CS412 HW4 Report

Pengyu Cheng

Netid: pcheng11

- ## For Decision Tree:

  I use GINI index to full split the root until there is no attribute left. There is no tuning for this part as I have tried to cut the tree depth but the accuracy stays relatively the same. In the process of debugging and building the tree, it occurs to me that in the led.train there are many rows which have exactly the same attributes but different class labels so I use the majority vote in this case. In other words, if a tree node has no attribute to split but the class labels are distinct then we use majority vote to choose the class label. For all the other data sets, I fully split the root until there is no attribute left.

- ## Below is the model evaluation for decision tree.

balance.scale data set (test set)

Overall Accuracy: 0.6267

Class Label	Accuracy	Specificity	Precision	Recall	F-1 Score	F-2 Score	F-0.5 Score
1	0.8267	0.9163	0	0	UNDEF	UNDEF	UNDEF
2	0.7244	0.7561	0.7000	0.6863	0.6931	0.6890	0.6972
3	0.7022	0.7016	0.6574	0.7030	0.6794	0.6934	0.6660

led data set (test set)

Overall Accuracy: 0.8589

Class Label	Accuracy	Specificity	Precision	Recall	F-1 Score	F-2 Score	F-0.5 Score
1	0.8589	0.8966	0.7705	0.7749	0.7727	0.7740	0.7714
2	0.8589	0.7749	0.8988	0.8966	0.8977	0.8970	0.8984

nursery data set (test set)

Overall accuracy : 0.9737

Class Label	Accuracy	Specificity	Precision	Recall	F-1 Score	F-2 Score	F-0.5 Score
1	0.9747	0.9810	0.9612	0.9618	0.9615	0.9617	0.9613
2	0.9834	0.9907	0.6788	0.7154	0.6966	0.7078	0.6858
3	0.9903	0.9949	0.9888	0.9805	0.9846	0.9821	0.9872
4	1	1	1	1	1	1	1
5	0.9990	0.9990	0	UNDEF	UNDEF	UNDEF	UNDEF

synthetic.social data set (test set)

Overall accuracy : 0.4760

Class Label	Accuracy	Specificity	Precision	Recall	F-1 Score	F-2 Score	F-0.5 Score
1	0.7240	0.8169	0.4846	0.4701	0.4773	0.4730	0.4817
2	0.7360	0.8411	0.4570	0.4122	0.4335	0.4205	0.4473
3	0.7570	0.8307	0.4779	0.5129	0.4948	0.5055	0.4845
4	0.7350	0.8121	0.4815	0.5098	0.4952	0.5039	0.4869

balance.scale data set (training set)

Overall Accuracy: 1

Class Label	Accuracy	Specificity	Precision	Recall	F-1 Score	F-2 Score	F-0.5 Score
1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1

led data set (training set)

Overall Accuracy: 0.8596

Class Label	Accuracy	Specificity	Precision	Recall	F-1 Score	F-2 Score	F-0.5 Score
1	0.8596	0.8992	0.7708	0.7696	0.7702	0.7698	0.7706
2	0.8596	0.7696	0.8986	0.8992	0.8989	0.8991	0.8987

nursery data set (training set)

Overall accuracy : 1

Class Label	Accuracy	Specificity	Precision	Recall	F-1 Score	F-2 Score	F-0.5 Score
1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1

synthetic.social data set (training set)

Overall accuracy : 1

Class Label	Accuracy	Specificity	Precision	Recall	F-1 Score	F-2 Score	F-0.5 Score
1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1

