**Implementation of Decision Tree**

**Project link:** <https://github.com/AviBomjan/31005-Advanced-Data-Analytics-Algorithms-Machine-Learning---Spring-2020> (if you have an issue opening the code, please view it raw. The raw button can be found at the row with the line numbers)

<https://colab.research.google.com/drive/12JoEA7q6tM07KTCg_E1-6eFtHVY4cJLR?usp=sharing> (Link to the colab file)

**Introduction**

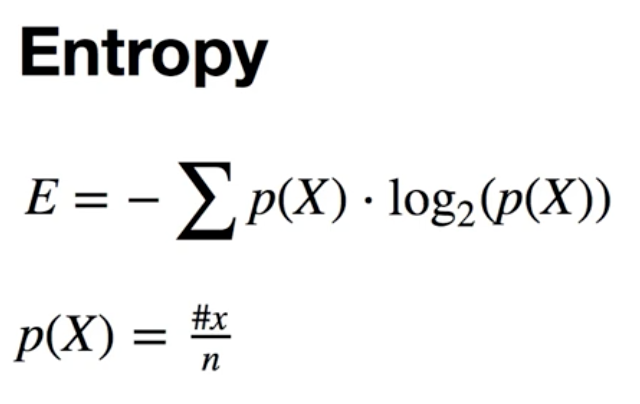
Decision tree is a decision support tool. It is structured and modeled like a tree. It includes possible consequences and chance event outcomes. It also includes resource costs and utility. A decision tree would be commonly used for predictive analysis. The decision tree is built from top to down. It’s main characteristics are internal nodes, branches and a terminal node. The internal nodes contains an attribute called test. The branches contains the conclusion of the previous test. The terminal nodes or the leaves classifies them with a label. Decision tree is the most popular algorithm as a supervised model. The decision Tree can be used both for classification and regression. The reason why decision tree is popular is because it is very stable and reliable.

I choose the dataset Iris for this assignment and I choose to implement a decision tree to tackle this dataset. I tested the dataset with liner and logistic regression but felt that the decision tree had given me the more reliable and stable accuracy compared to them.

**Decision Tree Algorithm**

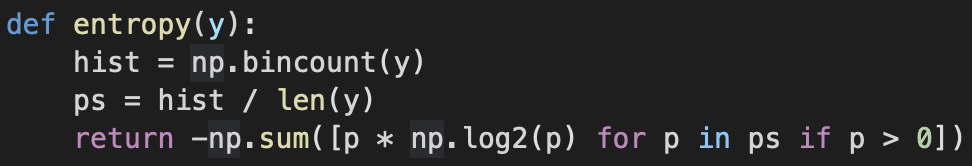
The algorithm consists of 2 modules: entropy computation and information gain computation.

**Entropy**

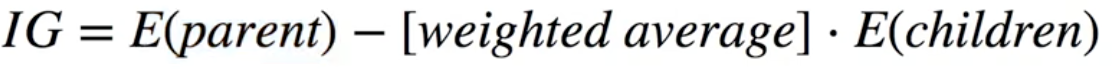


Entropy is used to measure undetermined. The formula that is I use for this assignment is - sum of p of x times log of p of x. The second formula is p of x equals to number of occurrences over the total number of samples

The entropy is located at section 3 as line 14. I calculated the number of occurrences with the function hist. “hist” utilises the numpy of bincount of y. To calculate p of x, I divided the hist by the length of y and named it as the variabl ps. Now that we have all the variables. To calculate the entropy i minused the numpy of sum and used list comprehension of p times numpy log 2 of p for all p in ps. I also implemented a condition that the it would only return a value if p is greater than 0 to prevent any errors.

