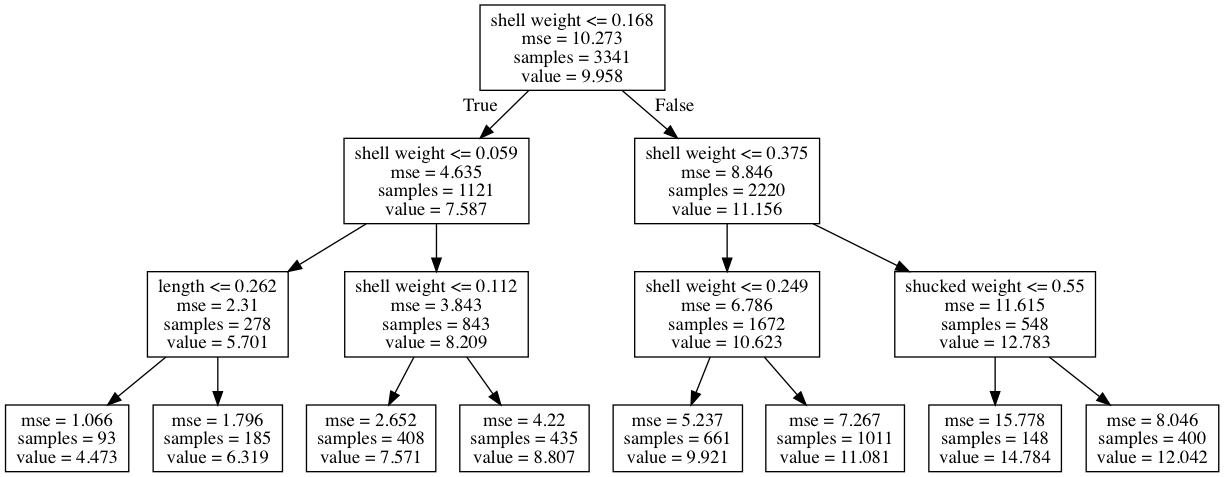
**Decision Tree**

When we heard about Decision Tree Ideally below kind of image will come into our mind.

* Decision Tree is nothing but predict a pattern or to classify the class of a data .
* Decision Tree is nothing but a group of rules it learned from InputData.



So while using Decision Tree basic questions are ,   
Which feature(attribute) we consider as Root Node ?  
Which attributes should consider in next level and at what is diving factor to decide next level ?  
When should we stop these rules formation ?

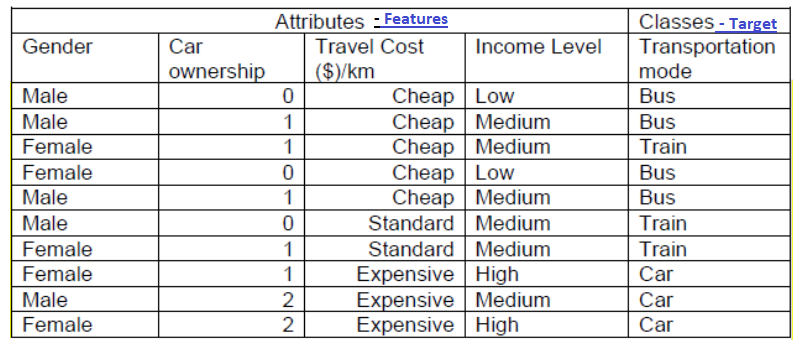
In this tutorial, we will answer to above questions by taking some sample datasets.

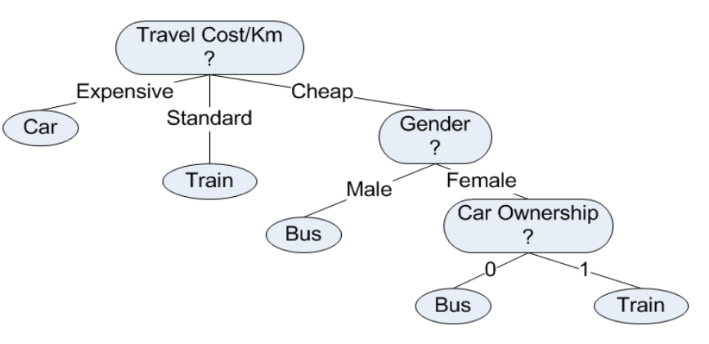
1. Decision Tree Algorithm theoretical explanation.
2. Impurity / Gini / Classification Error Calculations.
3. Build Decision Tree rules by applying Gini/Entropy calculations on sample data.
4. Implement Python API for Decision Tree.
5. Different Ways of Validations.

