Assignment 3A

Priyadarshi Hitesh

27 March 2020

1 Decision Tree and Random Forest

1.1 Decision Tree Construction

In this decision tree is constructed where at each node the attribute through which we are getting maximum information gain is used to split the tree further. The stopping criteria is when the tree cannot be split on any of the attribute or basically entropy is zero after splitting the dataset on any particular node.

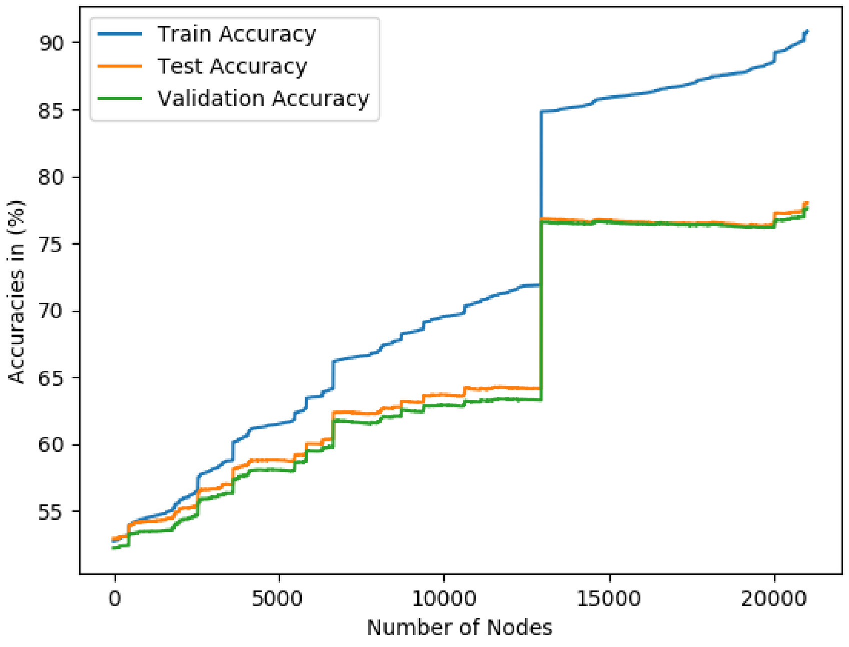

Figure 1: On the go Accuracy

1.1.1 Observations

1. Accuracy of training data is almost 90%
2. Accuracy of test and validation data is almost same and near 78%
3. Train accuracy is considerably high than test and validation tells us that our decision tree is overfitting the data.
4. It also tells us the importance of validation accuracy as test accuracy is almost close to validation accuracy, so we can't tell how our model would perform even if we have achieved the best train accuracy.
5. When number of nodes is approx 13000, there is sudden spike in accuracy of all train, test and valid. This is because now we have started to build the right sub-tree of the data as well, through which we have got the sudden spike in accuracy.
6. Our model is overfitting the data because it has fit the noise of the data as well.

1.2 Decision Tree Post Pruning

Since test accuracy was very less even after we had got best train accuracy, it clearly tells us that this is the case of over-fitting. So we will use post pruning to remove over-fitting from the tree obtained above. We will use validation accuracy as a parameter for post-pruning. We will continue to prune the nodes of the tree until there is decrease in validation accuracy.

I have used two approach for post pruning

1. **Bottom Up Approach** : Bottom up(post order) greedy approach for post pruning. We will traverse the tree in post order and check for every non-leaf nodes if making it a leaf would increase our current validation accuracy, we will do it until we have traversed the whole tree.

2. **Iterative Approach** : In this out of all the non-leaf nodes present in the tree, i am calculation validation accuracy after making every non-leaf node leaf, one at a time. The node through which i am getting maximum increasing in validation accuracy is considered to be used for pruning. This will be done until no such non-leaf node is present that would increase the validation accuracy.

