2019SIY7580 (AMAN BHARDWAJ)

ASSIGNMENT-3 PART_A

Q1 Decision Tree and Random Forest

a) Decision Tree Construction

Observations:

Total Nodes in Tree: 19987

Grow Time: 6.74Mins

Data Set	Max Accuracy	Argmax Node (Max Accuracy)	Accuracy (Full Grown Tree)
Train	90.40%	19900	90.40%
Validation	78.61%	7500	77.63%
Test	78.99%	9000	77.98%

Table: Accuracy at various levels of tree growth

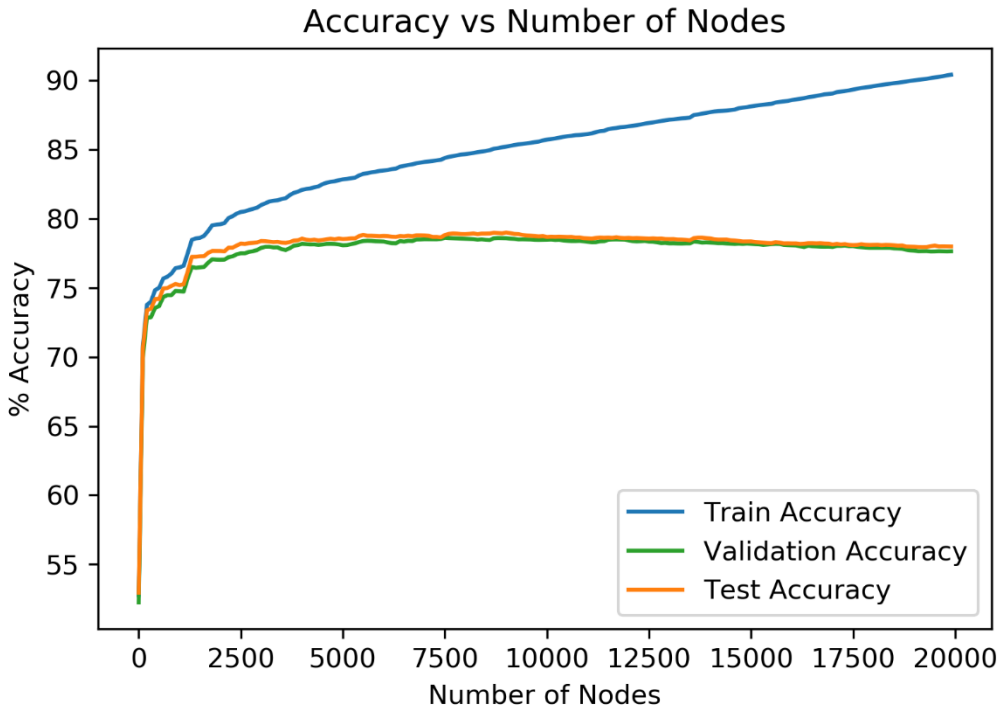

Figure: # Nodes vs Accuracy on every 100 nodes added to Tree (Before pruning)

Comments:

- Accuracy on Train Data is maximum and for Test and Validation is almost similar as train data is known to the model, i.e. it has seen even the noise of Train set. Test and Validation are unseen for the model.
- For Training dataset accuracy keeps on increasing and reaches maximum for fully grown tree. This is because the tree has learned even the noise of the Train data and predicts with high accuracy for train data.
- For validation and test data accuracy rises till a certain number of nodes and then decrease a bit till it reaches the leaves of the full-grown tree. This is due to the overfitting of Decision Tree. As we have tried to fit even the noise of the train data completely.

b) Decision Tree Post Pruning

Total Nodes in Tree: 4820

Data Set	Max Accuracy	Argmax Node (Max Accuracy)	Accuracy (Full Grown Tree)
Train	83.91%	4800	83.91%
Validation	81.08%	4400	81.04%
Test	79.90%	2800	78.35%

Table: Accuracy at various levels of tree growth (Post Pruning)

