COMPENG 4SL4 Assignment 4 Report

Raeed Hassan
hassam41
400188200

November 27, 2022

Data Set Loading and Splitting

The data set was saved and loaded using `dataset = pd.read_csv('Data/spambase.data', header=None)` and split into a training set (containing two thirds of the data sample points), and a test set (containing the other third of the data sample points). The training and test data sets were split using the `sklearn.model_selection.train_test_split` function, using the last 4 numbers of my student ID (8200) as the random state for the function.

K-Fold Cross-Validation Setup

For K-fold cross-validation in this lab, the value of K was set to 5. Arrays containing the training and test indexes for each fold were created using `sklearn.model_selection.KFold`. These indices are used for all K-Fold cross-validation done throughout the lab. These indices are used for row indexing to generate the feature and target matrices before doing regression.

Decision Tree Classifier

The decision tree classifier was implemented with `sklearn.tree.DecisionTreeClassifier`. The cross-validation error with 5-fold cross-validation was calculated for 2 to 400 maximum number of leaves, and the best n maximum number of leaves (least cross-validation error) was selected.

The cross-validation error versus the maximum number of leaves is plotted in Figure 1. The model producing the least cross-validation error was with 100 maximum number of leaves, with a cross-validation error of 0.0839546191247974 and a test error of 0.0901909150757077.

Bagging Classifier

The bagging classifier was implemented with `sklearn.ensemble.BaggingClassifier`. The test error for bagging classifiers with n predictors for $n = 50$ to 2500 (in increments of 50) was calculated.

The test error versus the number of predictors is plotted in Figure 2. The ensemble producing the least test error was with $n = 700$ predictors in the ensemble, with a test error of 0.05924950625411455.

Random Forest Classifier

The random forest classifier was implemented with the `RandomForestClassifier` classifier from `sklearn.ensemble`. The test error for random forest classifiers with n predictors for $n = 50$ to 2500 (in increments of 50) was calculated.

The test error versus the number of predictors is plotted in Figure 2. The ensemble producing the least test error was with $n = 1950, 2200 - 2500$ predictors in the ensemble, with a test error of 0.05200789993416721.

