|  |  |
| --- | --- |
| Faculty of Computer & Information Sciences  Ain Shams University  Subject: (SCC 234)  Statistical Analysis and Applications  Year: (2nd year) undergraduate  Academic year: 2nd term 2019-2020 |  |

**Research Topic (1)**

**Title: Using Probability to Build Decision Trees for Classification**

**تحذير هام: علي الطالب عدم كتابة اسمه أو كتابة اي شيء يدل علي شخصيته**

Contents

[Importance of Decision Trees 2](#_Toc40372930)

[What Are Decision Trees? 2](#_Toc40372931)

[Decision Trees and Machine Learning 2](#_Toc40372932)

[Advantages and Disadvantages of Decision Trees: 3](#_Toc40372933)

[Example on Decision Trees 7](#_Toc40372934)

[The Dataset: 15](#_Toc40372935)

[Random Forest: 21](#_Toc40372936)

[References: 23](#_Toc40372937)

**Decision Trees**

“The quality of decision is like the well-timed swoop of a falcon which enables it to strike and destroy its victim”

Importance of Decision Trees**:**

We can acquire two different wisdoms from this quote, that decisions must be well-timed that every aspect matters, and if the decision taken was not the optimal decision, the victim might be you, and that is why we have decision trees to help guide and map every aspect of our lives and evidently we use this method daily but in messy form.

# What Are Decision Trees?

Decision trees are a way to visualize all possible decisions along with their outcomes in order to understand all possible scenarios for better outcomes, so they are used in many fields such as (machine learning, data reconfiguration and statistical analysis, operations research, etc.), .

Decision Trees and Machine Learning**:**

Decision Trees turn out to have influenced many areas of Machine Learning, in both classification and regression.

And even though it’s used in many decision-making strategies to reach a proposed or an efficient goal, it’s preferable the most in machine learning.

As it was stated that Decision Trees are used in machine learning then then most preferable/common languages used to create their algorithms are Python and R Languages and both have their strengths and weaknesses, and through this thesis we’ll be using Python as our main language for processing a real life dataset and a common one which is heart disease dataset, this dataset combines symptoms and general information about 303 patients and their results whether positive or negative.

# Advantages and Disadvantages of Decision Trees:

* **Advantages:**
  + Trees are easily understandable to people.
  + Trees mirror man’s decision-making.
  + Trees have many forms including both flow and graphical forms.
  + Trees can easily handle qualitative predicators.
* **Disadvantages:**
  + Their Accuracy is lower than Regression and Classification approaches.
  + They are non-robust meaning a slight error in data can cause massive change in outcomes

The following will show evidently how important decision trees are and more importantly how to build them.

To understand decision trees we must understand a binary tree which is represented as an upside down tree where the root of the tree is at the top and it is the main/start node, every node has two branches specified as 0 and 1 or left and right, in our case of decision trees it’s yes or no depending on the type, then we have internal nodes which lead us to leaf nodes or leaves and they are the end point of the tree.

In our case of decision trees our nodes are like questions and each answer leads to another question until we reach the probability of all possible answers.

To calculate these probabilities we call them impurity measures.

What are impurities?

Suppose we have a basket full of apples and a bowl full of the label “apple”, if we draw one item from the basket and one item from the bowl then the probability that both items will match name is 1 and this means that impurity is Zero.

Now suppose that we have the same basket but full of three vegetables bananas, apples, kiwis and the same bowl but full of the labels of the three vegetables, the probability to get the same result and any random permutation or combination can be possible and won’t be 1 , so now we can say that impurity will not be Zero.

Which explains to us that if probabilities are not 1 then there must exist impurities.

There are multiple impurities measures used in decision trees of which are classification error rate, entropy, Gini-index, and decision trees algorithms use Information Gain (IG).

There are many decision tree algorithms available

Gini-Index is used by Classification and Regression Tree Algorithm or (CART), and entropy is used by Iterative Dichotomizer 3 Algorithm or (ID3),

There is another algorithm which is (C4.5) and which is a new version of (ID3),

Chi-Square automatic interaction detection or (CHAID) which performs multiple level splits when processing Classification trees.

And both (Gini-Index and Entropy) are different criterions for calculating information gain.

Each way has its advantages and disadvantages, the most non-preferable way is classification error which is not sufficiently sensitive for tree-growing.

First what is Information Gain?

Information gain is the measure of how much “information” a feature give us about the class, and it is based on the decrease of entropy.

Its mathematical formula:

Information Gain = Base Entropy – New Entropy

Then we have Classification Error Rate:

The Classification error rate is the fraction of training observations in a region that do not belong to the most common class.

