

Bt2101 – Tutorial 1: Decision Tree

Kaustubh Jagtap A0168820B

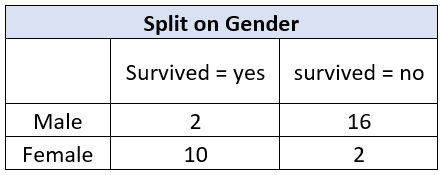
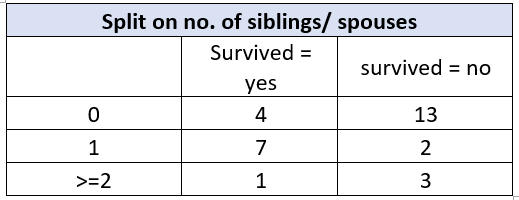
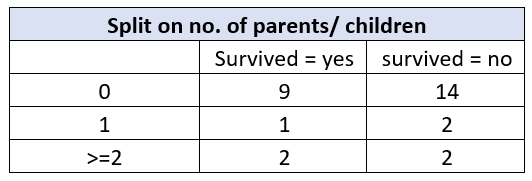
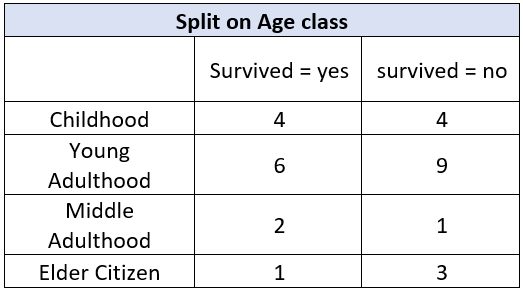
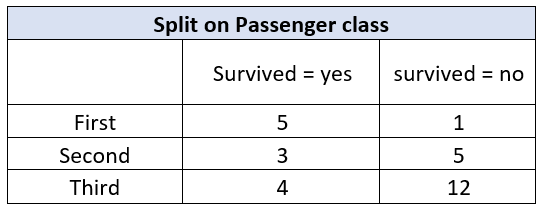


national university of singapore

**Overview**

In this tutorial, I first performed manual calculations for the information gain for all 5 attributes, by constructing frequency tables for all of them. Following this, I abstracted my methodology by writing python functions to calculate entropy and information gain (adapted and modified from the course github page). To address the third question, I directly defined python methods to calculate the gini index. All my findings are presented below.

**Questions 1 & 2 – Manual Calculation**

For this step, I had to break down the given dataset into frequency tables for each attribute. These frequency tables are given below. It is interesting to note that intuitively, we can hypothesize that *gender* would be the best attribute for the first split, since it gives us the cleanest partitioning – we can tell just by visual inspection of the data.

*Sample Calculation for split on passenger class*

Using the entropy formula, I calculated the entropies for passenger class split as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Split on Passenger class** | | | |
|  | Survived = yes | survived = no | Entropy |
| First | 5 | 1 | 0.65 |
| Second | 3 | 5 | 0.954 |
| Third | 4 | 12 | 0.811 |
| Weighted Entropy | | | 0.817 |

*Results*

The above calculations were performed for all 5 attributes, and the results are shown below.

