Programming Assignment 1: Decision Trees

Part 1: Group member and contribution

Bin Zhang (5660329599): implementation and display of decision tree without external library (part 2) .

Yihang Chen (6338254416): implementation of decision tree with (sklearn) and without library (part 3).

Hanzhi Zhang (4395561906): Applications of Decision Tree and implementation of decision tree without library (part 4).

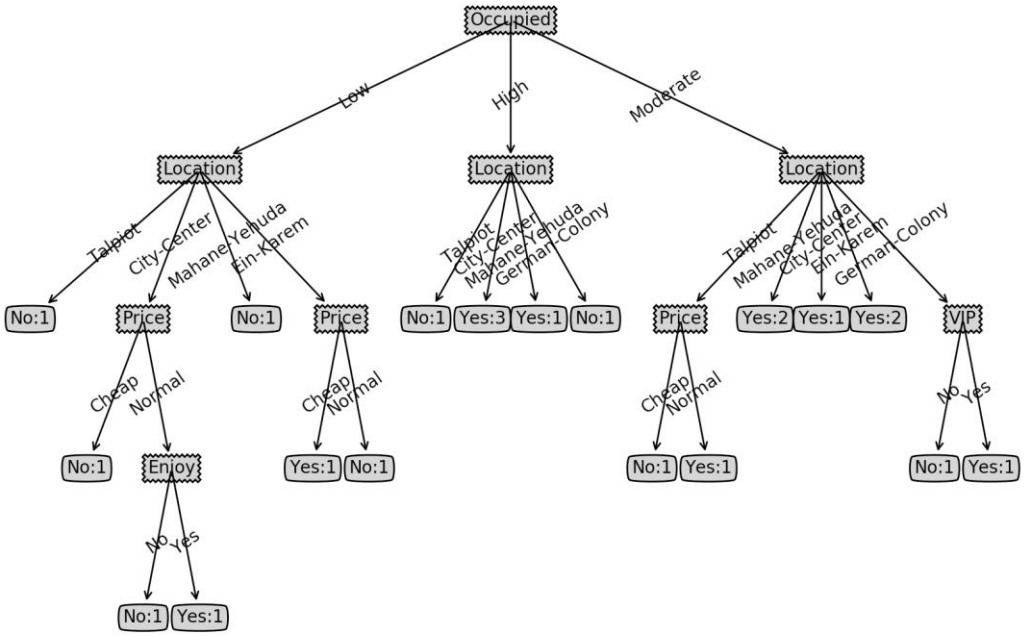
Actually: each of group member has developed their decision tree algorithm on their own, and we discussed each algorithm and finally chose the Bin's decision tree as the final tree to use.

Part 2: Implementation of decision tree

A decision tree uses a tree-like model of decisions and their possible consequences to display an algorithm that only contains conditional control statements. For the sake of convenience, we use nested dictionary rather than building a tree class from scratch to represent it. Here the root key of the dictionary represents the first best attribute that we choose based on information gain and the value of this key is a list of dictionaries, each whose key represents the condition of the attribute, and the structure is like the following Graph a:

Graph a

And when applied into our example dataset, the decision tree is like Graph b:



Graph b

The dictionary's format is:

{'Occupied': {'Low': {'Location': {'Talpiot': 'No:1', 'City-Center': {'Price': {'Cheap': 'No:1', 'Normal': {'Enjoy': {'No': 'No:1', 'Yes': 'Yes:1'}}}}, 'Mahane-Yehuda': 'No:1', 'Ein-Karem': {'Price': {'Cheap': 'Yes:1', 'Normal': 'No:1'}}}}, 'High': {'Location': {'Talpiot': 'No:1', 'City-Center': 'Yes:3', 'Mahane-Yehuda': 'Yes:1', 'German-Colony': 'No:1'}}, 'Moderate': {'Location': {'Talpiot': {'Price': {'Cheap': 'No:1', 'Normal': 'Yes:1'}}, 'Mahane-Yehuda': 'Yes:2', 'City-Center': 'Yes:1', 'Ein-Karem': 'Yes:2', 'German-Colony': {'VIP': {'No': 'No:1', 'Yes': 'Yes:1'}}}}}}

When it comes to code-level optimization, each node of the decision tree is built recursively, so in order to reduce the process time, we use reduce function to calculate each layer's entropy; Also, supposing that there is a subset which has one attribute but lots of label result, we make the rule that return the most frequent label for our tree.

The challenge we meet is the not about algorithm but the data. It is because we use the historical data to build a decision tree but the data is limited, so the decision tree we build cannot cover all the results, so when we come up the first decision tree and use our own test data, the error appears. So we write an error capture. Whenever this record cannot be

decided by the tree, we return “unknown result”.
When we use the decision tree to predict the test data, the result is ‘Yes’.

