CS6375 Project 2: Tree Classifiers

Author: Khoi Nguyen

Table of contents

Introduction
The CNF Dataset
    Experiments
        DecisionTreeClassifier
        BaggingClassifier
        RandomForestClassifier
        GradientBoostingClassifier
    Results and Discussion
The MNIST dataset
Conclusion
Appendix A: Random Forest on CNF
Appendix B: Bagging on CNF
Appendix C: Random Forest on CNF
Appendix D: Gradient Boosting on CNF

Introduction

This document details the experimental results when training tree-based classification algorithms on two datasets: CNF and MNIST. For each sub-dataset, I trained Decision Tree, Bagging, Random Forest, and Gradient Boosting on these datasets and swept through a comprehensive set of hyperparameters. The best set of hyperparameters along with the corresponding performance is reported for each dataset and algorithm. Finally, there will be some discussion on the performances.

The CNF Dataset

The CNF dataset is a collection of 15 sub-datasets. Each dataset is parametrized by C, the number of features (i.e., the number of clauses in the underlying CNF), and D, the number of examples.

Experiments

DecisionTreeClassifier

Parameter grid

```
{
    'criterion': ['gini', 'entropy'],
    'splitter': ['best', 'random'],
    'max_depth': [None, 4, 16, 64],
    'min_samples_split': [2, 8, 0.01, 0.1],
    'max_features': ['sqrt', None],
}
```

In total, 128 parameter sets were tried for each dataset. Results for each dataset is in Appendix A.

BaggingClassifier

Parameter grid

```
{
    'estimator': [DecisionTreeClassifier(
        criterion='gini',
        splitter='best',
        max_depth=None,
        min_samples_split=2,
        max_features=None,
    )],
    'n_estimators': [10, 50, 100],
    'max_samples': [0.1, 0.5, 1.0],
    'max_features': [0.1, 0.5, 1.0],
    'bootstrap': [True, False],
    'oob_score': [True, False],
    'warm_start': [True, False],
}
```

In total, 216 parameter sets were tried. Results are in Appendix B.

RandomForestClassifier

```
param_grid = {
    'criterion': ['gini'],
    'n_estimators': [10, 50, 100],
    'max_depth': [None, 4],
    'min_samples_split': [2, 0.01],
    'max_features': ['sqrt', None],
    'bootstrap': [True, False],
    'oob_score': [True, False],
```

```
}
```

There were 96 parameter sets in total. Results are in Appendix C.

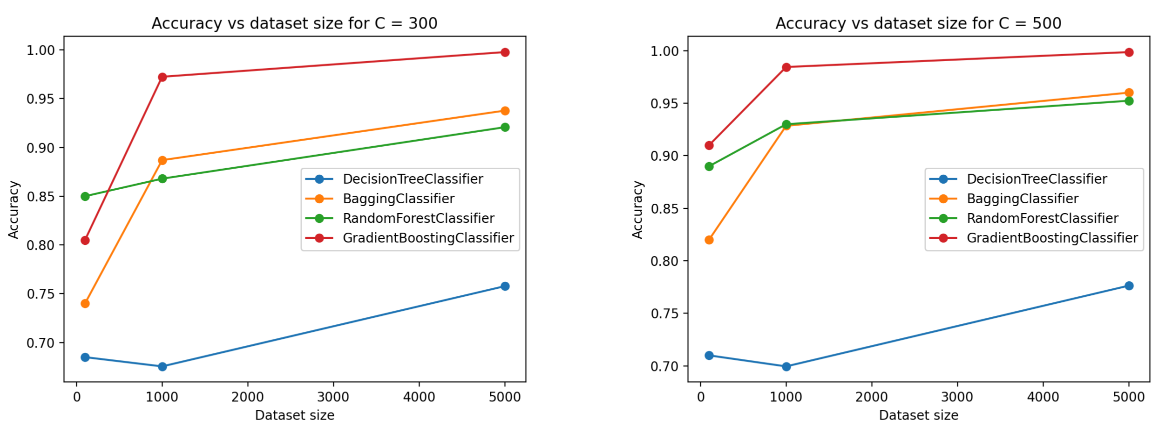
GradientBoostingClassifier

```
param_grid = {
    'n_estimators': [10, 50, 100],
    'learning_rate': [0.1, 0.01],
    'criterion': ['friedman_mse', 'squared_error'],
    'max_depth': [3, None],
    'min_samples_split': [2, 0.01],
    'max_features': ['sqrt', None],
}
```

96 parameter sets in total. Results are in Appendix D.