Its mathematical formula:

Where E refers to Error, and represents the proportion of training observations in the Th region that are from the Th class

After that we have Gini-Index:

This is a measure of total variance across all classes, it will be small if the proportion is close to 0 or 1, so it is a well measure of node purity.

Its mathematical formula:

Where G refers to the Gini-Index, and represents the proportion of training observations as mentioned above

Finally we have Entropy:

Entropy is the indicator of how messy the data is, it’s the measure of the randomness and unpredictability of the dataset.

Its mathematical formula:

Where D refers to Entropy, and represents the proportion of training observations as mentioned above.

And since, it follows

Another mathematical formula:

Entropy =

Where is the probability/percentage of class in a node

A lot of these measures are exploited within algorithms like CART and ID3 and even C4.5, now in machine learning there are two types of data (training data and testing data), training data is used to fit the model and testing data is used to validate this model, and since decision trees are mainly used in machine learning in our field of data science, these algorithms mainly revolve around training and testing the dataset, we’ll discuss each algorithm, how they operate and how they process data.

First we have ID3:

ID3 or Iterative Dichotomizer 3 was the first implementation of decision trees developed by Ross Quinlan in 1986.

What Iterative Dichotomizer means is that it recursively divides attributes into two completely opposite groups which are the most dominant attribute and others in constructing the tree, then it calculates the Entropy and information gain (IG) of each attribute since ID3 Algorithm uses Entropy, that’s how we find the most dominant attribute and put it as a node in the tree, and this process is being repeated as stated by iterative notation until the tree is completed and we’ve reached a decision for that branch and the same process is done in all of the other branches in order to complete the tree.

After that we have CART:

CART or Classification and Regression Tree was a term introduced by Leo Breiman to refer to decision tree algorithms that can be used in classification and regression prediction modelling problems.

It is very similar to C4.5, though the only notable difference is that CART constructs the tree based on numerical splitting criterion recursively applied to the data.

CART Modelling is composed of selecting input variables and splitting points on those variables in order to reach the final tree form, how to select these input variables and how to split them are done using a greedy algorithm to minimize the cost function surely, ending the tree with a stopping criterion such as the minimum number of training instances assigned to each leaf node in that tree.

Lastly we have C4.5:

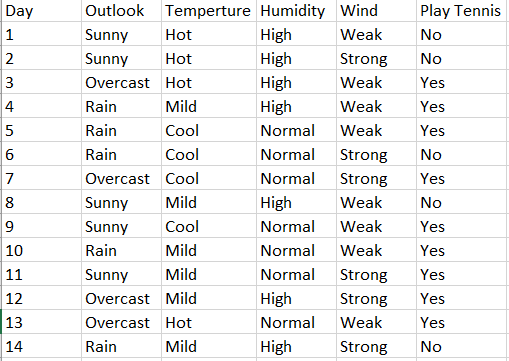
C4.5 or J48 are Ross Quinlan’s next generation of ID3 and the C in C4.5 is due to the fact that this algorithm is developed in C language and J in J48 is due to being developed in Java, though they’re the same algorithm but they’re written in different languages.

C4.5 introduced new features on ID3, such as accepting both continuous and discrete features, handling incomplete data points, solving over-fitting problems by using the bottom-up technique usually referred to as (pruning) and that different weights can be applied to the features that comprise the training data.

# Example on Decision Trees

The Following is an example of how decision trees are being built we’ll be using a dataset about weather, the use of this dataset is to determine whether a group of people are going to play tennis outside or not based on the weather.

The dataset will also be given with the thesis.



Above is our Dataset containing 14 rows and 5 attributes, 4 of them are outlook, temperature, humidity, wind, and the last attribute is the final decision.

We’ll solve this dataset now by using the ID3 Algorithm.

First, there are 9 binary classes of “yes” and 5 of “no”

Using the Entropy Rule,

Now that we figured out the entropy of the decision.

Let’s look at every Attribute Information Gain to figure out which attribute will be at the root.

First Attribute: Outlook

Now that we calculated the categorical values of entropy of sunny, overcast, rain in outlook, we then get the Average Entropy information for outlook.

Now that we’ve calculated information gain of the first attribute we then calculate the second attribute and so on until we finish gathering the information gain for each attribute.

Second attribute: Temperature

Then we calculate the Average Information Entropy for the second attribute.

