Random Deforestation

Henry and Eric

Problem

How do we prune random forests?

- Decision trees can be pre-pruned and post-pruned
    - Pre-pruning is easier but risks underfitting
    - Post-pruning is time intensive
- To maximize the benefits of each method, we decided to try pre-pruning half the decision trees and post-pruning the other half
- We call this random deforestation

Related Works

- Maximum Risk Minimization with Random Forests [1]
  - Weights each individual tree based on risk, “ensemble level”
- Pruning Random Forests for Prediction on a Budget [2]
  - Post-pruning, but for a different goal
- An analysis of ensemble pruning methods under the explanation of Random Forest [3]
  - Prunes entire trees, trying to improve efficiency and reduce redundancy
- Cost-complexity pruning of random forests [4]
  - Extremely similar to our project, except they rely only on post-pruning
- Pruning Random Forest with Orthogonal Matching Trees [5]
  - Is still “post pruning” for the sake of reducing the size of the forest like us, but prunes entire trees and then weights the remaining trees after based on what subsets of trees have the best combined predictions

Datasets

- The Iris dataset contains 150 instances, where each instance is a flower. Each instance has four attributes and the class:
  - Sepal length
  - Sepal width
  - Petal length
  - Petal width
  - Species - Iris setosa, Iris virginica, or Iris versicolor
- We split into 105 training and 45 testing
- The diabetes dataset has 768 instances, each representing one person
  - Each instance has a variety of health data
  - Prediction target is binary, diabetes or not
- We split it into 537 testing instances and 231 testing instances

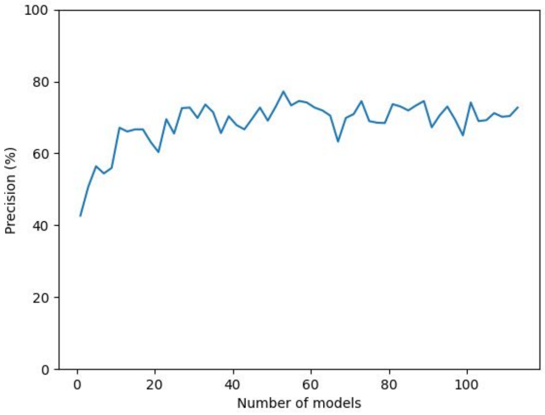
Methods

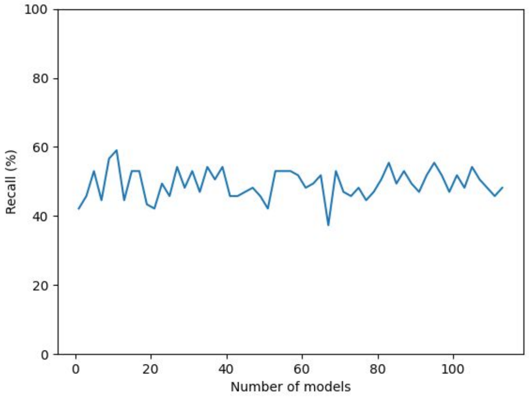
- Pre-pruning: maximum height
  - Caps tree building at a specified height
- Post-pruning: cost complexity pruning
  - Recursively removes the node with the highest cost complexity
  - Until the tree’s cost complexity falls below a threshold called alpha
  - Cost complexity is the sum of cost (number of incorrect instances) and complexity (branching factor)
- Three parameters: maximum height, ccp_alpha, and number of trees

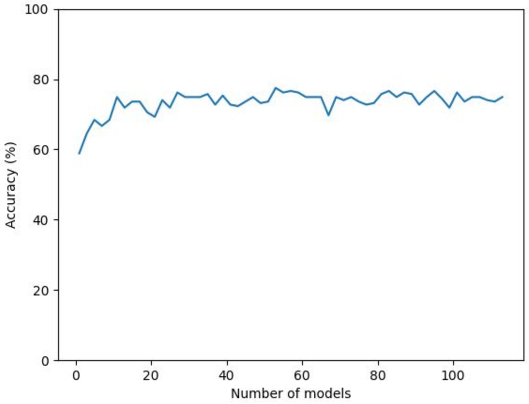
Results

Diabetes baseline random forest:

Average accuracy: 73.46%



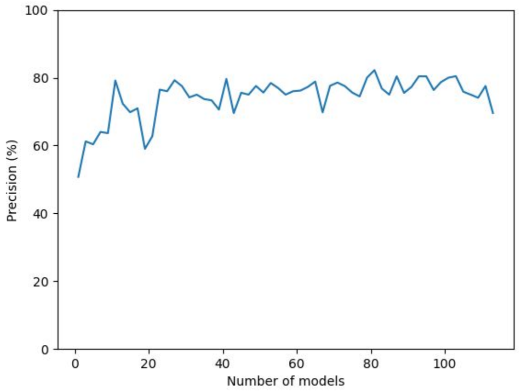


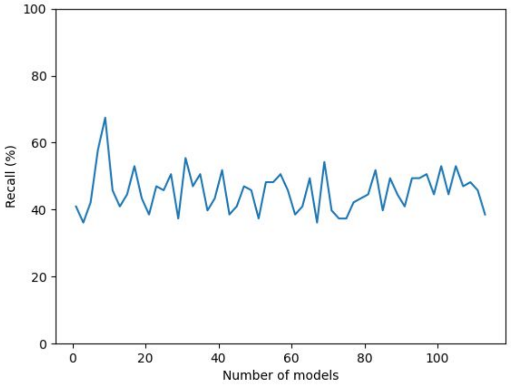


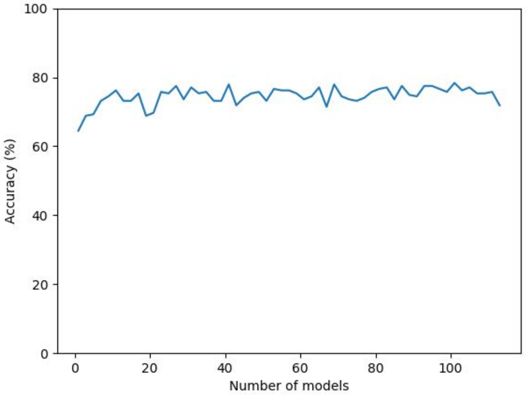
Results

Diabetes best model was with half of the trees pre-pruned with max height of 6 and half post-pruned with an alpha value of 0.004.

