**Report of Decision Tree**

Authors : Na Li, Yash Naik

In this assignment, we implemented the ID3 decision tree with python3. Na is mainly responsible for pre-processing the dataset, selecting features, and building the decision tree, and making a prediction. Yash is mainly responsible for calculating the entropy, validated the effectiveness of our implementation, and did research on the application of decision trees.

The steps we took are：

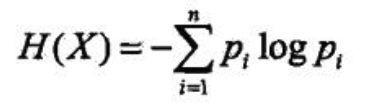
**(1) Pre-processing the dataset:**

****

There are some special characters in this dataset, such as [ ‘(‘, ‘:’, ‘)’, ‘;’], whitespace, etc. so firstly, we removed all these invalid characters from the dataset. Then we used lists data structure to store the attribute and labels respectively, and return them. If necessary, we could also use Pandas to import the dataset as a dataframe.

**(2) Calculate the entropy:**

We used the below formula to calculate it, and the input values are label values.



**(3) Feature Selection and Engineering:**

Firstly, we used the function split\_dataset() to help us split the dataset into children sets, which is much easier for us to calculate the entropy of each feature. Then we used the information gain value to help us pick up the best feature to split.

**(4) Decision Tree Model:**

This is the most challenging part in this assignment, since we used recursion to help us build the tree, and we had to figure out the termination condition, and how to break up the ties among attributes. Luckily, we figured all these out, and stored the tree in a python dictionary, which could easily be converted into a json format.

**Tree:**

**{**

**"Occupied": {**

**"High": {**

**"Location": {**

**"Mahane-Yehuda": "Yes",**

**"Talpiot": "No",**

**"German-Colony": "No",**

**"City-Center": "Yes"**

**}**

**},**

**"Moderate": {**

**"Location": {**

**"Mahane-Yehuda": "Yes",**

**"City-Center": "Yes",**

**"German-Colony": {**

**"VIP": {**

**"Yes": "Yes",**

**"No": "No"**

**}**

**},**

**"Talpiot": {**

**"Price": {**

**"Cheap": "No",**

**"Normal": "Yes"**

**}**

**},**

**"Ein-Karem": "Yes"**

**}**

**},**

**"Low": {**

**"Location": {**

**"Ein-Karem": {**

**"Price": {**

**"Cheap": "Yes",**

**"Normal": "No"**

**}**

**},**

**"Talpiot": "No",**

**"City-Center": {**

**"Price": {**

**"Cheap": "No",**

**"Normal": {**

**"Favorite Beer": {**

**"Yes": "Yes",**

**"No": "No"**

**}**

**}**

**}**

**},**

**"Mahane-Yehuda": "No"**

**}**

**}**

**}}**

**(5) Predict:**

After we trained the data, we printed out the decision tree in json format, and made a prediction based on input test data.

The test data included the following attributes:

**Occupied = Moderate**

**price = Cheap**

**music = Loud**

**location = City-Center**

**VIP = No**

**favorite beer = No**

**Enjoy = [To Predict]**

The result of our prediction is **“yes”**.

**(6) Conclusion:**

After predicting the given scenario, we can further validate the effectiveness of our ID3 algorithm using ‘DecisionTreeClassifier’ class from ‘Sklearn’ library. However, one important thing to note is that this DecisionTreeClassifier only works on numeric data and not on ‘String’ categories like we are dealing with in this assignment. To tackle this issue we need to convert the given attributes into numeric data using ‘OneHotEncoder’. Following this, we can now fit our X and y labels into the DecisiontreeClassifier and get a prediction on the test data described above.

There are many applications of Decision trees in the real-world. Few such advantages are handling data which is non-linear in nature, handling categorical data, continuous variable data, etc.