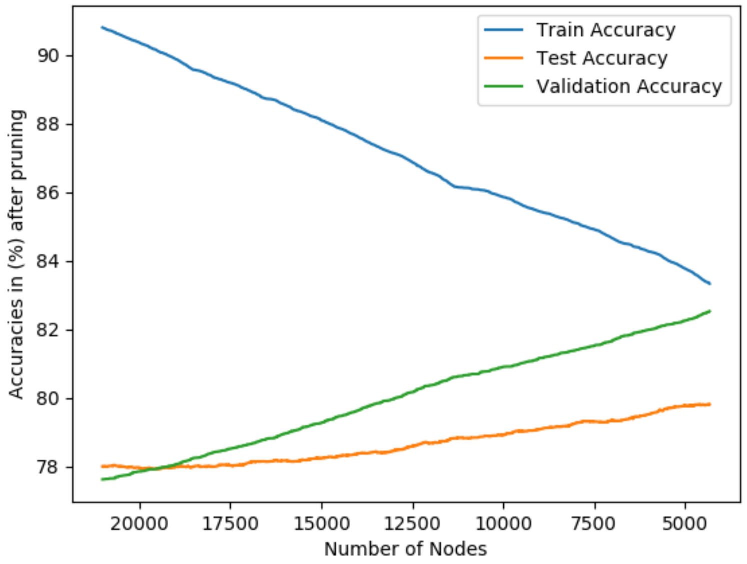

Figure 2: Accuracies after post pruning using bottom up approach

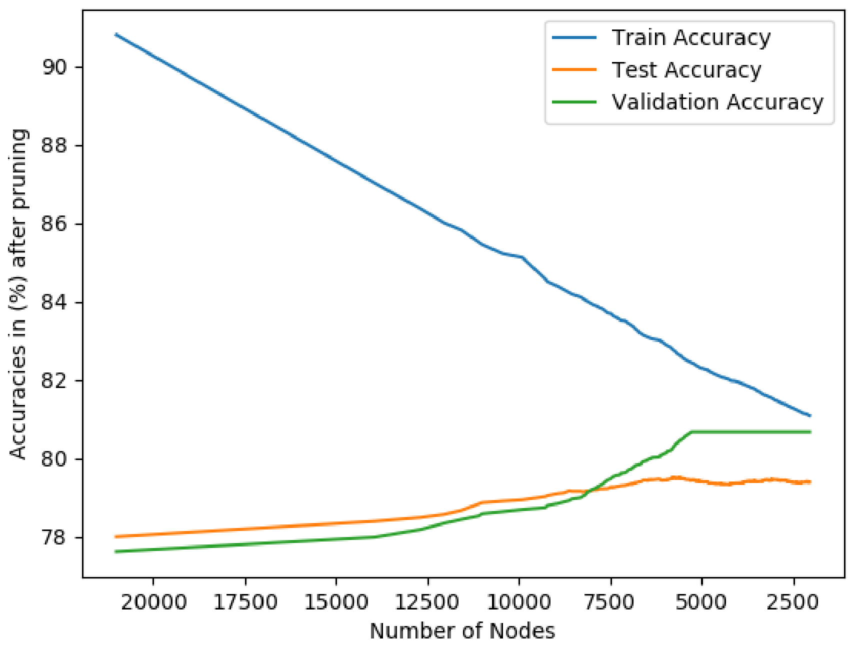

Figure 3: Accuracies after post pruning using iterative approach

1.2.1 Accuracy obtained on whole tree

Train Accuracy	Test Accuracy	Validation Accuracy
83.33101540648711%	79.81085716934774%	82.52364175783423%

Table 1: Accuracy after post-pruning using bottom up approach

Train Accuracy	Test Accuracy	Validation Accuracy
81.08417165020938%	79.3889944833341%	80.66938624142406%

Table 2: Accuracy after post-pruning using iterative approach

1.2.2 Observations

1. Through bottom up approach, test accuracy is greater than what we are getting through iterative approach. So, we will use bottom up for post-pruning.

2. As we prune, we can see our test accuracy is increasing which clearly tells we are removing over-fitting from our fully grown tree.

3. The fact that our validation and test accuracy is increasing as we prune the nodes tells that the noise that we had fit while training the model is getting removed.

4. After post pruning, there isn't significant difference between train, test and validation accuracy.

1.3 Random Forests

In this we have trained our dataset with Random Forest method by varying the paramters like n_estimators, max_features and min_samples_split using Scikit Learn of Python. We'll take the model that will give the best accuracy on a particular set of values of parameters.

