Plan:
1. Work through an ML example
2. Introduce overfitting

Machine Learning: Example

Shannon E. Ellis, Ph.D
UC San Diego

Department of Cognitive Science
sellis@ucsd.edu

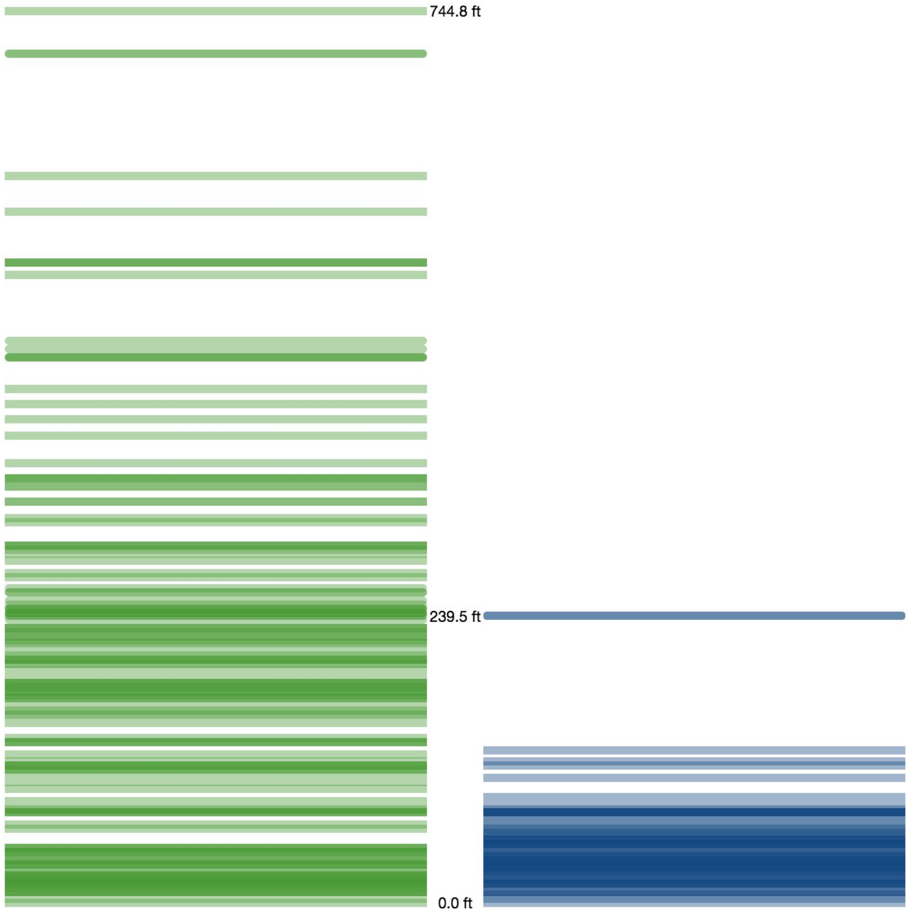
Can we build a model that distinguishes a house in NY from a house in San Francisco?

First, some intuition

Let's say you had to determine whether a home is in **San Francisco** or in **New York**. In machine learning terms, categorizing data points is a **classification** task.

San Fran is hilly …so elevation may be a helpful feature.

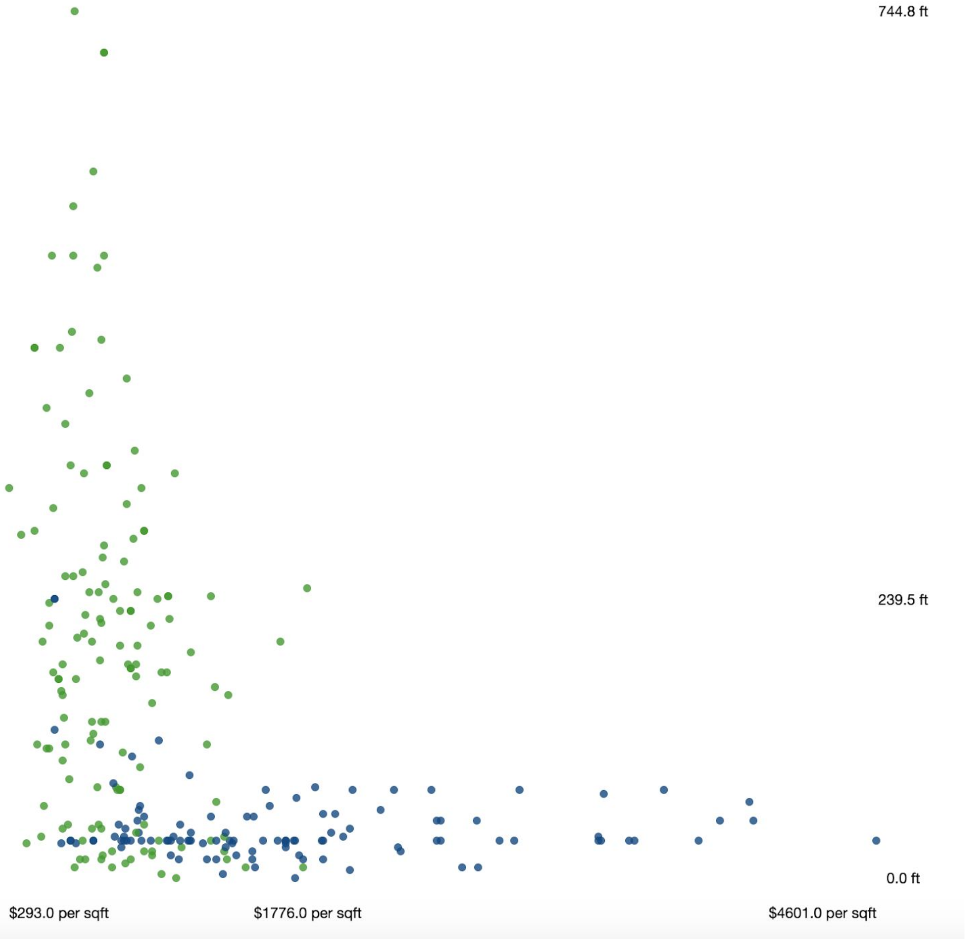
With the data here, homes > ~73m should be classified as San Fran homes