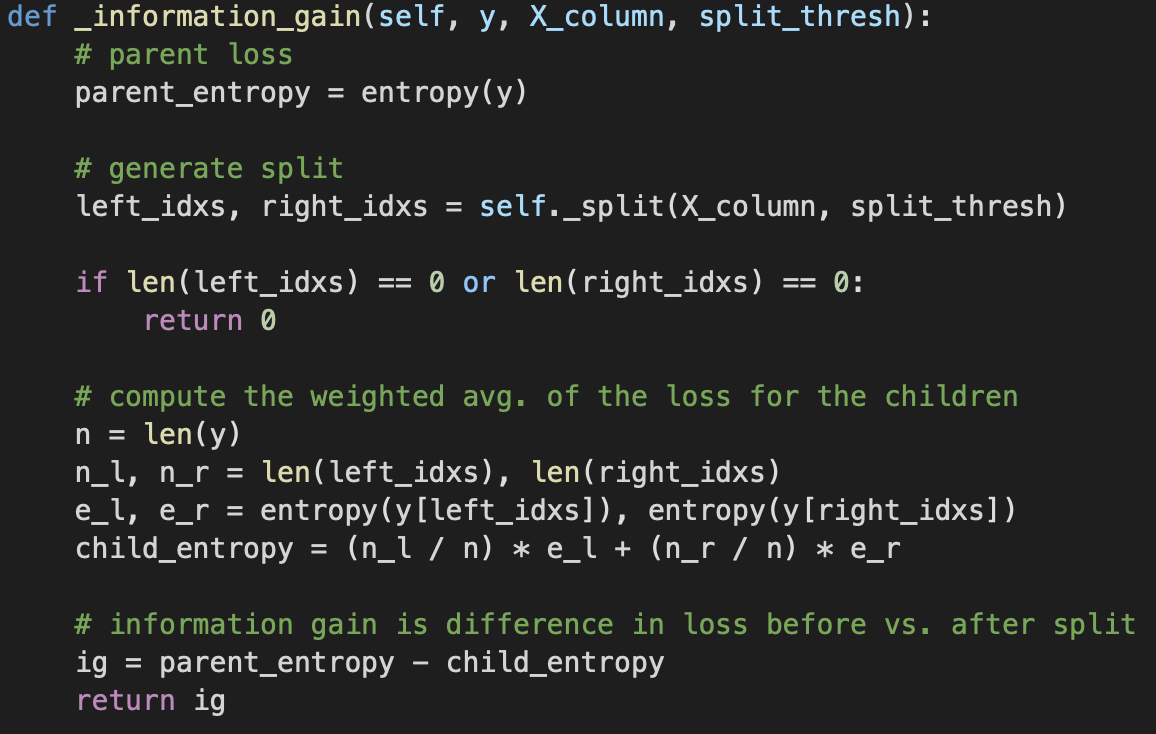


**Information Gain**



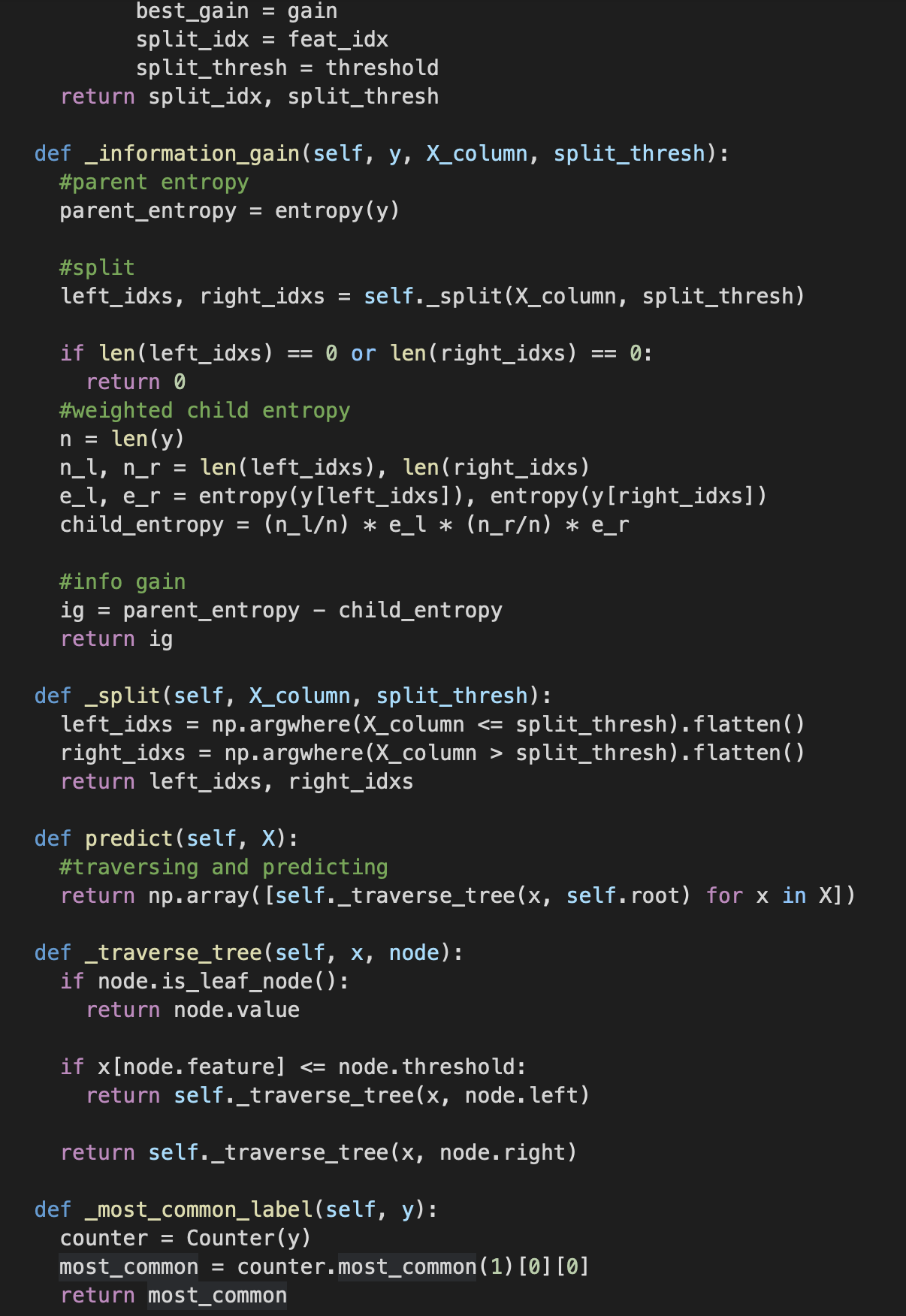
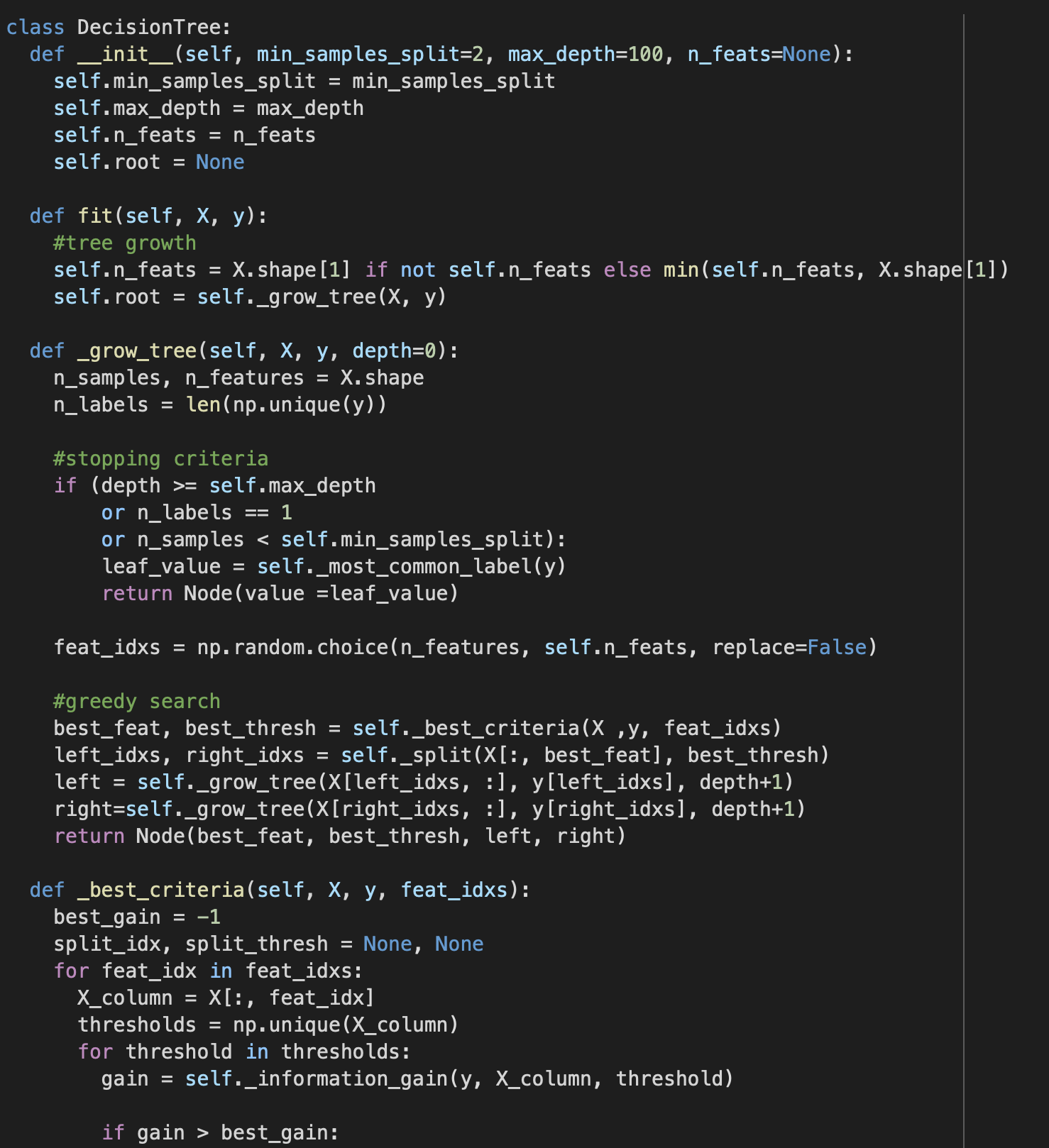
Information gained is calculated by entropy of the parent minus a weighted average of all the child entropy.