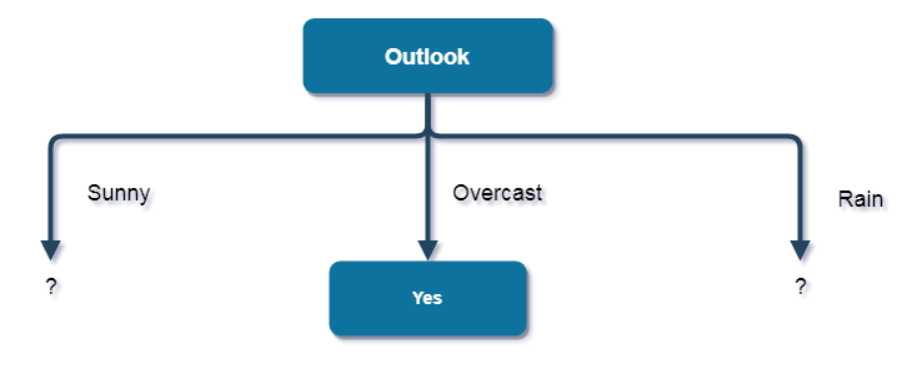
Third attribute: Humidity

Then we calculate the Average Information Entropy for the third attribute.

Fourth Attribute: Wind

Then we calculate the Average Information Entropy for the fourth attribute.

Now we’ve finished calculating the information gain of each and every attribute, notice that the attribute with the maximum information gain is Outlook, this means that the root of the tree will be Outlook, and since that Overcast Entropy measure is 0 then this means that the decision will be of pure class “Yes”.



Now that we figured out the root as the figure above we now repeat the same process but for the other nodes of Sunny and Rain.

Entropy of Sunny:

First Attribute: Temperature

Then we calculate the Average Information Entropy for the second attribute.

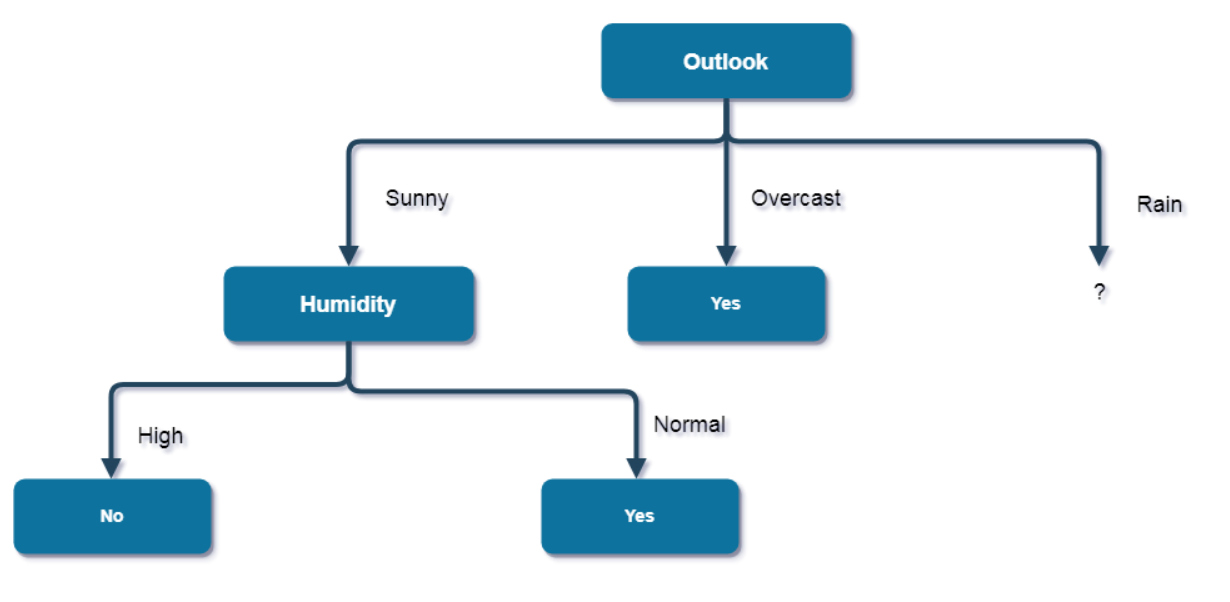
Second Attribute: Humidity

Then we calculate the Average Information Entropy for the third attribute.

Third Attribute: Wind

Then we calculate the Average Information Entropy for the fourth attribute.

So, the attribute that holds the maximum information gain is humidity, so, the attribute that goes in the sunny branch is humidity as shown in the figure below.



When outlook is sunny and humidity is high, this is a pure class of category “no”, and vice versa, when outlook is sunny and humidity is normal, this is a pure class of category “Yes”.

Now let’s figure out which attribute will be put in the branch “Rain”, but first we also get the entropy of “Rain”.