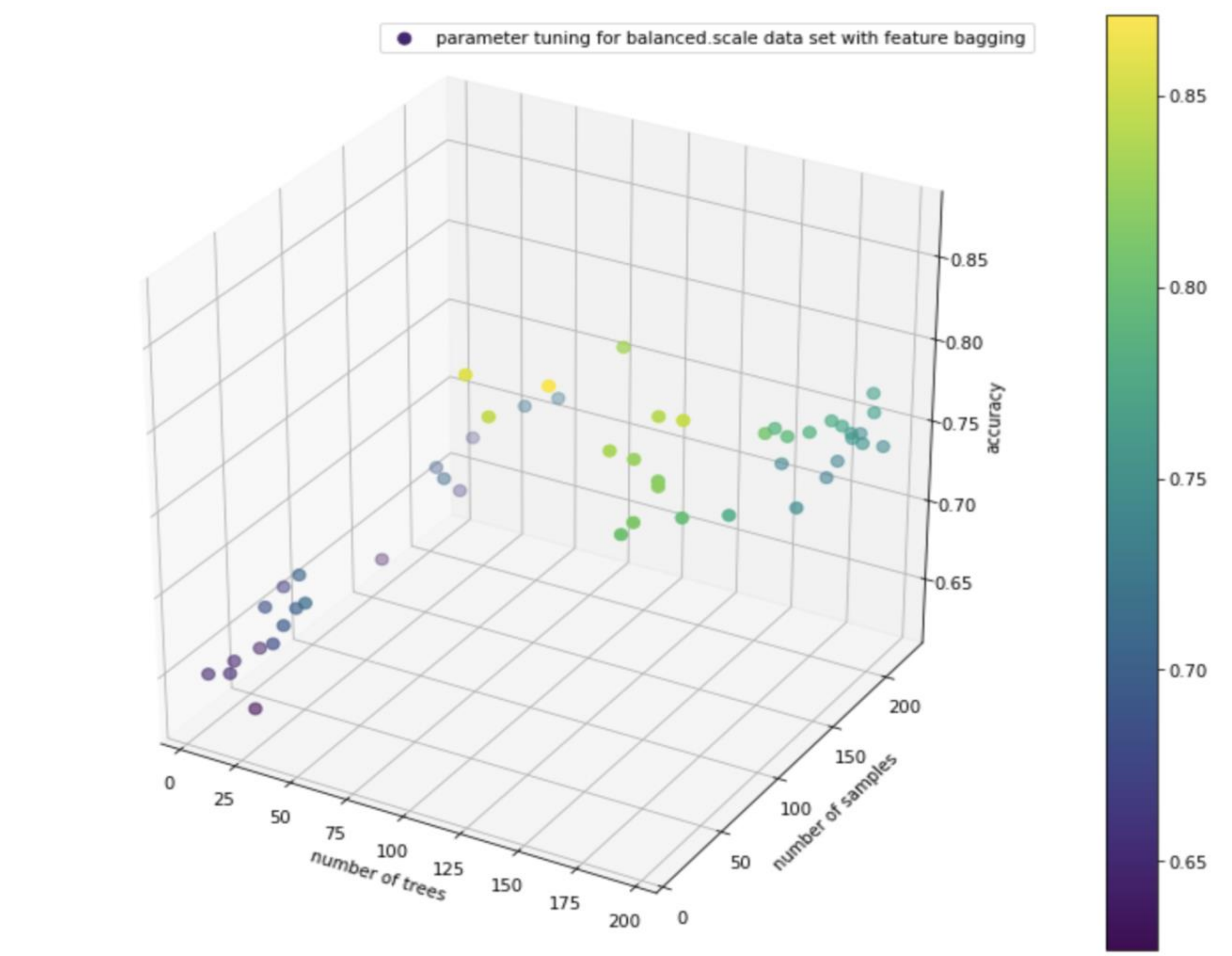
- **For Random Forest:**

Firstly, before building each tree, I use data bagging instead of bootstrap method as I observe that randomly choosing samples of relatively smaller size with replacement achieves a better overall accuracy.

Secondly, I also use feature bagging for each tree node. I randomly select square root of the original number of attributes and then use GINI index to split them. Note that although three of the data set have relatively less attributes, from my tuning process, turning feature bagging option on somehow increases the accuracy by a small portion and this is still to be studied further for me.

