# Predicting Term Deposits Subscriptions with Tree Ensemble Methods

Brandon Novak

April 26th, 2021

## Introduction

In this exploration of categorical data from a Portuguese bank, we investigate the accuracy of decision trees and its respective ensemble methods. Decision trees are highly effective supervised machine learning techniques that split data into subgroups that will decrease a specified error metric. Decision trees greatest strength is its interpretability, but its downfall is its susceptibility to overfitting. Ensemble methods are strategies to keep the interpretability of the trees, but decrease the likelihood of overfitting the model. In this particular examination, we are implementing decision trees on a binary categorical variable. We use the decision tree classifier provided by the scikit-learn machine learning Python module. The ensemble methods implemented are also provided by the scikit-learn module. The goal of this investigation is to demonstrate which ensemble method is the most effective classifier.

### Methodology and Data

#### Decision Tree

As mentioned before, decision trees are very popular and effective supervised learning techniques to predict a target variable. Decision trees' nodes start as observations and conclude as predictions. As a single observation traverses down the tree, the prediction is the value of the leaf node at the bottom of the tree. The tree is built by splitting data into subgroups that decrease a spec-

ified error metric. For regressors, the error metric can be mse, friedman_mse, mae, or poisson. Classifiers typically use gini impurity or entropy.

### Ensemble Methods

Ensemble methods are strategies of solving artificial intelligence problems with a group of solutions. Each ensemble method reduces the likelihood of overfitting through different methods. The typical structure of ensemble methods is to create multiple models and then combine their results to produce an aggregated result. The three tree ensemble methods that we discuss are random forest, bagging, and boosting. Each method will be explained and their results will follow.

### Random Forest

Random forest is a tree ensemble method that builds trees by selecting a random subset of attributes from the original dataset. Each subtree is a decision tree that uses different features of the dataset. Typically, the number of attributes is $\sqrt{d}$, where $d$ is the number of attributes. For every observation, each tree will produce a prediction, and the aggregated result will either be the average, or the category that was predicted the most in the subtrees.

### Bagging

The bagging method is an effective strategy when the size of the observation is not very large. Bagging involves bootstrap sampling which is a method of random sampling with replacement. Bagging uses every attribute to build trees, but only uses a subset of the observations. This method is similar to random forest in that we limit input. In random forest, each tree uses a different subset of attributes. In bagging, each tree uses a different subset of observations. Similar to random forest, the output is the aggregated result of all the subtrees.

### Boosting

Boosting is the final ensemble method that we are evaluating in this examination. There are many different types of boosting. We chose to imple-

ment gradient boosting in our investigation. All boosting algorithms intend to increase training on observations that are misclassified. This is done through weights. In a typical boosting algorithm, each observation has weight $\frac{1}{n}$, where $n$ is the number of samples. If the observation is predicted correctly, its corresponding weight remains unchanged. But if the observation is misclassified, the corresponding weight increases by a pre-specified $\alpha$. However, gradient boosting in particular tries to fit the new predictor to the residual errors made by the previous predictor. Furthermore, gradient boosting uses a new cost function in each iteration to increase the probability of each observation being classified correctly.

### Data

The dataset was acquired from the UCI machine Learning Repository. There are 20 variables in the dataset, 19 predictors and 1 response. Most variables are categorical. The dataset is client data from a bank in Portugal. The data derives from a marketing campaign where clients were called if they were interested in opening a term deposit which is a contract for the bank to pay the client interest until the term ended. The predictors are descriptions of the clients. The response variable is if they subscribed to a term deposit. A table with a description of each variable can be found at Table 1 in the Appendix.

## Decision Tree results

The results for the decision trees are quite impressive. We create two decision trees: one to minimize the gini impurity, and the other to minimize the entropy. Gini impurity is a measurement of the likelihood of an incorrect classification of a new instance of a variable. It is calculated with the following equation:

$$G_k = \sum_{i=1}^{C} \sum_{j=1} N(j)N(i)$$

where $N(i)$ is the fraction of sample in class $i$ and $C$ is the categories.

Gini Impurity is typically the measurement used in classification trees. The accuracy of the decision tree when using the Gini Impurity is **.90118**. This

means that our model was about 90% effective at predicting the correct response from the clients' data.

