The Case for Decision Trees

CUNY 622 - Assignment 2

Eric Hirsch

10/23/2022

Contents

Introduction

Random Forest and Decision Trees are both non-parametric machine learning algorithms for predicting target variables from a set of independent variables. They may be used for classification or regression. Decision trees work by splitting a source set into subsets, which may be further split depending on the data. Random Forest is an ensemble learning algorithm which constructs a multitude of trees and takes the majority (in classification) or average (in regression) to make its prediction. By pooling the information from multiple trees, Random Forests compensate for the tendency of Decision Trees to overfit the data. For this reason, Random Forests generally significantly outperform single trees in terms of prediction.

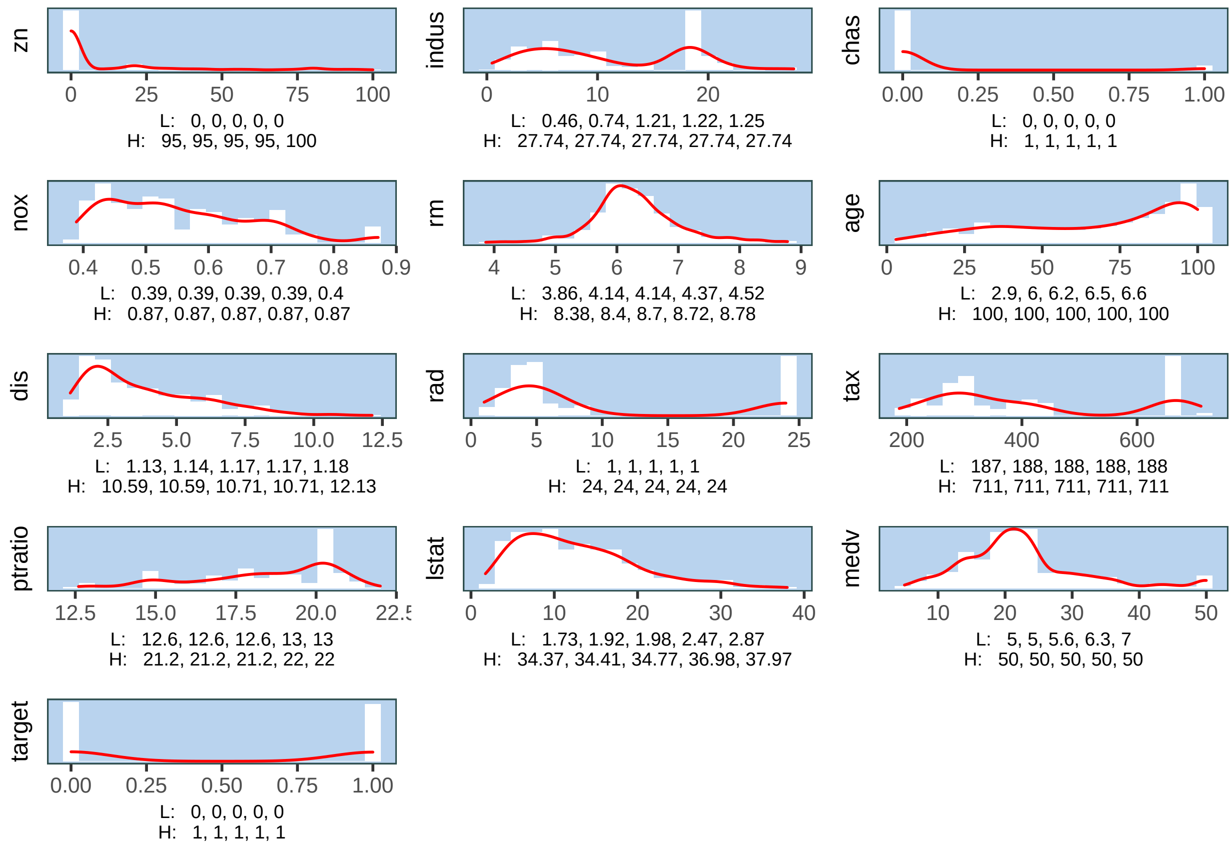
However, the evaluation of algorithm performance can only be made in the context of a use case. For many use cases, particularly those where time and other resources are scarce and/or interpretability is more important, Decision Trees will be the better choice. We illustrate this with a dataset of data related to suburbs in the Boston area.