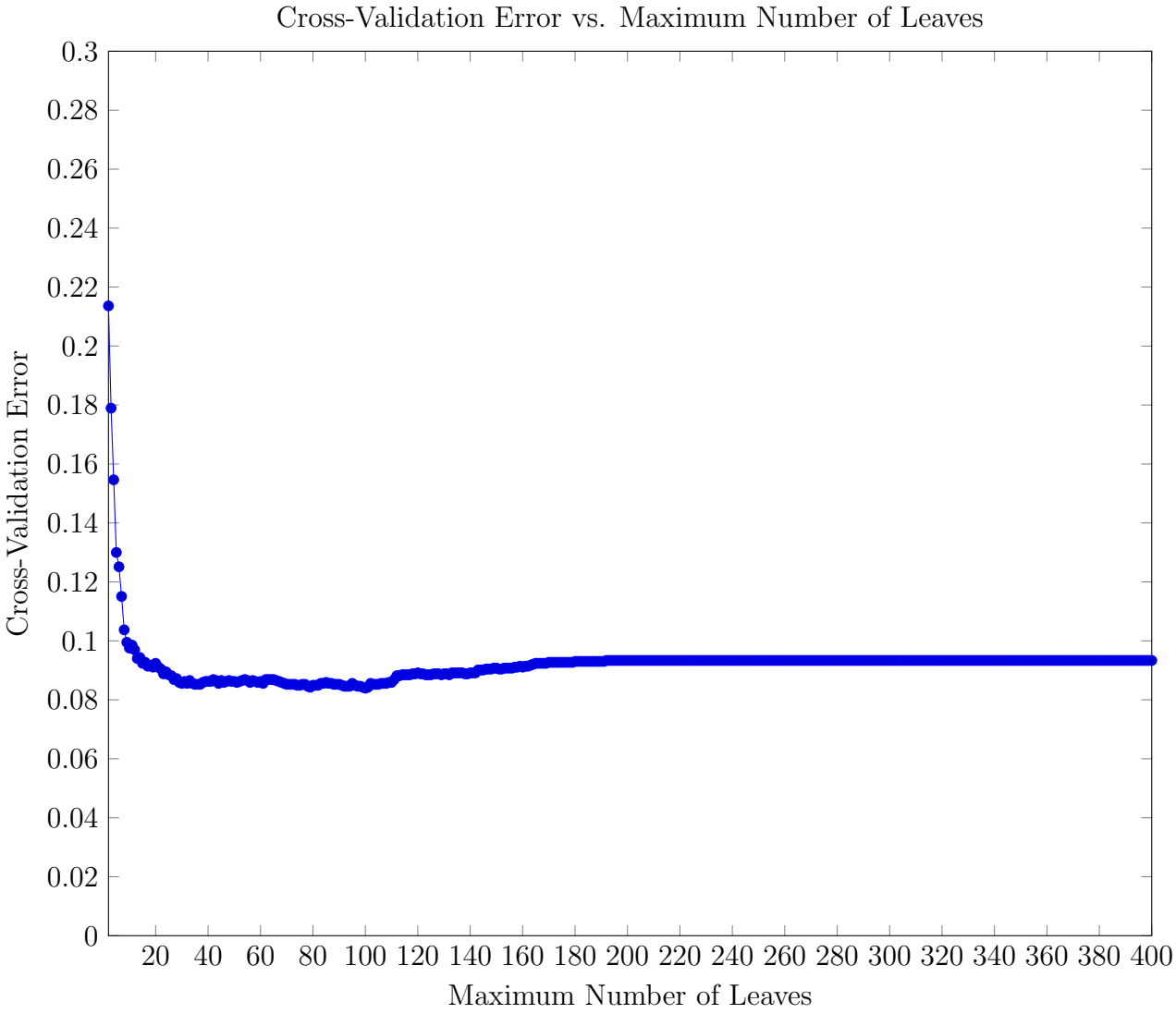

Figure 1: Cross-Validation Error versus Maximum Number of Leaves

Adaboost

The Adaboost classifiers were implemented with `sklearn.ensemble.AdaBoostClassifier`. The test error for Adaboost classifiers with n predictors for $n = 50$ to 2500 (in increments of 50) was calculated. This was performed for decision trees with decision stumps (max_depth = 1), decision trees with at most 10 leaves (max_leaf_nodes = 10), and decision trees with no restriction on depth or node number.

The test errors versus the number of predictors for all three Adaboost scenarios are plotted in Figure 2. The ensemble producing the least test error with decision stumps was with $n = 400$ predictors in the ensemble, with a test error of 0.05924950625411455. The ensemble producing the least test error with a maximum of 10 leaves was with $n = 750, 1250$ predictors in the ensemble, with a test error of 0.03620803159973667. The ensemble producing the least test error with no restriction on depth or node number was with $n = 100-400, 500$ predictors in the ensemble, with a test error of 0.05069124423963134.

Ensemble Comparison

The test errors versus n number of predictors for all 5 ensemble methods are plotted in Figure 2. We can observe that all ensemble methods outperformed the single decision tree classifier. The Adaboost classifiers with a maximum of 10 leaves performed the best (least test error) at every number of predictors. The performance of the ensemble methods can be ordered as (from best to worse):

1. Adaboost (max 10 leaves)
2. Adaboost (no restrictions)
3. Random Forest
4. Bagging
5. Adaboost (decision stumps)

This performance ranking is maintained for most n number of predictors, with the test errors for any ensemble method crossing another's test error at most twice. The performance is very similar (generally less than 0.05 difference in test error) between the 2nd and 3rd best methods, and the 4th and 5th best methods. The classifiers can be roughly split into three groups that are about 0.01 test error apart (1st best, 2nd and 3rd best, 4th and 5th best).