Results and Discussion

Below are the results of the four algorithms across settings. Each subplot is for a value of C (number of features). In each subplot, the x-axis is about the dataset size, while the y axis are for accuracy. For detailed results, please see Appendix A-D.



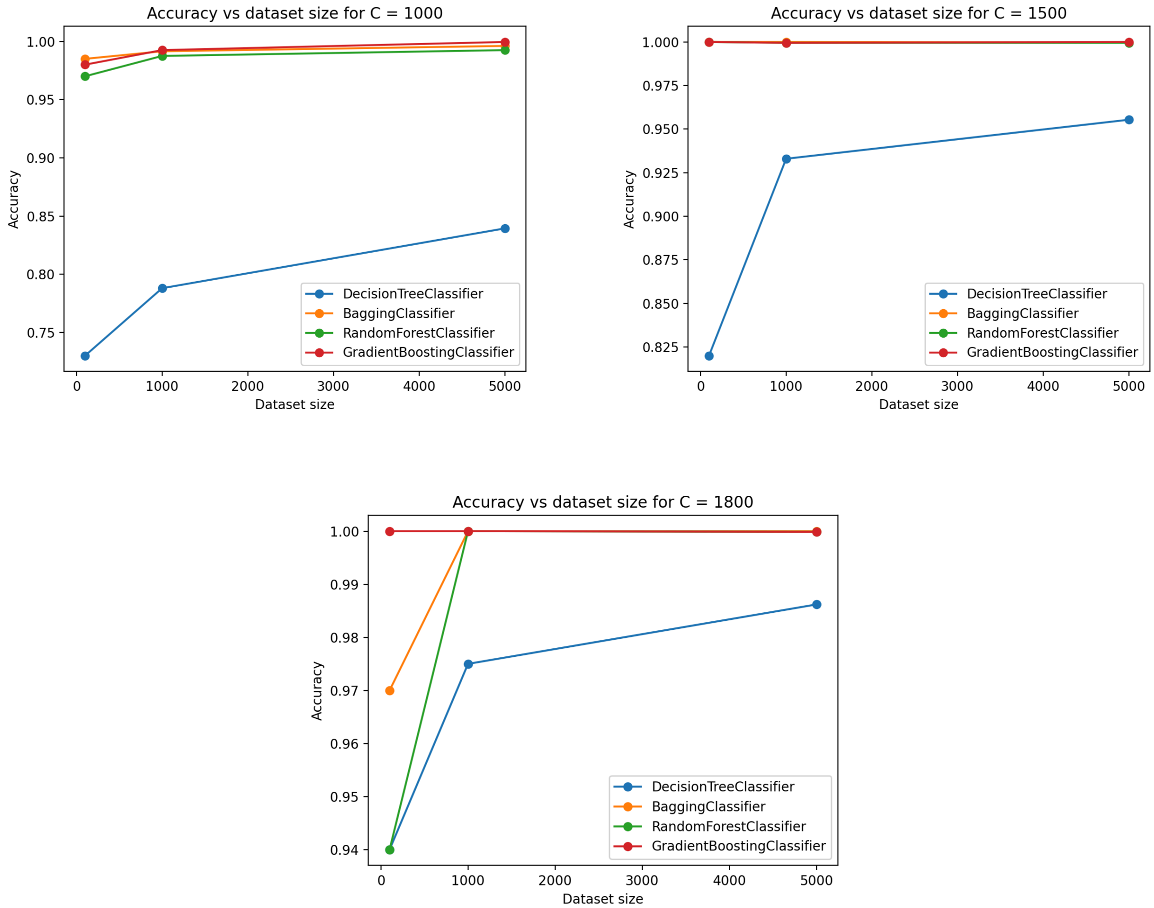


Figure 1: Summary of results of four algorithms on various settings

I now discuss the results by answering the professor's questions.

Question 1: Which classifier (among the four) yields the best overall generalization accuracy/F1 score? Based on your ML knowledge, why do you think the "classifier" achieved the highest overall accuracy/F1 score?

Gradient Boosting has the best overall generalization. Even though this algorithm has not been covered in class, I learned that it combines techniques from boosting (adjusting the weights of the examples for better classification accuracy) with gradient descent. Therefore, it is expected to be more robust.