http://www.r2d3.us/visual-intro-to-machine-learning-part-1/?from=@

Adding nuance

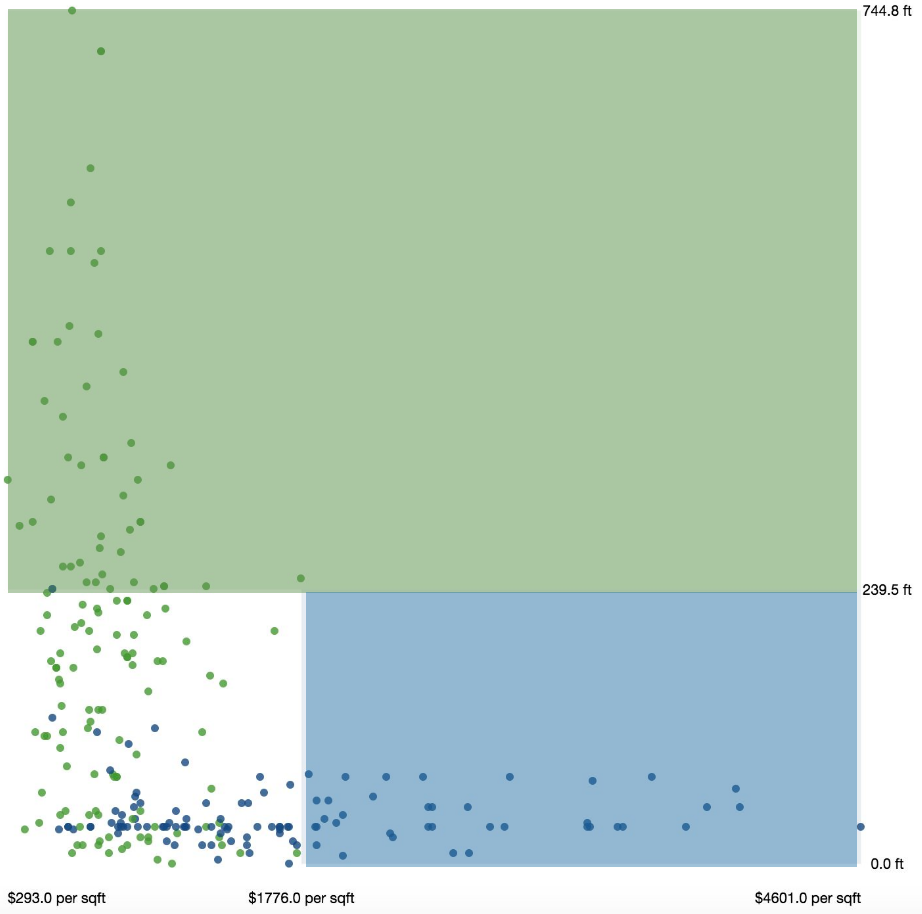
Elevation isn't a perfect feature for classification, so we can look at its relationship to other features, like *price per square foot*



http://www.r2d3.us/visual-intro-to-machine-learning-part-1/?from=@

Drawing boundaries

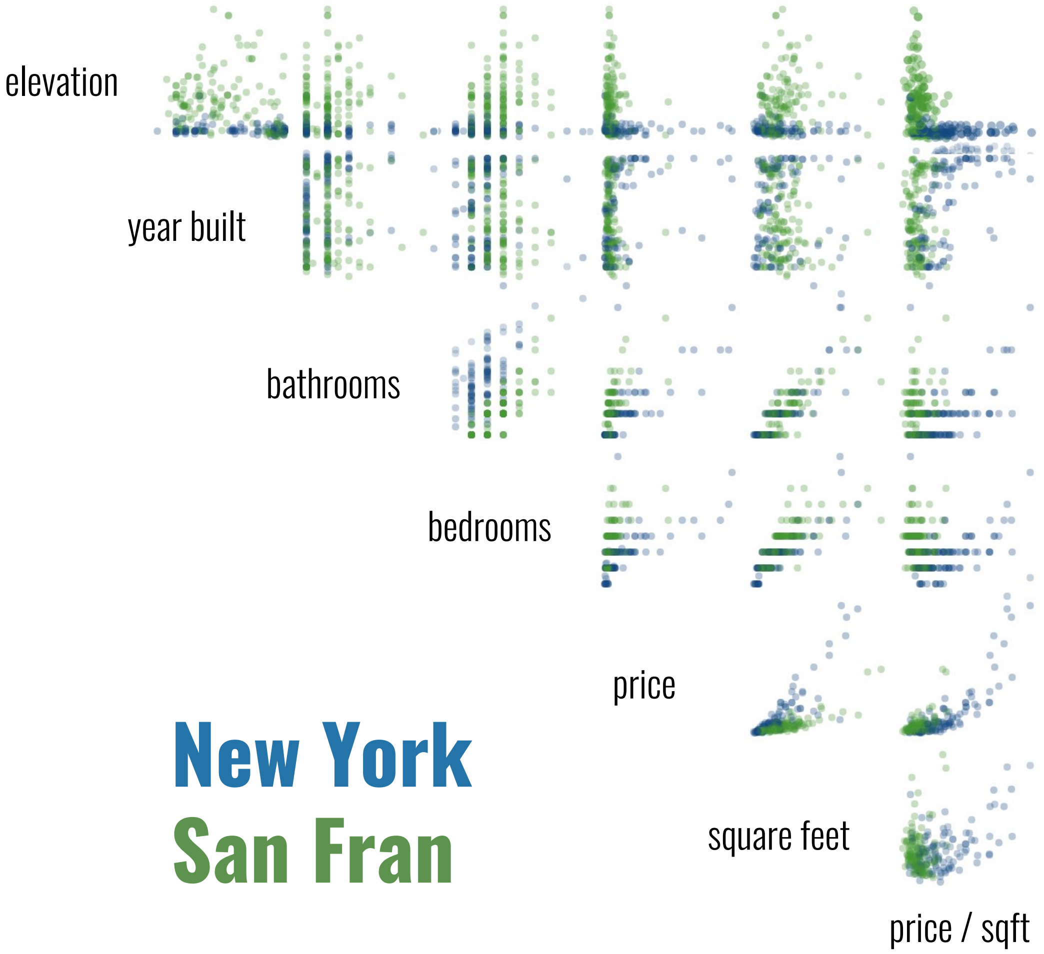
Boundaries can be drawn so that if a house falls in the green box, it's classified as a San Fran home. Blue box, New York. Statistical learning figures out how to best draw these boxes.



