Analysis of the Tree Algorithms In Wheat Seeds DATASET

Using all features

	Decision Tree	Random Forest	Gradient Boosting
Training Accuracy	100	100	100
Test Accuracy	92.5	92.5	92.5
Wall Time for finding optimal parameters	144ms	32.8s	23.5s

After feature selection

	Decision Tree	Random Forest	Gradient Boosting
Training Accuracy	98.74	98.74	100
Test Accuracy	92.5	92.5	90
Wall Time for finding optimal parameters	66ms	31.1s	7.98s

Observations

- All three methods had lower training accuracy after highly correlated features were removed.
- The Decision Tree Classifier had the same performance with the Random Forest Classifier.
- Gradient Boosting performed slightly worse on the test set
- Feature Selection did not improve the accuracy much, but it reduced the time it took the algorithm to run significantly.
- Time taken to run the Decision Tree Classifier regardless of the feature engineering method < Time taken to run the Random Forest Classifier regardless of the feature engineering method < Time taken to run the Gradient Boosting Classifier regardless of the feature engineering method.

Expectation Vs. Reality

- I expected the Decision Tree to have a high training accuracy and this happened
- I did not expect the Decision Tree classify to perform the same way as the Random Forest Classifier, I expected the Random Forest Classifier to be better. This may have happened because the dataset is quite small.
- Random Forest and Gradient Boosting can be better depending on the situation
- I expected Gradient Boosting to outperform Decision Trees but this is not the case

Analysis of the Tree Algorithms In COVID DATASET

Using all features

	Decision Tree	Random Forest	Gradient Boosting
Training Accuracy	66.25	69.8	68.52
Test Accuracy	65.81	67.4	67.23
Wall Time for finding optimal parameters	1.22s	88s	174s

Dropping the City column

	Decision Tree	Random Forest	Gradient Boosting
Training Accuracy	66.14	67.08	67.19
Test Accuracy	65.91	66.15	66.52
Wall Time for finding optimal parameters	656ms	72s	134s

Aggregating the categories of the features

	Decision Tree	Random Forest	Gradient Boosting
Training Accuracy	66.14	66.94	68.14
Test Accuracy	65.91	67.16	67.33
Wall Time for finding optimal parameters	739ms	79s	124s

Observations

- The test accuracy of the Decision Tree Accuracy improved while the test accuracy scores for the other two algorithms experienced fractional reductions
- There was a drastic reduction in computation time for all algorithms
- Gradient Boost performed best with the aggregated features.
- Decision Tree performed better in both models where the city column was excluded or aggregated.
- The training and test set accuracy was only slightly better when the city column was included as compared to when the features were aggregated.
- Time taken to run the Decision Tree Classifier regardless of the feature engineering method < Time taken to run the Random Forest Classifier regardless of the feature engineering method < Time taken to run the Gradient Boosting Classifier regardless of the feature engineering method.

Expectation Vs. Reality

- I expected Random Forest and Gradient Boosting to outperform Decision trees and that happened
- I expected Gradient Boosting and Random Forest to take much longer than Decision trees and that happened.

Overall, for the covid dataset, aggregating the features seemed better as it produced comparatively good results and it also saves a lot of computation time.

Conclusion from both analysis

For simple datasets with little variations and bias, Decision Trees perform excellently well and can rival Random Forest and Gradient Boosting Trees.

But for Complex datasets with high variations and bias,ensemble methods seem to perform better than a single Decision Tree.

Considering that random forest had to do 20 iterations in finding optimal parameters compared to Gradient Booting Tree that had to do only 5 iteration, it can be concluded the Gradient Boosting Tree is more computationaly expensive.

In [ ]: