# Decision Tree Classifier

For this experiment, I am using random state 42. I first tune the criterion for either gini or entropy. Then I performed post pruning by tunning the ccp\_alpha using the cost\_complexity\_pruning\_path function on either gini or entropy. All rest of the variable are kept as default. The following table shows the results of the experiment.

Table

Description automatically generated

The accuracy and f1-score increase as the number of features and examples increase.

To experiment further, I also try min\_sample\_split to preform pre-pruning. The results are in the table below

Table

Description automatically generated

However, the result remains largely the same. The score increases as the number of clause and example increases.

# Bagging Classifier

For this experiment, I am using random state 42. I fine tune the max\_sample, and max\_features for the Bagging Classifier. The following table shows the result.

Table

Description automatically generated

As the number of samples increases there is a quite noticeable improvement in both accuracy and f1-score. The same can also to be said for the size of the clause. As the number of clause and example increases, the accuracy and f1-score reach nearly 100 percent.

# Random Forest Classifier

For this experiment, I am using random state 42. I fine tune the criterion, min\_samples\_split, and max\_features for the Bagging Classifier. The result is shown in the table below.

Table

Description automatically generated

The score improves as the number of feature and sample increases. The effect of the feature size and sample size of the scoring of the model seem to be comparable.

# Gradient Boosting Classifier

For this experiment, I am using random state 42. I fine tune the loss function for the Gradient Boosting. To speed up the training process, I set the n\_iter\_no\_change to 3. The result is shown in the table below.

Graphical user interface, table

Description automatically generated

The Gradient Boosting converges extremely fast. The scores increase with the number of the feature and the number of the examples. As shown in the table above, 1000 examples are enough for the Gradient Boosting to converge even with only 300 features.

# Final Evaluation

For all four of the classifiers, an increase in the number of the features and examples corresponds directly to an increase in the accuracy and f1 scores.

It makes sense that Gradient Boosting is performing well on this dataset. Since the samples were generated directly from the solution and no solution, meaning there is no noisy on the data.