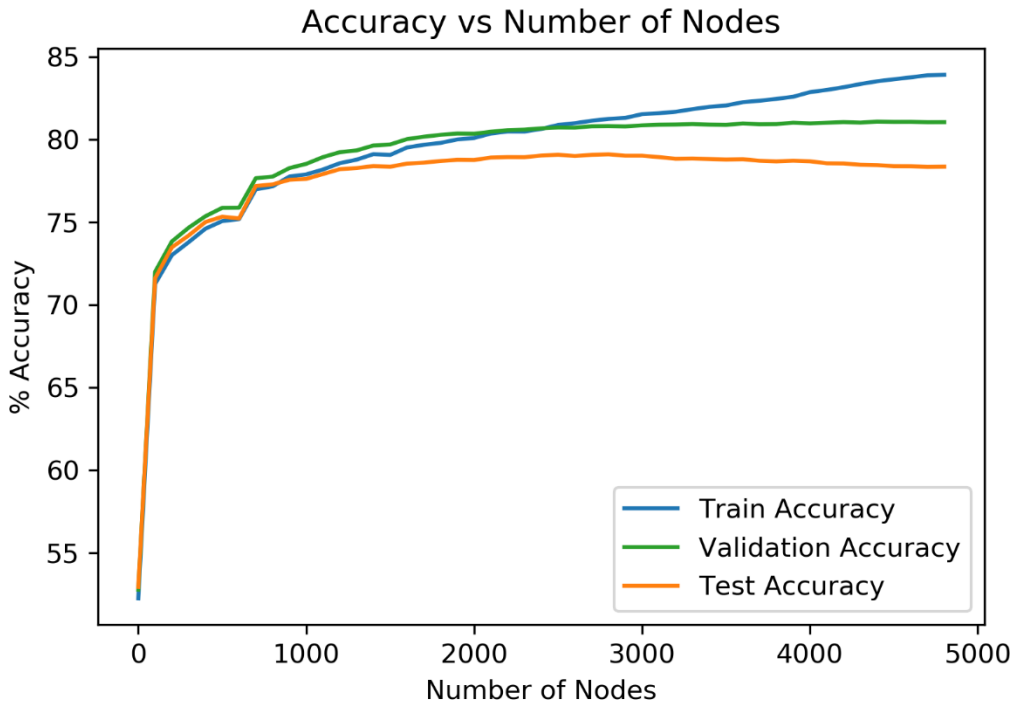

Figure: # Nodes vs Accuracy on every 100 nodes added to Tree (Post pruning)

Data Set	Gain/Loss in Max Accuracy	Gain/Loss in Full & Pruned Tree Accuracy
Train	- 6.49%	- 6.49%
Validation	+ 2.71%	+ 3.41%
Test	+ 0.91%	+ 0.37%

Table: Accuracy gain/loss for various datasets (Post Pruning)

Observations:

- Max Training Accuracy came down by 6.49% and Validation Accuracy went up by 3.41%.
- After pruning the number of nodes left in the tree are only 4820 while before pruning there were a total of 19987 Nodes in fully grown tree.
- Test accuracy also went up.
- This proves that after pruning the accuracy increases.

Comments:

- As the accuracy increases post pruning. This proves that tree does not give best accuracies when fully grown. This is because of overfitting over the noise and outliers of the Train data.
- Best accuracy could be achieved either by limiting the height of the tree while growing it. Or by post pruning it. In our case we have done Reduced Error Post Pruning over the validation data.
- Reduction in Train Accuracy is expected as now we have removed the subtrees to the nodes which were increasing the error. Since post-pruning we have just 4820 nodes vs 19987 nodes the model would reduce efficiency over Train Data. The results are intuitively correct.
- Increase in score over Validation data by 3.41% is also intuitively correct. As we have pruned the full tree over Validation dataset. Therefore, we have removed the subtrees which were increasing the error and made their parent nodes as leaf nodes. Therefore, for Validation data model would show significant improvement as we have got in our results.
- There is also increase in test accuracy score as now we have removed the overfitting of our model over the train dataset. But it didn't increase over the Test data significantly as it was still unseen by the tree. The increase in accuracy was expected and we have achieved the same here.

c) Random Forest:

GridSearchCV Results (For CV = 5):

- **Total tasks = (5*5*5) * 5 = 625**
- **Total Search time = 977.47Mins**
- **Best Params (Using GridSearchCV and CustomSearch Both):**
  - **max_features: 0.1**
  - **min_samples_split: 10**
  - **n_estimators: 250**

Accuracies:

Accuracy	Accuracy RFC Optimal Param	Accuracy DTree Post Pruning	Difference(RFC - DTree)
Train	**87.39%**	**83.91%**	**+3.48%**
Validation	**80.70%**	**81.04%**	**- 0.34%**
Test	**80.77%**	**78.35%**	**+2.43%**
Out-of-bag	**80.99%**	-	-

Table: Accuracy Sklearn RandomForestClassifier v/s Post Pruned Decision Tree

Observations:

- Accuracy for Random Forest Classifier is more in case of Train and Test datasets, but it is slightly less in case of Validation dataset.
- Results are quite close in both the cases.