http://www.r2d3.us/visual-intro-to-machine-learning-part-1/?from=@

Our training set will use 7 different **features**. At the right we see the **scatterplot matrix** of the relationship between these features.

Patterns are clear, but boundaries for delineation are not obvious.



http://www.r2d3.us/visual-intro-to-machine-learning-part-1/?from=@

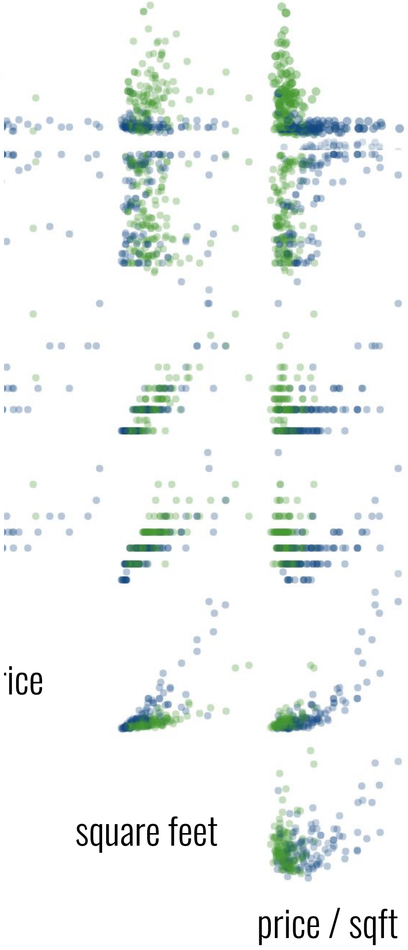
Our training set will u
features. At the righ
scatterplot matrix
relationship between

Patterns are clear, bu
delineation are not ob

And now, machine learning

Determining the best boundary is where **machine learning** comes in.

Decision trees are one example of machine learning method for classification tasks.



http://www.r2d3.us/visual-intro-to-machine-learning-part-1/?from=@

Finding better boundaries

We guessed ~73m before. Let’s improve on that guess…

http://www.r2d3.us/visual-intro-to-machine-learning-part-1/?from=@

A **histogram** helps display frequency of homes by elevation more easily.

73m is the highest home in New York, but most of them have lower elevations

http://www.r2d3.us/visual-intro-to-machine-learning-part-1/?from=@

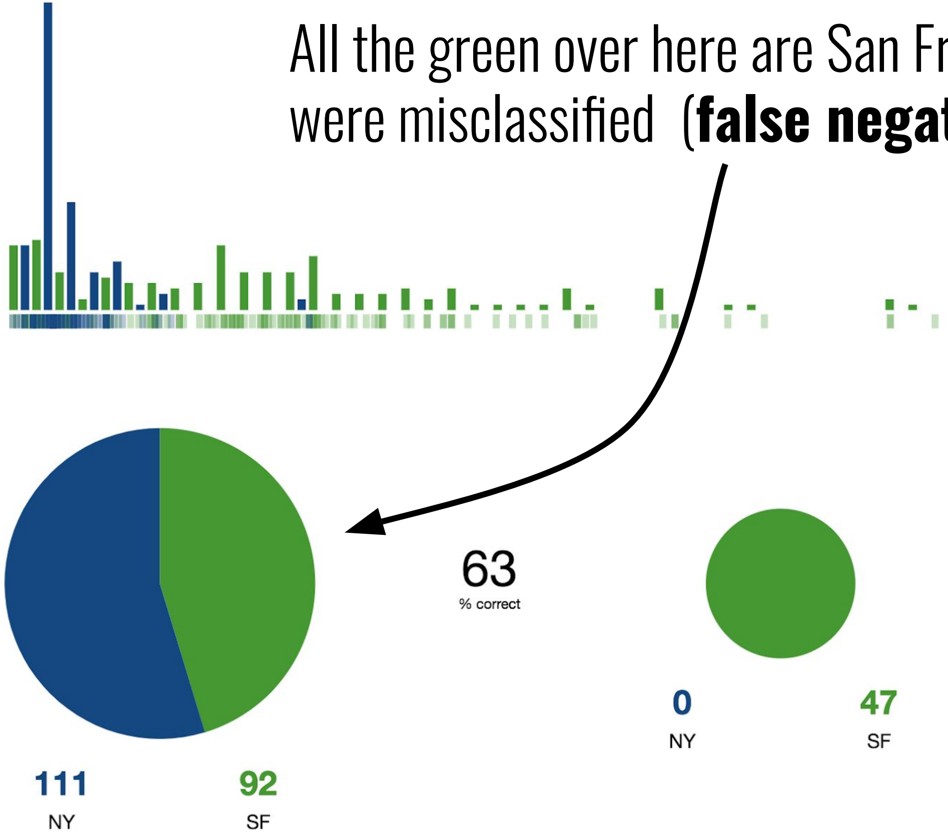
Your first fork

A decision tree uses if-then statements to define patterns in the data.