* Local Validation (Cross Validation)
* Kaggle Submission.

1. **Overall Decision Tree Algorithm theoretical explanation.**

To explain theoretically how decision Tree works I considered below sample records.



Now I’ve dataset So I just created some decision Tree (Below Image) based on above input data. 

Just assume now as by Reading above dataset we / Python API created below decision Tree.

Decision Tree is nothing but a **group of rules** it learned from Input Data.

**Rule 1**: If Travel cost/km is expensive then mode = car

**Rule 2:** If Travel cost/km is standard then mode = train

**Rule 3:** If Travel cost/km is cheap and gender is male then mode = bus

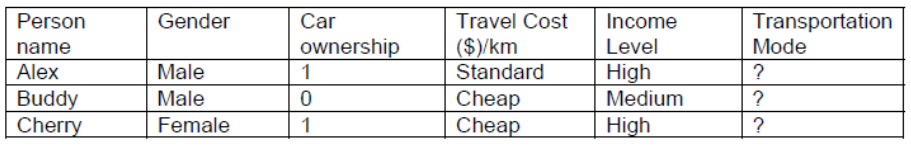
**Rule 4:** If Travel cost/km is cheap and gender is female and she owns no car then mode =

bus

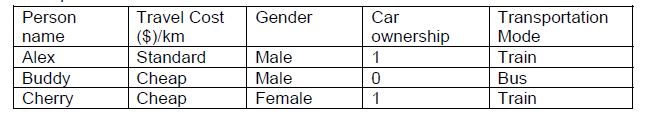
**Rule 5:** If Travel cost/km is cheap and gender is female and she owns 1 car then mode =

Train

Now we are ready with Rules we learned from Sample Data, We can predict the any new records target variable value based on above rules.



If we have Some sample records (test data) as above Can we predict the target value based on above rules ? Yes. Below is the output.



Ohhhh Nice. But we have few difficulties as mentioned below.

1. However, In above example we have only 10 Samples (training data only 10 records) based on that we formed decision tree, Based on Just 10 Samples we cannot generalize the rules of the decision tree above to be applicable for other cases in your city.
2. The sequence of rules generated by the decision tree is based on priority of the attributes.

For example, there is no rule for people who own more than 1 car because based on the data it is already covered by attribute travel cost/km. For those who own 2 cars the travel cost/km are always expensive, thus the mode is car.

To overcome these general and Important problems we should rely on few important concepts.

1. **Impurity / Gini / Classification Error Calculations.**

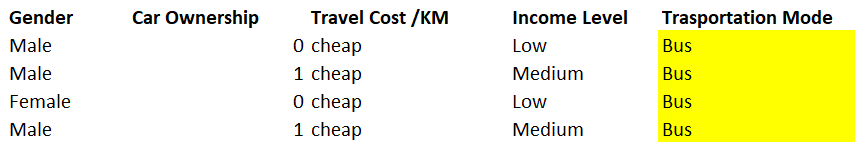
What is Impurity ?

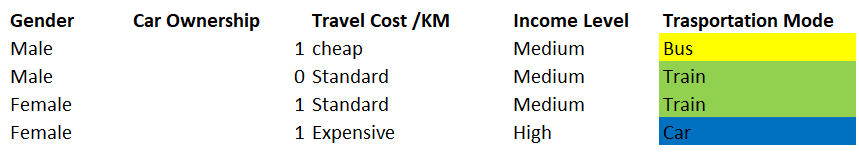
Given a data table that contains features and Target (Final Predicting column/Feature), we can measure homogeneity (or heterogeneity) of the table based on the classes (Target Value).

We say a data table is pure (homogenous) if it contains only a single class.

