

# Part 1: Implementation

---

The code is in *Jiashi\_Chen\_Homework\_1.ipynb* file.

The data format that I use to print decision tree is **dictionary**. Key of the dictionary could represent both the value of the attribute and the name of the attribute. Also the value of dictionary represents the next branch or leaf. I also plot the decision tree at the end of part 1.

Code-level optimizations I perform is that I use **recursion** to get a decision tree to keep the code tidy. Also I use the functions to **encapsulate the code**, which will help me improve the reusability of the code.

The most challenge I face is that I am not familiar with numpy package and pandas package, leading to many compilation errors. At the beginning I used a lot of if statements which made the code look messy.

The prediction for (occupied = Moderate; price = Cheap; music = Loud; location = City-Center; VIP = No; favorite beer = No) is **yes**.

# Part 2: Software Familiarization

---

**Sklearn package** offers a good implementation of the decision tree learning algorithm.

## How to use

```
from sklearn import tree

clf = tree.DecisionTreeClassifier(criterion='entropy') #Selecting decision tree
algorithm
clf = clf.fit(attributes_train, label_train) #Fitting training data
label_test = clf.predict(attributes_test) #Predicting test data
```

I used this package to fit the data in *Jiashi\_Chen\_Homework\_1.ipynb* file too.

## Comparsion and Improvement

- It encapsulates functions into the tree class and let people only need to use *DecisionTreeClassifier*, *fit* and *predict* functions to finish the whole process, which is quite convenient and friendly for users. I can also do this, which will benefit my future use and maintenance.
- It passes attributes and labels into the function as two variables while in my code I just input the whole dataset and default the last column to be labels. So my function may lack applicability.
- When calculating entropy, my function only works when there are just two values in the label at

the beginning. Now I have modified the function so that it can adapt to various situations.

## Part 3: Applications of Decision Trees

---

### Daily Life

- Suppose we need to select a flight for our next travel. We will check first if the flight is available on that day or not. If it is not available, we will look for some other date but if it is available then we look for maybe the duration of the flight. If we want to have only direct flights then we look whether the price of that flight is in your pre-defined budget or not. If it is too expensive, we look at some other flights else we book it!

### Software Products

- Email clients use many spam filtering methods. To make sure that these spam filters are constantly updated, they use a lot of machine learning algorithms, because after rule-based spam filtering is complete, it can't track the latest techniques adopted by spammers. Some spam filtering technologies such as multi-layer perceptron and C4.5 decision tree are supported by the machine.