The data set consists of 466 observations with 11 numeric variables and two binary variables. There are no missing values. The variables include the level of industrialization, average tax rates, pollution levels, and so on. This data set is often used to predict crime rates, but we won't use it for that purpose. A summary appears below:

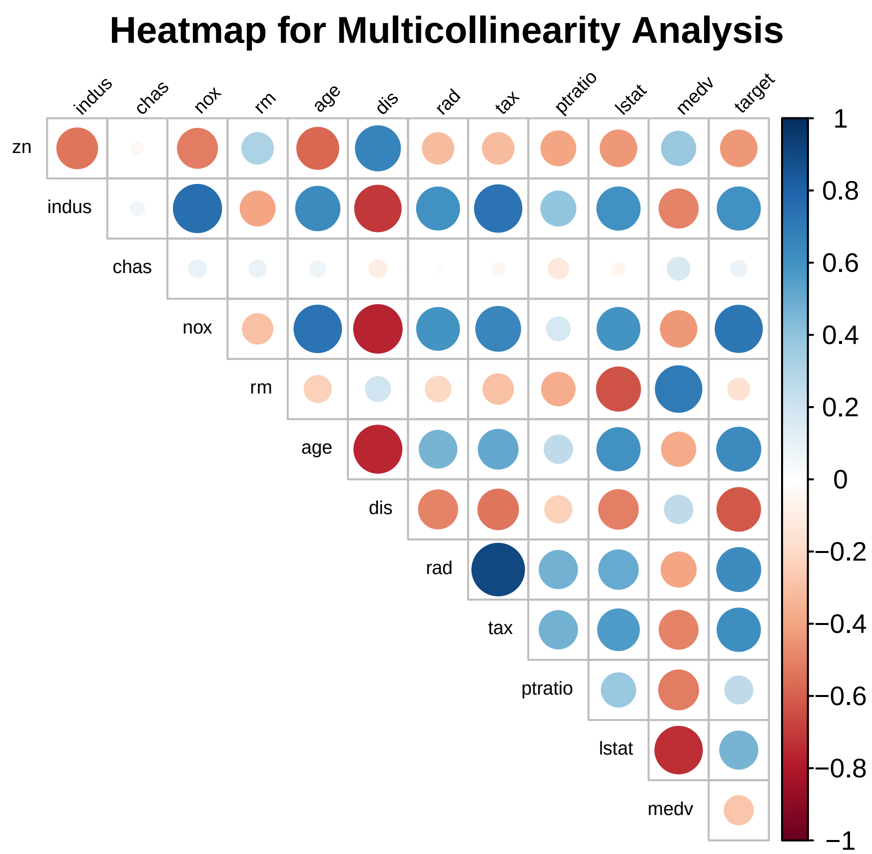
```
##        zn             indus            chas              nox
##  Min.   :  0.00   Min.   : 0.460   Min.   :0.00000   Min.   :0.3890
##  1st Qu.:  0.00   1st Qu.: 5.145   1st Qu.:0.00000   1st Qu.:0.4480
##  Median :  0.00   Median : 9.690   Median :0.00000   Median :0.5380
##  Mean   : 11.58   Mean   :11.105   Mean   :0.07082   Mean   :0.5543
```

```
##  3rd Qu.: 16.25   3rd Qu.:18.100   3rd Qu.:0.00000   3rd Qu.:0.6240
##  Max.   :100.00   Max.   :27.740   Max.   :1.00000   Max.   :0.8710
##        rm             age              dis              rad
##  Min.   :3.863   Min.   :  2.90   Min.   : 1.130   Min.   : 1.00
##  1st Qu.:5.887   1st Qu.: 43.88   1st Qu.: 2.101   1st Qu.: 4.00
##  Median :6.210   Median : 77.15   Median : 3.191   Median : 5.00
##  Mean   :6.291   Mean   : 68.37   Mean   : 3.796   Mean   : 9.53
##  3rd Qu.:6.630   3rd Qu.: 94.10   3rd Qu.: 5.215   3rd Qu.:24.00
##  Max.   :8.780   Max.   :100.00   Max.   :12.127   Max.   :24.00
##       tax           ptratio         lstat             medv
##  Min.   :187.0   Min.   :12.6   Min.   : 1.730   Min.   : 5.00
##  1st Qu.:281.0   1st Qu.:16.9   1st Qu.: 7.043   1st Qu.:17.02
##  Median :334.5   Median :18.9   Median :11.350   Median :21.20
##  Mean   :409.5   Mean   :18.4   Mean   :12.631   Mean   :22.59
##  3rd Qu.:666.0   3rd Qu.:20.2   3rd Qu.:16.930   3rd Qu.:25.00
##  Max.   :711.0   Max.   :22.0   Max.   :37.970   Max.   :50.00
##      target
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.4914
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

When we examine histograms we see that a number of variables have distributions that are broken and uneven (zn, indus, nox and rad), suggesting possible hidden groupings. This may lend itself well to decision tree/random forest algorithms. Many of the distributions are also skewed.



There is also a great deal of multicollinearity. The highest correlation (over 90%) is between rad and tax. We will drop the tax rate to avoid problems with interpretation later on.



When Random Forest Works Better

The table below illustrates how much more effective random forests are than decision trees in making predictions. Random forests and decision tree models were constructed for seven selected variables in dataset, predictions were made on an evaluation set, and RMSEs were calculated. The random forest models were superior at predicting in every case. The package used here is the CARET.