If a data table contains several classes, then we say that the table is impure or heterogeneous.

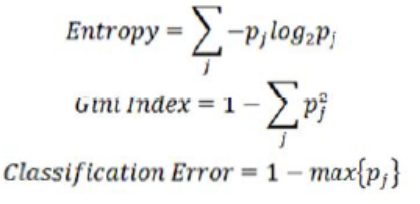
**Pure (or) Homogeneous As it has only one Bus class Value (Target Value contain only one distinct value)**

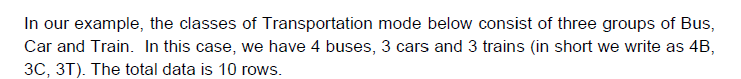


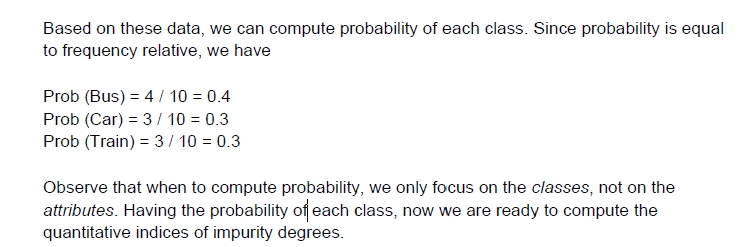
**Impure or Heterogeneous**.

**How to Measure Impurity? (Entropy / Gini Index / Classification Error)**

Most well-known indices to measure degree of impurity are **entropy, gini index, and classification error.**







Entropy:   
https://lh4.googleusercontent.com/RbRHJ9KPLNGbHex2HCUkHe6d0NQdOCa7ytJIze7_4ujeriz6B6KTZYHbItaA22h3WVBi5ZoPg66FmuuZ7QEgjCFO75VguqV9mqvHxMulUG2k3ixZXq8oiJwTT04gmZQ9btiniIS5

Example:

Given that Prob (Bus) = 0.4, Prob (Car) = 0.3 and Prob (Train) = 0.3,

Entropy = – 0.4 log (0.4) – 0.3 log (0.3) – 0.3 log (0.3) = 1.571

The logarithm is base 2.

Note:

1. Entropy of a pure table(Consisting of single class) is Zero. Because the probability is 1 and log(1) = 0.
2. Entropy reaches maximum value when all classes in the table have equal probability.

Gini Index

Another way to measure impurity degree is using Gini index.

Example: Given that Prob (Bus) = 0.4, Prob (Car) = 0.3 and Prob (Train) = 0.3, we can

now compute Gini index as

Gini Index = 1 – (0.4^2 + 0.3^2 + 0.3^2) = 0.660

Gini index of a pure table (consist of single class) is zero because the probability is 1 and

1-(1)^2 = 0. Similar to Entropy, the Gini index also reaches maximum value when all classes

in the table have equal probability.

Similarly, We can calculate Classification error also .

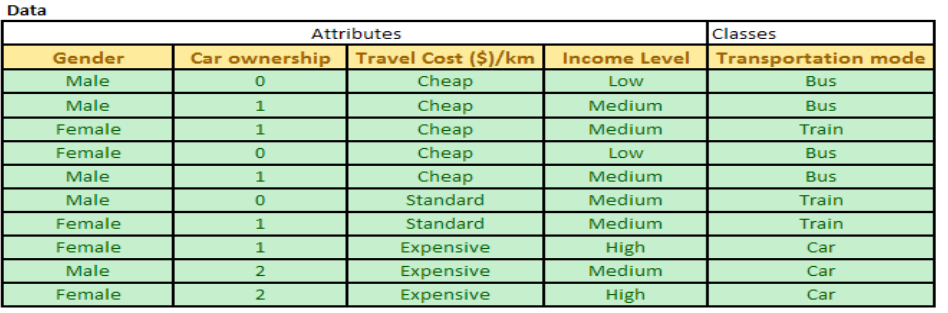
1. **Build Decision Tree rules by applying Gini/Entropy calculations on sample data.**

As we are doing manual activity I am calculating Entropy. similarly, you can calculate Gini or Classification error If you are interested. Steps of calculations.