Part 3: Software Familiarization

The most popular python library for machine learning study is *sklearn*. It offers a function to implement decision tree. The parameter is shown below:

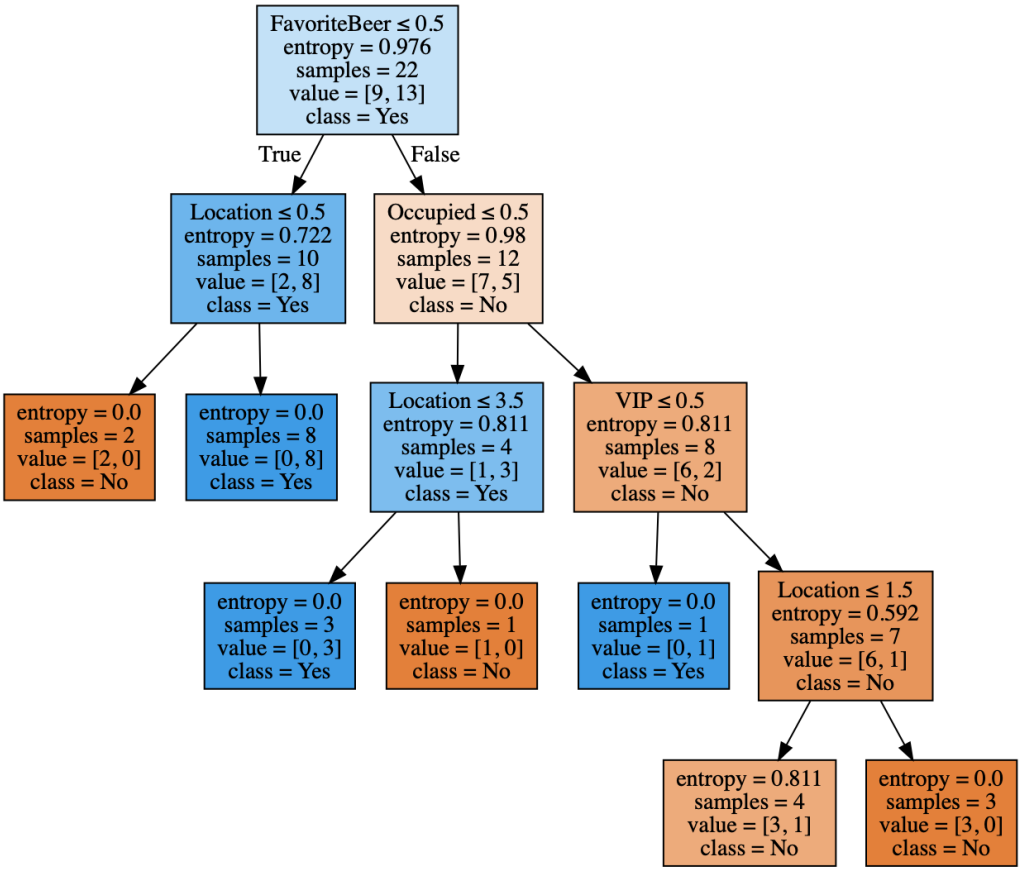
class sklearn.tree.**DecisionTreeClassifier**(*criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None, presort='deprecated', ccp_alpha=0.0*)

For the parameter *criterion,* we can set as “entropy”, so it can build the decision tree using information gain. However, because this classifier uses CART (Classification and Regression Trees) as the algorithms to generate node and tree, it will always split the node into two “child” node. That is to say, the tree it generated is binary tree. As is clarified in the document: *scikit-learn uses an optimised version of the CART algorithm; however, scikit-learn implementation does not support categorical variables for now.* the tree it generated can be used as decision tree, but the threshold in each node may lose the meaning in reality. The value of the attributes can be encoded into numbers, like “blue”: 0, “red”:1, “green”:2. If the threshold is 1.5, it does not represent a specific categorical color. In the color node, it can only be split into node (color: “blue”, red), and node (color: “green”), not three nodes (color: “blue”), (color: “red”), (color: “green”).

However, the decision tree generator in sklearn can set many parameters to control the tree it generated. It can set the max_depth, max_leaf_nodes, which can be used to control the size of the decision tree. We can add some parameters in our program to set the limit of the depth of the recursion. That is to say, we can control the depth of the generating tree. Besides, the value of the attributions are encoded in the sklearn. We

can also encode data to optimize time and space occupation.

The decision tree is like following Graph c by applying the sklearn into the example dataset:



Graph c

Part 4: Applications

Classification

Decision Tree can be used to determine whether the received email is a spam email. We can use the feature including whether it is from correct domain name, content type, whether it is reply message, whether it has subjects, whether subject contains vulgar word and so on. We can collect some spam email and normal email to build a train set.

Heart disease is the main death causes in the world for decades. Decision Tree could help to diagnosing the heart risk (low, medium high) of patients. The age, sex, chest pain type, resting blood pressure, serum cholesterol, fasting blood sugar, resting electrocardiographic, maximum heart rate achieved, exercise induced angina, depression induced by exercise relative

to rest, and the slope of the peak exercise segment as features to determine the heart risk. The patient can use this model t0 obtain their evaluation of heart health quickly without the help of doctor.

Regression

In China, one of the most general problems, that the young generation in big cities should face, is expensive house price. For many families, they have to spend lots of time to determine whether this price of house worth to buy. Therefore, there is a program that uses the information of the house to predict the house price. According to the comparison between predicted house price and actual house price in the housing agency, the price-performance ratio can be obtained to help people make a decision.

The feature of this model contains Linear feet of street connected to property, total size in square feet, number of kitchens, number of bedrooms, number of bathrooms, General shape of property, Original construction date and so on. The labels are the housing price.