```
##          Decision Tree-RMSE Random Forest-RMSE
## zn                    13.60               5.70
## indus                  3.80               1.10
## nox                    0.07               0.03
## lstat                  4.80               3.40
## tax                   62.30              27.10
## ptratio                1.50               0.75
## medv                   6.00               3.20
```

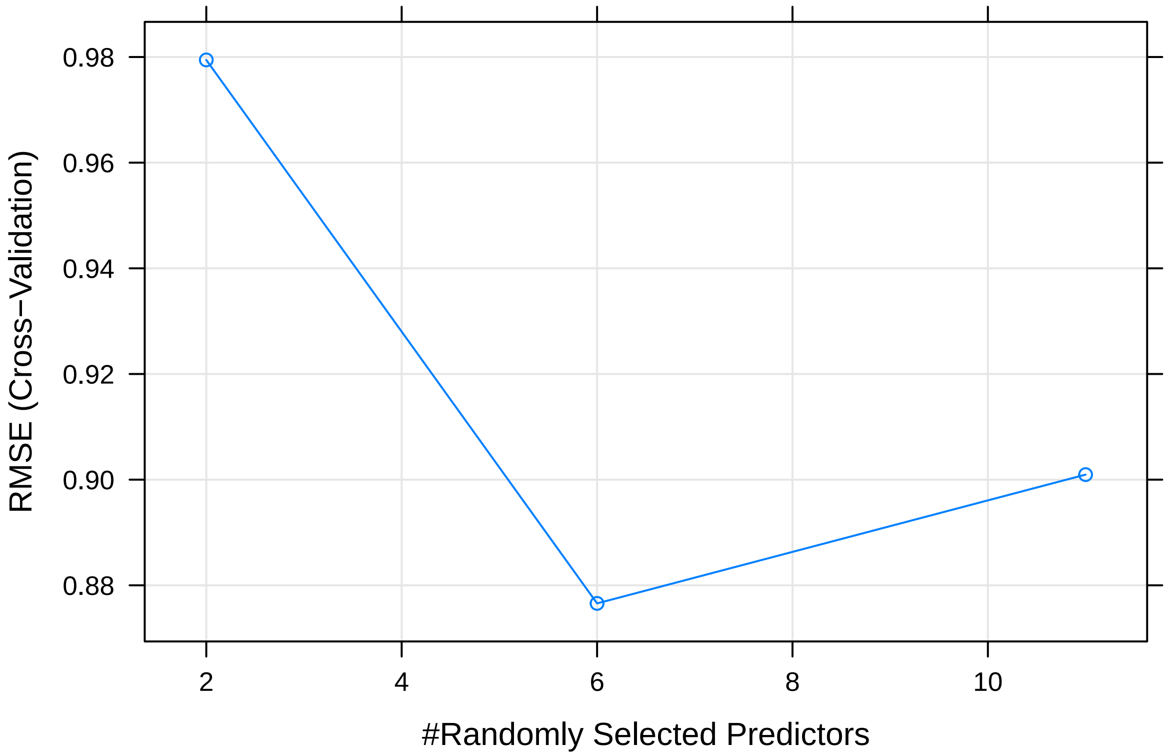
Now imagine you are a data scientist working for the Department of Education tasked with predicting the pupil-student ratio in various suburbs where the information is not readily available. Hundreds of thousands of tax dollars to support underserved students depend on the calculation so you need to be as accurate as possible. Your department has the time and resources to apply whatever model you create to any new data you receive. Given the table above, random forest is the best choice as it outperforms a single decision tree when predicting ptratio (RMSE = 7.5 vs. RMSE = 1.5 for the decision tree)..

When creating a random forest algorithm in R, there are a number of parameters we can tune. The most important of these are mtry (the number of variables drawn randomly for each split), ntree (the number of trees to grow) and maxnode (the maximum amount of terminal nodes in the forest). While the caret package automatically optimizes parameters for random forest, the parameters can also be tuned manually. However, manual tuning of the parameters did not result in a lower RMSE in the evaluation set than out of the box tuning.

Below is the result of a Random Forest analysis of pupil-student ratio using ten fold cross validation:

```
## Random Forest
##
## 374 samples
##  11 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 336, 337, 336, 337, 337, 337, ...
## Resampling results across tuning parameters:
##
##   mtry  RMSE       Rsquared   MAE
##    2    0.9794509  0.8272820  0.6900778
##    6    0.8765771  0.8512309  0.5333115
##   11    0.9009539  0.8390698  0.5279173
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mtry = 6.
## [1] "Random Forest - RMSE on evaluation set:  0.750961639827483"
```

The analysis chooses the model with the lowest RMSE. We can see that the lowest RMSE was .88 at an mtry of 6. The plot below shows how mtry was minimized at 6 and began to climb thereafter. Interestingly, when the model was applied to the evaluation set, the RMSE was lower at .75. Assuming that the errors are normally distributed, an RMSE of .75 suggests our predictions will be within 1.5 of the actual value 95% of the time. Since the range for ptratio is 12.6 to 22 and the mean is 18.4, this is quite reasonable.



Below we see the variable importance table. Levels of industrialization and air quality are the two most important factors determining the predicted ptratio. In general, indications of poverty – pollution, factories, low housing prices – all appear to influence the ptratio. However, for the purposes of our prediction algorithm, it may not matter what influences the ptratio, as long as we can predict it.

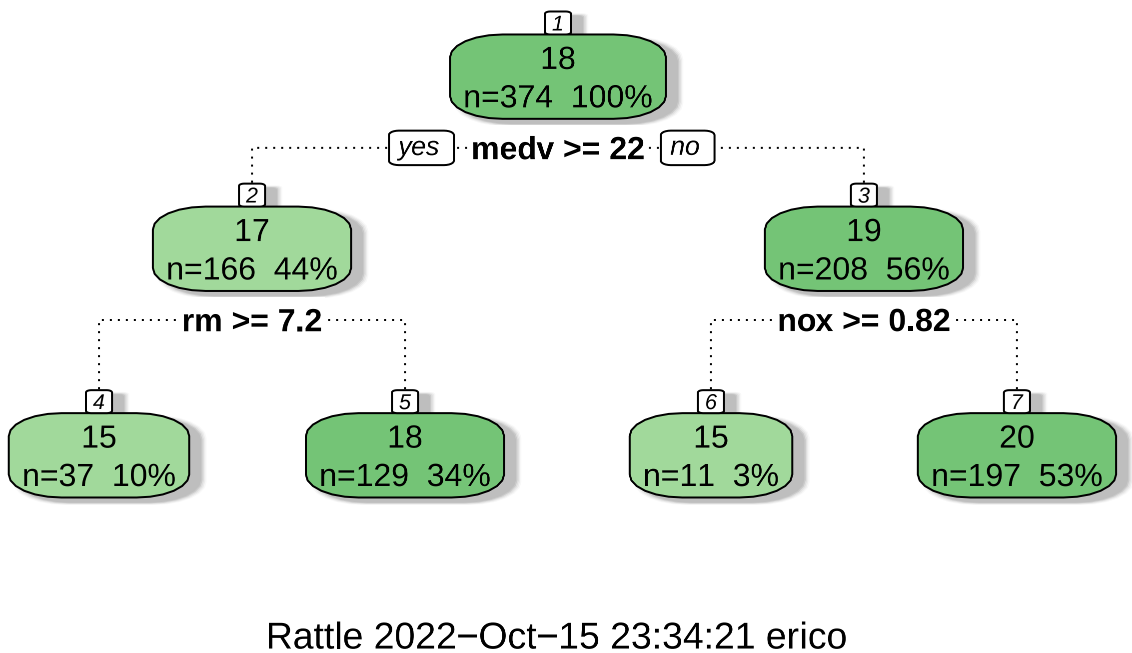
```
## rf variable importance
##
##        Overall
## nox    100.000
## indus   84.217
## medv    62.311
## rad     53.745
## zn      38.555
## dis     37.063
## rm      16.453
## lstat   12.132
## age     11.045
## target   6.236
## chas     0.000
```

The Case for Decision Trees

Sometimes when we leave our house in the morning it's foggy and gray, and so we take our umbrella just in case. We could run a random forest algorithm over all of the relevant variables and improve our prediction of whether it is going to rain, but this normally wouldn't be appropriate for this situation.