* 1. Calculate Entropy for the whole dataset on the target variable
  2. Calculate Entropy on each feature.
  3. Find the maximum Information Gain Attribute.
  4. Whichever attribute is given Max Information(Step-3) consider it as Root Node and repeat the activity.

Iteration – 1

1. Calculate Entropy for whole dataset on target variable



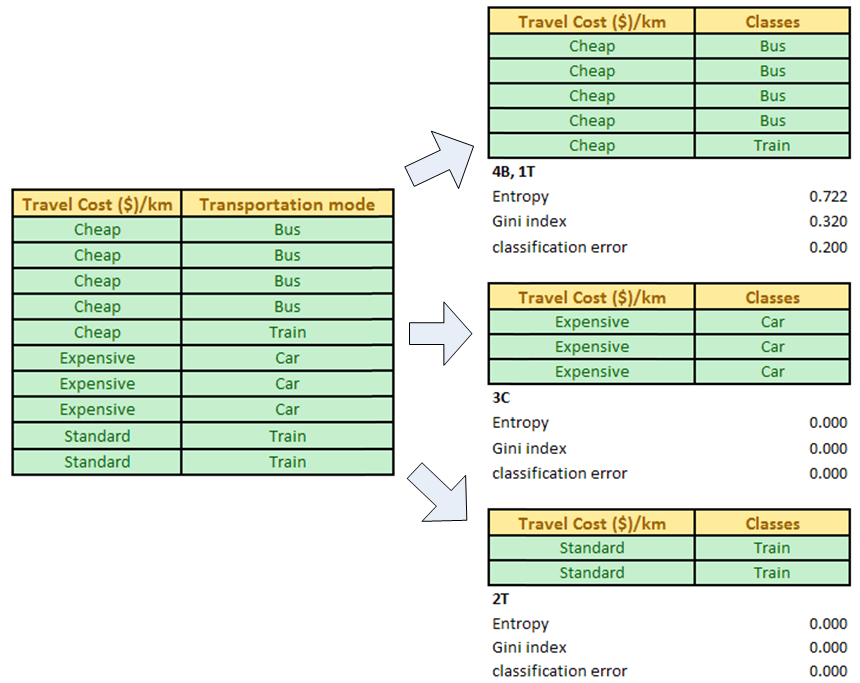
Calculate Entropy or Gini or Classification error for whole dataset as mentioned above formals.

4 Busses , 3 Trains and 3 Cars .

Whole Entropy: – 0.4 log (0.4) – 0.3 log (0.3) – 0.3 log (0.3) = 1.571

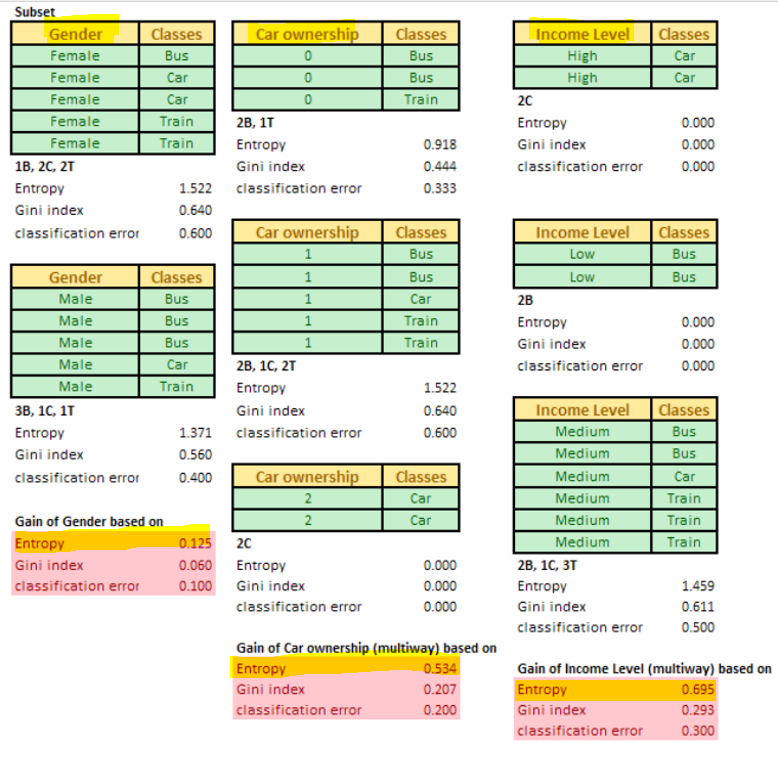
1. Calculate Entropy on each feature.