In machine learning, the splits are called **forks** and they split the data into **branches** based on some value.

The value that splits the branches is the **split point.** Homes to the left get categorized differently than those on the right.

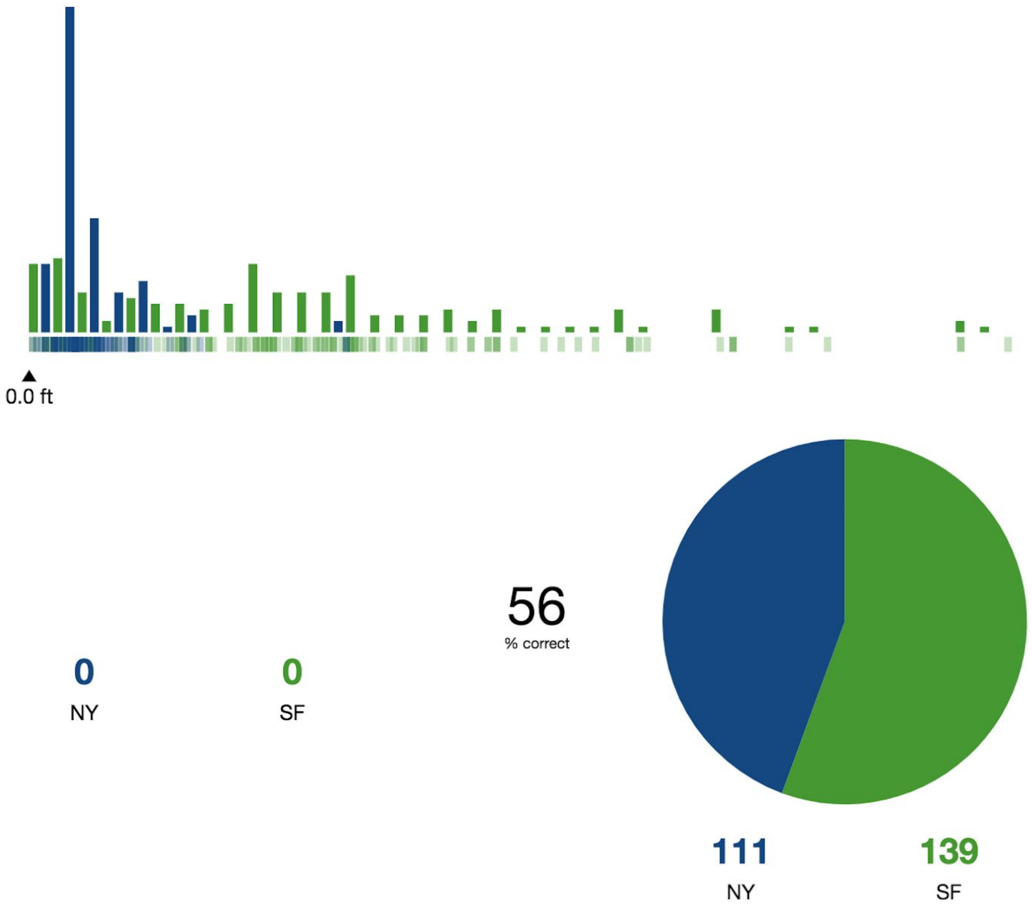
All the green over here are San Fran homes that were misclassified (**false negatives**)



Tradeoffs

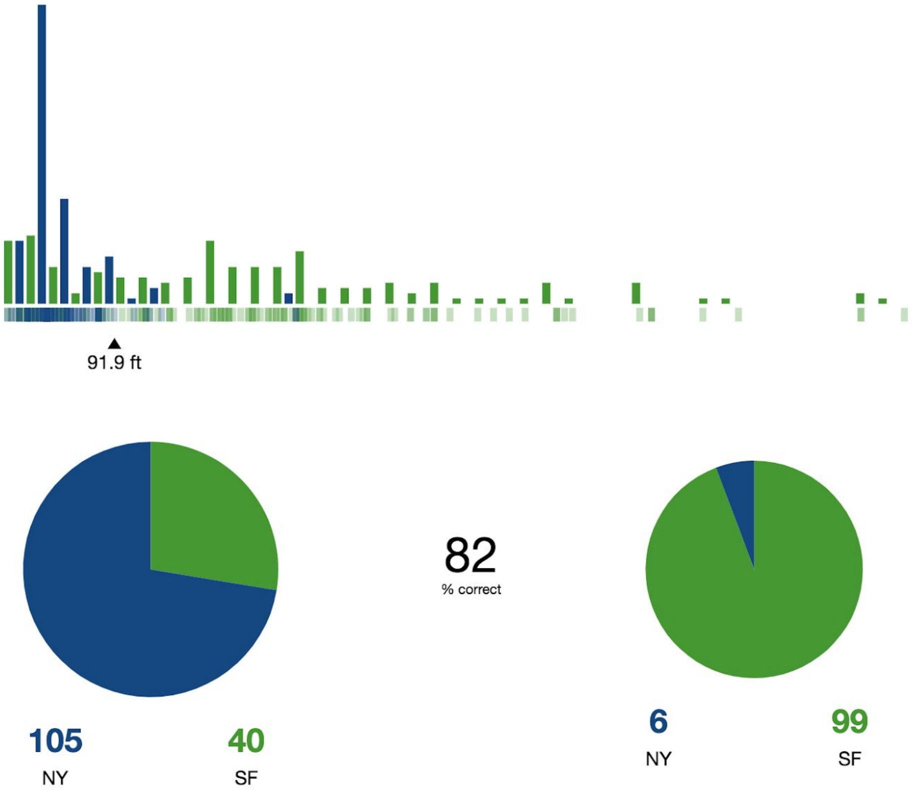
Splitting at ~73m incorrectly classifies some San Francisco homes as New York homes.