Comments:

- Validation accuracy is more in case of post pruned DTree as pruning of our tree based on Validation dataset therefore this bias is expected.
- Train accuracy would have been more in case of Fully-grown tree as we had trained over all the Train dataset which includes even its noise. But in case of pruned tree it has reduced as we have removed the subtrees which were reducing the accuracy over Validation dataset. Therefore, the bias towards Train dataset has been reduced significantly, resulting in reduction of accuracy in case of pruned tree. Random forest it is more as it trains over certain portion of dataset. It has not pruned based on Validation dataset.
- Test Accuracy has increased in case of RFC which is also expected as RFC from sklearn is quite optimized hence provides better results for test dataset. Also, the same in case of rest of the datasets.
- We have received same results with sklearn GridSearchCV and our custom Hyperparameter Tuning using oob_score as the scoring parameter.

d) Random Forests - Parameter Sensitivity Analysis:

n_estimator value	validation accuracy	test accuracy
50	80.4%	80.7%
150	80.62%	80.74%
250	80.64%	80.77%
350	80.72%	80.86%
450	80.71%	80.8%

Table: Validation & Test Accuracy for different n_estimators values keeping other params to optimal value using Sklearn RandomForestClassifier

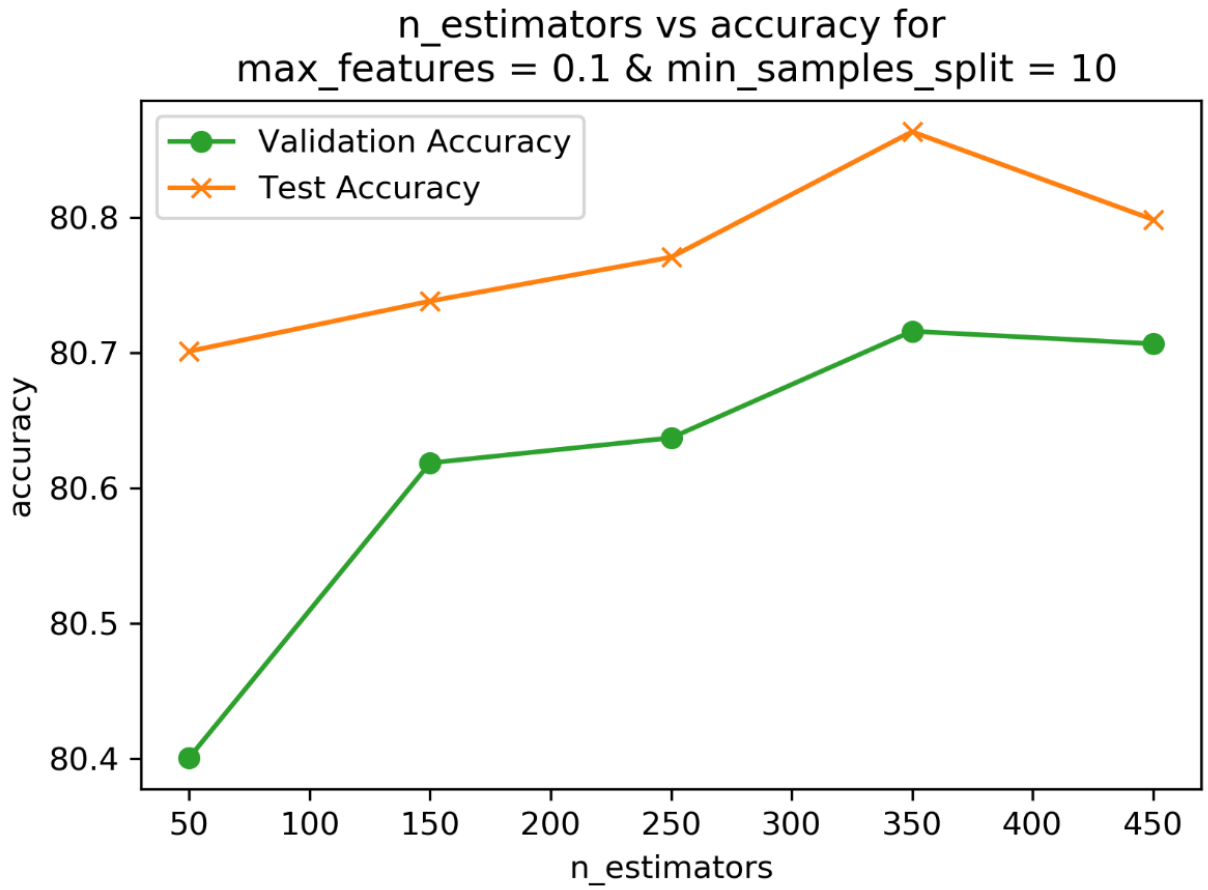

Figure: Sensitivity towards n_estimators values keeping others to their optimal values.

