0





The data already split into two regions, we add up the squared residual for every index data. furthermore we calculate **RSS** each node using equation 2.0

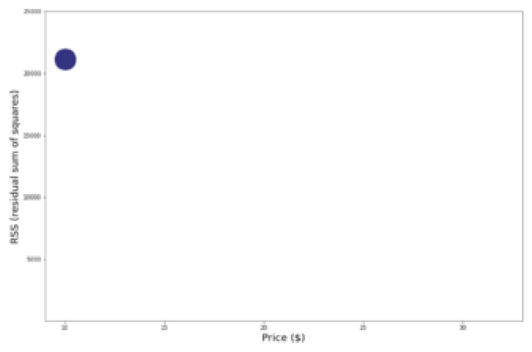


$$\varepsilon = \sum_{i=1}^{n} (actual\ value - average\ value\ in\ each\ region)^2$$

$$RSS = \varepsilon_1^2 + \varepsilon_2^2 + .. + \varepsilon_n^2$$

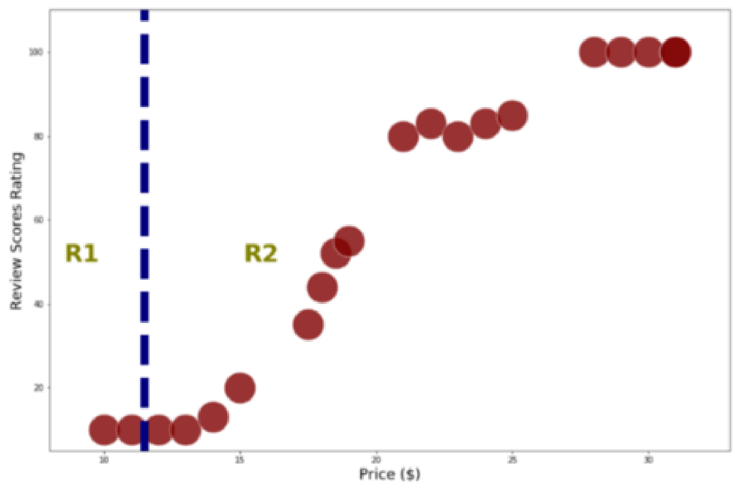
$$RSS = (10 - 10)^2 + (10 - 58.9)^2 + (10 - 58.9)^2 \ldots (10 - 58.9)^2 = 21139.78$$

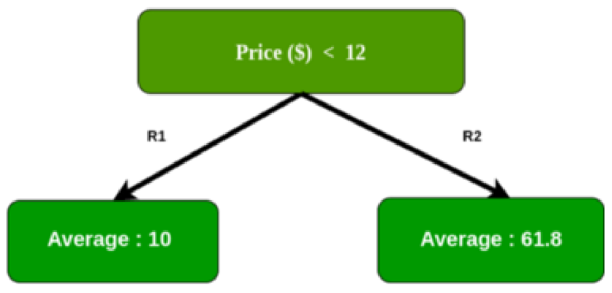
The next step is RSS analysis in graphics as following



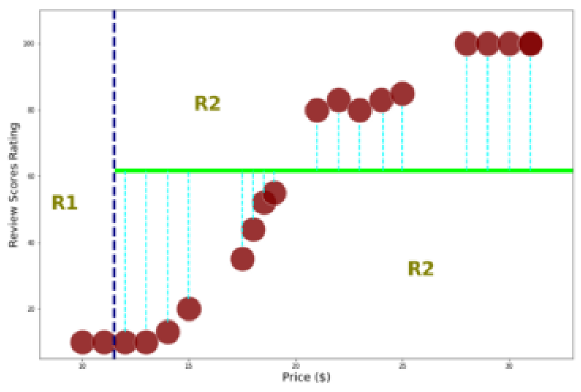
Start within index 1

calculate **RSS** by split into two regions within index 1





after the data is divided into two regions then calculate **RSS** each node using equation 2.0