Travel Cost / Km feature Entropy



Information gain (i) = Entropy of parent table D **–** Sum (nk/n \* Entropy of each value k of subset table Si)

1.571 –     (5/10 \*0.722  + 2/10\*0 +   3/10\*0) = **1.210**



Now We calculated Information Gain for Whole Dataset and Individual Features. Values are below.

Entropy of Initial Whole DataSet : 1.571

Information Gain of **Travel Cost /Km** Feature: 1.210

Information Gain of **Gender** Feature : 0.125

Information Gain of **Car OwnerShip** Feature : 0.534

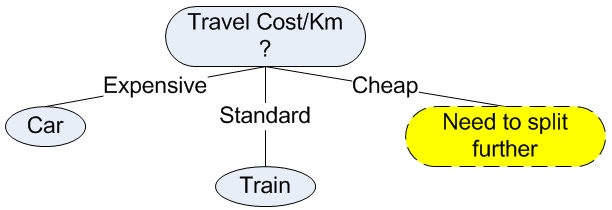
Information Gain of **Income Level** Feature: 1.210

1. Find the maximum Information Gain Attribute.

As calculated above Max Information Gain Attribute/Feature is : Travel Cost /Km.

So with this step , we are concluding as Travel Cost /Km is Maxmimum Information Gain Attribute. So consider this as Root Node and Proceed the calculations to find next level nodes.

By completing Iteration- 1 our Decision Tree Look as below.



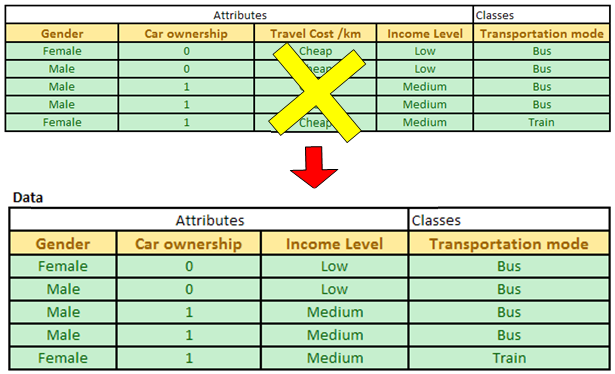
Now challenging is which attribute (Feature) we should consider in next level ? To decide that repeat the same activity In next Iteration. For **Cheap** travel cost/km, the classes are not pure, thus we need to split further in the next iteration.

Iteration – 2

For second iteration, our data table D is only come from the Cheap Travel cost/km.

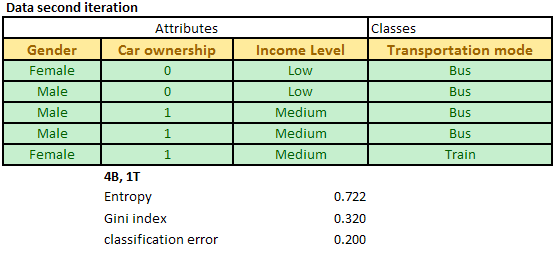
We remove attribute travel cost/km from the data because they are equal and redundant.

Now Our Source Data is Below.



