Project2: Classification

E54036219 張富嘉

I. Design Data and Rules:

I have designed a set of rules to classify data, and then generate 1200(M=1200) data based on these rules. The classification question is designed as making a decision of hanging out or not according to the weather and time. The data contains 4 features(k=4), which are outlook, humidity, temperature, and time, respectively. In addition, I designed these rules with reasonable and intuitive concept, as the way normal people thinks, in order to bring it closer to the real world. All rules are as follows:

```
1.  If 0 <= time <= 5, then hangingOut = No
2.  If outlook == Rainy, then hangingOut = No
3.  If temperature < 9 or temperature > 32, then hangingOut = No
4.  If temperature > 30 and humidity > 0.85, then hangingOut = No
5.  If temperature < 12 and outlook == Windy, then hangingOut = No
6.  Otherwise, hangingOut = Yes
```

II. Construct Models:

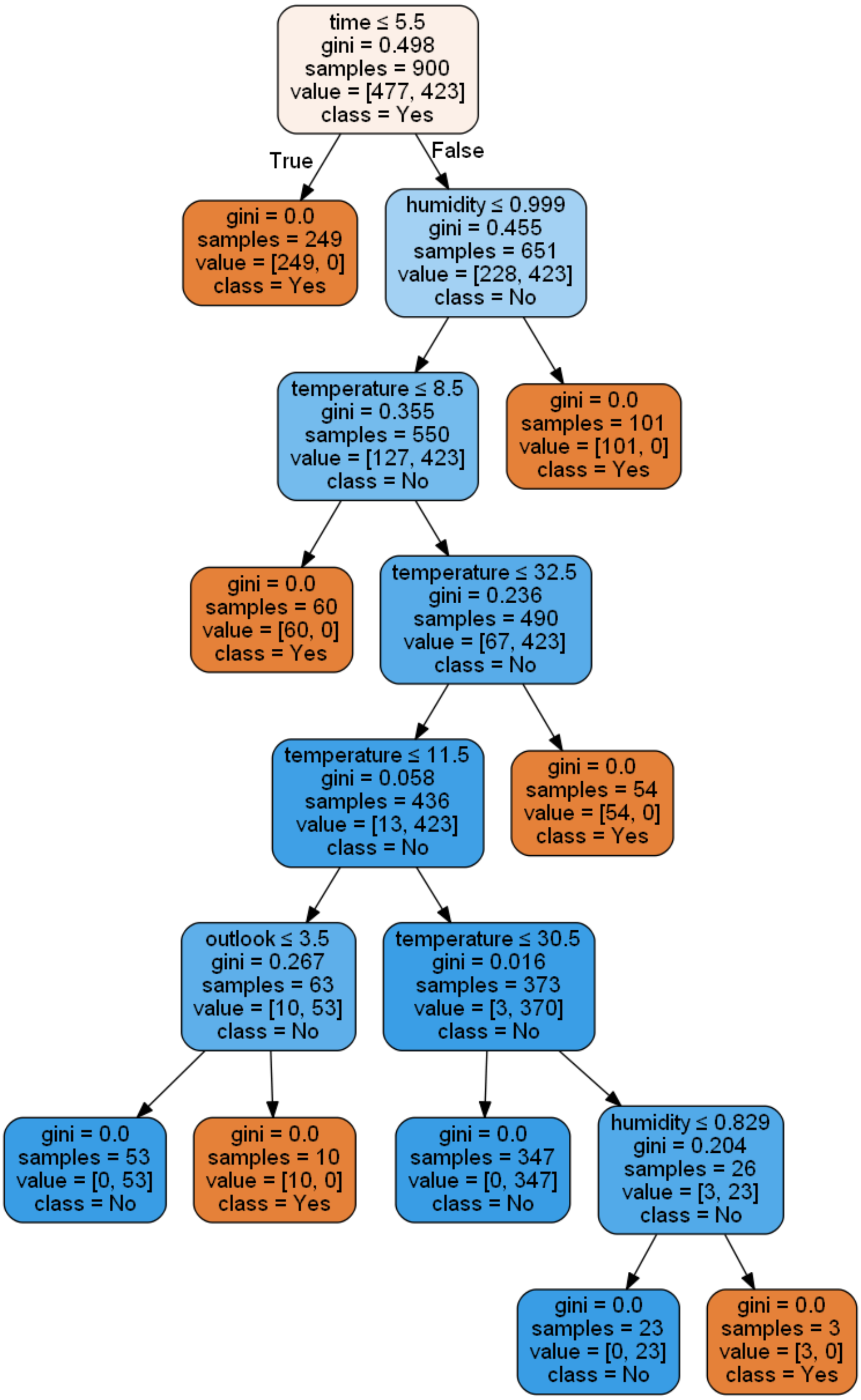
I have tried 4 models based on sickit-learn package on python. Three of the models are only used for testing and comparison, so there is no actual adjustment of the parameters to reach the best precision. The 4th model is the decision-tree classifier. Next, I will briefly talk about decision-tree classifier and compare it with the other models. First is to preprocess the data, including following steps: 1. Encode labels with values. 2. Splitting data into training data and test data. 3. Standardize features by removing the mean and scaling to unit variance. Then we can easily construct a decision-tree classifier model to fit the training data by using sickit-learn package. Finally, I found that the decision-tree classifier can be perfectly fit, according to nearly 100% precision when making prediction on test data. However, other than decision-tree classifier, supported vector classifier(SVC) and Naive-bayes classifier got only about 90% of precision. There are several reasons of getting this result. One possible reason is that SVC is a linear kernel, but this problem is non-linear.

III. Rules Comparison:

The picture on the next page is the decision tree diagram, showing the rules that mined from the data. Obviously, these rules are roughly consistent with the absolutely-right rules. Furthermore, we can easily find out that these rules are more concise compare to absolutely-right rules. Since there are some overlapping (Rainy day, for example, cause the same result as Humidity == 1) in the absolutely-right rules, the decision-tree classifier will consider them as redundancy.

IV. Discussion:

It is easy to evaluate classification rules learning algorithms by generated standard data set. However, these data are normally not close to the real-world data. Therefore, I think that it might be a feasible way to generate data based on the rules that is from the real world and found by decision tree algorithm.



Picture1. Decision-Tree diagram