1.3.1 Optimal Set of parameters obtained

max_features	**min_samples_split**	**n_estimators**
0.1	10	450

1.3.2 Accuracy Obtained after training on parameters obtained above

Training	**Out-of-bag**	**Validation**	**Test**
87.3642081189251%	81.07644522738863%	80.73428518449842%	80.82610912799592%

1.3.3 Comparison of Accuracy obtained with post-pruned accuracy

1. There is a difference of approx 4% in train accuracy with accuracy obtained after post pruning in previous part, Train accuracy is greater using Random Forest.
2. There is a difference of approx 2% in validation accuracy with accuracy obtained after post pruning in previous part, Validation accuracy is greater after post pruned model in previous part.
3. There is a difference of approx 1% in Test accuracy with accuracy obtained after post pruning in previous part, Test accuracy is greater using Random Forest.

1.3.4 Observations

1. We can clearly see Test accuracy is greater using Random Forest than post-pruning of previous decision tree.
2. Validation and Test accuracy in random forest part is almost same in Random Forest, whereas there is difference of approx 3% in both these accuaracy in post-pruning of previous decision tree.
3. So, through random forest validation accuracy we are getting more idea of test accuracy than in post pruning of decision tree in previous part.
4. Also, through random forest, we are getting greater train accuracy.

1.4 Random Forests - Parameter Sensitivity Analysis

In this after we have obtained optimal parameters, we are fixing two parameters and varying the other one and getting different plots of train, test and validation accuracy.

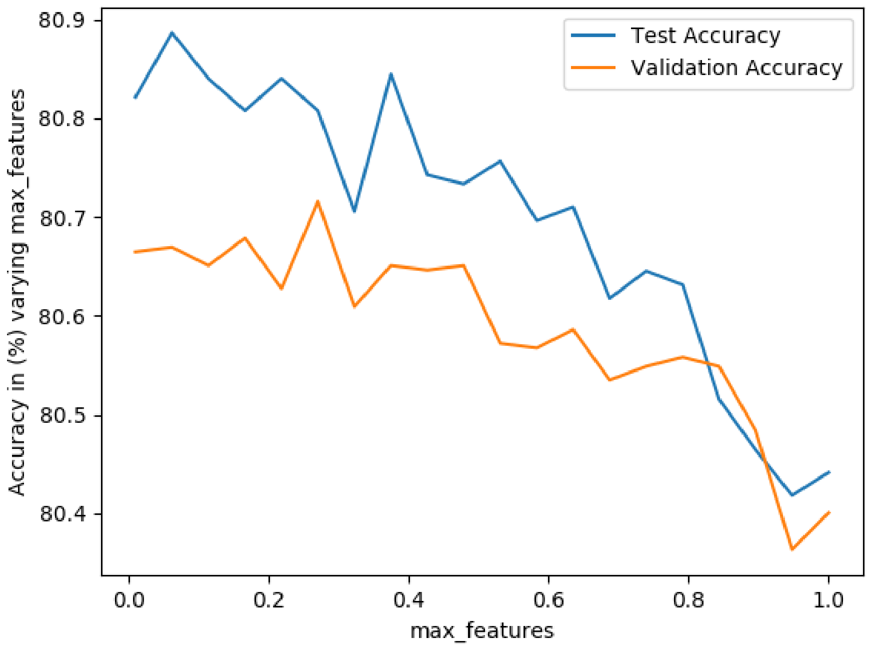

Figure 4: Accuracy varying max_features parameter keeping rest fixed to optimum

1.4.1 Observations from above by varying max_features

1. Test accuracy is maximum at 0.1
2. As we vary max_feature keeping others fixed, the accuracy is varying between 80 and 81, not such significant difference
3. When we are taking more features into consideration, the accuracy is decreasing, not by significant amount though.