First Attribute: Temperature

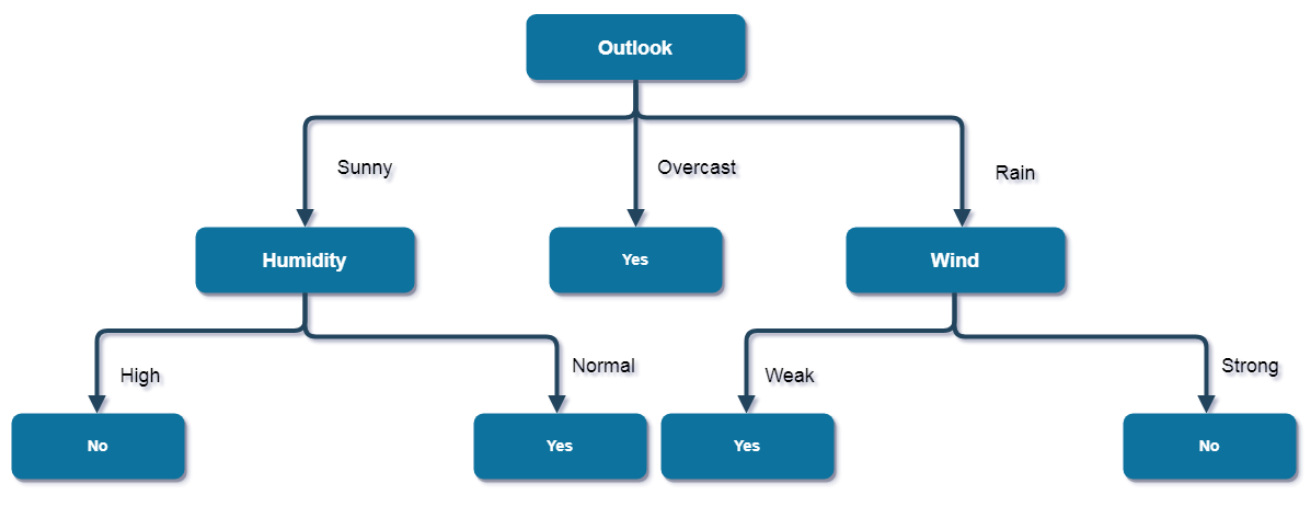
Notice that we didn’t use Hot Temperature since there’s no possibility of having hot temperature while in rain.

Then we calculate the Average Information Entropy for the second attribute.

Second Attribute: Wind

Then we calculate the Average Information Entropy for the fourth attribute.

So, as it is clear the “Wind” attribute has the maximum information gain so this makes “Wind” attribute go in the branch of “Rain” as shown in the figure below.



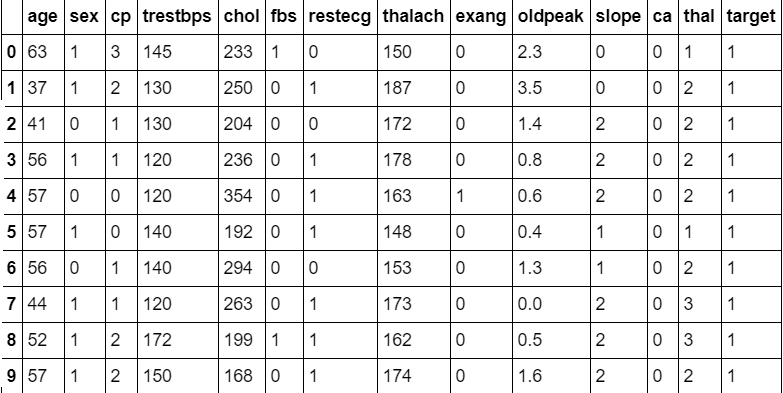
As we have notice we didn’t put temperature in the tree, this is due to the fact that the decision doesn’t depend in any way on temperature, so, when outlook is rain and wind is strong, then it’s a pure class of category “No”, and when “Outlook” is “Rain” and “Wind” is “Weak”, then it’s a pure class of category “Yes”.

And that concludes our example of decision trees.

The following is a real life dataset of decision trees being used in machine learning we’ll be using python programming language as stated above, for future reference I’ve made a repository containing both the dataset and the program along with other csv files which are created by the program, for better performance it is recommended that the program be used in a python interpreter environment in form of [in, out] clauses, each section will be marked using comments, but first we must discuss the dataset and reveal its attributes.

# The Dataset:

By using the python interpreter and importing many modules in python such as [“NumPy”, “Seaborn”, “Pandas”, “Matplotlib”, “Scikit-learn”, etc.], we were able to establish connection and read the dataset using the .read.csv() built-in function in pandas to first read our dataset and .head(10) to look at our dataset and it looks like this.