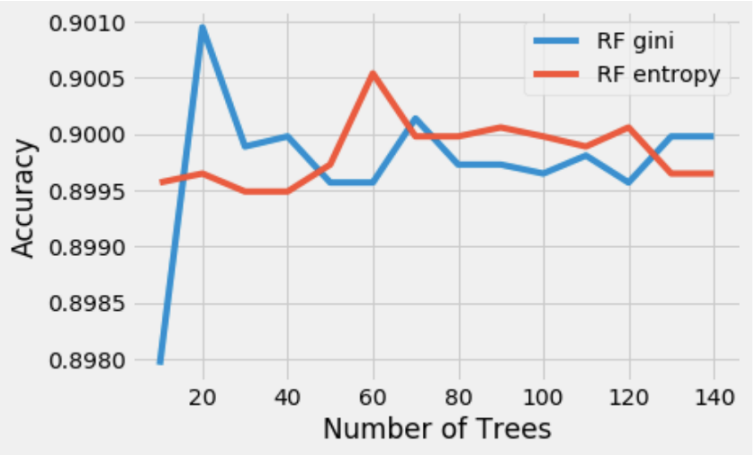
Entropy is the measure of uncertainty in the dataset. Entropy is measured on a scale from 0-1. The goal of the decision tree is to minimize the entropy or the amount of uncertainty. Entropy is also used to calculate the amount of information gained in each iteration of the tree. The mathematical formula for entropy is

$$s = \sum_{i=1}^{n} -p_i - log_2(p_i)$$

The results of the decision tree using entropy is **.89900**. This is not as effective at predicting the target variables as the decision tree using gini. However, both are extremely similar and effective at predicting if the client subscribes to a term deposit. Now that the accuracies of the decision trees have been established, we begin to test the ensemble methods on the data. The goal of the ensemble methods is to improve or stay consistent with the accuracy of the decision trees, but reduce the likelihood of overfitting the model.

## Random Forest

The first method used was the random forest classifier. The random forest classifier is the only ensemble method to use a criterion that we can modify to build the tree. Similar to the decision trees, we use the gini impurity and the entropy to build the subtrees. Since the number of predictors is 20, the number of predictors used in each tree will be $\sqrt{20} = 4.47 \approx 4$. Additionally, we run the random tree classifier multiple times limiting its number of subtrees to a different model. Each iteration of the classifier ran with 10 more subtrees than the prior iteration. We start the function with 10 subtrees, and it continues to 140. The goal is to evaluate the trends of the impact of the number of subtrees on the accuracy. This will also help us find the most accurate model for our dataset. Below are the results of our random forest model for gini impurity and entropy.



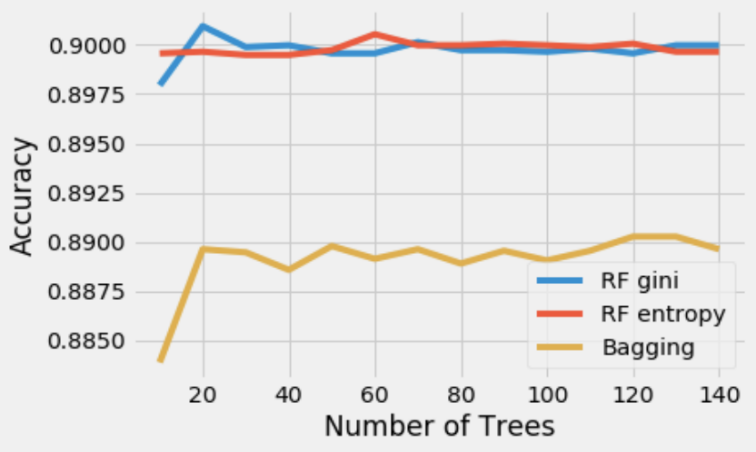

| Number of Trees | RF Accuracy w/ gini | RF Accuracy w/ entropy |
|---|---|---|
| **10** | 0.89795 | 0.89957 |
| **20** | 0.90095 | 0.89965 |
| **30** | 0.89989 | 0.89949 |
| **40** | 0.89998 | 0.89949 |
| **50** | 0.89957 | 0.89973 |
| **60** | 0.89957 | 0.90054 |
| **70** | 0.90014 | 0.89998 |
| **80** | 0.89973 | 0.89998 |
| **90** | 0.89973 | 0.90006 |
| **100** | 0.89965 | 0.89998 |
| **110** | 0.89981 | 0.89989 |
| **120** | 0.89957 | 0.90006 |
| **130** | 0.89998 | 0.89965 |
| **140** | 0.89998 | 0.89965 |

As evident in the graphics above, the results are very similar to the decision trees. The highest accuracy score on random forest model using the gini impurity metric is **.90095**. As for the random forest model using the entropy metric, the accuracy is **.90054**. Again, the gini impurity model scores a higher accuracy than the entropy model. The gini impurity model accuracy remains just a little bit below the accuracy of the gini impurity decision tree metric which was .90118. Of the four models used so far to make predictions, all four are very impressive. There is virtually no difference between the models. The random forest accuracy is slightly smaller than the decision tree accuracy, but

our confidence in the random forest models is much higher due to its smaller likelihood to overfit the model.