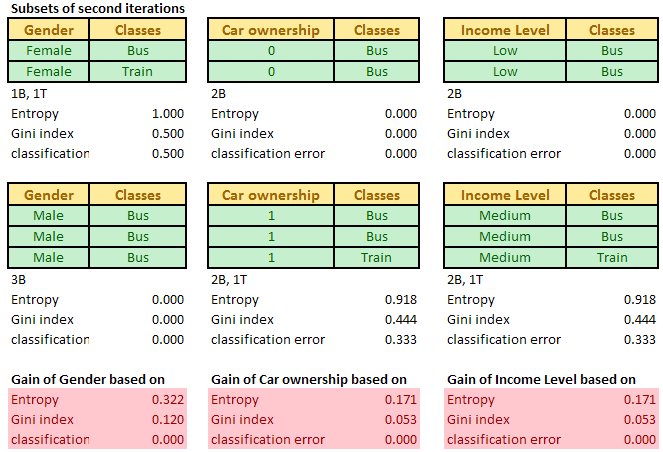
Now we have only three attributes: Gender, car ownership and Income level. The degree

of impurity of the data table D is shown in the picture below

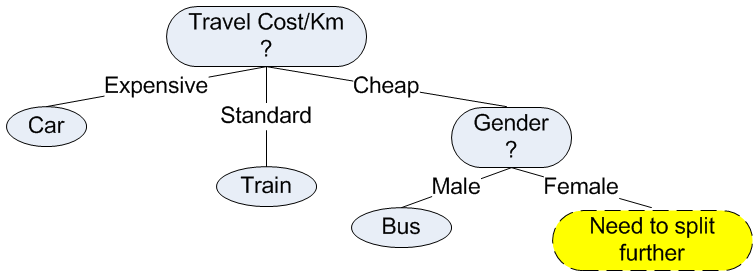


Now this is our Whole Data Now. Repeat Below steps on this subset of Data to decide next Branching element in above Tree.

* 1. Calculate Entropy for whole dataset on target variable
  2. Calculate Entropy on each feature.
  3. Find the maximum Information Gain Attribute.
  4. Whichever attribute given Max Information(Step-3) consider it as Root Node and repeat the activity.

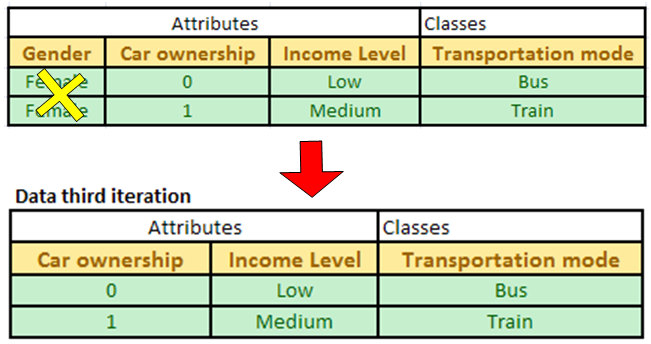


The maximum gain is obtained for the optimum attribute Gender.



Third Iteration:

Since attribute **Gender** has been used in the decision tree, we can remove the attribute and focus only on the remaining two attributes: Car ownership and Income level.



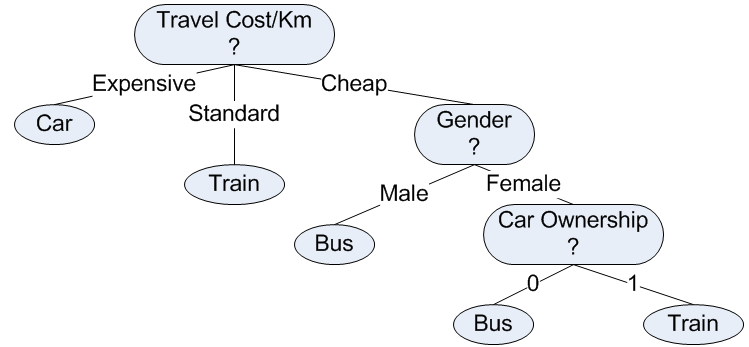
If you observed the data table of the third iteration, it consists only two rows. Each row has

distinct values. If we use attribute car ownership, we will get pure class for each of its

value. Similarly, attribute income level will also give pure class for each value. Therefore,