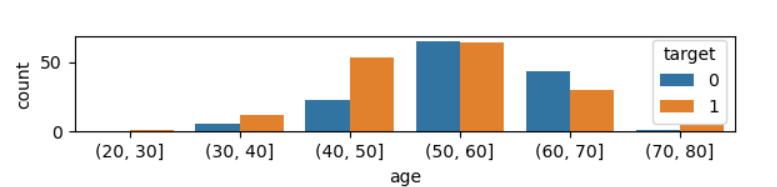


Contents of the table:

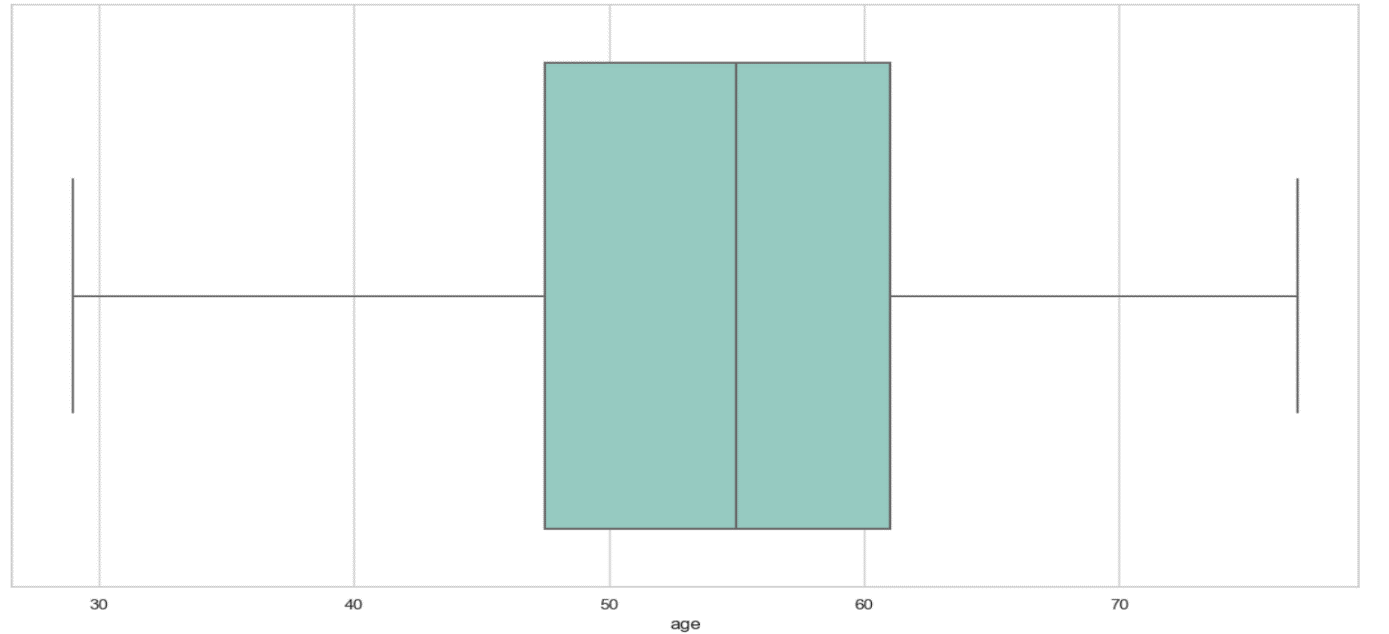
* “Age”: this column represents the ages of patients.
* “Sex”: this column represents the sex of patients of 1 being a male and 0 being a female.
* “Cp”: this column represents the patients on a scale from 1 to 3, 3 being the highest in severity.
* “Trestbps”: this column represents the resting blood pressure (measured in mm Hg).
* “Chol”: this column represents the cholesterol reading (measured in mg/dl)
* “Fbs”: this column represents the fasting blood sugar being more than 120 mg/dl refers to value 1 and 0 value for being less than or equal to 120.
* “Restecg”: this column represents resting electrocardiographic results.
* “Thalach”: this column represents maximum heart rate achieved.
* “Exang”: this column represents exercise induced angina being 1 for yes and 0 for no.
* “Oldpeak”: ST depression induced by exercise ST segment.
* “Slope”: this column represents the slope of the peak exercise ST segment.
* “Ca”: this column represents the number of major vessels ranging from 0 to 3 which are colored by fluoroscopy.
* “Thal”: this column represents normality of thalassemia an inherited blood disorder being 1 for normal, 2 for fixed defect, 3 for reversible defect.
* “Target”: this column represents if the patients were diagnosed with heart disease or not being 1 for positive and 0 for negative results.

Now that we figured out each element in the table we can now begin analyzing each and every column using the indexing operator, .describe() and others built-in functions to get the results for the median, mode, mean, standard deviation, quartiles and minimum and maximum values.

First we look at the shape of the first column which is Age.



So, this our shape for age and it looks more centered but we can double check using boxplots as figured below.