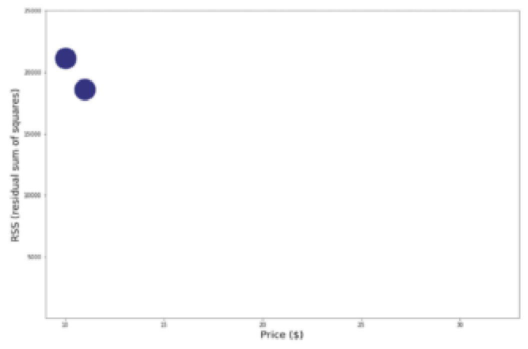


$$\varepsilon = \sum_{i=1}^{n} (actual\ value - average\ value\ in\ each\ region)^2$$

$$RSS = \varepsilon_1^2 + \varepsilon_2^2 + .. + \varepsilon_n^2$$

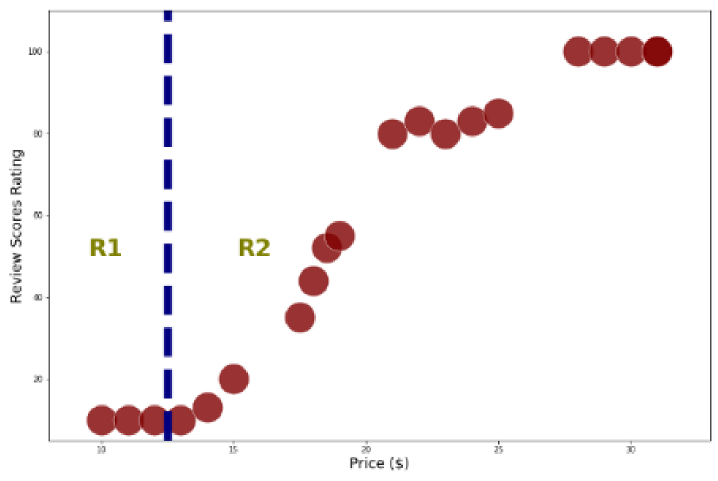
$$RSS = (10 - 10)^2 + (10 - 10)^2 + (10 - 58.9)^2 \ldots (10 - 58.9)^2 = 18609.08$$

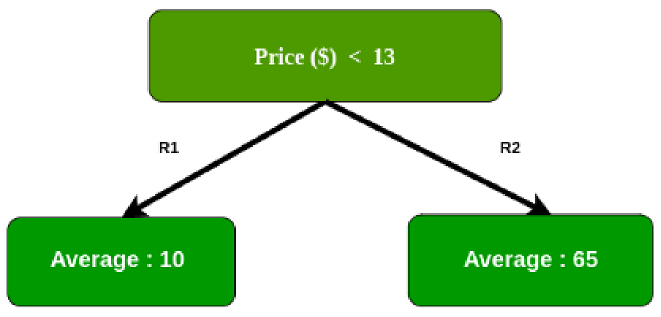
The next step is RSS analysis in graphics as following



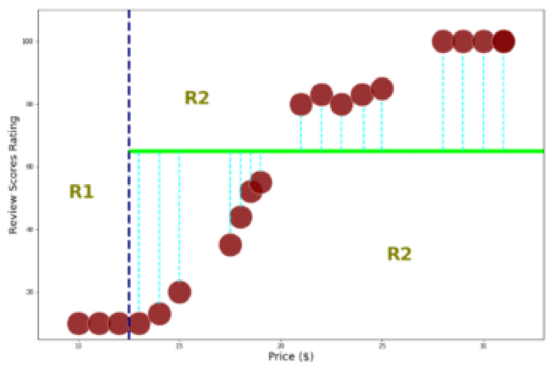
Start within index 2

calculate **RSS** by split into two regions within index 2





calculate **RSS** each node

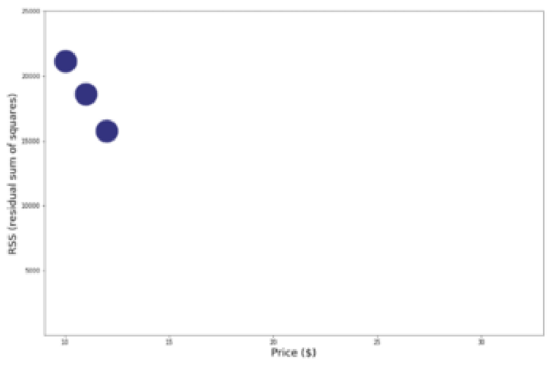


$$\varepsilon = \sum_{i=1}^{n} (actual\ value - average\ value\ in\ each\ region)^2$$