http://www.r2d3.us/visual-intro-to-machine-learning-part-1/?from=@



If you split to capture *every* home in San Fran, you'll also get a bunch of New York homes (**false positives**)

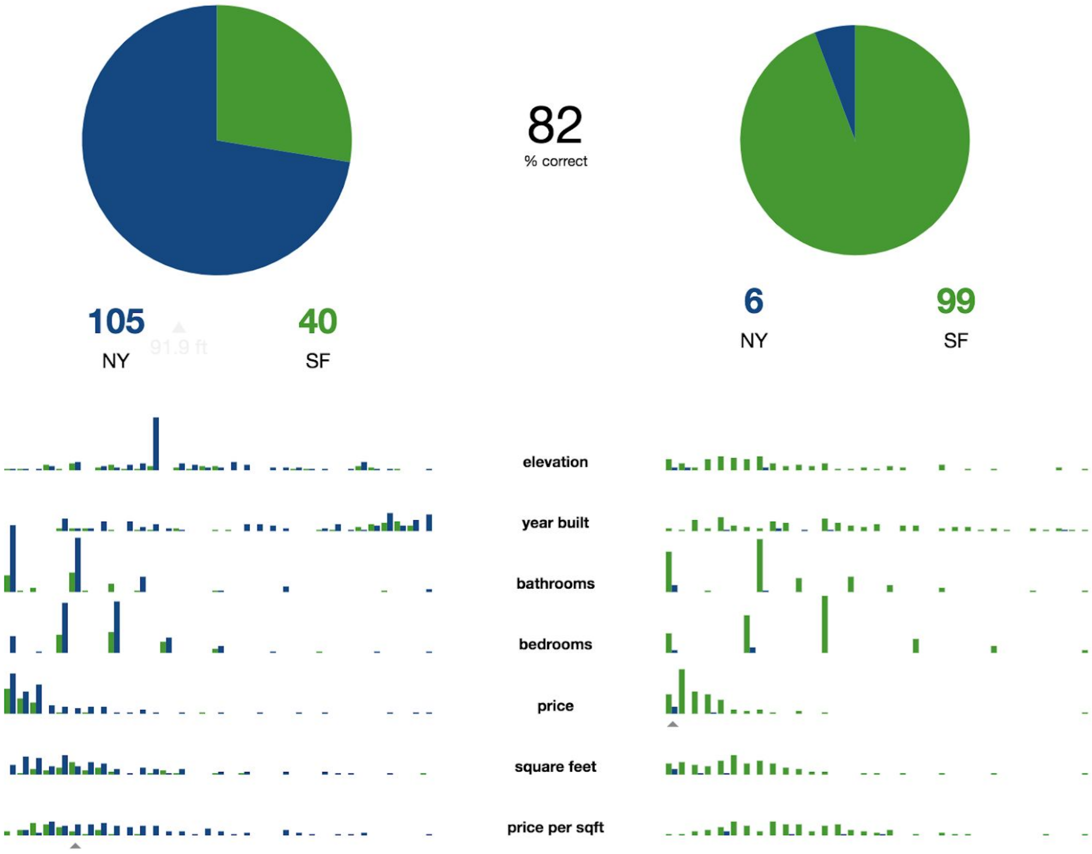
http://www.r2d3.us/visual-intro-to-machine-learning-part-1/?from=@



The best split

The best split point aims for branches that are as homogenous (pure) as possible

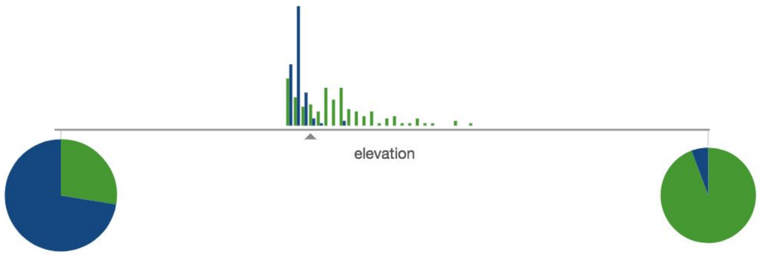
http://www.r2d3.us/visual-intro-to-machine-learning-part-1/?from=@



Recursion

Additional split points are determined through repetition (**recursion**)