I used the code as shown below to calculate the information gain.



**Decision Tree**

This is the code of the Decision Tree that i choose to implement to this assignment.

****

**Data Preparation**

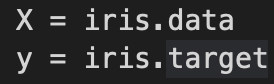
The dataset that i used for this assignment is the Iris Dataset. The datatype of the iris is classified as Bunch. The feature names are sepal length (cm), sepal width (cm), petal length (cm), petal width (cm). The target names are setosa, versicolor and virginica.

The first preprocessing task for the dataset was to import the dataset. The second task would be to load it. The third task would be to classify the variables with the data and the target. The fourth task would be to classify the the arrays. The fifth task would be to classify the train and test variable and determine the size and the random state.

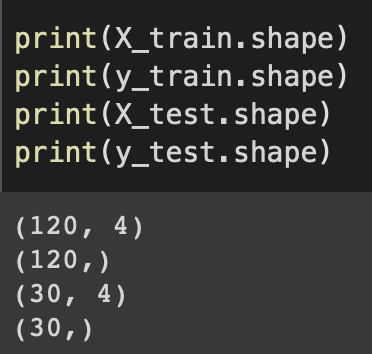
I split the data into test size as 0.2 and random state as 1.

**Experiment Design and Evaluation**

Inputs and targets are needed for the implementation of the decision tree. I named them as shown below:



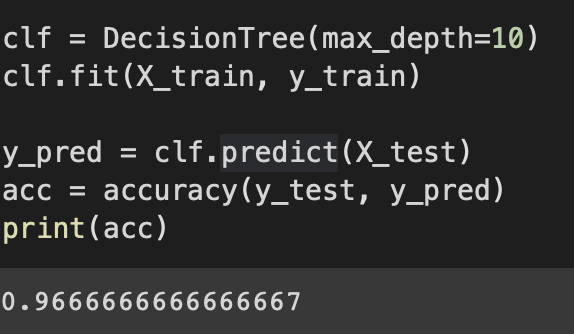
I split the train and test portions. As shown below:



The expected result of the accuracy for the trained set should be as close as possible to the number 1.

**Evaluation Results**

After implementing everything, I ran the accuracy test. The result of accuracy for the train set was 0.9666666666666667. That accuracy result is very good. It’s inaccurate rate is 0.03333333333333.

****

**Conclusion**

In conclusion, the decision tree is very useful to classifying the data. It also very reliable and conclusive. In this case, the decision tree handles the iris dataset very well. Although creating a decision was very difficult for me, it was well worth the effort because it was able to become a very good tool.