Question 2: What is the impact of increasing the amount of training data on the accuracy/F1 scores of each of the four classifiers?

For all four algorithms, more data means higher accuracy. Interesting, when giving only 100 examples, Random Forest out performs Bagging (at C being 300 and 500) and Gradient

Boosting (C = 300). This shows that Bagging and Gradient Boosting can only perform well with a lot of data, while Random Forest seems to be more data efficient.

Question 3: What is the impact of increasing the number of features on the accuracy/F1 scores of each of the four classifiers?

With more features, it is apparent that the performance of all algorithms improve across the board, regardless of the training set size.

The MNIST dataset

For hyperparameter tuning, I split the original training set into 50K:10K for training and evaluation. The hyperparameter grid is kept the same as the CNF experiments.

Results are as follows.

algorithm	acc	best_params
DecisionTreeClassifier	0.8802	{"random_state": 42, "criterion": "gini", "splitter": "best", "max_depth": 16, "min_samples_split": 8, "max_features": null}
BaggingClassifier	0.9674	{'random_state': 42, 'n_jobs': 24, 'estimator': DecisionTreeClassifier(random_state=42), 'n_estimators': 100, 'max_samples': 1.0, 'max_features': 0.1, 'bootstrap': True, 'oob_score': True, 'warm_start': False}
RandomForestClassifier	0.9739	{'random_state': 42, 'n_jobs': 24, 'criterion': 'gini', 'n_estimators': 100, 'max_depth': None, 'min_samples_split': 2, 'max_features': 'sqrt', 'bootstrap': False, 'oob_score': False}
GradientBoostingClassifier	0.9184	{'random_state': 42, 'n_estimators': 50, 'learning_rate': 0.1, 'criterion': 'friedman_mse', 'max_depth': 3, 'min_samples_split': 2, 'max_features': 'sqrt'}

Here, surprisingly, Random Forest performs the best. Following the CNF discussion, it seems like we are in the data-scarce setting for MNIST. Therefore, Random Forest tends to perform better.

Conclusion

The tree-based classifiers are highly effective in classification tasks, even the complex ones such as image recognition. Among the variants, Gradient Boosting consistently performs the best.

Appendix A: Random Forest on CNF

dataset	acc	f1	best_params
C=300, D=100	0.685	0.7123	{'criterion': 'gini', 'splitter': 'best', 'max_depth': 4, 'min_samples_split': 2, 'max_features': None}
C=300, D=1000	0.6755	0.669	{'criterion': 'gini', 'splitter': 'best', 'max_depth': None, 'min_samples_split': 0.1, 'max_features': None}
C=300, D=5000	0.7578	0.7779	{'criterion': 'gini', 'splitter': 'best', 'max_depth': None, 'min_samples_split': 0.01, 'max_features': None}
C=500, D=100	0.71	0.6979	{'criterion': 'entropy', 'splitter': 'best', 'max_depth': None, 'min_samples_split': 0.1, 'max_features': None}
C=500, D=1000	0.6995	0.6978	{'criterion': 'gini', 'splitter': 'best', 'max_depth': None, 'min_samples_split': 0.1, 'max_features': None}
C=500, D=5000	0.7764	0.7778	{'criterion': 'entropy', 'splitter': 'random', 'max_depth': None, 'min_samples_split': 2, 'max_features': None}
C=1000, D=100	0.73	0.7477	{'criterion': 'gini', 'splitter': 'random', 'max_depth': 4, 'min_samples_split': 2, 'max_features': None}
C=1000, D=1000	0.788	0.8004	{'criterion': 'entropy', 'splitter': 'best', 'max_depth': None, 'min_samples_split': 2, 'max_features': None}
C=1000, D=5000	0.8394	0.8452	{'criterion': 'gini', 'splitter': 'random', 'max_depth': None, 'min_samples_split': 0.01, 'max_features': None}
C=1500, D=100	0.82	0.8125	{'criterion': 'gini', 'splitter': 'best', 'max_depth': None, 'min_samples_split': 8, 'max_features': 'sqrt'}
C=1500, D=1000	0.933	0.9322	{'criterion': 'entropy', 'splitter': 'best', 'max_depth': None, 'min_samples_split': 0.01, 'max_features': None}
C=1500, D=5000	0.9554	0.9555	{'criterion': 'entropy', 'splitter': 'best', 'max_depth': None, 'min_samples_split': 8, 'max_features': None}
C=1800, D=100	0.94	0.9417	{'criterion': 'gini', 'n_estimators': 10, 'max_depth': 4, 'min_samples_split': 2, 'max_features': None, 'bootstrap': True, 'oob_score': True}

