Classifying Homeownership Using Decision Trees

Nitharsan Sivakanthan

4/12/2022

Abstract:

This study uses data from the American Housing Survey to predict whether an individual owns or rents their home. We will use a subset of the hundreds of variables available through the American Housing Survey to classify the housing status variable called TENURE. These variables are considered in random forest and boosting decision tree models. We find that certain variables from the survey contain enough information to perfectly predict TENURE. When these variables are removed, we can achieve accuracy of roughly 93%. We expect to be able to improve these results with more intensive decision tree models.

Problem Statement:

Using data collected from the American Housing Survey, we will determine if an individual owns or rents their home. The data contains hundreds of variables. However, we have reduced the number of variables used in this classification problem. We will use various decision trees to classify an individual's homeownership status. The data contains a variable that classifies each individual on their homeownership called TENURE.

Background:

We will implement two different types of decision tree models, random forest models and gradient boosting models.

Random Forest Models –

The random forest algorithm is an ensemble approach to decision tree modeling. This means the algorithm will create many decision trees and consider each of them to create the best decision tree. The algorithm creates different trees by taking a sample of m of the p possible predictors to develop each tree. This allows the algorithm to search for more possible ensembles compared to other approaches to decision trees that only consider the best predictor at each step.

Boosting Models –

Boosting models slow down the decision-making process to consider information from previous steps and use the error at previous steps to inform the next split on the decision tree model. In effect, the algorithm is learning from its mistakes to improve predictions.

Methodology:

Data-

The data is collected by the U.S. Census Bureau in the American Housing Survey (AHS). It provides current information on the size, composition, and quality of the nation's housing. It includes the conditions of the homes and neighborhoods, the costs of financing and maintaining homes, and the characteristics of the people who live in these homes.

For this study, we use the person and household data that collects information for each individual living at a residence. We narrowed down the variables in the AHS to 21 possible predictors. The TENURE variable classifies an individual by their homeownership status.

Random Forest and Boosting Algorithms –

We used the R package called randomForest to create our random forest models and the R package called gbm to create the boosting models. These libraries allowed us to train the models on our training data for the predictors we were considering. With these trained models, we were then able to test its accuracy on our testing data to determine the model's performance.

To create the random forest model, the function requires us to specify the number of m predictors out of the p possible predictors to sample. We chose to use the square root of possible predictors as m.

For the boosting model, the function requires us to specify the distribution, the number of trees, interaction depth, and shrinkage. In our case, the distribution is multinomial because the TENURE variable we are predicting has three possible classes to predict. For different models we use different number of trees, interaction depth and shrinkage to obtain the best results. The interaction depth refers to the number of predictors used at each step of the process. The shrinkage parameter is related to the speed at which the boosting model reduces error.

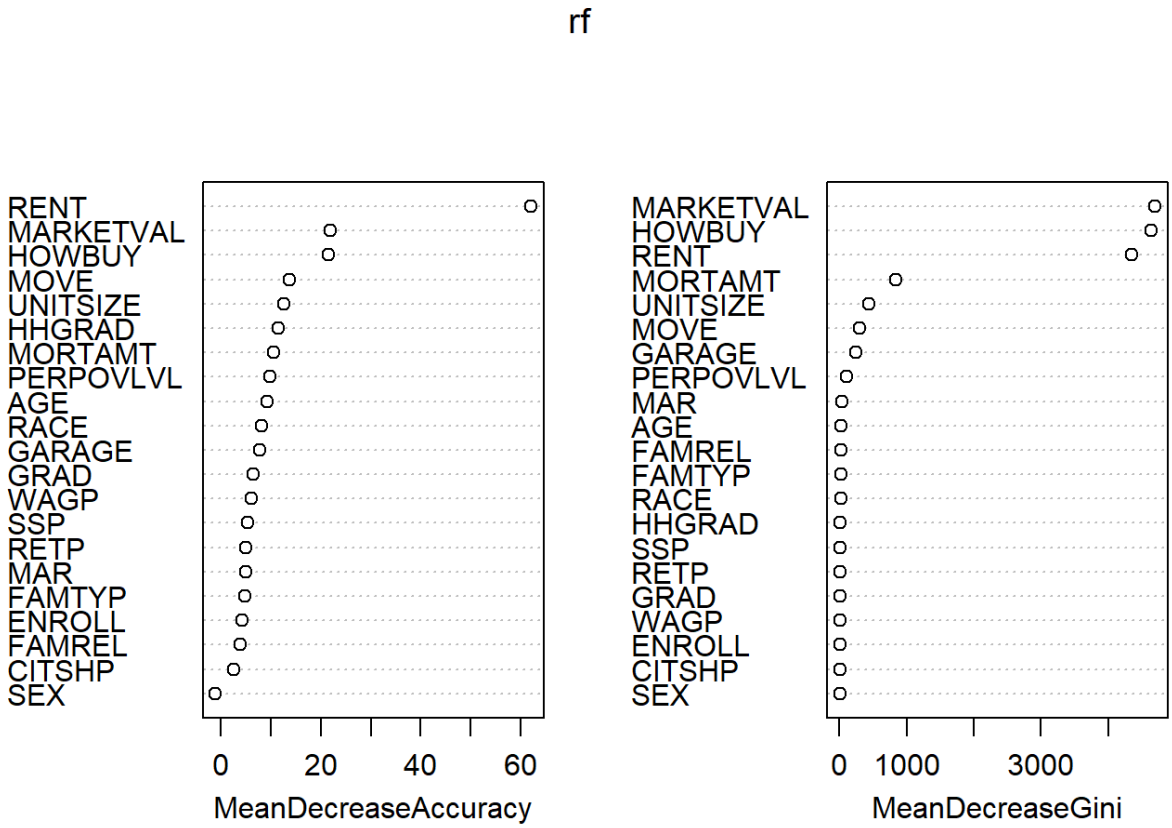
Results:

First, we use a random forest model to predict TENURE using the subset of predictors we chose. Here is a confusion matrix of the results of the trained model on the testing data:

```
## yhat   '1'  '2'  '3'
##   '1' 6333    0    0
##   '2'    0 2764    0
##   '3'    0    0   81
```

This model predicted TENURE without any errors.

