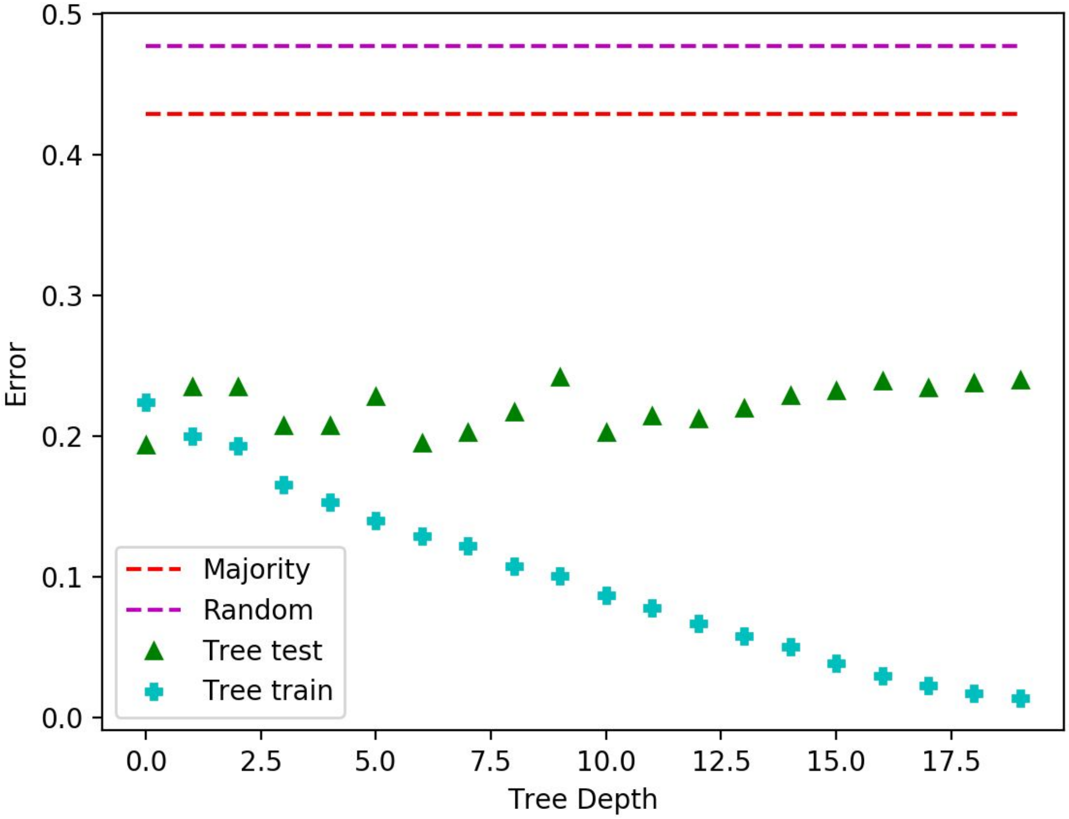
1.

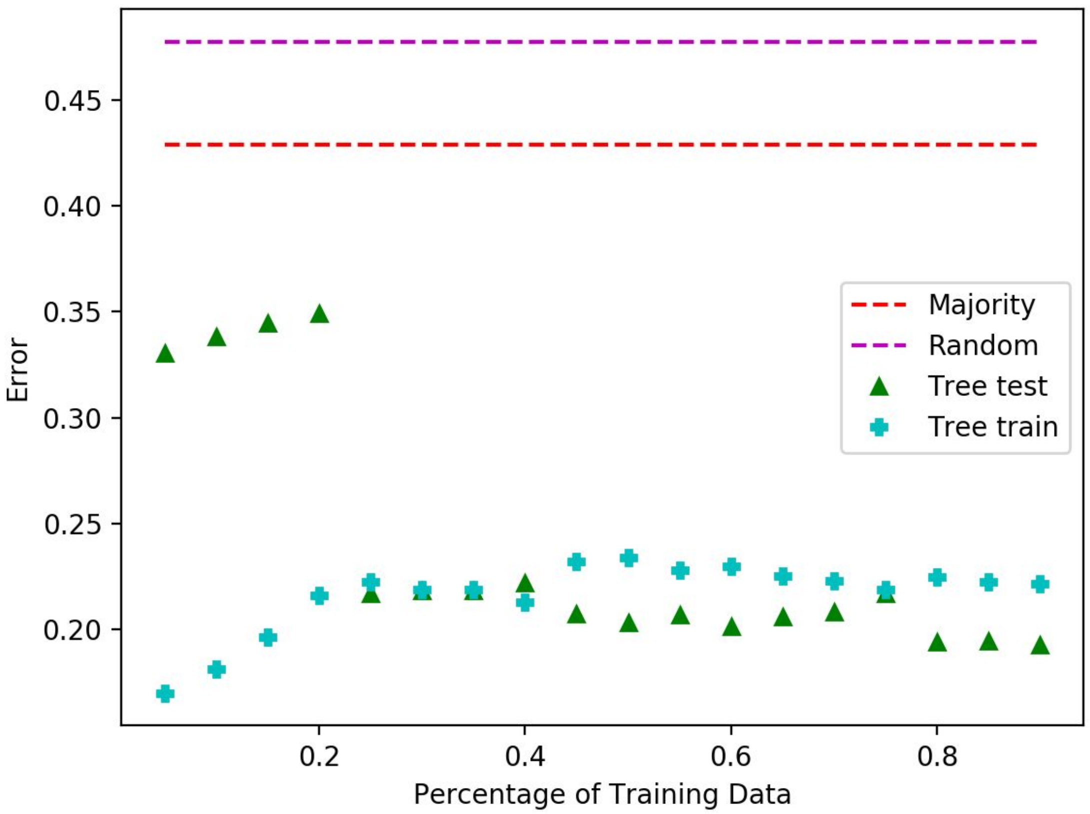
a. Training error of the DecisionTreeClassifier = 0.014

b. Average training error of the MajorityVoteClassifier = 0.393
   Average testing error of the MajorityVoteClassifier = 0.429
   Average training error of the RandomClassifier = 0.512
   Average testing error of the RandomClassifier = 0.478
   Average training error of the DecisionTreeClassifier = 0.012
   Average testing error of the DecisionTreeClassifier = 0.239

c. The best tree depth to use for this data is 1. The tree depth of 1 corresponds to the lowest testing error so if we assume the testing data is representative of the overall distribution, a tree depth of 1 will minimize error. We see a bit of overfitting because as the max depth increases, the training error for the tree decreases significantly while the testing error for the tree increases slightly.



d. We see that for very small percentages of training data (<20%), the testing error is very high while the training data is very low. This is likely because for very small datasets, it is easy for the classifier to build a model that reflects the training data very well; however, because the training data is so small the model does not reflect the test data. For percentages of training data >20%, we see relatively uniform training error and testing error.



Extra Credit Contest Write-up:

For our contest decision tree we decided to alter 3 hyperparameters: the maximum depth, the minimum impurity decrease, and the min samples split. To determine the best values for these hyperparameters, we wrote a function, make_contest_predictions that tests all possible combinations of the hyperparameters. We varied min samples split from 2 to 20, max depth from 1 to 10, and minimum impurity decrease from .05 to .25. We then evaluated the performance of each combination using cross-validation error. We found the best parameters to use were 3 for max depth, 10 for min samples split and .15 for minimum impurity decrease.