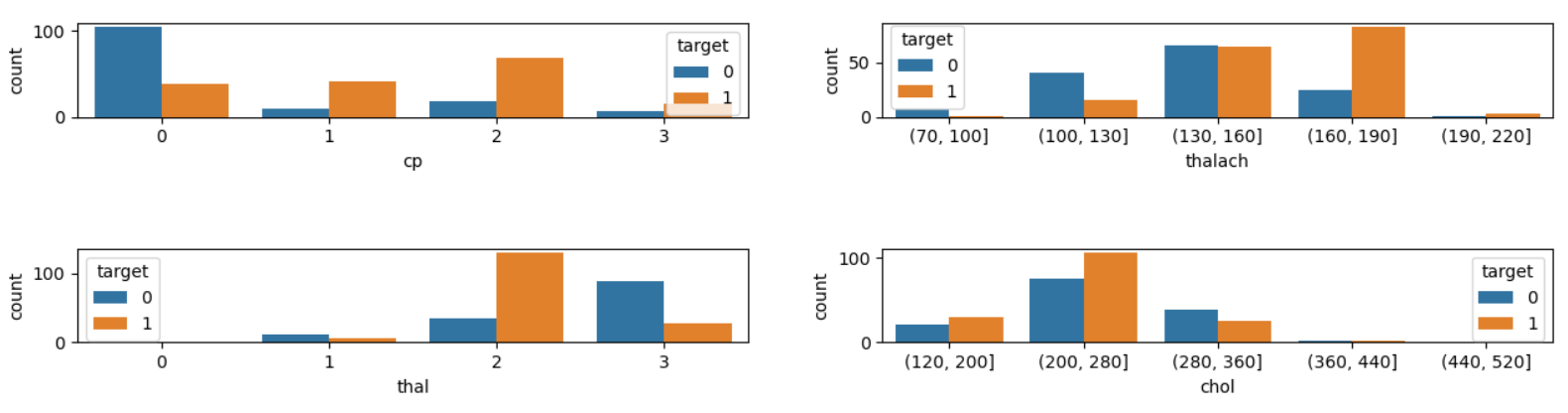


We can determine from our built-in functions that the measures of central tendencies are:

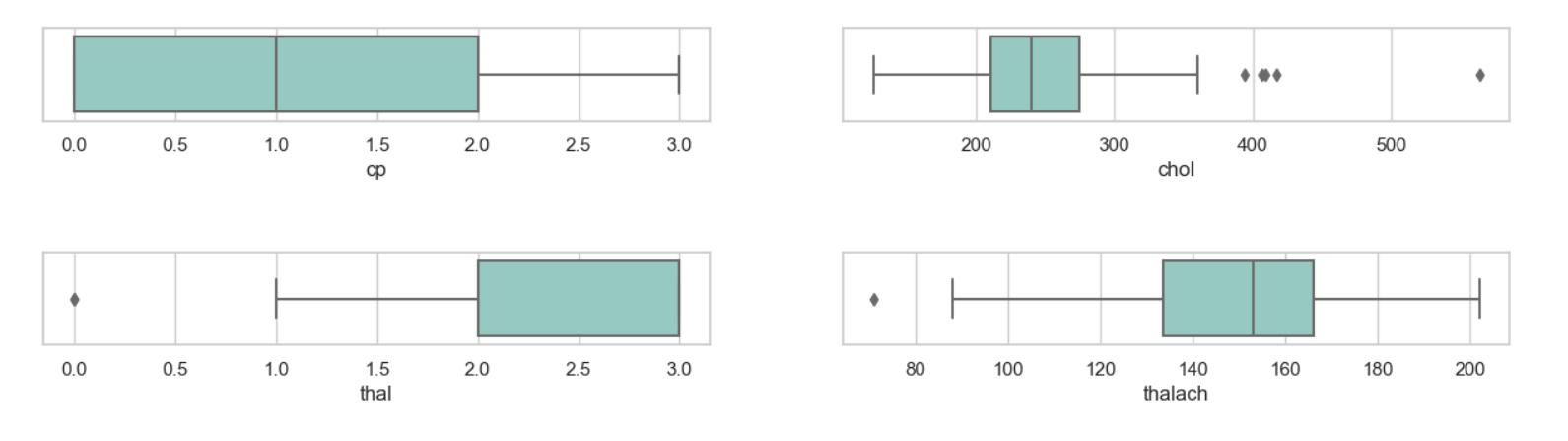
* The mean value is 54.366.
* The minimum value is 29.
* The maximum value is 77.
* The median value is 55.
* The first quartile Q1 equals 47.5 and the third quartile Q3 equals 61
* The outliers are (27.25,81.25)
* The standard deviation calculates at 9.08.
* The Variance is 82.44

So this is our analysis of Age data, each of the remaining 13 attributes will be analyzed the same way in order to continue with our decision tree but since 13 attributes are more than enough we’ll state only 5 important attributes including Age attribute for just proving visualization.

