Connor Kutz

Bird Classification Using Random Forest

I. PROJECT OVERVIEW

Though automated bird classification is not a major issue in the world today, the information that we gather by solving datasets such as these can be very valuable. In this project, I used Random Forest to classify birds from the Caltech-UCSD Birds-200 2011 dataset[1]. In classifying this dataset, I also wanted to show how changing parameters and data affects performance and accuracy of the results from Random Forest. The project was inspired by Dr. Caliskan's lecture on bagging.

II. PROBLEM STATEMENT

The goal is to create a Random Forest classifier using Matlab and the tasks involved are as follows:

1. Download and process the Caltech-UCSD Birds dataset
2. Train a random forest classifier on the binary attributes
3. Measure performance including accuracy, misclassifications, and runtime
4. Optimize data and classifier parameters
5. Measure performance again and analyze results

III. RELATED WORK

There have been many projects which aim to classify birds, for example Vidaña-Vila and Navarro aimed to classify birds from recordings[2] of their songs and were quite successful. Later on, there's been research on classifying birds solely on color features[3]. Many have used random forest for similar classification problems

IV. METRICS

For this classifier, accuracy is measured in terms of "estimated out-of-bag" error. This means that for each bootstrap sample taken from the data, the mean prediction error is measured on each tree without the given bootstrap sample.

This can be written as:

---

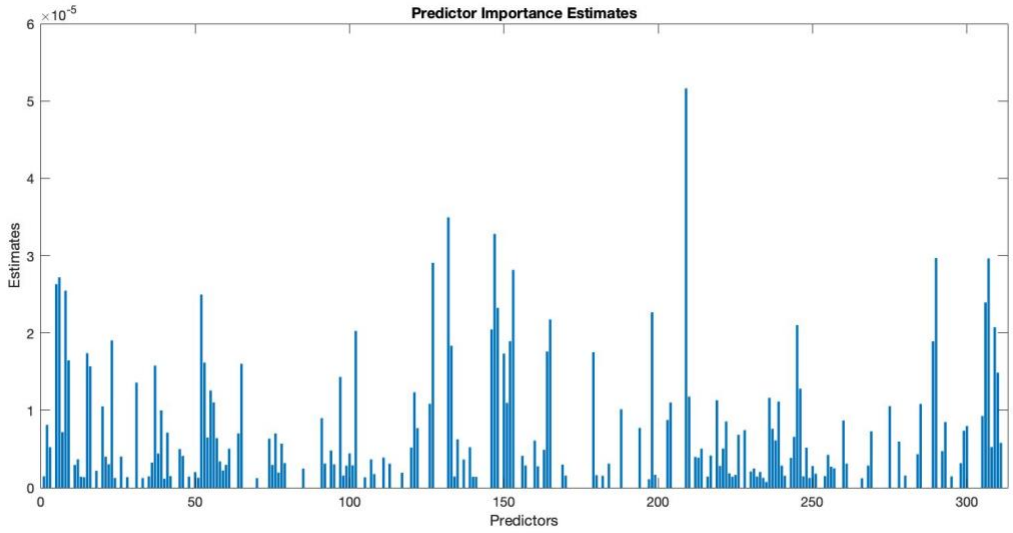
[1] https://pdfs.semanticscholar.org/c069/629a51f6c1c301eb20ed77bc6b586c24ce32.pdf
[2] https://www.mdpi.com/2306-5729/2/2/18/htm
[3] https://pdfs.semanticscholar.org/c069/629a51f6c1c301eb20ed77bc6b586c24ce32.pdf

$$E = \sum_t q * (t)|p * (t) - p(t)|$$

Where t is a terminal node, and $q * (t) = P(X \in t)$ is the chance that a bird is gets into that node of the tree.

We measure using out-of-bag error because we want to know how the trees and forest perform on the data without the observations from the most recent sample. Not only does this allow us to measure accuracy increase with each additional tree, but it gives a more accurate error percentage and negates the need for a separate validation data split. The only downside of this method is that it can underestimate the actual performance of the random forest.

Feature importance was calculated at the end of each forest training. The method used was to find the sum of risk changes due to splits on each predictor. After that, the sum was divided by the number of branch nodes on each tree. Using this information, unused predictors could be removed from the data set to improve running time. Later, less important features could also be removed, improving overall accuracy.



The original runtime to create a model using the entire dataset was over one hour. Since this project was on a deadline, I opted to shrink the dataset by ~90 percent, classifying 18 species of birds with about 30,000 attributes total. This was done to allow time to perform the necessary optimization and testing for the project.

V. DATASET

The Caltech Birds dataset includes ~11,000 images of birds, each image also comes with 312 labeled binary attributes one-hot encoded. Only the binary features were used to classify, not the images. Labels included features such as “has yellow throat,”

Yellow Breasted Chat

and “has black belly feathers.” In this dataset, images were labeled by the researchers, but each of the 312 binary attributes included with each image was labeled by Mechanical Turkers. Because of this, there were quite a few discrepancies in the data that caused issues classifying.