C=1800, D=1000	1.0	1.0	{'criterion': 'gini', 'n_estimators': 50, 'max_depth': None, 'min_samples_split': 2, 'max_features': 'sqrt', 'bootstrap': True, 'oob_score': True}
C=1800, D=5000	0.9999	0.9999	{'criterion': 'gini', 'n_estimators': 50, 'max_depth': None, 'min_samples_split': 2, 'max_features': 'sqrt', 'bootstrap': False, 'oob_score': False}

Appendix B: Bagging on CNF

dataset	acc	f1	best_params
C=300, D=100	0.74	0.7476	{'estimator': {'params': {'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'max_leaf_nodes': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'monotonic_cst': None, 'random_state': 42, 'splitter': 'best'}}, 'n_estimators': 100, 'max_samples': 0.5, 'max_features': 0.1, 'bootstrap': False, 'oob_score': False, 'warm_start': True}
C=300, D=1000	0.887	0.8898	{'estimator': {'params': {'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'max_leaf_nodes': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'monotonic_cst': None, 'random_state': 42, 'splitter': 'best'}}, 'n_estimators': 100, 'max_samples': 1.0, 'max_features': 1.0, 'bootstrap': True, 'oob_score': True, 'warm_start': False}
C=300, D=5000	0.9378	0.9384	{'estimator': {'params': {'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'max_leaf_nodes': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'monotonic_cst': None, 'random_state': 42, 'splitter': 'best'}}, 'n_estimators': 100, 'max_samples': 1.0, 'max_features': 0.5, 'bootstrap': True, 'oob_score': True, 'warm_start': False}
C=500, D=100	0.82	0.8182	{'estimator': {'params': {'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'max_leaf_nodes': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'monotonic_cst': None,

			'random_state': 42, 'splitter': 'best'}}, 'n_estimators': 100, 'max_samples': 0.5, 'max_features': 0.1, 'bootstrap': False, 'oob_score': False, 'warm_start': True}
C=500, D=1000	0.9285	0.9292	{'estimator': {'params': {'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'max_leaf_nodes': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'monotonic_cst': None, 'random_state': 42, 'splitter': 'best'}}, 'n_estimators': 100, 'max_samples': 0.5, 'max_features': 0.5, 'bootstrap': False, 'oob_score': False, 'warm_start': True}
C=500, D=5000	0.9601	0.9602	{'estimator': {'params': {'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'max_leaf_nodes': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'monotonic_cst': None, 'random_state': 42, 'splitter': 'best'}}, 'n_estimators': 100, 'max_samples': 1.0, 'max_features': 0.5, 'bootstrap': True, 'oob_score': True, 'warm_start': False}
C=1000, D=100	0.985	0.9851	{'estimator': {'params': {'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'max_leaf_nodes': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'monotonic_cst': None, 'random_state': 42, 'splitter': 'best'}}, 'n_estimators': 100, 'max_samples': 0.5, 'max_features': 0.1, 'bootstrap': False, 'oob_score': False, 'warm_start': True}
C=1000, D=1000	0.9915	0.9915	{'estimator': {'params': {'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'max_leaf_nodes': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'monotonic_cst': None, 'random_state': 42, 'splitter': 'best'}}, 'n_estimators': 100, 'max_samples': 1.0, 'max_features': 0.1, 'bootstrap': True, 'oob_score': True, 'warm_start': False}
C=1000, D=5000	0.9961	0.9961	{'estimator': {'params': {'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'max_leaf_nodes': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'monotonic_cst': None, 'random_state': 42, 'splitter': 'best'}}, 'n_estimators': 100,