$$RSS = \varepsilon_1^2 + \varepsilon_2^2 + .. + \varepsilon_n^2$$

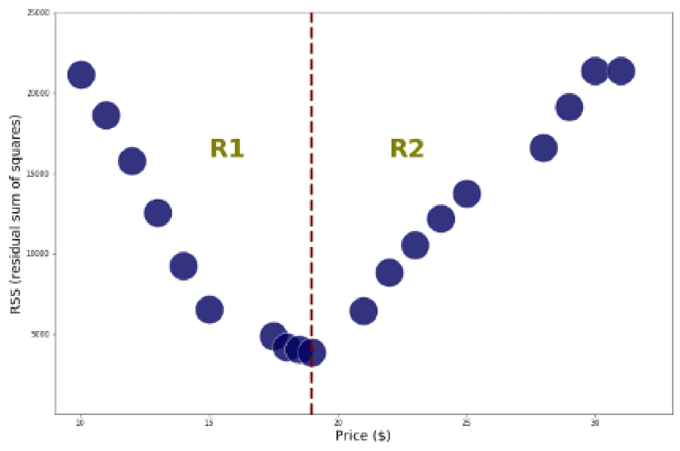
$$RSS = (10 - 10)^2 + (10 - 10)^2 + (10 - 10)^2 ... (10 - 58.9)^2 = 15762.0$$

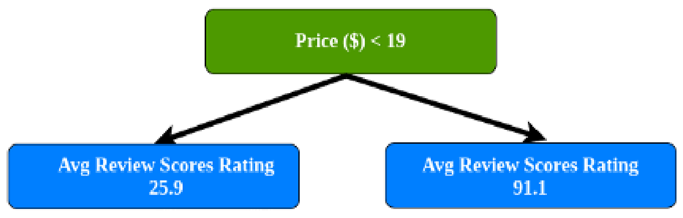
The next step is RSS analysis in graphics as following