The next attributes are as follows [“cp” (chest pain), “chol” (cholesterol), “thalach” (max heart rate), “thal” (inherited blood disorder)].



Now that we’ve generated the shape we can now look at the boxplots of the given attributes.



As we can see these are the boxplots of our other attributes, the following is an explanation.

Content of the Figure:

1. “Cp” (Chest Pain):
   * Mean is 0.966
   * Median is 2
   * Minimum value and first quartile are 0
   * Maximum value is 3
   * third quartile are 2
   * Outliers are (-3,5)
   * standard deviation is 1.032
   * Variance is 1.06
2. “Chol” (Cholesterol):
   * Mean is 246.26
   * Median is 240
   * Minimum value is 126
   * Maximum value is 564
   * First and third quartiles are 211 and 274
   * Outliers are (115.75,369.75)
   * standard deviation is 51.83
   * Variance is 2,686.34
3. “Thal” (Inherited Blood Disorder):
   * Mean is 2.313
   * Minimum value is 0
   * Median and first quartile are 2
   * Maximum and third quartile are 3
   * Outliers are (0.5,4.5)
   * Standard deviation is 0.612
   * Variance is 0.374
4. “Thalach” (Maximum Heart Rate):
   * Mean is 149.646
   * Median is 153
   * Minimum value is 71
   * Maximum value is 202
   * First and third quartiles are 133.5 and 166
   * Outliers are (84.75,214.75)
   * Standard deviation is 22.905
   * Variance is 524.63

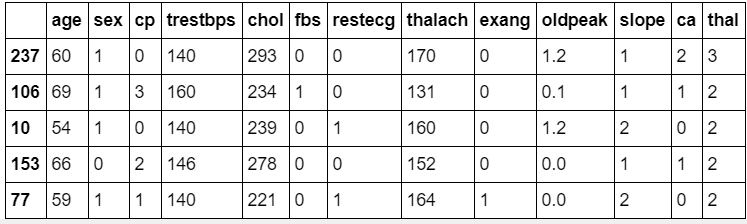
As mentioned above there are other attributes calculated, after analyzing the dataset we now can begin building the tree.

A Must is seeing the correlation between all attributes to see which ones affect the output stated as “target” to start splitting the data.

Using a heat-map we can find the correlation as figured below.



From the above figure we realize that “exang”, “cp”, “thalach”, “ca” are the attributes having the stronger effect on the output, so now we can start splitting the data and balancing the tree along every split.



In the above table this is the head of the training set.

If we check the balance of the tree we find that the mean of this training set is about 0.5619 while it is a good balance but it is essential to balance the tree even more for better outcomes, And at the IN[6] we’ve balanced that tree and the mean became 0.5

Now we can start building the models we built Decision-Tree Model and Random Forest Model, so what is Random Forest?

## Random Forest:

Or so called Random Decision Forests, are an ensemble learning method for classification and regression, they operate by constructing a multitude of decision trees and correct their habit of overfitting to their training sets.

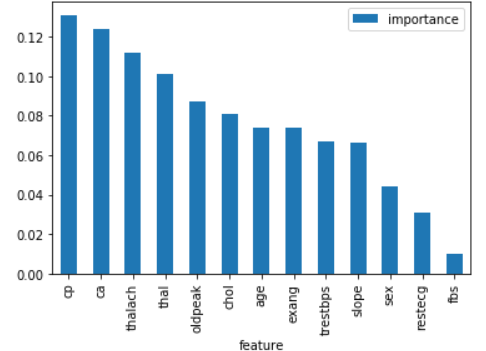
They are essential for building a better form, correct decision tree for better use.

So we’ll be using them to build our tree.

And after building the models we have to tune them for better results at IN[8].