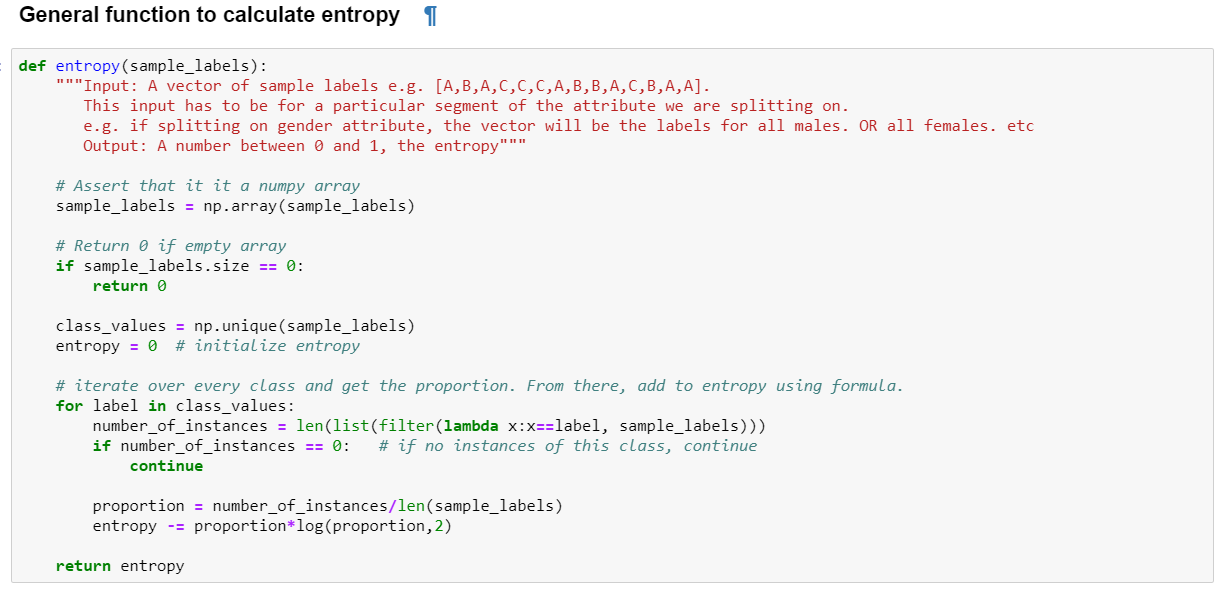
|  |  |  |
| --- | --- | --- |
| **Entropy Method** | | |
| **Attribute** | **Weighted Entropy** | **Gain in Information** |
| Age Class | 0.868 | 0.103 |
| Passenger Class | 0.817 | 0.154 |
| Gender | 0.562 | 0.409 |
| No of Siblings or spouses | 0.783 | 0.187 |
| No of parents or children | 0.965 | 0.005 |
| Note: Initial Entropy = 0.971 | | |

We want to maximize the gain in information, hence we shall choose the *Gender* attribute for our first split. This is in line with our initial hypothesis.

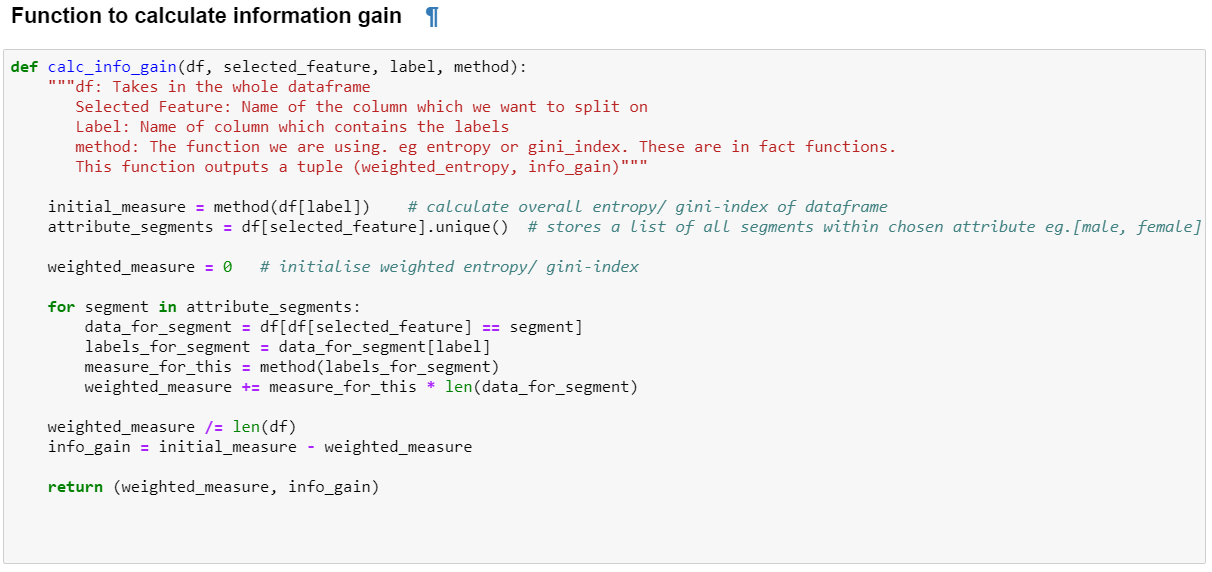
**Questions 1 & 2 – Python scripts**

I thought it would be interesting to redo the above using python. I adapted some of the scripts from our course github and came up with an automated process to calculate information gain.