			'max_samples': 1.0, 'max_features': 0.1, 'bootstrap': True, 'oob_score': True, 'warm_start': False}
C=1500, D=100	1.0	1.0	{'estimator': {'params': {'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'max_leaf_nodes': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'monotonic_cst': None, 'random_state': 42, 'splitter': 'best'}}, 'n_estimators': 100, 'max_samples': 0.1, 'max_features': 0.5, 'bootstrap': True, 'oob_score': True, 'warm_start': False}
C=1500, D=1000	1.0	1.0	{'estimator': {'params': {'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'max_leaf_nodes': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'monotonic_cst': None, 'random_state': 42, 'splitter': 'best'}}, 'n_estimators': 50, 'max_samples': 0.5, 'max_features': 0.1, 'bootstrap': True, 'oob_score': True, 'warm_start': False}
C=1500, D=5000	0.9997	0.9997	{'estimator': {'params': {'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'max_leaf_nodes': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'monotonic_cst': None, 'random_state': 42, 'splitter': 'best'}}, 'n_estimators': 50, 'max_samples': 1.0, 'max_features': 0.1, 'bootstrap': True, 'oob_score': True, 'warm_start': False}
C=1800, D=100	0.97	0.97	{'estimator': {'params': {'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'max_leaf_nodes': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'random_state': 42, 'splitter': 'best'}}, 'n_estimators': 10, 'max_samples': 1.0, 'max_features': 1.0, 'bootstrap': True, 'oob_score': True, 'warm_start': False}
C=1800, D=1000	1.0	1.0	{'estimator': {'params': {'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'max_leaf_nodes': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'random_state': 42, 'splitter': 'best'}}, 'n_estimators': 50, 'max_samples': 0.1, 'max_features': 0.1, 'bootstrap': True, 'oob_score': True, 'warm_start': False}

C=1800 , D=5000	1.0	1.0	{'estimator': {'params': {'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'max_leaf_nodes': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'random_state': 42, 'splitter': 'best'}}, 'n_estimators': 50, 'max_samples': 0.5, 'max_features': 0.1, 'bootstrap': True, 'oob_score': True, 'warm_start': False}

Appendix C: Random Forest on CNF

dataset	acc	f1	best_params
C=300, D=100	0.85	0.85	{'criterion': 'gini', 'n_estimators': 100, 'max_depth': 4, 'min_samples_split': 2, 'max_features': 'sqrt', 'bootstrap': True, 'oob_score': True}
C=300, D=1000	0.868	0.8712	{'criterion': 'gini', 'n_estimators': 100, 'max_depth': None, 'min_samples_split': 0.01, 'max_features': None, 'bootstrap': True, 'oob_score': True}
C=300, D=5000	0.9208	0.9251	{'criterion': 'gini', 'n_estimators': 100, 'max_depth': None, 'min_samples_split': 2, 'max_features': None, 'bootstrap': True, 'oob_score': True}
C=500, D=100	0.89	0.8866	{'criterion': 'gini', 'n_estimators': 100, 'max_depth': 4, 'min_samples_split': 2, 'max_features': 'sqrt', 'bootstrap': False, 'oob_score': False}
C=500, D=1000	0.93	0.9297	{'criterion': 'gini', 'n_estimators': 100, 'max_depth': None, 'min_samples_split': 0.01, 'max_features': 'sqrt', 'bootstrap': True, 'oob_score': True}
C=500, D=5000	0.9524	0.9524	{'criterion': 'gini', 'n_estimators': 100, 'max_depth': None, 'min_samples_split': 2, 'max_features': 'sqrt', 'bootstrap': False, 'oob_score': False}
C=1000, D=100	0.97	0.97	{'criterion': 'gini', 'n_estimators': 100, 'max_depth': 4, 'min_samples_split': 2, 'max_features': 'sqrt', 'bootstrap': False, 'oob_score': False}
C=1000, D=1000	0.9875	0.9875	{'criterion': 'gini', 'n_estimators': 100, 'max_depth': None, 'min_samples_split': 2, 'max_features': 'sqrt', 'bootstrap': True, 'oob_score': True}

C=1000, D=5000	0.9925	0.9925	{'criterion': 'gini', 'n_estimators': 100, 'max_depth': None, 'min_samples_split': 2, 'max_features': 'sqrt', 'bootstrap': True, 'oob_score': True}
C=1500, D=100	1.0	1.0	{'criterion': 'gini', 'n_estimators': 50, 'max_depth': None, 'min_samples_split': 2, 'max_features': 'sqrt', 'bootstrap': True, 'oob_score': True}
C=1500, D=1000	0.9995	0.9995	{'criterion': 'gini', 'n_estimators': 50, 'max_depth': None, 'min_samples_split': 2, 'max_features': 'sqrt', 'bootstrap': True, 'oob_score': True}
C=1500, D=5000	0.9995	0.9995	{'criterion': 'gini', 'n_estimators': 100, 'max_depth': None, 'min_samples_split': 2, 'max_features': 'sqrt', 'bootstrap': True, 'oob_score': True}
C=1800, D=100	0.94	0.9417	{'criterion': 'gini', 'n_estimators': 10, 'max_depth': 4, 'min_samples_split': 2, 'max_features': None, 'bootstrap': True, 'oob_score': True}
C=1800, D=1000	1.0	1.0	{'criterion': 'gini', 'n_estimators': 50, 'max_depth': None, 'min_samples_split': 2, 'max_features': 'sqrt', 'bootstrap': True, 'oob_score': True}
C=1800, D=5000	0.9999	0.9999	{'criterion': 'gini', 'n_estimators': 50, 'max_depth': None, 'min_samples_split': 2, 'max_features': 'sqrt', 'bootstrap': False, 'oob_score': False}

