CMPT 459: Data Mining, Spring 2024

Assignment 1

Tushrima Kelshikar 301357928

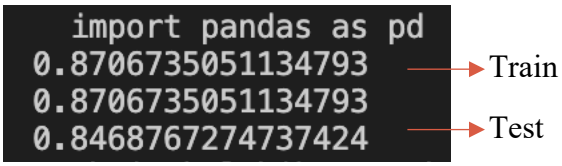
---

a) Pseudocode

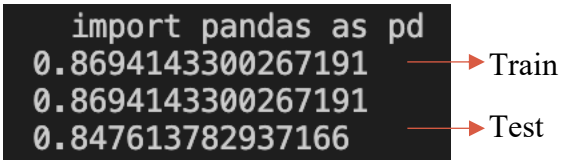
```
function LearnDecisionTree(node n, dataset D, attributes A):
    if termination condition(A,D) then
        n.label := majority label in D
    else
        for all a in A do
            q(a) := quality of splitting D using a
            b := attribute a with maximum q(a)
            n.attribute := b
            p := number of partitions for attribute b
            for i from 1 to p do
                Di := {d in D|d belongs to partition i of b}
                create new node ni as child of n
                LearnDecisionTree(ni,Di,A-{b})

function quality of splitting D using a:
    if a is categorical then
        calculate Gini index for each partition of a
        return minimum Gini index
    else if a is numeric then
        calculate Gini index for each partition of a
        return minimum Gini index

function termination condition(A,D):
    if number of distinct classes in D is 1 then
        return true
    else if number of attributes in A is 0 then
        return true
    else
        return false

function number of partitions for attribute b:
    if b is categorical then
        return number of distinct values in b
    else if b is numeric then
        calculate optimal number of partitions for b
        return optimal number of partitions
```

b) Random Forest

1. python3 main.py --n-classifiers 10 --train data/train.csv --test data/test.csv --criterion gini --maxdepth 10 --min-sample-split 20 --max-features 11



2. python3 main.py --n-classifiers 10 --train data/train.csv --test data/test.csv --criterion entropy --maxdepth 10 --min-sample-split 20 --max-features 11



3. The accuracy of the training data may not be 100% due to several reasons:

   Complexity of the Data: The data might not be perfectly separable by the features used for training. There could be overlapping regions between classes, making it impossible for any model to achieve 100% accuracy.

   Noise in the Data: The presence of noise or outliers in the data can affect the model's ability to perfectly fit the training data.

   Model Complexity: The model might not be complex enough to capture all the nuances and intricacies of the data, especially if the decision boundaries are highly complex.

a. ***Is there any way to train a tree that yields 100% accuracy on the training dataset? How would that affect the accuracy of the test data?***

While it's theoretically possible to train a decision tree that achieves 100% accuracy on the training dataset, especially if the tree is allowed to grow without any constraints (i.e., unlimited depth), doing so may lead to overfitting. Overfitting occurs when the model learns to memorize the training data instead of generalizing well to unseen data. As a result, the accuracy of the test data is likely to be lower than the training data, as the model may fail to generalize to new data points.

b. ***What parameters should you change to get a tree with perfect training accuracy? Explain if such a classification model has a high variance or bias?***

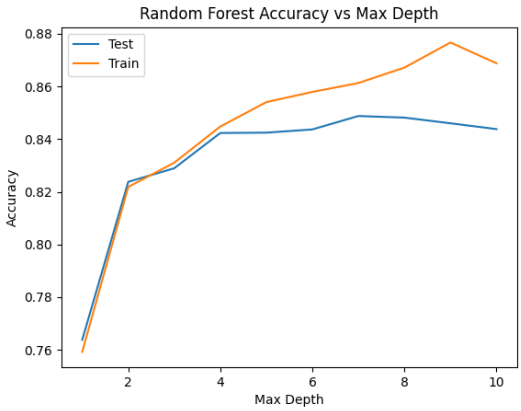
To achieve perfect training accuracy, one could potentially adjust several parameters:

Max Depth: Increasing the maximum depth of the tree allows it to make more complex decisions, which could potentially lead to better fitting of the training data. However, this might also increase the likelihood of overfitting.

Min Samples Split: Decreasing the minimum number of samples required to split a node might allow the tree to create more decision nodes, potentially leading to better fitting of the training data. Again, this could increase the risk of overfitting.

Max Features: Increasing the maximum number of features considered for each tree might allow the model to capture more information from the data, potentially improving training accuracy. However, this could also increase the risk of overfitting.

4. As the max_depth increases, the model becomes more complex, potentially leading to overfitting on the training data. There is an increase in accuracy on the training data with increasing max_depth. However, this may not lead to higher accuracy on the test data due to overfitting.



max_depth	train_scores	test_scores
1	0.7591904425539756	0.7637737239727289
2	0.8218420810171678	0.8237823229531356
3	0.8310555572617548	0.8289417111971009
4	0.8447836368661896	0.8423315521159634
5	0.8540585362857406	0.8424543946932007
6	0.8579281963084672	0.8436828204655734
7	0.8613064709314825	0.8487807874209201
8	0.8671723841405362	0.8481665745347338
9	0.8766929762599429	0.8460168294330815
10	0.8688000982770799	0.8438056630428107

References

Lecture notes, Martin Ester

https://www.geeksforgeeks.org/decision-tree-implementation-python/

https://medium.com/@cristianleo120/building-random-forest-from-scratch-in-python-16d004982788

https://towardsdatascience.com/improving-random-forest-in-python-part-1-893916666cd

https://www.kaggle.com/code/faressayah/decision-trees-random-forest-for-beginners