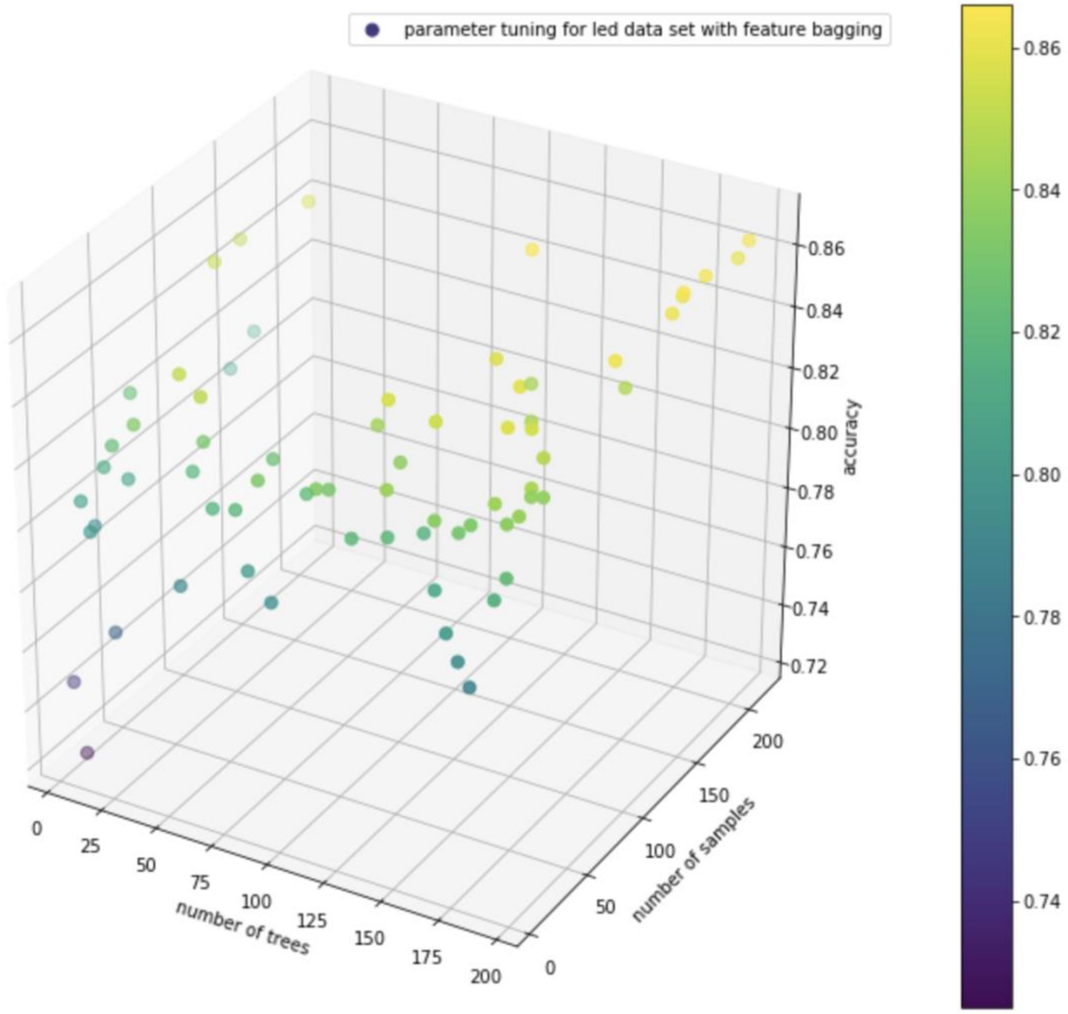
Thirdly, for the number of trees in a certain forest, I create some visualization graph for your better understanding. As I observe, with the number of trees increasing and the sample size staying at a relatively small level comparing to the size of its test data set, the overall accuracy arrives at the peak and converges.

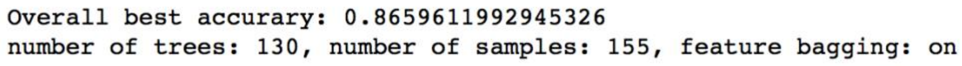
◊ For balance.scale data set, using feature bagging, I set the number of trees to be 150 and number of samples to be 20 to boost my overall accuracy. The direct tuning process can be viewed below. Note that this overall accuracy is consistent within 6% of this peak.



```
Overall best accurary: 0.8711111111111111
number of trees: 150, number of samples: 20, feature bagging: on
```

