Assignment 3 Report

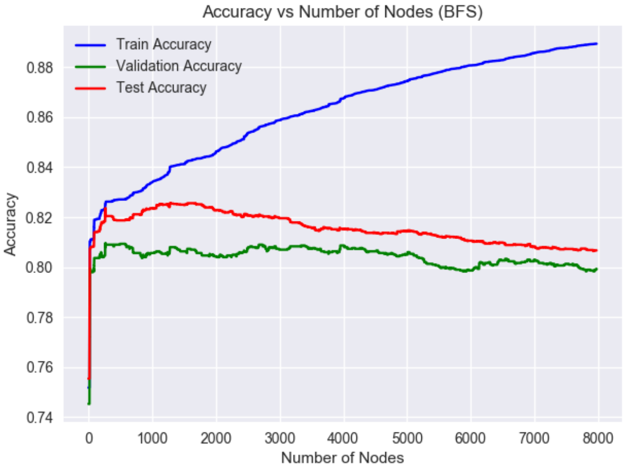
Anoop (2015CS10265)

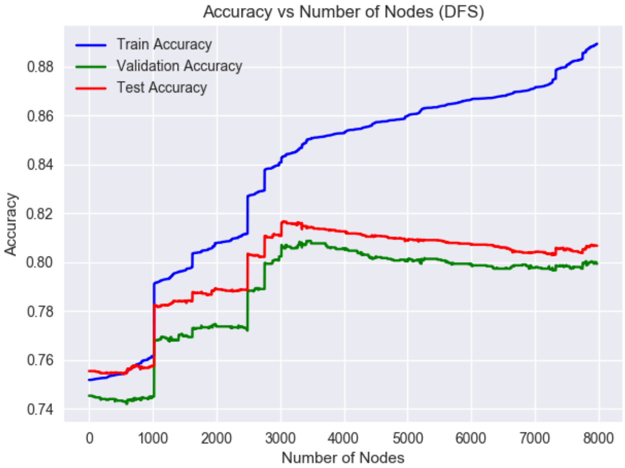
April 11, 2018

Decision Trees

Decision Tree Implementation

Number of Nodes	= 7972
Train Accuracy	= 0.88933
Validation Accuracy	= 0.79933
Test Accuracy	= 0.80671

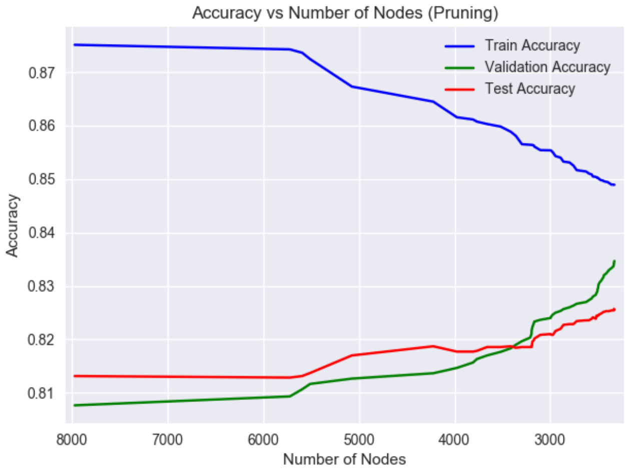




From both BFS and DFS ways of growing the decision tree, we can observe that the training accuracy increases with number of nodes. Also from the validation and test accuracy curves we can say that the decision tree is overfitting the data. The decision tree is not able to fit the data completely because of the pre-processing that has been done on the data (There are examples where for the same features, the labels are different).

Post-Pruning

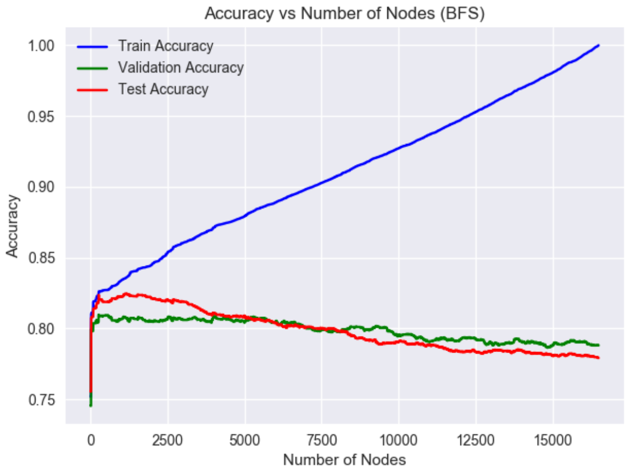
Number of Nodes	= 2322
Train Accuracy	= 0.84893
Validation Accuracy	= 0.83467
Test Accuracy	= 0.82557