max_features value	validation accuracy	test accuracy
0.1	80.58%	80.85%
0.3	80.56%	80.72%
0.5	80.64%	80.65%
0.7	80.51%	80.58%
0.9	80.37%	80.51%

Table: Validation & Test Accuracy for different max_features values keeping other params to optimal value using Sklearn RandomForestClassifier

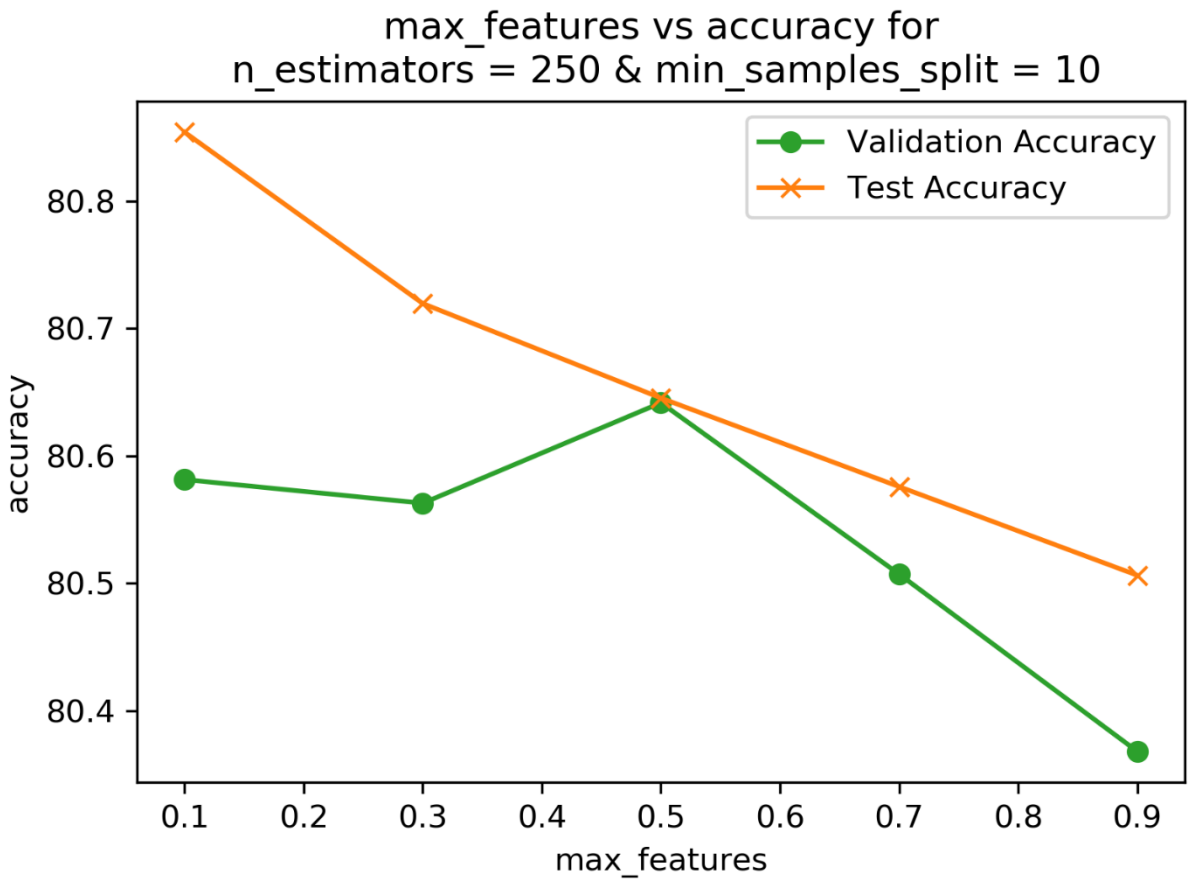

Figure: Sensitivity towards max_features values keeping others to their optimal values.

min_sample_split value	validation accuracy	test accuracy
2	79.65%	79.90%
4	79.99%	80.27%
6	80.30%	80.57%
8	80.47%	80.63%
10	80.60%	80.89%

Table: Validation & Test Accuracy for different min_samples_split values keeping other params to optimal value using Sklearn RandomForestClassifier