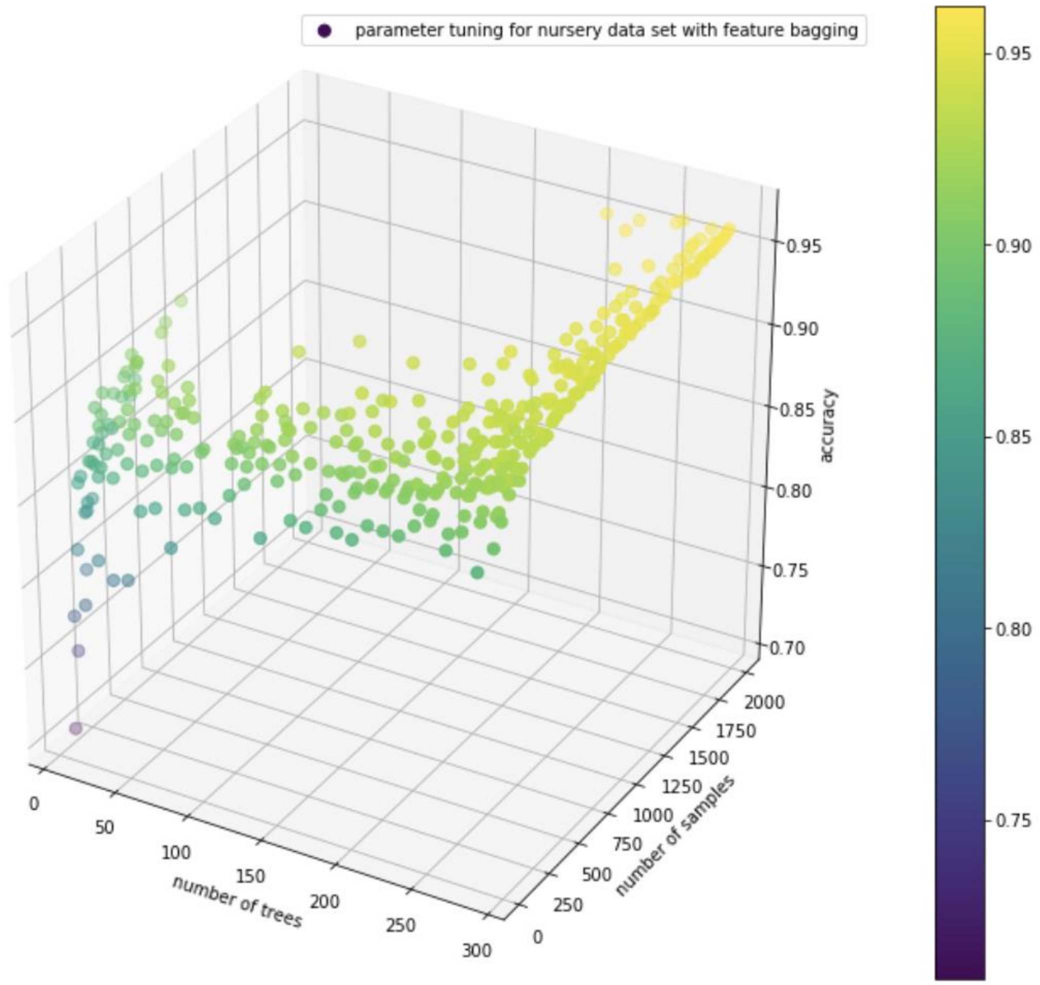
◊ For led data set, using feature bagging, I set the number of trees to be 130 and number of samples to be 155. The direct tuning process can be viewed below. Note that this overall accuracy is consistent within 5% of this peak.

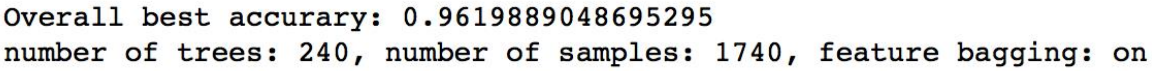




```
Overall best accurary: 0.8659611992945326
number of trees: 130, number of samples: 155, feature bagging: on
```

◊ For nursery data set, using feature bagging, I set the number of trees to be 240 and number of samples to 1740. The direct tuning process can be viewed below. Note that this overall accuracy is consistent within 2% of this peak.





```
Overall best accurary: 0.9619889048695295
number of trees: 240, number of samples: 1740, feature bagging: on
```

- For synthetic.social data set, using feature bagging, I set the number of trees to be 300 and number of samples to 2000. Note that this overall accuracy is consistent within 3% of this peak.

  I did not have a graph for tuning process of this data set as it is time consuming to perform such tuning process.

  I start with 10 trees and 100 sample size and the accuracy is about 44.7%. When I increase the number of trees to 50 with the same sample size, the accuracy improves to about 63%. I think that the accuracy will increase with more trees but the relationship with sample size is still to be determined so I tried 50 trees with 1000 sample size and get the accuracy to be 67.5%. From here I conclude that the accuracy of this data set has more weight in the number of trees rather than with the sample size, although increasing sample size indeed boost up the accuracy a little bit. Because of the time limit of the auto grader, I finally settle down at the parameters provided above.

