**Decision Tree Classifier**

110505259 陳柏燊

**Function implementation**

* \_feature\_split

If np.unique(y) == 1 means that’s all instance in the node have the same target, so we don’t need to split.

O.W. we go through all the feature and try all the treadhold to find which split method will get the least entropy and return the index of the split feature and the threadhold.

* \_build\_tree



If the depth didn’t exceed the limitation, we call \_feature\_split function to find how to split the tree. IF the return value aren’t NULL we split the data and call \_build\_tree function to find the left and right children nodes of this node and link to it.

* \_find\_min\_alpha



By the definition of alpha, it is numerical error to define an alpha of a leave node since it well cause to divide something with “Zero”. So we return “inf” with the leave node.

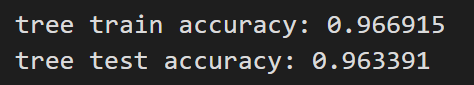
Otherwise, we computer the alpha of now node and compare with the minimum alpha of the left sub tree and right sub tree which is calculate by calling \_find\_min\_alpha.

* \_prune



We use \_find\_min\_alpha to find which node we should cut and cut it.

**Decision tree before post-pruning accuracy (depth = 8, random\_state = 1)**



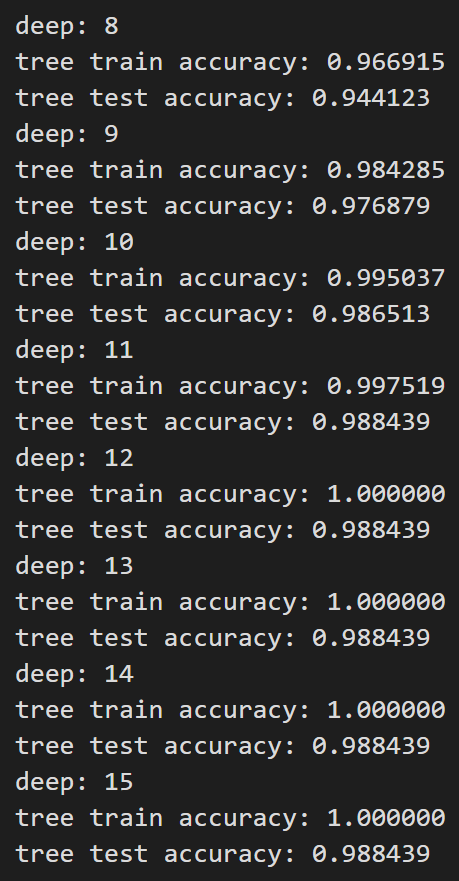
**Decision tree after post-pruning accuracy (dep = 8, random\_state = 1)**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Cut | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| trainac | 0.966915 | 0.966915 | 0.966915 | 0.966915 | 0.966915 | 0.965261 | 0.964433 | 0.961125 | 0.955335 | 0.951199 |
| test ac | 0.963391 | 0.963391 | 0.955684 | 0.955684 | 0.955684 | 0.949904 | 0.95183 | 0.95183 | 0.94605 | 0.942197 |

**The effect of different parameters**

* Depth

Depth = 8 to 15 (random\_state = 1)

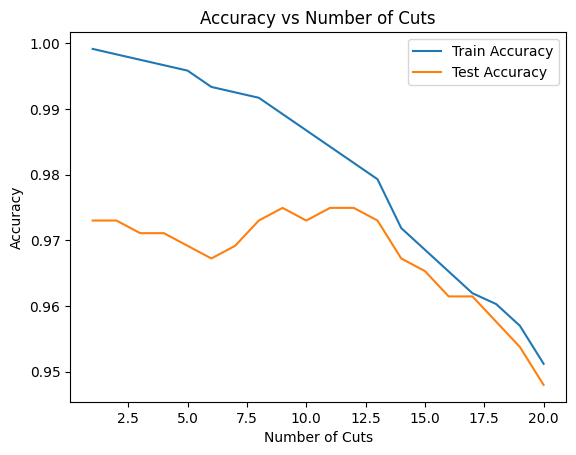


We can observe that if we make tree go more deeper, we will finally get pefect predition on training data.

Testing data accuracy will also increase but will have a upper boundary.

* Prune\_times

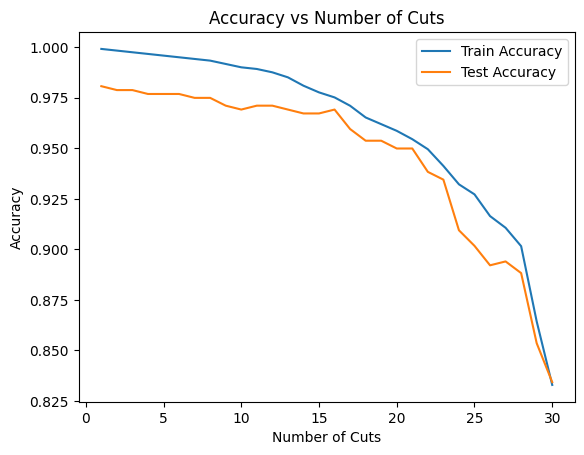
Depth = 15, random\_state = 1355373270



We can see training accuracy keep decrease with more cut. When test accuracy increase a little and decrease. We consider the test accuracy with 5 cut lower with 10 cut cause by overfitting.

**Conclusion**

It hard to find the random\_state to present overfitting. I have try a lot of random\_state that most the of the different split with testing data and training data lead to testing accuracy and testing accuracy both decrease (like the picture below).

(random\_state = 1903942106)

In my opinion, it seems to be cause by the data size. Maybe some bigger data set will lead to present overfitting easier.