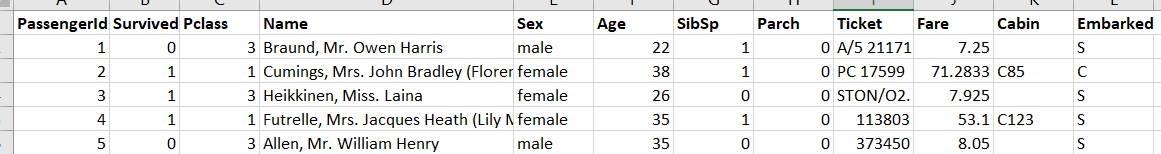
we can use either one of the two attributes. Suppose we select attribute car ownership, we

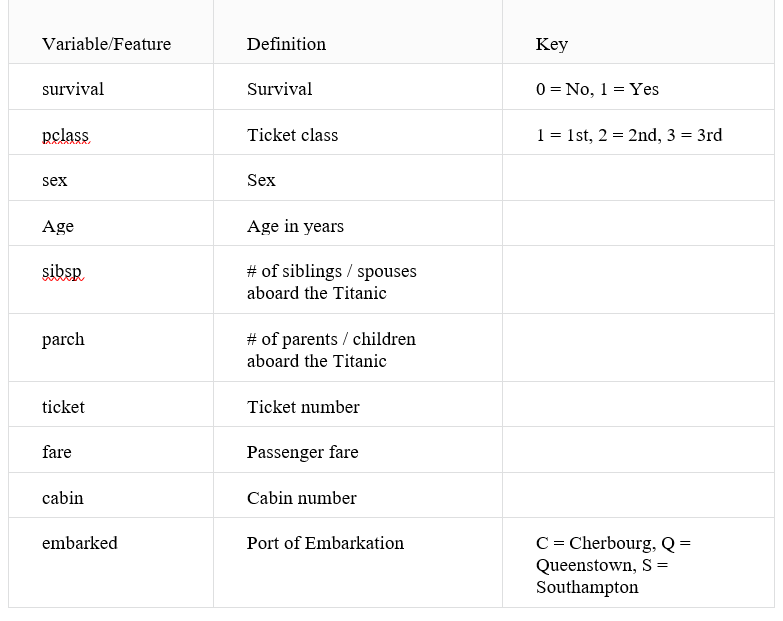
can update our decision tree into the final version.



1. **Implement Python API for Decision Tree.**Note : In below code implementation I choose different data set to demonstrate .   
   Titanic Dataset collected From Here. **: https://www.kaggle.com/c/titanic/data**

Sample Dataset





Code Implementation:

1. Started Jupyter Notebook.
2. Downloaded Titanic Train and Test datasets from Kaggle to local machine

**Train** Dataset : 'E:\\MLPractise\\Datasets\\ titanic\_**train**.csv'

**Test** Dataset : 'E:\\MLPractise\\Datasets\\ titanic\_**test**.csv'

| All import statements Goes here. |
| --- |
| Load Dataset and Print Sample Records . Once inputdataset loaded , now we have whole input data as Pandas DataFrame. |
| Explore about your data. |
| This is One of the essential step .   1. Considered ‘Parch’ , ‘SibSp’ as fetures to build DecisionTree . 2. Created DecisionTree Object. 3. Provided InputFeatures and TargetVariable value as input to DecisionTree Algo.     As we have not given any tuning parmeters while instantiating DecisionTree , Default values it considered. Due to that criterion value take as deault ‘gini’. If we want we can change while construcing DecisionTree object. Same case for other parameters aswell. We will see how to tune these parameter values in upcoming articles. |
| Now we know our decision tree learned the rules based on input data. But my doubt is what kind of rules (Decision Tree) it learned ? Can I print those rules or Tree ?  Yes. Those rules are nothing but our model now. You can print the decision tree by using below code.    Note : For this in your system dot.exe , graphviz such external components need to install.  Output : Below is the tree it leanred from our input data. Now this is our model. |
| Can I test few sample records by using above rules.   * + - Yes .  1. Load Test Data (titnaic\_test.csv) and analyze by calling shape and predict . 2. Prepare testdataframe (titnaic\_test\_data) with only trainined features. 3. Call predict method on model object by giving test data as input. 4. Write the predicted result into submission.csv . |
| Result :    Load this data into Kaggle and check the accuracy. |

Above Code Snippet:

Ref:

<https://people.revoledu.com/kardi/tutorial/index.html>