This process continues until the calculation of RSS in the last index

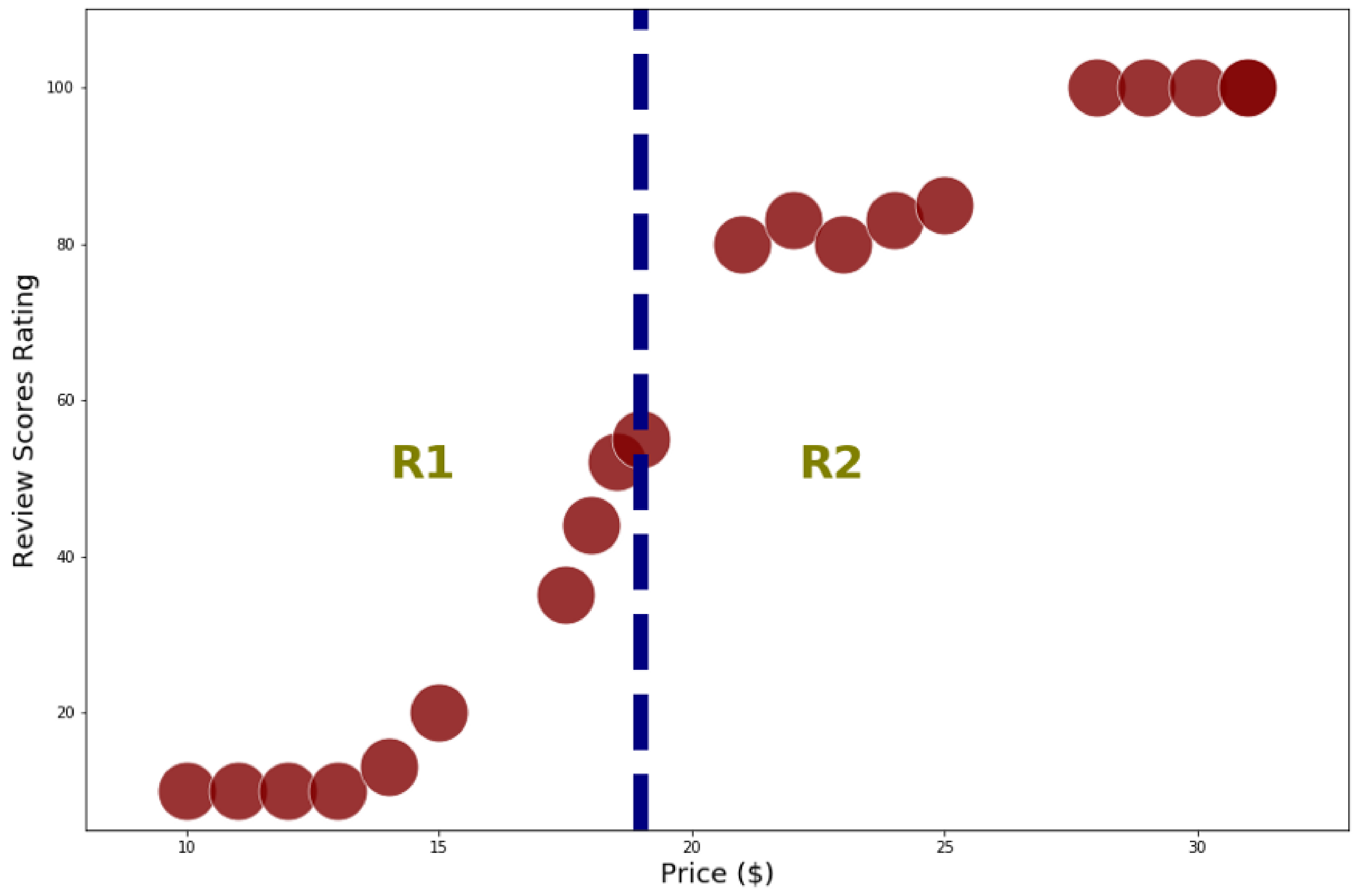
Last Index



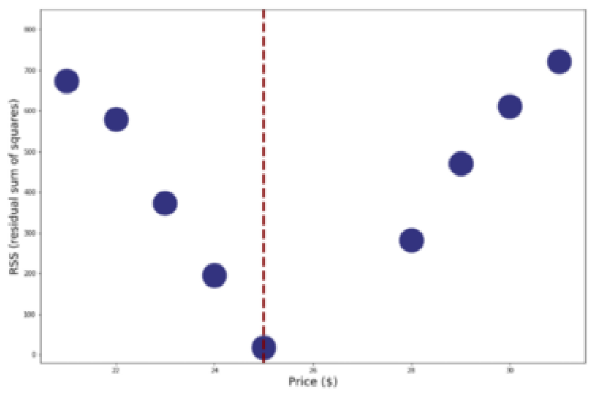


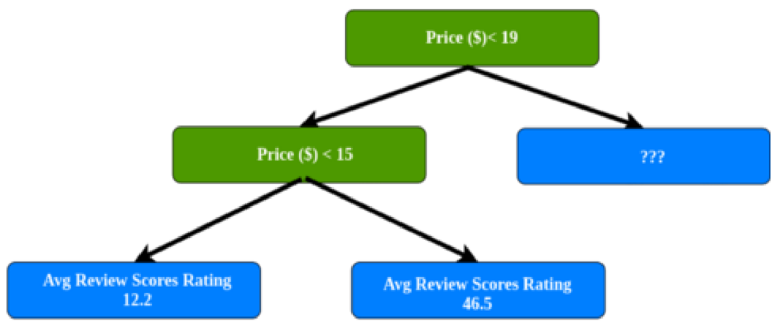
Price with threshold 19 has a smallest RSS, in R1 there are 10 data within price < 19, so we'll split the data in R1. In order to avoid overfitting, we define the minimum data for each region >= 6. If the region has less than 6 data, the split process in that region stops.

Split the data with threshold 19

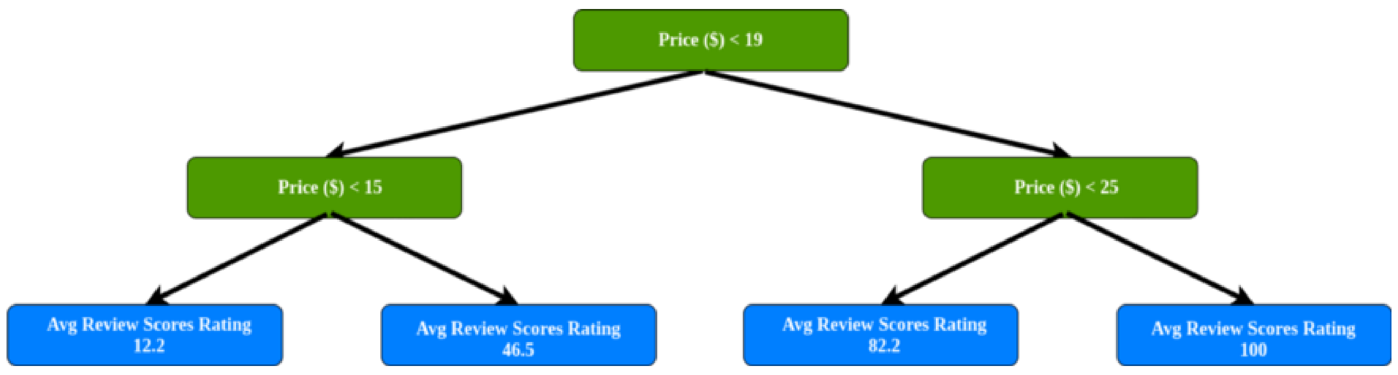


calculate RSS in R1, the process in this section is the same as the previous process, only done for R1