Average accuracy: 74.60%



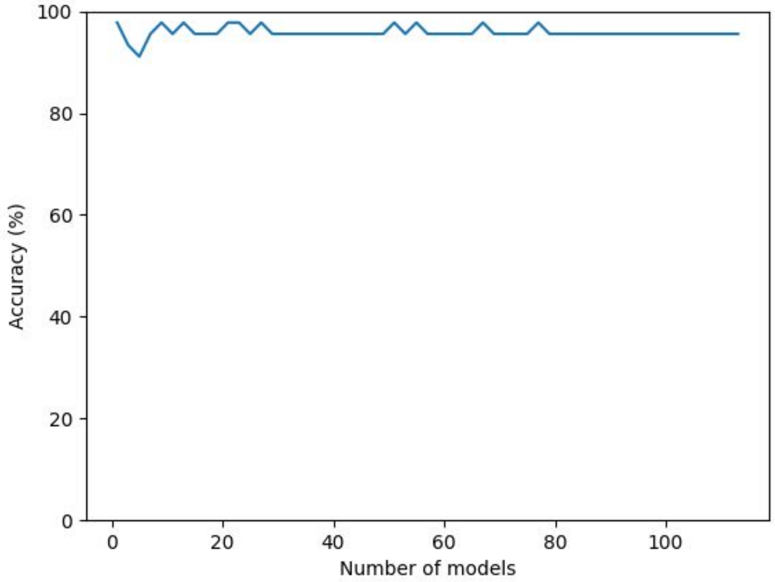




Results

Iris dataset baseline:

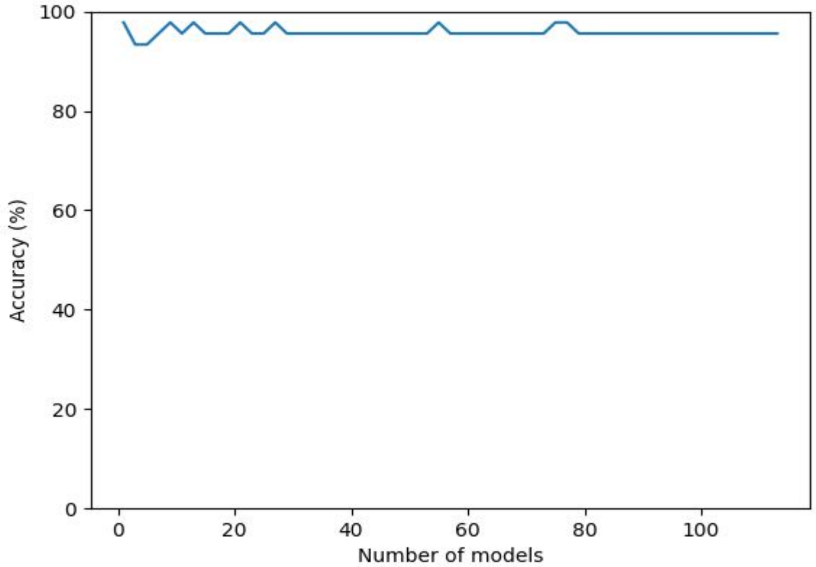
Average accuracy: 95.83%



Results

All Iris dataset pruned trees:

Average accuracy ~95.7%



Discussion and conclusions

- No performance improvement
- Future work could look at alternative pruning methods
- Random forests are already designed to overcome the problem of decision tree overfitting
    - Pruning may be trying to solve a problem that doesn’t exist
- There are still benefits!
    - For the largest Iris model, 300kb unpruned vs 160kb pruned
    - Same performance, smaller model
    - Benefits in specific use cases when model size and classification speed are worth the sacrifice of the computational cost to prune the trees

References

[1] F. Freni, A. Fries, L. Kühne, M. Reichstein, and J. Peters, “Maximum Risk Minimization with Random Forests,” 2025, arXiv. doi: 10.48550/ARXIV.2512.10445.

[2] F. Nan, J. Wang, and V. Saligrama, “Pruning Random Forests for Prediction on a Budget,” Jun. 16, 2016, arXiv: arXiv:1606.05060. doi: 10.48550/arXiv.1606.05060.

[3] F. A. Khalifa, H. M. Abdelkader, and A. H. Elsaid, “An analysis of ensemble pruning methods under the explanation of Random Forest,” Information Systems, vol. 120, p. 102310, Feb. 2024, doi: 10.1016/j.is.2023.102310.

[4] K. B. Ravi and J. Serra, “Cost-complexity pruning of random forests,” Jul. 19, 2017, arXiv: arXiv:1703.05430. doi: 10.48550/arXiv.1703.05430.

[5] L. Giffon, C. Lamothe, L. Bouscarrat, P. Milanesi, F. Cherfaoui, and S. Koço, “Pruning Random Forest with Orthogonal Matching Trees,” 2020.