Machine Learning

Prediction using Decision Tree and Random Forest Models

Abhishek Goyal
2016csb1027

Preprocessing

Used the publicly available large movie review dataset from stanford. Randomly sampled 1000 instances from the train folder to create the training set and 1000 instances from test folder for the test set. Also sampled another 1000 instances from test folder for validation set required for pruning. Chose 5000 features (words) from the imdb vocab list with half of those with the highest imdbErr value and half with the lowest (highest negative values), thus the 5k words with comparatively more influence on the sentiment of the review were considered. Took 1k features from the first 10k words in the list and the remaining 4k from the other 79k left in the word list as the earlier words in the vocab list have a higher frequency of occurrence so less sparse datasets would be considered.

Experiment 2: Decision Tree

Created a decision tree model, trained on the 1000 sample training set. Considered only the presence or not of a particular word as a feature (tried continuous splitting but that gave similar or worse results as the feature matrix is very sparse for all reviews). Used maximum information gain as the criteria for selecting the feature to split the dataset on at every node. Used the depth of tree as early stopping criteria and tried different depths to compare the train and test set accuracy results.

Fully Grown Tree Results:-

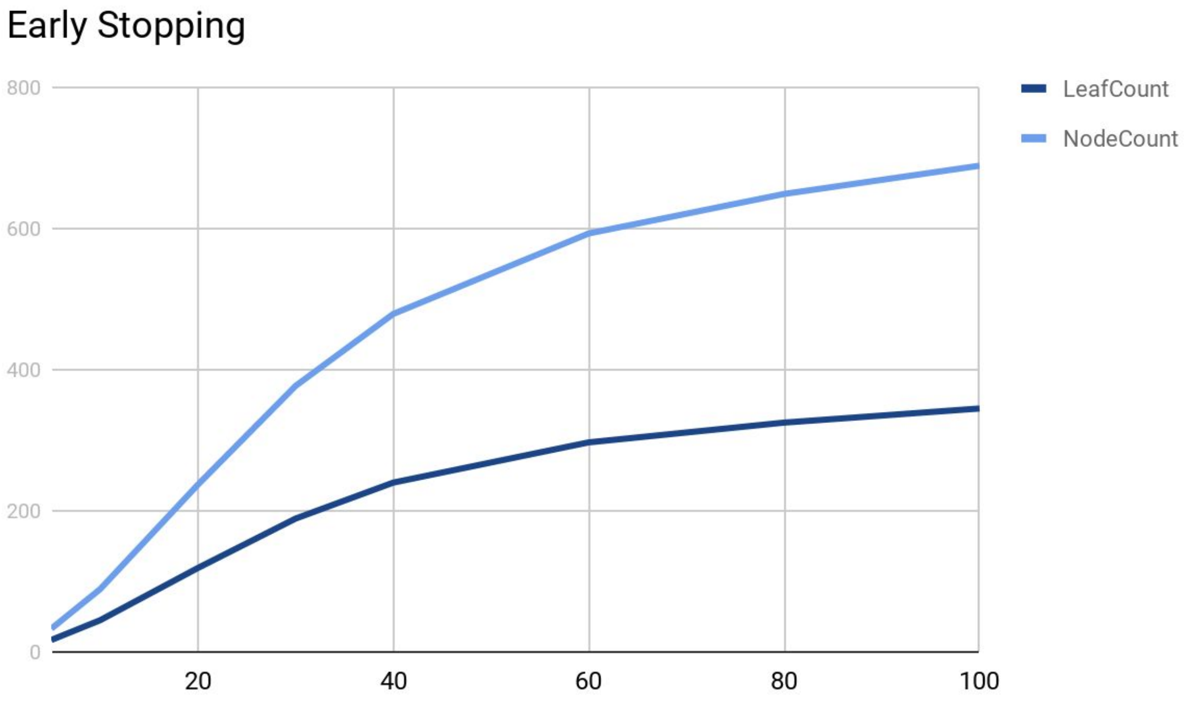
TrainSet Accuracy	TestSet Accuracy	LeafCount	NodeCount	Depth of Tree
99.2	73.2	347	693	103