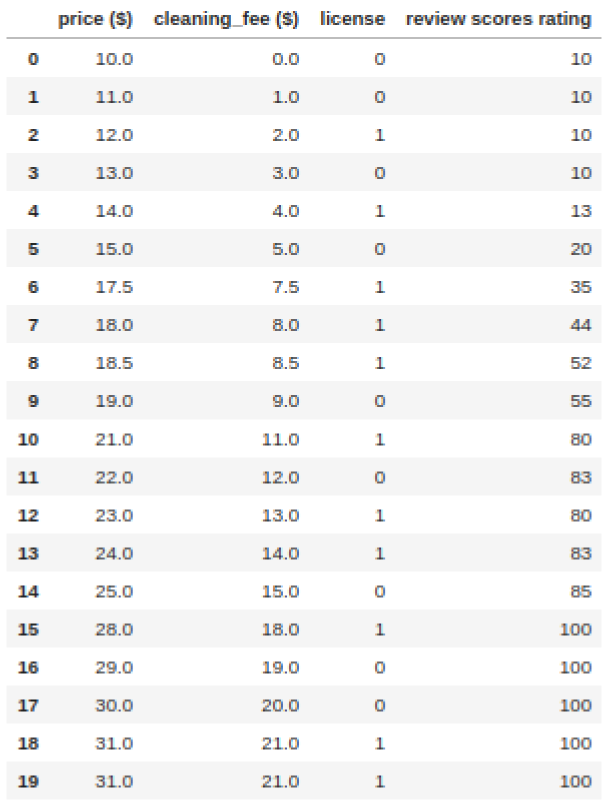


Do the same thing on the right branch, so the end result of a tree in this case is


Price ($) < 19
Price ($) < 15
Price ($) < 25
Avg Review Scores Rating 12.2
Avg Review Scores Rating 46.5
Avg Review Scores Rating 82.2
Avg Review Scores Rating 100

2.3 How does CART process the splitting of the dataset (predictor > 1)

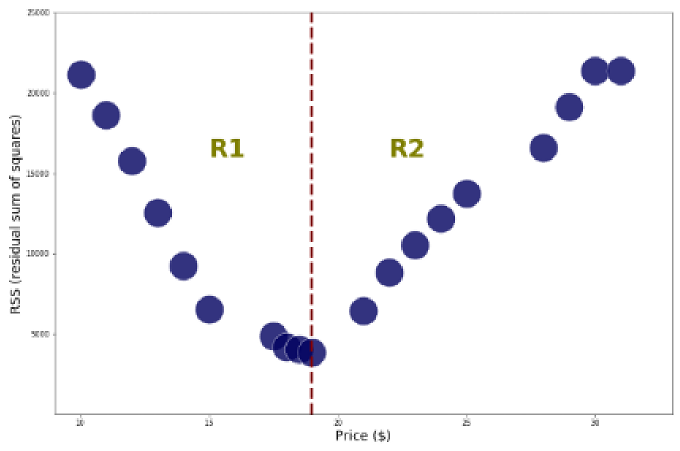
This simulation uses a **dummy data** as following

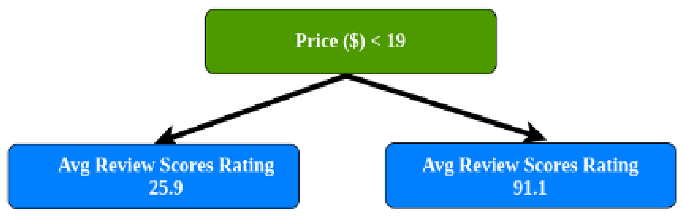


	price ($)	cleaning_fee ($)	license	review scores rating
0	10.0	0.0	0	10
1	11.0	1.0	0	10
2	12.0	2.0	1	10
3	13.0	3.0	0	10
4	14.0	4.0	1	13
5	15.0	5.0	0	20
6	17.5	7.5	1	35
7	18.0	8.0	1	44
8	18.5	8.5	1	52
9	19.0	9.0	0	55
10	21.0	11.0	1	80
11	22.0	12.0	0	83
12	23.0	13.0	1	80
13	24.0	14.0	1	83
14	25.0	15.0	0	85
15	28.0	18.0	1	100
16	29.0	19.0	0	100
17	30.0	20.0	0	100
18	31.0	21.0	1	100
19	31.0	21.0	1	100