Here is a plot showing the importance of predictors in this model:



From this plot, we can see there are a few predictors that highly influence the results of the model: RENT, MARKETVAL, and HOWBUY.

With further inspection of the data, the manner in which these three variables were created provides almost all the information needed to determine a person's tenure. Therefore, if this information is accessible with the goal of determining TENURE, use it. However, we will now shift to predicting TENURE assuming this information is not available.

After removing the three predictors RENT, MARKETVAL, and HOWBUY from consideration, we trained another random forest model. Here is a confusion matrix of the results of the trained model on the testing data to predict TENURE:

```
## yhat   '1'  '2'  '3'
##   '1' 5972  203   32
##   '2'  361 2561   46
##   '3'    0    0    3
```

This model had an accuracy of approximately 93%. Most notably, the model has trouble predicting the third class in the TENURE variable.

Now, we train a boosting model using the same predictors to predict TENURE. Here is a confusion matrix of the results of the trained model on the testing data:

```
## yhat  '1'  '2'  '3'
##    1 5899  245   31
##    2  434 2519   50
```

This time, the model did not predict anyone to be in class 3 and this model did worse than the random forest model in predicting class 1 & 2. We assume this is because of the parameters we chose for the boosting model. We need more trees to reduce the error to get a better model.

We now train the boosting model with more trees. Many more trees. With 10,000 trees, the boosting model starts to classify one person to class 3, however it is incorrect. An attempt was made with 100,000 trees, but unfortunately, we did not have enough computing power to obtain the results before R was forced to close.

We were able to train a better boosting model by changing the depth to 3 and increasing the shrinkage parameter to .1. By increasing the shrinkage parameter, we will require less trees to obtain lower error. Here is a confusion matrix of the results of the trained model on the testing data using 5000 trees:

```
yhat  '1'  '2'  '3'
   1 5999  226   30
   2  330 2533   34
   3    4    5   17
```

This model improves the accuracy to 93.15%, most notably this model does the best at predicting class 1 & 3 at the expense of doing slightly worse at predicting class 2.

Conclusion:

Using all the variables available to us from the AHS, we can predict TENURE with perfect accuracy. However, this is because of the way certain variables in the survey were created. If we do not consider those variables, we are able to obtain a random forest model with roughly 93% accuracy and a boosting model with roughly 93.15% accuracy of predicting TENURE. A concern with the current best model is in predicting class 3 of TENURE. Using boosting, we can theoretically achieve better results with the proper computing requirements.

It is important to be cautious with choosing variables from the AHS. In addition to the three predictors we found to perfectly predict TENURE, there may be more variables we did not consider that can achieve the same results.

Appendix:

```
#import libraries
library(tidyverse)
library(tree)
library(randomForest)
library(gbm)

#read in the data
person = read.csv('person.csv', header = TRUE)
household = read.csv('household.csv', header = TRUE)
mortgage = read.csv('mortgage.csv', header = TRUE)
project = read.csv('project.csv', header = TRUE)

#join tables
data = person %>%
  left_join(household, by = 'CONTROL')

#choose columns
data = data %>%
  select(PERSONID,AGE, MAR, SEX, MOVE, CITSHP,FAMTYP,FAMREL, RACE, GRAD,
HHGRAD,MORTAMT,WAGP,SSP,RETP,ENROLL, PERPOVLVL,RENT, GARAGE, HOWBUY,UNITSIZE,
MARKETVAL, TENURE)
data = data %>%
  transform(TENURE = as.factor(TENURE), MAR = as.factor(MAR), SEX = as.factor(SEX), CITSHP =
as.factor(CITSHP), FAMTYP = as.factor(FAMTYP), FAMREL = as.factor(FAMREL), RACE = as.factor(RACE),
GRAD = as.factor(GRAD), HHGRAD = as.factor(HHGRAD), ENROLL = as.factor(ENROLL), GARAGE =
as.factor(GARAGE), HOWBUY = as.factor(HOWBUY), UNITSIZE = as.factor(UNITSIZE))

#create train/test split
set.seed(1)
```

```
index = sample(1:nrow(data), nrow(data)*.8)
train = data[index,]
test = data[-index,]
#number of predictors; do not use PERSONID or TENURE as predictors
npredictors = length(data)-2

#random forest decision tree model
rf = randomForest(TENURE ~ .-PERSONID,
                  data= train,
                  mtry=sqrt(npredictors),
                  importance=TRUE)
rf
yhat = predict(rf, newdata = test, type = 'response')
table(yhat, test$TENURE)
mean(yhat == test$TENURE)
importance(rf)
varImpPlot(rf)

#decision tree model with boosting
boost = gbm(TENURE ~ .-PERSONID, data = train, distribution = "multinomial",
            n.trees = 500, interaction.depth = 2, shrinkage = 0.01)

yhat = predict(boost, newdata = test, n.trees = 100, type = 'response')
yhat = apply(yhat, 1, which.max)
table(yhat, test$TENURE)
summary(boost)
```