For example, one such problematic label I noticed was “has Spatulate shaped beak.” About half of Black Footed Albatross were labeled as having a spatulate shaped beak, though it’s incorrect.

True Spatulate

Black Footed Albatross

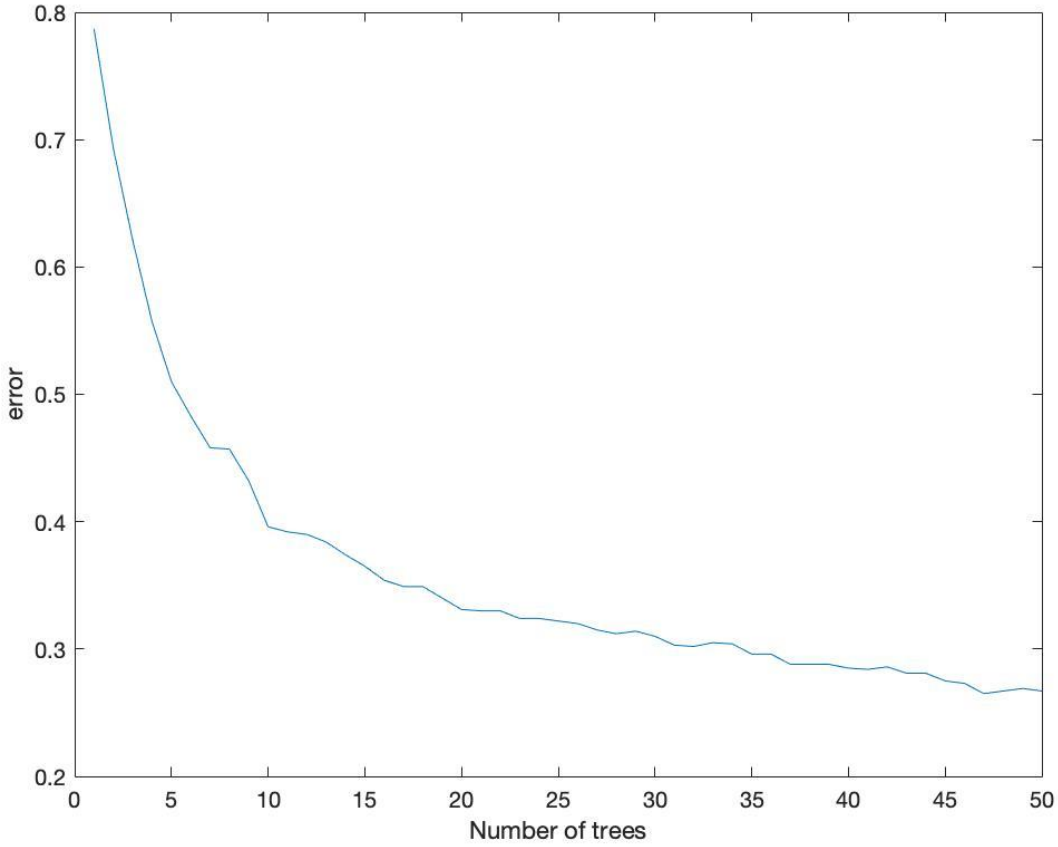
Because of this mislabeling of attributes, Black Footed Albatross were almost never classified correctly. The best way I was able to solve issues like this was to remove this feature from the model completely. As another example, Brewer Blackbirds were classified equally often with “has black eyes,” and “has white eyes,” which caused a similar problem.

VI. METHODS

Prior to loading the data into Matlab for processing, a large amount of preprocessing was done in Microsoft Excel. Here, data was organized and resized to work on my personal computer. After importing the data to Matlab, it was further separated into class labels and attributes then used in the Random Forest model.

The model was trained on bagged training sets from the data. Bagged sets are taken from the dataset and are the same size as the original dataset, the difference is that examples are chosen with replacement. For large datasets, the data can be estimated to be 1-1/e = 63.2% unique examples, per bootstrap sample. Bagging is used because it trains on a more realistic estimation of testing data thereby reducing variance and overfitting.

Before parameter refinement, the model was already fairly accurate at ~30% error. After performing hyperparameter optimization, classification accuracy converged around 75%



VII. CONCLUSION

It is well known that Random Forest is a very effective classifier and can be proven by Condorcet's Jury Theorem, but the goal of my project was to analyze the effect of good data and parameters on the results. Random Forest ended up being very insensitive to hyperparameters and only improved classification accuracy by 5% after optimization.

The data had a larger effect on the results of the classifier than anything else. This project ended up being a tool to investigate just how much having good clean data can produce a very strong classifier.