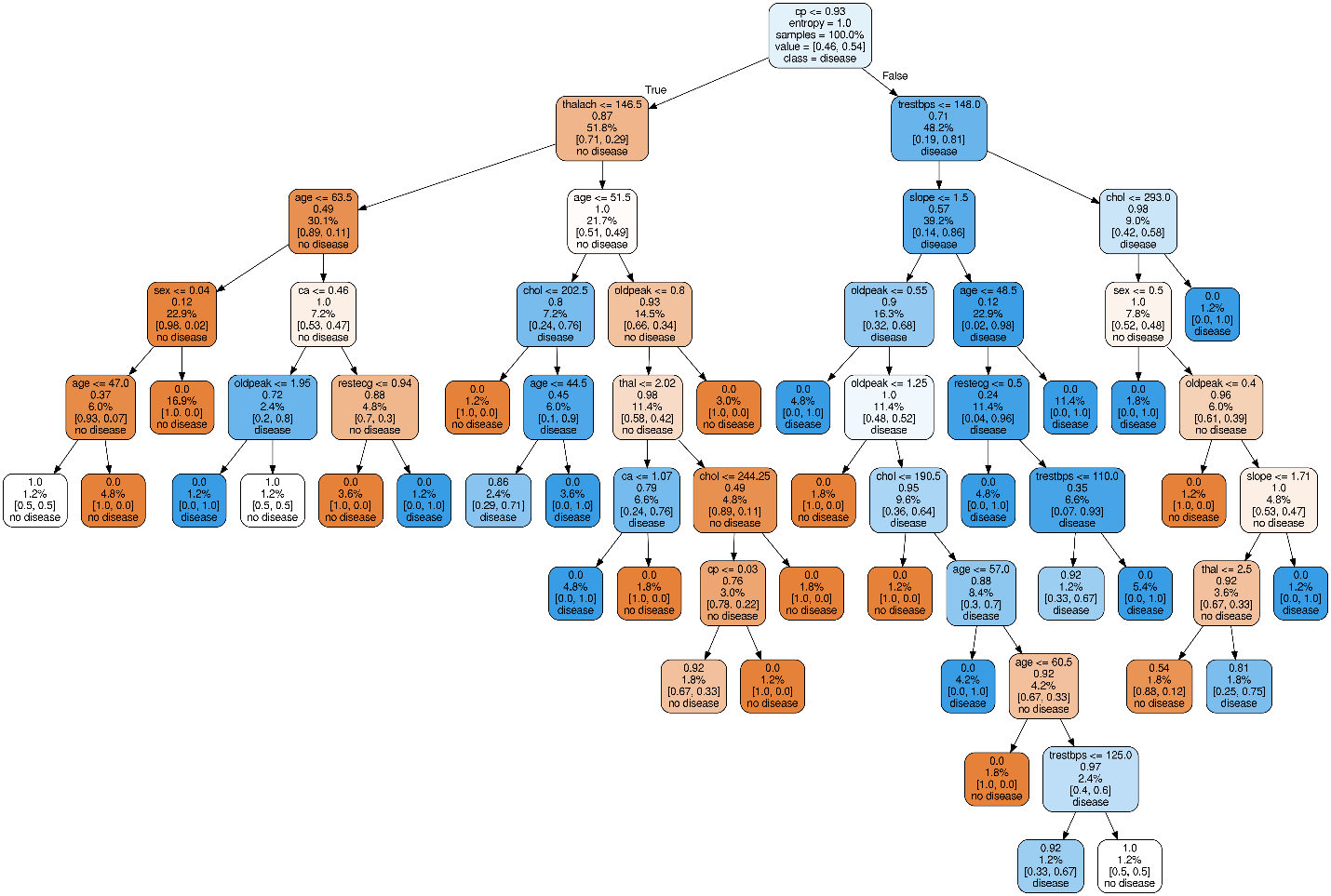
After tuning them we now can explore their results to check if we tuned correctly or not.

At exploring these models we finally reached the importance of each attribute using a plot bar to use them in our tree as figured below.



So now finally we now know that cp (Chest Pain) is the most important attribute which means that this attribute will be our root of the tree.

Finally at IN[10], We Build the tree Balanced and ready for use.



So, In conclusion, decision trees help describe all possible outcomes and even the most preferred so, we used it in all possibly related fields to help reach the verdict, after using our real life dataset which was on a critical subject which is heart disease, we now realized that these trees are in fact essential.

We first mentioned the definition of decision trees and their structure and phases then we explained the multiple algorithms such as ID3 and CART and C4.5, and we explained the information gain what it is and when we use it, and we also demonstrated entropy and the Gini-Index.

After that we solved an example and drew our first decision tree.

After that we brought our real life dataset which was on heart disease and we explained each and every attribute and used a referenced code to help plot our boxplot and count-plot to view our shape, center and spread of our data, containing the mode, mean, median, quartiles, and then we started building the models and tuning them to get the best balance and results, after that we finally reached our decision tree.

Finally, below are our references if you’d like to check our thesis.

# References:

GitHub. 2020. *Yeimsf/HD-Dataset*. [Online] Available at: <https://github.com/yeimsf/HD-DataSet>

Archive.ics.uci.edu. n.d. *UCI Machine Learning Repository: Heart Disease Data Set*. [Online] Available at: <https://archive.ics.uci.edu/ml/datasets/Heart+Disease>

Kaggle.com. 2020. *Heart Disease UCI*. [Online] Available at: <https://www.kaggle.com/ronitf/heart-disease-uci/data>

Gareth James, et al., “An Introduction to Statistical Learning with Applications in R”, Springer

GitHub. 2020. *Vinici Usov/Heart-Disease*. [Online] Available at: <https://github.com/viniciusov/heart-disease>