- **Below is the model evaluation for random forest**

balance.scale data set (test set)

Overall Accuracy: 0.8356

Class Label	Accuracy	Specificity	Precision	Recall	F-1 Score	F-2 Score	F-0.5 Score
1	0.9022	1	UNDEF	0	UNDEF	UNDEF	UNDEF
2	0.8844	0.8780	0.8585	0.8922	0.8750	0.8852	0.8650
3	0.8844	0.8226	0.8151	0.9604	0.8818	0.9273	0.8406

led data set (test set)

Overall Accuracy: 0.8651

Class Label	Accuracy	Specificity	Precision	Recall	F-1 Score	F-2 Score	F-0.5 Score
1	0.8651	0.9208	0.8075	0.7407	0.7727	0.7532	0.7932
2	0.8651	0.7407	0.8879	0.9208	0.9041	0.9140	0.8943

nursery data set (test set)

Overall accuracy = 0.9579

Class Label	Accuracy	Specificity	Precision	Recall	F-1 Score	F-2 Score	F-0.5 Score
1	0.9579	0.9511	0.9065	0.9718	0.9380	0.9580	0.9188
2	0.9762	1	1	0.1077	0.1945	0.1311	0.3764
3	0.9817	0.9865	0.9707	0.9714	0.9710	0.9713	0.9708
4	1	1	1	1	1	1	1
5	1	1	UNDEF	UNDEF	UNDEF	UNDEF	UNDEF

synthetic.social data set (test set)

Overall accuracy = 0.7910

Class Label	Accuracy	Specificity	Precision	Recall	F-1 Score	F-2 Score	F-0.5 Score
1	0.8970	0.9426	0.8313	0.7724	0.8808	0.7835	0.8188
2	0.8890	0.9417	0.8018	0.7265	0.7623	0.7404	0.7855
3	0.9040	0.9323	0.7833	0.8103	0.7966	0.8048	0.7886
4	0.8920	0.9047	0.7543	0.8549	0.8015	0.8327	0.7725

balance.scale data set (training set)

Overall Accuracy: 0.8825

Class Label	Accuracy	Specificity	Precision	Recall	F-1 Score	F-2 Score	F-0.5 Score
1	0.9325	1	UNDEF	0	UNDEF	UNDEF	UNDEF
2	0.9250	0.9346	0.9239	0.9140	0.9189	0.9160	0.9219
3	0.9075	0.8451	0.8472	0.9786	0.9082	0.9492	0.8706

led data set (training set)

Overall Accuracy: 0.8572

Class Label	Accuracy	Specificity	Precision	Recall	F-1 Score	F-2 Score	F-0.5 Score
1	0.8572	0.9103	0.7833	0.7367	0.7593	0.7456	0.7735
2	0.8572	0.7367	0.8870	0.9103	0.8985	0.9055	0.8916

nursery data set (training set)

Overall accuracy = 0.9886

Class Label	Accuracy	Specificity	Precision	Recall	F-1 Score	F-2 Score	F-0.5 Score
1	0.9888	0.9860	0.9722	0.9944	0.9832	0.9899	0.9766
2	0.9914	0.9999	0.9923	0.6515	0.7866	0.6996	0.8983
3	0.9974	0.9973	0.9940	0.9976	0.9958	0.9969	0.9947
4	1	1	1	1	1	1	1
5	0.9998	1	UNDEF	0	UNDEF	UNDEF	UNDEF

synthetic.social data set (training set)

Overall accuracy = 1

Class Label	Accuracy	Specificity	Precision	Recall	F-1 Score	F-2 Score	F-0.5 Score
1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1

- **Conclusion**

From the above tuning process, I find that although using random forest achieves a better overall accuracy for balance.scale and synthetic.social data set, this method is not suitable for the other two.

One plausible explanation is that the decision tree built for balance.scale is sensitive to overfitting. It is unstable for the unseen values of an attribute and can lead to low accuracy in such case. Random forest is much better for this kind of situation as the randomness mitigates this kind of issue.

For synthetic.social, there are 128 attributes and as I observe, many of the nodes have a label when there are almost 100~110 attributes which have not been split yet. So the decision tree built for this kind of data set is extremely unstable and thus random forest is more suitable for data set with a lot of attributes.