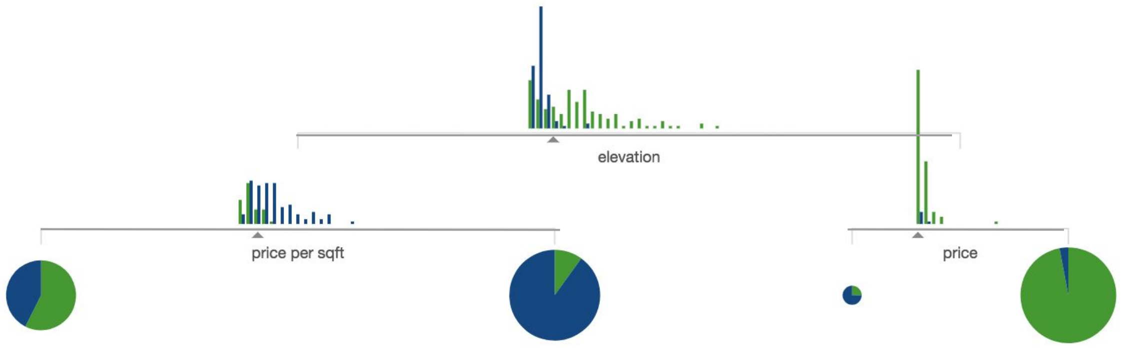
http://www.r2d3.us/visual-intro-to-machine-learning-part-1/?from=@



Growing a tree

Additional forks add new information to improve **prediction accuracy**.

Accuracy: 82%



Growing a tree

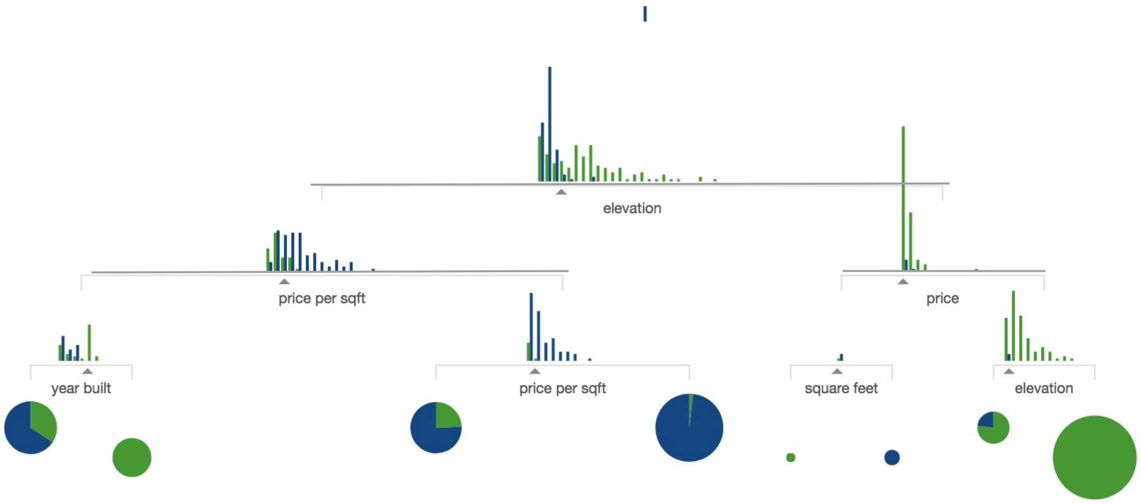
Additional forks add new information to improve **prediction accuracy**.