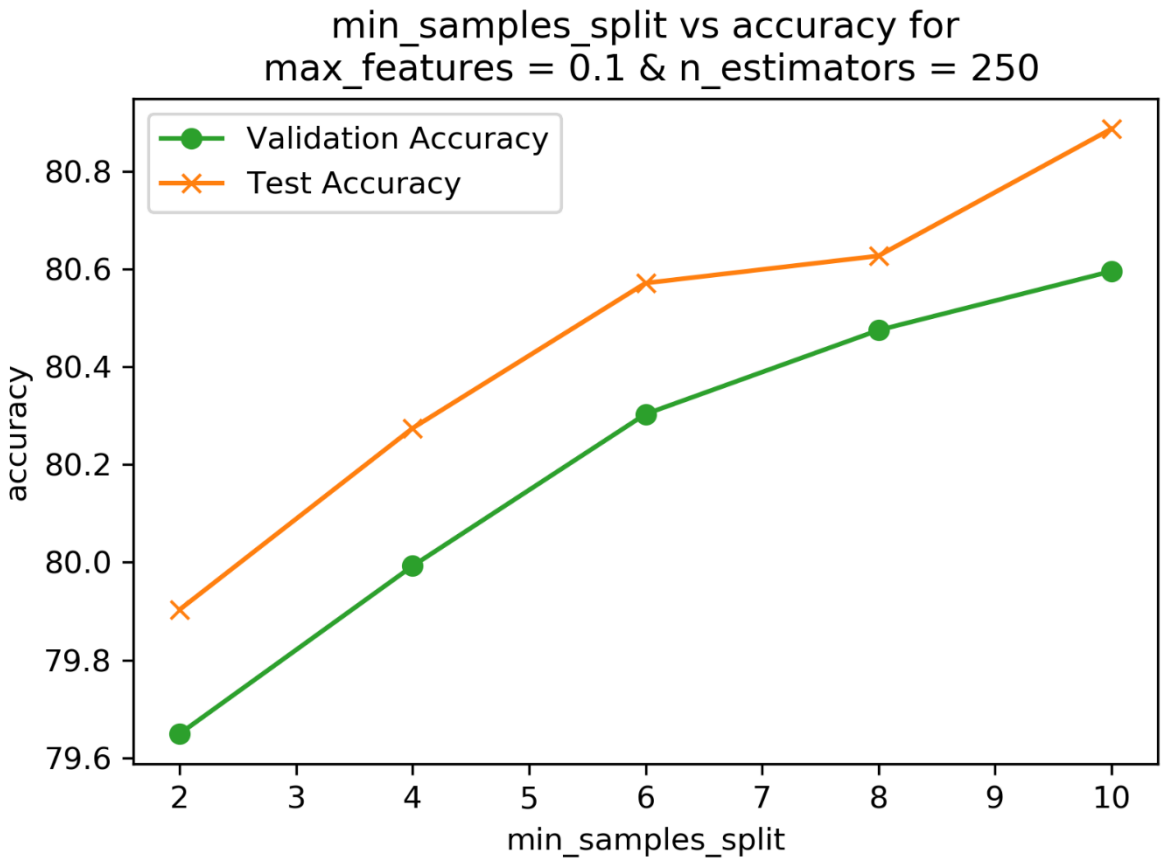

Figure: Sensitivity towards min_samples_split values keeping others to their optimal values.

Observations & Comments:

- For n_estimators: best param was 250 but max accuracy has been achieved at 350 for both test and validation dataset. This could be because the parameter tuning was done over Train dataset. They can perform slightly different over other datasets. And as the variation in the accuracy is very less over n_estimators so model is less sensitive for values of n_estimators from 50 – 450. A significant variation can be seen for the values from 10 – 50. Therefore, for smaller values of n_estimators the model becomes sensitive. But for higher ones it is very less sensitive.
- For max_features, the accuracy will decrease with increase in max_features. This is because for 0.1 features the accuracy is supposed to be highest as we are considering only 10% of the total features for the split purpose. Therefore, at each node only 10% of the features will be considered for split. So, each feature will have to compete with less feature to get its chance to split. Therefore, best tree will be for less features to split at each node. There is just one exception in value of validation accuracy at 0.5. Since the variation is not significant therefore this can be considered exception. Rest trends follows our analysis and are as per expectations.
- For min_sample_split gave best accuracy for 10, this shows that the nodes with samples less than 10 will be declared as leaf nodes and the majority class will be used for prediction at that node. This helps in preventing overfitting of the tree. Therefore, if we reduce the value of this param then it will split and grow the tree further which will result in overfitting of the tree.
- Model is more sensitive towards min_sample_split values as compared to the other two params

****