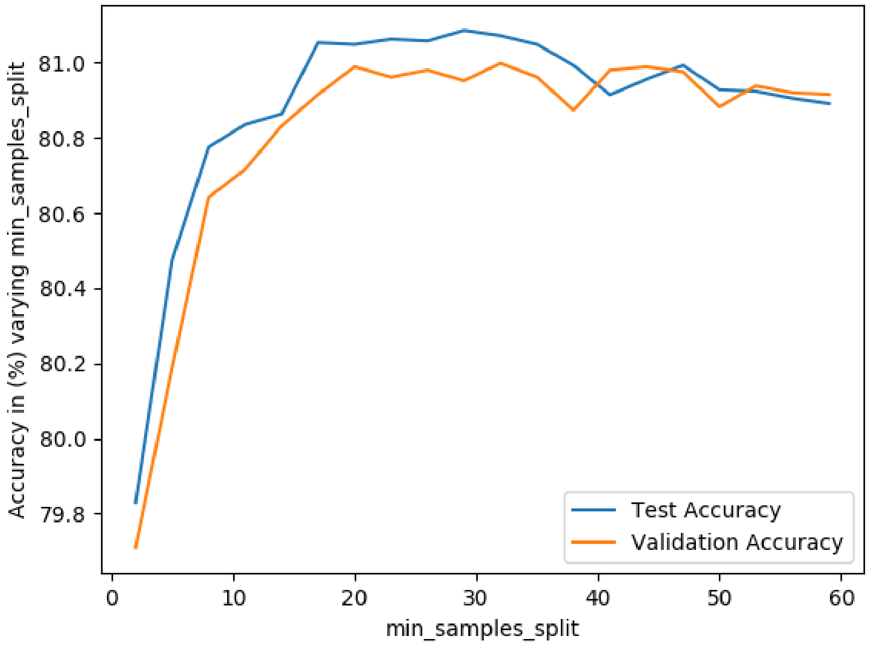

Figure 5: Accuracy varying min_samples_split parameter keeping rest fixed to optimum

1.4.2 Observations from above by varying min_samples_split

1. As we increase min_samples_split, accuracy is increasing by small amount upto some point but after that it decreases by small amount.
2. Accuracy is maximum when min_samples_split is around 27.
3. Initially when min_samples_split is less than 10 accuracy is less as it is the case of over-fitting which is handled as we increase min_sample_split.
4. If we keep on increasing min_samples_split, then our model would underfit for sure, thereby decrease in accuracy.

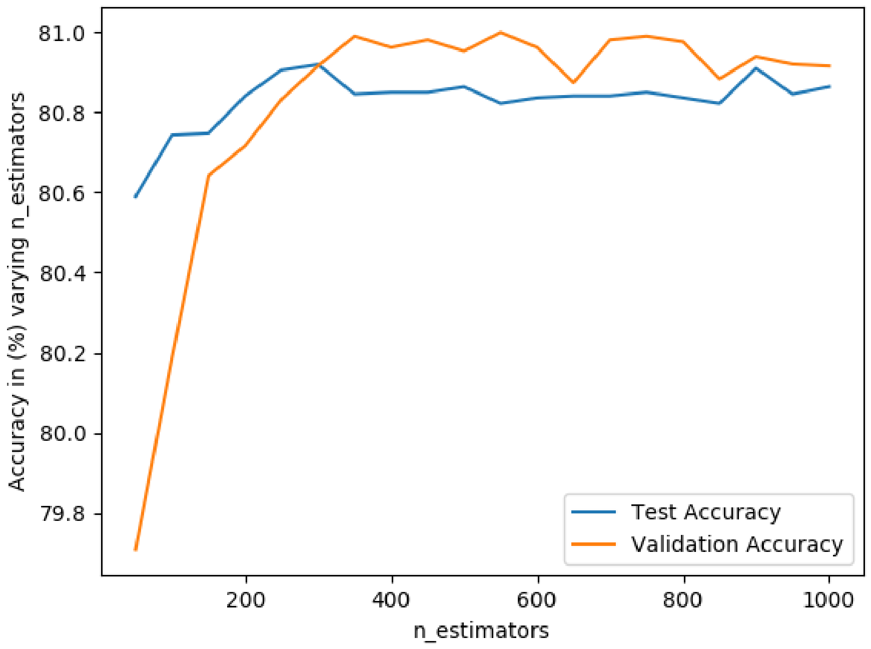

Figure 6: Accuracy varying n_estimators parameter keeping rest fixed to optimum

1.4.3 Observations from above by varying n_estimators

1. As we increase n_estimators, first accuracy is increasing but after sometime it is oscillating between 80.8 and 81.0, this change won't be considered significant.

2. As we are increasing n_estimators our accuracy is saturating.

3. If value of n_estimators is small, then it won't give the best results as we are taking less number of trees in forest for consideration.

1.4.4 Sensitivity of each parameter

1. Model from random forest is sensitive to max_features as if we take more number of features after a point for splitting, then we may not get the best the result as we saw in our case.
2. Model from random forest is sensitive to min_samples_split as if we it is very low , it is kind of over-fitting and if it is high it is kind of under-fitting.
3. Model from random forest is sensitive to n_estimators as if it less then we are taking less number of trees in the forest and that won't give us the best result. It will saturate after some point if we keep on increasing n_estimators.