Likewise, imagine that in addition to working for the Department of Education, you also volunteer in support of a nonprofit tutoring program. The program wants to strategically offer tutoring services in suburbs where pupil-teacher ratios are high. Since pupil-teacher ratios are not readily available, they ask you for a simple rule-of-thumb to predict them based on information that is readily available, like the age of owner-occupied units, housing prices and pollution levels. They wouldn't have the resources to implement a random forest algorithm, and really don't need to – they just need to make some good, educated guesses about where best to deploy their staff.

Assuming the data in this dataset is generalizable, you can offer such rules easily with a decision tree. Consider the decision tree below, which looks at pupil-teacher ratio against the other variables in the dataset:



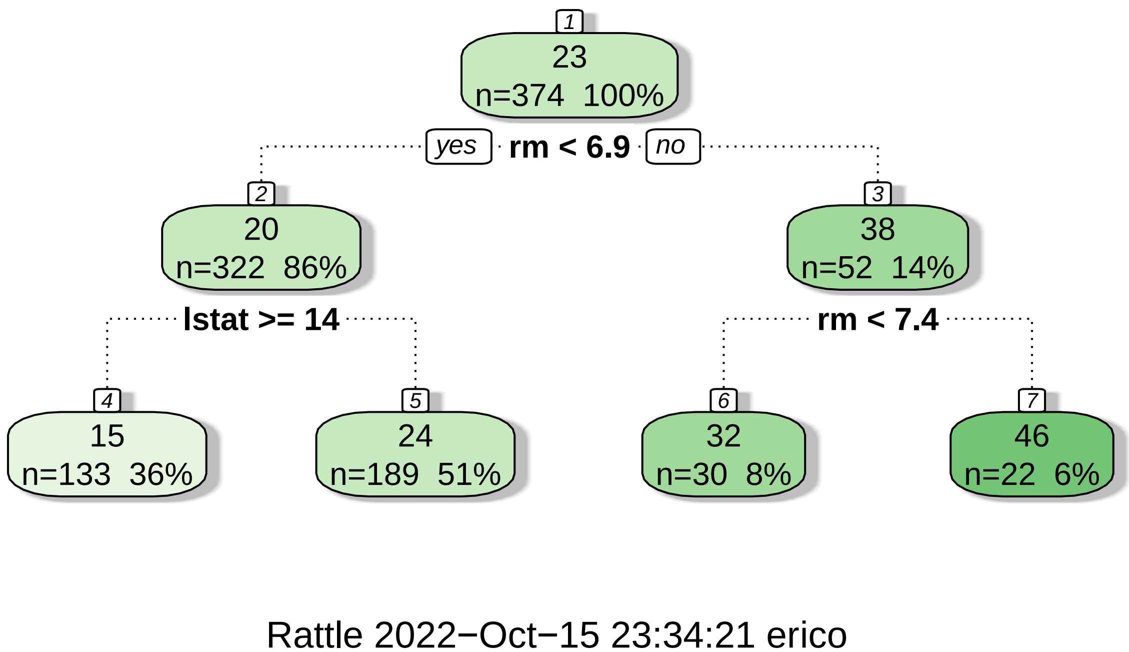
This decision tree tells us that median home price is a reliable indicator of the pupil-teacher ratio. Moreover, it tells us where the split is ($22,000). After that, we can use air-quality and number of rooms per house to better determine where tutors might be needed. This tree can become a handy rubric for helping the nonprofit determine where to put its resources when certain information isn't available.

If the decision tree is likely to overfit the data, how do we know this particular tree is appropriate? First, the tree had an RMSE of 1.5. The mean pupil-teacher ratio is 18.4. While some predictions will be incorrect, the majority of the time the rubric will do a perfectly good job of at least distinguishing between high pupil-teacher ratios and lower ones.