Accuracy: 86%

http://www.r2d3.us/visual-intro-to-machine-learning-part-1/?from=@

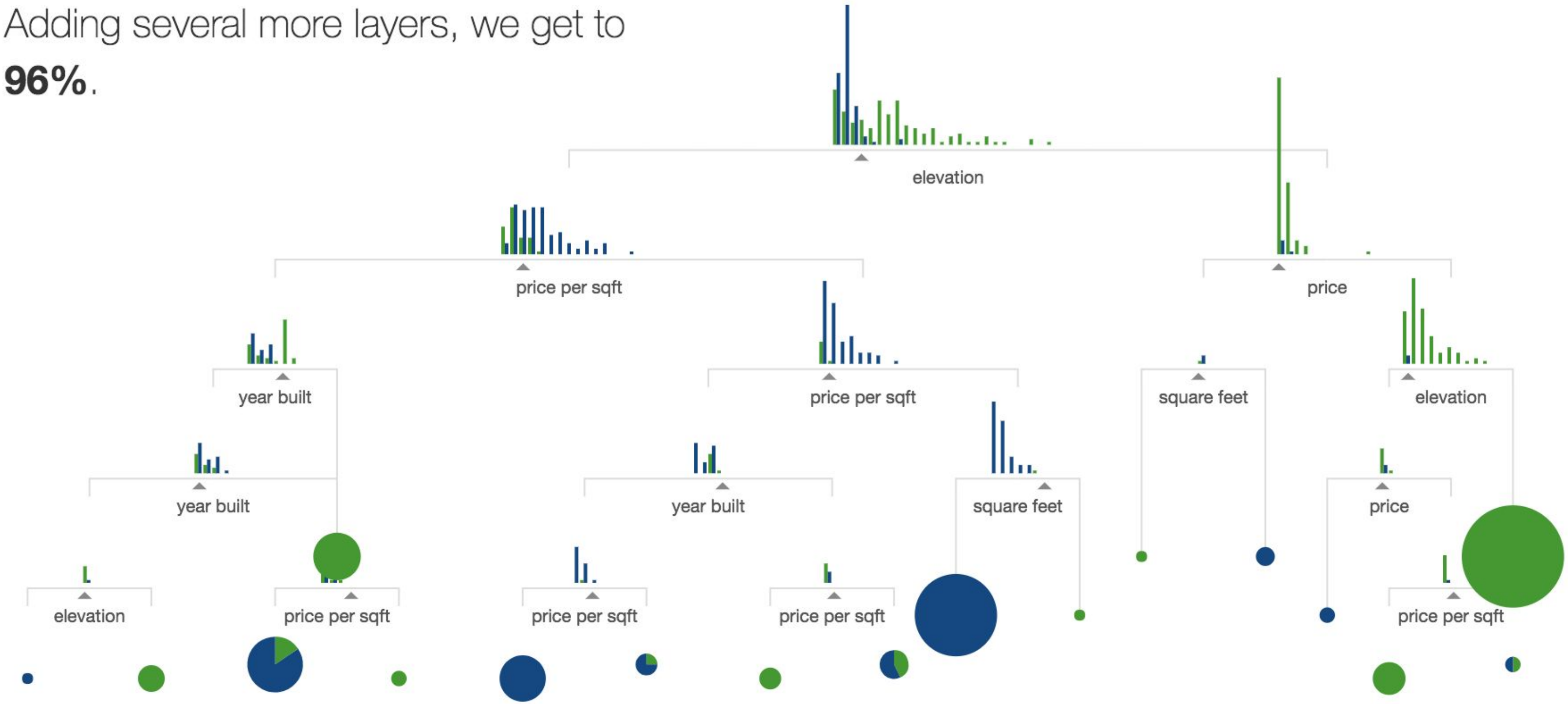
Growing a tree

Additional forks add new information to improve **prediction accuracy**.



http://www.r2d3.us/visual-intro-to-machine-learning-part-1/?from=@

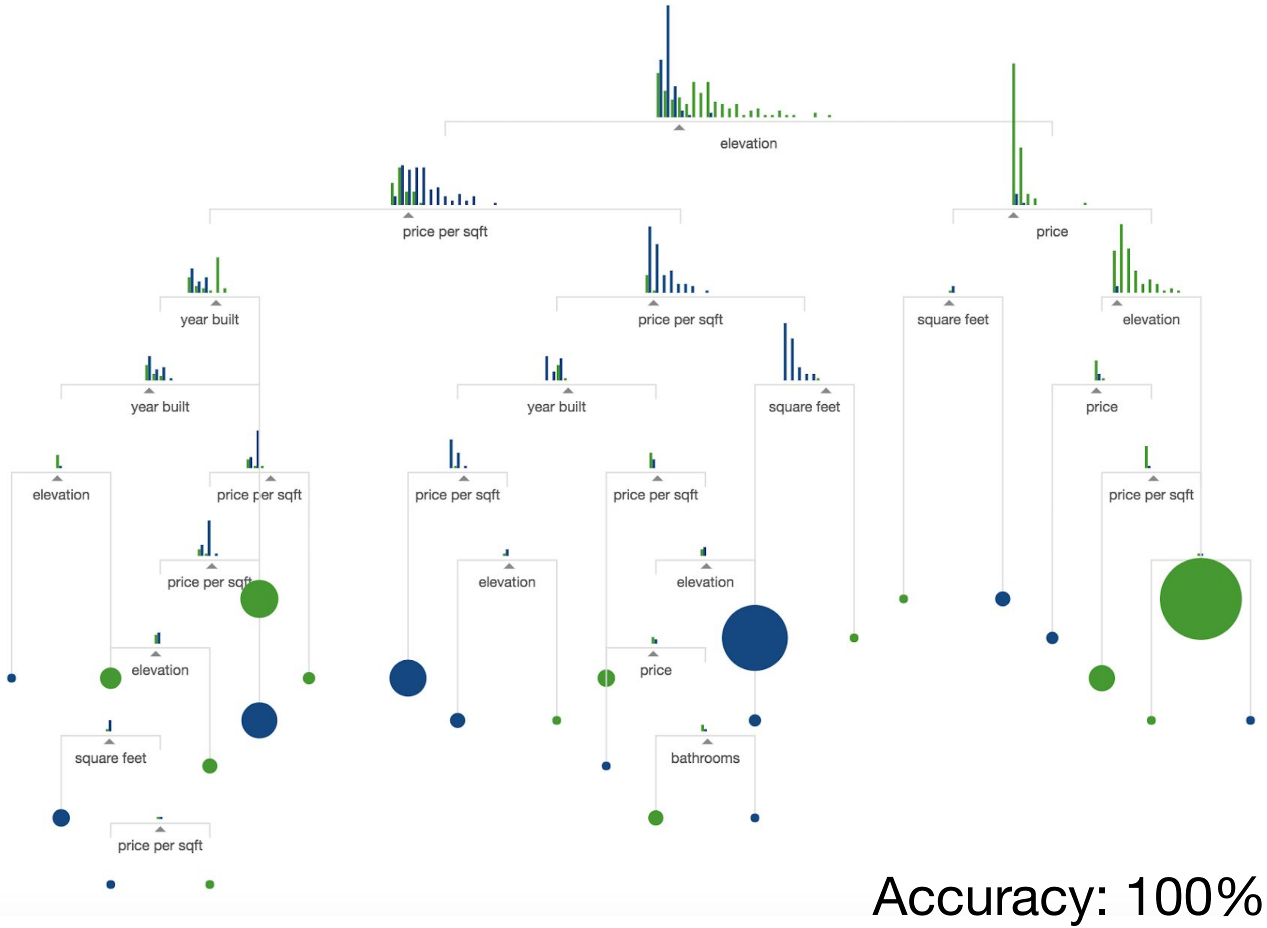
Adding several more layers, we get to **96%**.



Accuracy: 96%

http://www.r2d3.us/visual-intro-to-machine-learning-part-1/?from=@

It's possible to add branches until your model is **100% accurate**.