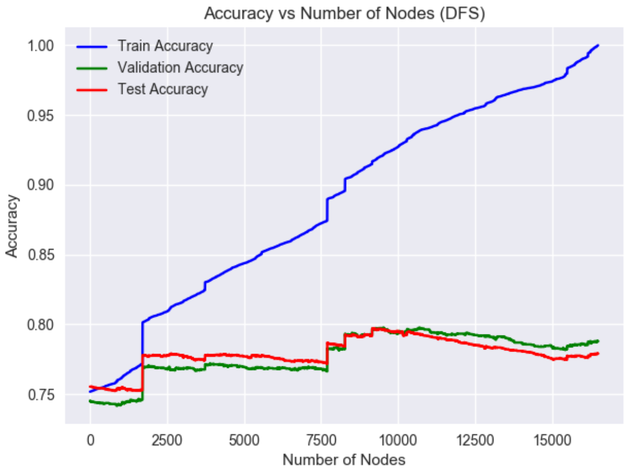


From the graph, we can observe that post-pruning is reducing the overfitting. Also, as the validation accuracy is increasing, the test accuracy is also increasing.

Decision Tree Implementation (Handling numerical features)

Number of Nodes	= 16495
Train Accuracy	= 0.99989
Validation Accuracy	= 0.78800
Test Accuracy	= 0.77914

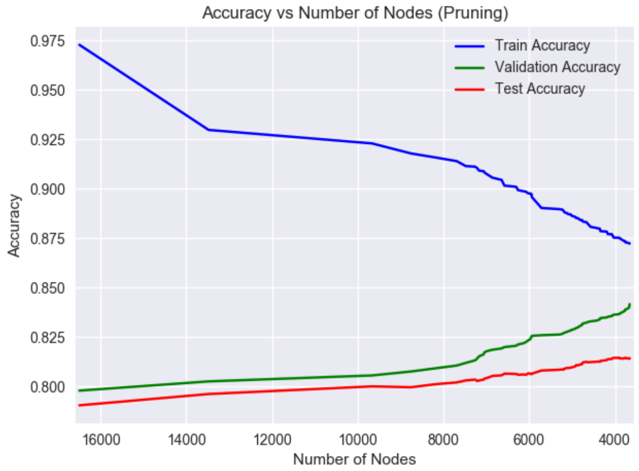




In this modified implementation of decision trees, we are better able to handle numerical attributes. This can be seen in the training accuracy which is close to 100%. The decision tree has almost completely fit the training data. This overfitting has brought down the validation and test accuracies.

Table 1: Number of splits per numerical feature

Age	**Fnlwgt**	**Education Number**	**Capital Gain**	**Capital Loss**	**Hour per Week**
3446	2194	0	18	6	536



Number of Nodes	= 3650
Train Accuracy	= 0.87230
Validation Accuracy	= 0.84167
Test Accuracy	= 0.81429

Post-Pruning the fully grown decision tree (modified) gives even better validation accuracy but lesser test accuracy.

Decision Tree Implementation - Scikit-Learn

Table 2: Best Parameters (from 2160 models)

Model	criterion	max_depth	min_samples_split	min_samples_leaf	max_features
1	gini	12	0.005	0.001	None
2	entropy	10	0.005	0.001	None

Table 3: Accuracies

Model	Train Accuracy	Validation Accuracy	Test Accuracy
1	0.83507	0.82633	0.82914
2	0.83574	0.82633	0.83000

max_depth restricts the depth of the decision tree, **min_samples_split** restricts the splitting of small nodes and **min_samples_leaf** enforces a minimum size requirement for a node to be a leaf. All these hyperparameters help in reducing overfitting. As compared to greedy post-pruning, these hyperparameters need to be tuned manually. In this case, searching for the hyperparameters helps rather than greedy post-pruning marginally (over test accuracies).

Random Forest Implementation - Scikit-Learn

Table 4: Best Parameters (from 17280 models)

Model	criterion	n_estimators	max _depth	min _samples _split	min _samples _leaf	max _features	bootstrap
1	gini	2	8	0.01	1	10	False
2	entropy	2	12	0.01	0.001	10	True

Table 5: Accuracies

Model	**Train Accuracy**	**Validation Accuracy**	**Test Accuracy**
1	0.83263	0.82867	0.82586
2	0.83119	0.82933	0.82771

n_estimators specifies the number of decision trees in the random forest , **max_features** restricts the number of features searched over while choosing the split and **bootstrap** specifies if bootstrap samples are used or not. As compared to greedy post-pruning, these hyperparameters need to be tuned manually. In this case also, searching for the hyperparameters helps rather than greedy post-pruning marginally (over test accuracies).

Neural Networks