## Bagging

The second ensemble method is the bagging method. Similar to the random forest models, we use many iterations of models to classify the target data. In each iteration, the numbers of trees increase by 10 from 10 up to 140. In the results below, notice the difference between the random forest and bagging accuracies. Although the difference may seem very large, we must consider the scale of the graph. More analysis is given below.

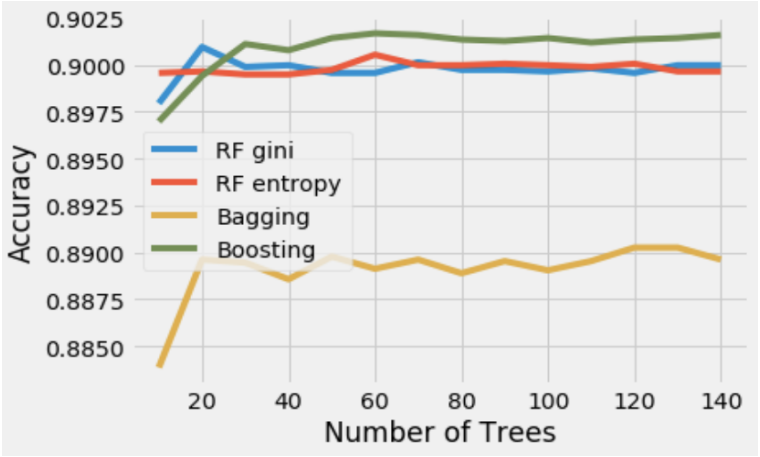


| Number of Trees | RF Accuracy w/ gini | RF Accuracy w/ entropy | Bagging Accuracy |
|---|---|---|---|
| **10** | 0.89795 | 0.89957 | 0.88387 |
| **20** | 0.90095 | 0.89965 | 0.88962 |
| **30** | 0.89989 | 0.89949 | 0.88946 |
| **40** | 0.89998 | 0.89949 | 0.88857 |
| **50** | 0.89957 | 0.89973 | 0.88978 |
| **60** | 0.89957 | 0.90054 | 0.88913 |
| **70** | 0.90014 | 0.89998 | 0.88962 |
| **80** | 0.89973 | 0.89998 | 0.88889 |
| **90** | 0.89973 | 0.90006 | 0.88954 |
| **100** | 0.89965 | 0.89998 | 0.88905 |
| **110** | 0.89981 | 0.89989 | 0.88954 |
| **120** | 0.89957 | 0.90006 | 0.89026 |
| **130** | 0.89998 | 0.89965 | 0.89026 |
| **140** | 0.89998 | 0.89965 | 0.88962 |

The difference between the accuracies seems very large, but their accuracies are only about 1.7 percentage points from each other. The highest accuracy scored for the bagging method is **.89026**. The bagging method was still very successful in determining if a client would subscribe to a term deposit. However, in this case, we would prefer to use the random forest model because of its accuracy being slightly higher.

## Boosting

The third ensemble method is the boosting method. As mentioned earlier, there are many different types of boosting. The boosting method that it is implemented in this section is gradient boosting. This type of boosting builds an additive model in a forward stage-wise fashion. It uses an arbitrary differential loss function to optimize the results. The loss function use in our model is the deviance function. This is similar to a logistic function as it calculates the probability of each observation as the output. Similar to the other ensemble methods, we perform the model multiple times with different numbers of subtrees. Below are the results for the gradient boosting method.