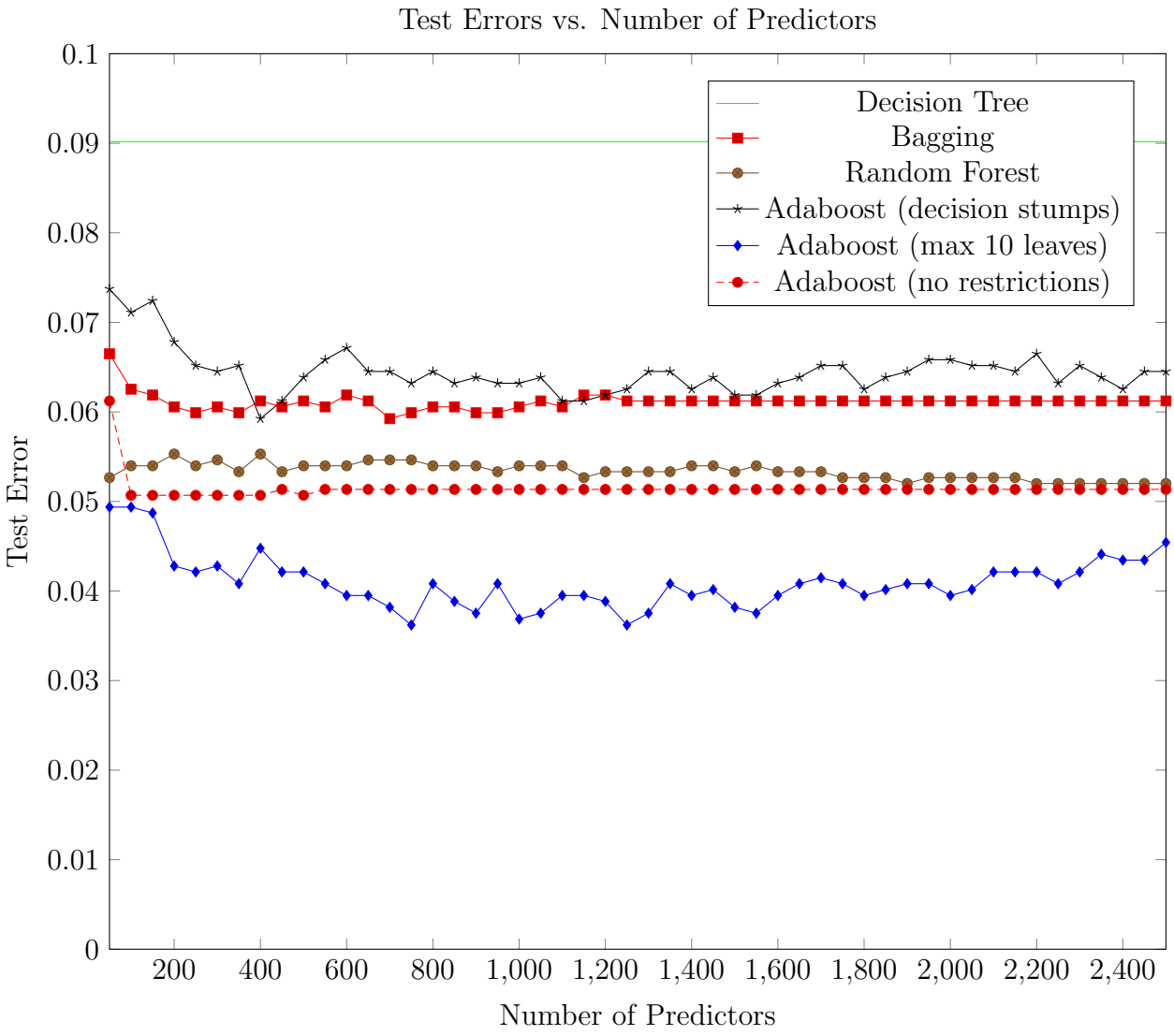

Figure 2: Test Errors versus Number of Predictors for Multiple Ensemble Methods