Second, the VIF factors for the random forest result also report median home price, pollution levels and average number of rooms as important factors in determining pupil-teacher ratios (see above). However, they are in a different order and include other factors as well. It is possible if we removed more of the multicollinear variables we would have more consistency between individual decision trees and the random forest analysis.

Now we are given a second request. Tutors deployed to areas in need may not necessarily want to live in those areas, as the areas are likely to be economically disadvantaged with few services. How might they be directed to suburbs with home prices that meet their modest tutor budget but are not in the most impoverished areas?

The decision tree below suggests a simple rubric based primarily on the average number of rooms in the house – suburbs where houses tend to have more rooms are going to have a more expensive housing stock. While this makes intuitive sense without needing a decision tree analysis, it is handy to have a simple formula – the neighborhoods that are affordable have less than seven rooms per house, and the neighborhoods that are not in the most impoverished areas have a lower status index less than 14. These are the neighborhoods for the tutors to begin their search.



In this case, the VIF factors agree with the decision tree. The RMSE is 6.0. Because medv has a mean of 22 and ranges from 5 to 50, this rubric might be thought of more as a rule of thumb – it is a good starting place to avoid the most expensive and most impoverished suburbs.

Conclusion

In short, random forest and decision tree algorithms both have their uses. Always reflexively choosing an algorithm because it predicts best is akin to always choosing a Ferrari over a rickety school bus because it goes faster. It's fine until you have to transport 150 11-year-olds to the local zoo. Algorithms don't stand on their own but are used to solve problems, and the nature of the solution needs to match the nature of the problem.