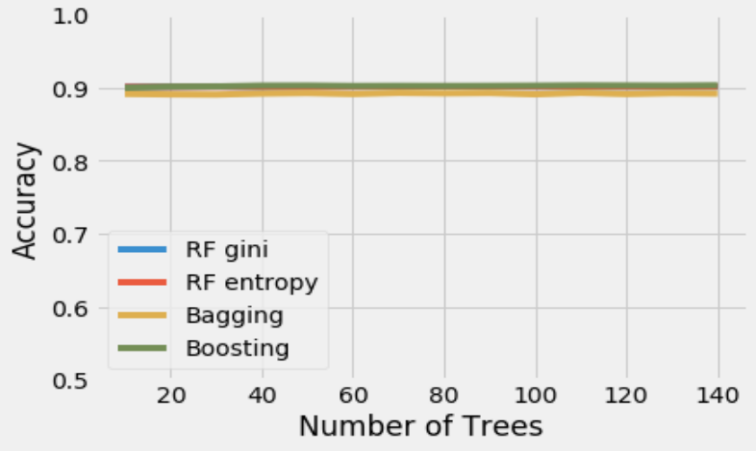



| Number of Trees | RF Accuracy w/ gini | RF Accuracy w/ entropy | Bagging Accuracy | Boosting Accuracy |
|---|---|---|---|---|
| 10 | 0.89795 | 0.89957 | 0.88387 | 0.89698 |
| 20 | 0.90095 | 0.89965 | 0.88962 | 0.89941 |
| 30 | 0.89989 | 0.89949 | 0.88946 | 0.90111 |
| 40 | 0.89998 | 0.89949 | 0.88857 | 0.90078 |
| 50 | 0.89957 | 0.89973 | 0.88978 | 0.90143 |
| 60 | 0.89957 | 0.90054 | 0.88913 | 0.90168 |
| 70 | 0.90014 | 0.89998 | 0.88962 | 0.90159 |
| 80 | 0.89973 | 0.89998 | 0.88889 | 0.90135 |
| 90 | 0.89973 | 0.90006 | 0.88954 | 0.90127 |
| 100 | 0.89965 | 0.89998 | 0.88905 | 0.90143 |
| 110 | 0.89981 | 0.89989 | 0.88954 | 0.90119 |
| 120 | 0.89957 | 0.90006 | 0.89026 | 0.90135 |
| 130 | 0.89998 | 0.89965 | 0.89026 | 0.90143 |
| 140 | 0.89998 | 0.89965 | 0.88962 | 0.90159 |

As we can see in the results above, the boosting method outperforms all other ensemble methods and the original decision trees. The accuracy of the boosting method is **.90168**. This is higher than the highest ensemble method's accuracy (random forest with gini) of .90095 and the highest decision tree's accuracy of .90118. As a result, this appears to be the most effective ensemble method to classify the target variable.

## Final Insights

With the final results provided above, it is clear that the most effective method of predicting if client subscription to a term deposit is the gradient boosting method. Even though the boosting accuracy was higher, random forest and bagging are both viable options that are effective at classifying the target variable. The marginal difference in accuracy between each method is so small that any method is useful for classification. There is virtually no difference between the accuracies of the random forest methods and boosting. The scale of graphs can be deceiving in interpreting the results. Below is a graph scaled to from .5 to 1. It is should be noticeable how similar the results are in actuality when given a scale that is used more often to interpret accuracy results.




As we can see in the graph above, the accuracies are incredibly similar. Therefore, our results culminate to relatively inconclusive. Without verification from other data sets, it is unclear if boosting will always be the dominant ensemble method. While boosting may provide consistent results in our data, they are not marginally impressive from the other methods. Our final insight is that boosting is the most effect ensemble method in this classifying investigation. Furthermore, boosting was the only method to outperform the decision trees. Therefore, we can be confident that boosting is the best option considering its accuracy and small likelihood of overfitting.

# Appendix

| Variable Name | Description | Type | Example |
|---|---|---|---|
| age | client's age | numeric | 30 |
| job | type of job | categorical | blue-collar |
| marital | marital status | categorical | married |
| education | client's education | categorical | high.school |
| default | has credit default? | categorical | yes,no |
| housing | has housing loan? | categorical | yes,no |
| loan | has personal loan? | categorical | yes,no |
| contract | contact communication type | categorical | cellular |
| month | last contact month of year | categorical | feb |
| day_of_week | last contact day of week | categorical | mon |
| campaign | total number of contacts performed | numeric | 3 |
| pdays | number of days since last contact | numeric | 15 |
| previous | number of contacts beforehand | numeric | 2 |
| poutcome | outcome of previous contact | categorical | success |
| emp.var.rate | employment variation rate | numeric | 1.1 |
| cons.price.idx | consumer price index | numeric | 93.994 |
| euribor3m | euribor 3 month rate | numeric | -36.4 |
| nr.employed | number of employees | numeric | 5191 |
| y | did client subscribe to term deposit? | categorical | yes,no |

Table 1: Data set Description