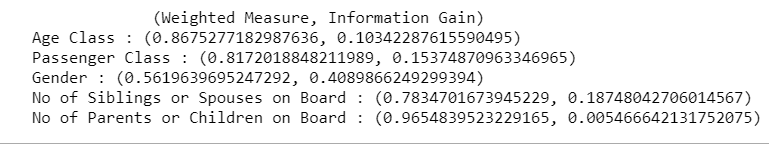
The first script I rewrote is that for entropy calculation. Below is an abstracted version of the function that can calculate entropy for *any* number of input sample labels.



I then defined a function to calculate information gain – this was done from scratch. Here, it takes a parameter called *method*. This is actually a function to be passed in – either entropy or gini\_index. The function uses this *method* to calculate the information gain. Doing this allows for better abstraction.



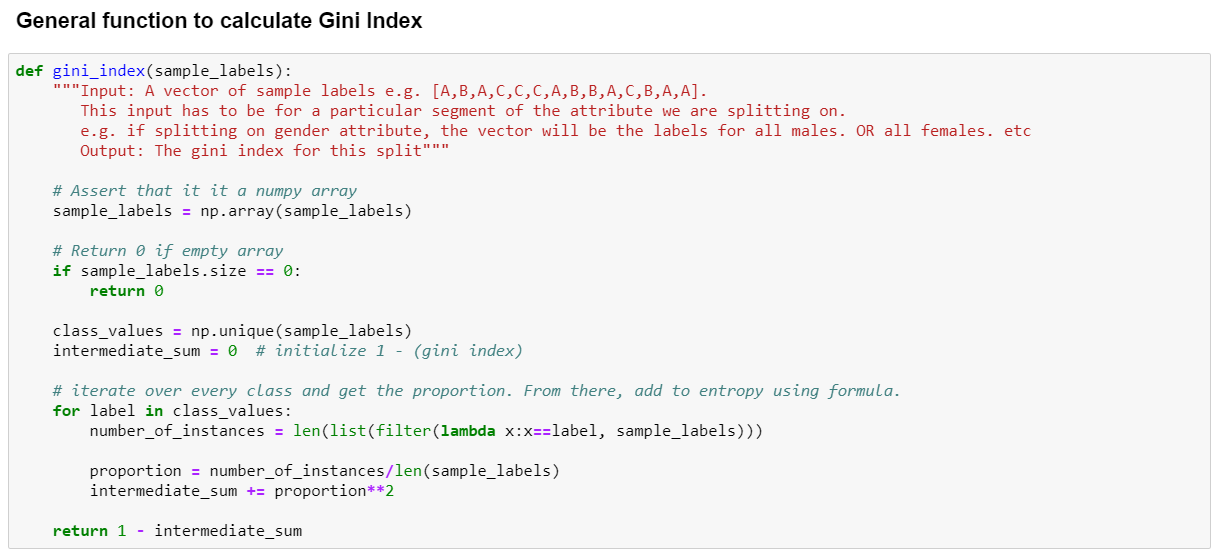
Running the function gives the following result.



It is good to note that these results match with the ones calculated by hand above. Once again, it is affirmed that *Gender* would be the best attribute to split on.

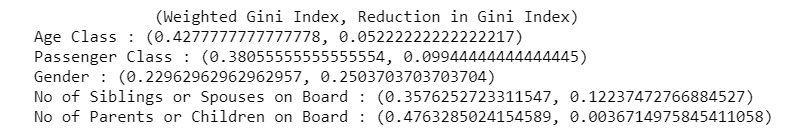
**Question 3 – Using the Gini Index**

Below is a function to calculate the gini index for an attribute, given *any* number of class labels. This was written from scratch.



*Results*

Running the above script on the data produces the following raw results:



The consolidated results are presented in the table below:

|  |  |  |
| --- | --- | --- |
| **Gini Index Method** | | |
| **Attribute** | **Weighted Gini Index** | **Reduction from start** |
| Age Class | 0.428 | 0.052 |
| Passenger Class | 0.381 | 0.099 |
| Gender | 0.230 | 0.250 |
| No of siblings or spouses | 0.358 | 0.122 |
| No of parents or children | 0.476 | 0.004 |
| Note: Initial Gini Index = 0.480 | | |

The split on the *Gender* attribute produces the highest reduction in Gini Index. Hence, even using this method, *Gender* is the best root attribute to split upon.