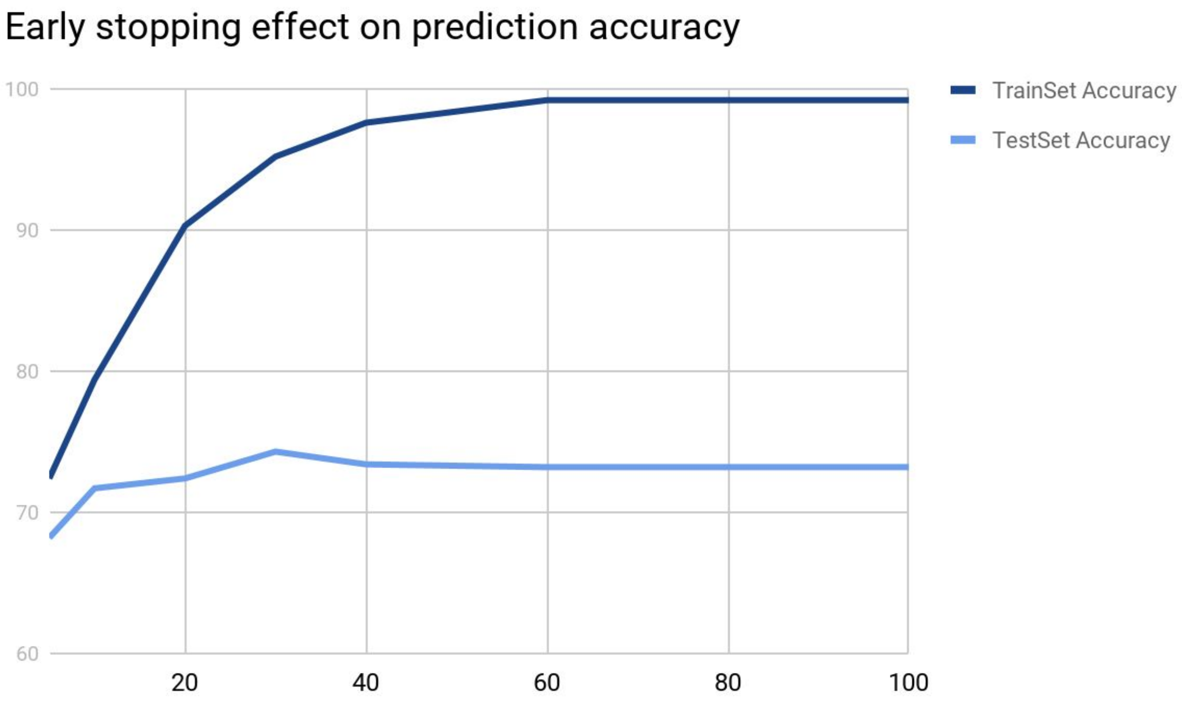
Early Stopping Results:-

Depth of Tree	TrainSet Accuracy	TestSet Accuracy	LeafCount	NodeCount
5	72.4	68.2	17	33
10	79.4	71.7	45	89
20	90.3	72.4	119	237
30	95.2	74.3	189	377
40	97.6	73.4	240	479
60	99.2	73.2	297	593
80	99.2	73.2	325	649
100	99.2	73.2	345	689

Observations:-

Only through excessive hit and trial depth of tree trials was a 1% increase in test set accuracy possible through early stopping. Even then as a validation set was not used and the test set results were compared directly to estimate the best possible depth of tree, this 1% increase should be theoretically lower for the scenario if a validation set was used for choosing the best possible depth. As expected, the leaf and node count increases rapidly with depth and slows down as the depth is about to reach the fully grown tree depth. Since the number of splits is minimal for less depth of trees, the train set and test set accuracy is very poor initially as most of the instances have to be predicted with a few amount of classifying features considered in these trees.





Experiment 3: Adding noise to the training set

Results:-

NoisePercent	TrainSet Accuracy	TestSet Accuracy	LeafCount	NodeCount
0	99.2	73.2	347	693
0.5	99.2	73.1	354	707
1	99.3	72.8	360	719
5	99.3	71.4	371	741
10	99.1	67.3	417	833
20	98.3	57.6	404	807

Observations:

As the noise percent increases, the test accuracy suffers badly. This is due to the erratic and inconsistent nature of the training set since for many reviews with supposedly positive features, the sentiment prediction is flipped to be negative. For more such modified training set instances, the model trained on them suffers and so the accuracy of test set predictions on this model decreases. The number of nodes and leaves also seems to increase with an increase in noise since the noise makes the splits at every node less homogenous so more number of splits are required for fully growing the tree.

Experiment 4: Post-Pruning

Pruned the tree using a validation set of 1k instances picked from the test folder.

Accuracy results on the TestSet and TrainSet (Not validation):

Tree	TrainSet acc	TestSet acc	LeafCount	NodeCount	TreeDepth
Fully Grown	99.2	73.2	347	693	103
Pruned	90.6	74.7	115	229	42

Observations:

Pruning resulted in the removal of all the nodes from the tree which caused the overfitting of the model using the validation set. This caused a great reduction in the number of leaves, nodes and tree depth of the model tree. The accuracy of predictions on the training set also decreased since the model is no longer overfit so some deviations in the training set predictions from the actual arose. Importantly, the test set accuracy improved by >1% as the model is not overfitted on the training set now. Tried pruning on different samples of the provided 25k instances and found that even if the initial accuracy was 67-70 for the test set instances, pruning increased them to near 71-74%, thus pruning gives much better results for OOB instances.

Experiment 5: Random Forest

Number of Trees	Train Set Accuracy	Test Set Accuracy
-- (Normal Single DTree)	99.2	70.5
5	96.0	74.5
10	95.6	76.8
15	96.8	76.0
20	95.8	78.6
30	96.0	79.0

Observations:

Random forest led to much more increase in accuracy than normal post-pruning with about (76-80)% accuracy achieved with this model, tested on multiple test sets. The increase of random forest trees generally increased accuracy up to a point (optimum at about 30 trees) but the increase of trees also led to a great increase in computation time (about 30 minutes required for 30 trees).