Appendix D: Gradient Boosting on CNF

dataset	**acc**	**f1**	**best_params**
C=300, D=100	0.805	0.8079	{'n_estimators': 100, 'learning_rate': 0.01, 'criterion': 'friedman_mse', 'max_depth': 3, 'min_samples_split': 2, 'max_features': 'sqrt'}
C=300, D=1000	0.9725	0.9732	{'n_estimators': 100, 'learning_rate': 0.1, 'criterion': 'friedman_mse', 'max_depth': 3, 'min_samples_split': 2, 'max_features': None}
C=300, D=5000	0.9978	0.9978	{'n_estimators': 100, 'learning_rate': 0.1, 'criterion': 'friedman_mse', 'max_depth': None, 'min_samples_split': 0.01, 'max_features': None}
C=500, D=100	0.91	0.9062	{'n_estimators': 100, 'learning_rate': 0.1, 'criterion': 'friedman_mse', 'max_depth': 3, 'min_samples_split': 2, 'max_features': 'sqrt'}

C=500, D=1000	0.9845	0.9846	{'n_estimators': 100, 'learning_rate': 0.1, 'criterion': 'friedman_mse', 'max_depth': 3, 'min_samples_split': 2, 'max_features': None}
C=500, D=5000	0.9987	0.9987	{'n_estimators': 100, 'learning_rate': 0.1, 'criterion': 'friedman_mse', 'max_depth': None, 'min_samples_split': 0.01, 'max_features': None}
C=1000, D=100	0.98	0.9804	{'n_estimators': 100, 'learning_rate': 0.1, 'criterion': 'squared_error', 'max_depth': None, 'min_samples_split': 2, 'max_features': 'sqrt'}
C=1000, D=1000	0.9925	0.9925	{'n_estimators': 100, 'learning_rate': 0.1, 'criterion': 'friedman_mse', 'max_depth': None, 'min_samples_split': 0.01, 'max_features': 'sqrt'}
C=1000, D=5000	0.9995	0.9995	{'n_estimators': 100, 'learning_rate': 0.1, 'criterion': 'friedman_mse', 'max_depth': None, 'min_samples_split': 0.01, 'max_features': None}
C=1500, D=100	1.0	1.0	{'n_estimators': 50, 'learning_rate': 0.1, 'criterion': 'friedman_mse', 'max_depth': 3, 'min_samples_split': 2, 'max_features': 'sqrt'}
C=1500, D=1000	0.9995	0.9995	{'n_estimators': 100, 'learning_rate': 0.1, 'criterion': 'friedman_mse', 'max_depth': 3, 'min_samples_split': 2, 'max_features': 'sqrt'}
C=1500, D=5000	1.0	1.0	{'n_estimators': 100, 'learning_rate': 0.1, 'criterion': 'friedman_mse', 'max_depth': None, 'min_samples_split': 0.01, 'max_features': 'sqrt'}
C=1800, D=100	1.0	1.0	{'n_estimators': 50, 'learning_rate': 0.1, 'criterion': 'friedman_mse', 'max_depth': 3, 'min_samples_split': 2, 'max_features': 'sqrt'}
C=1800, D=1000	1.0	1.0	{'n_estimators': 50, 'learning_rate': 0.1, 'criterion': 'friedman_mse', 'max_depth': 3, 'min_samples_split': 2, 'max_features': 'sqrt'}
C=1800, D=5000	0.9999	0.9999	{'n_estimators': 50, 'learning_rate': 0.1, 'criterion': 'friedman_mse', 'max_depth': None, 'min_samples_split': 2, 'max_features': 'sqrt'}