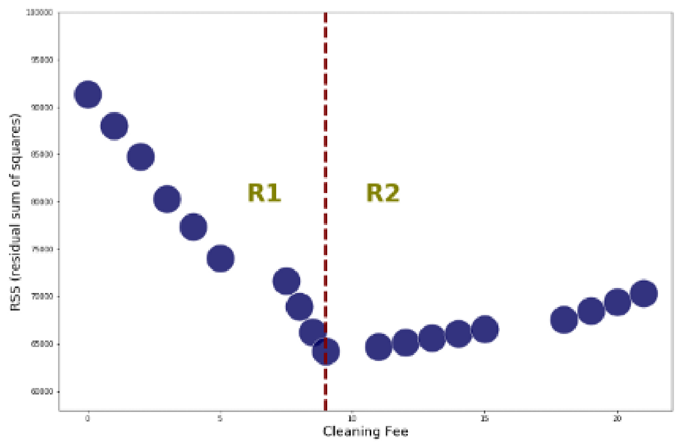
Find out the minimum RSS each predictor

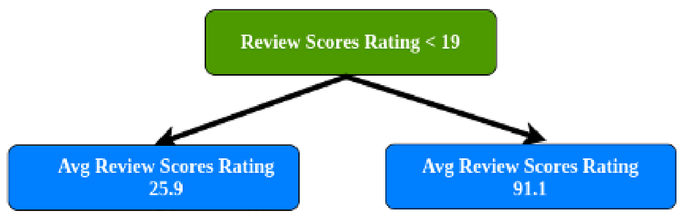
Price with RSS = 3873.79



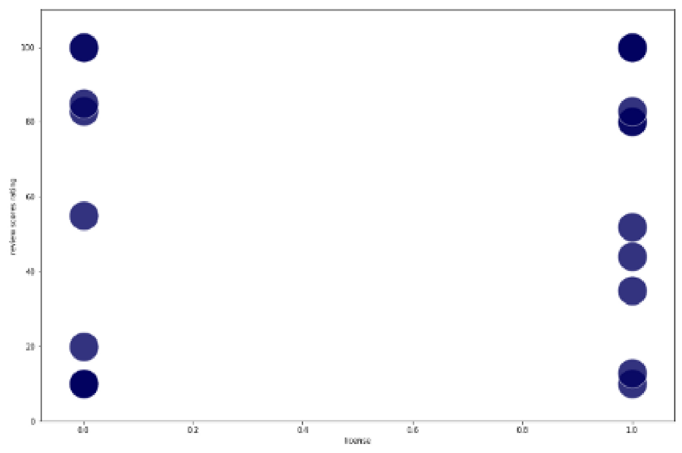


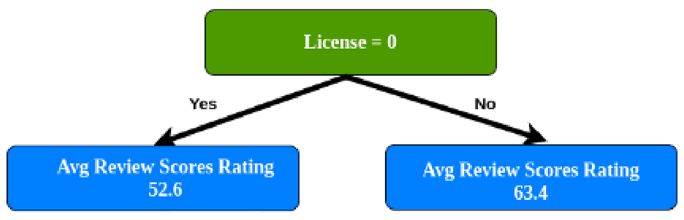
Cleaning fee with RSS = 64214.8




Review Scores Rating < 19
Avg Review Scores Rating 25.9
Avg Review Scores Rating 91.1

There is only one threshold in License, 1 or 0. So we use that threshold to calculate RSS. **License with RSS = 11658.5**





We already have RSS every predictor, compare RSS for each predictor, and find the lowest RSS value. If we analyze, License has the lowest value so it becomes root.

The next step can follow the intuition of the Classification in Decision Tree, in the case of classification calculates Gini Impurity, while in the case of regression calculates the minimum RSS. So this is a challenge for you if want to calculate RSS to the end :)

About Me

I'm a Data Scientist, Focus on Machine Learning and Deep Learning. You can reach me from Medium and Linkedin

My Website : https://komuternak.com/

Reference

1. Introduction to Statistical Learning
2. Ecological Informatics — Classification and Regression Trees
3. *Adapted from YouTube Channel of "StatQuest with Josh Stamer"*