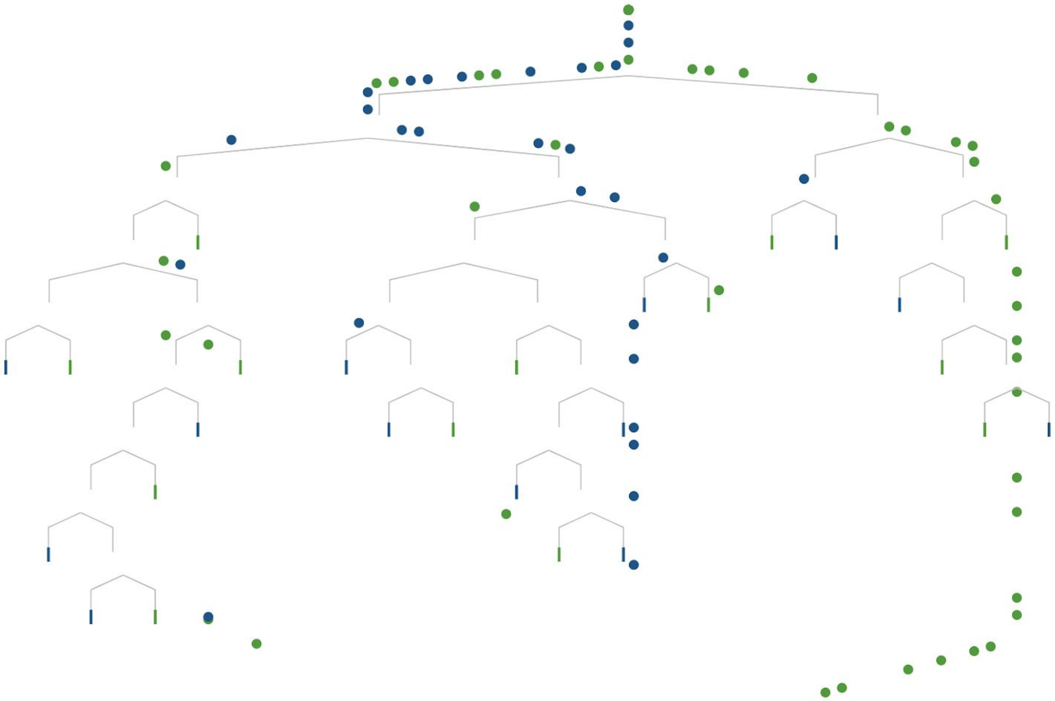


http://www.r2d3.us/visual-intro-to-machine-learning-part-1/?from=@

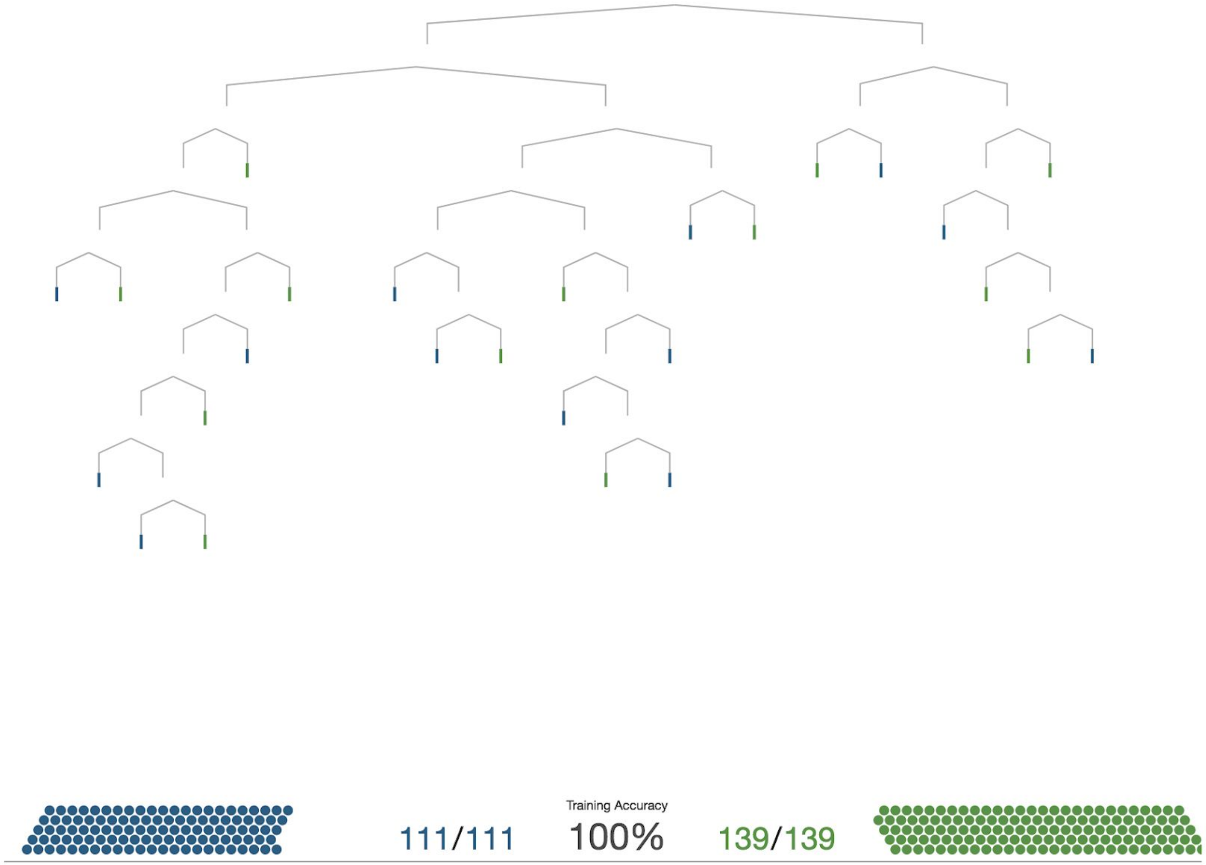
Making predictions

The decision tree **model** can then predict which homes are in which city.

Here, we're using the **training data**.



Because our tree was trained on this data and we grew the tree to 100% accuracy, each house is perfectly sorted

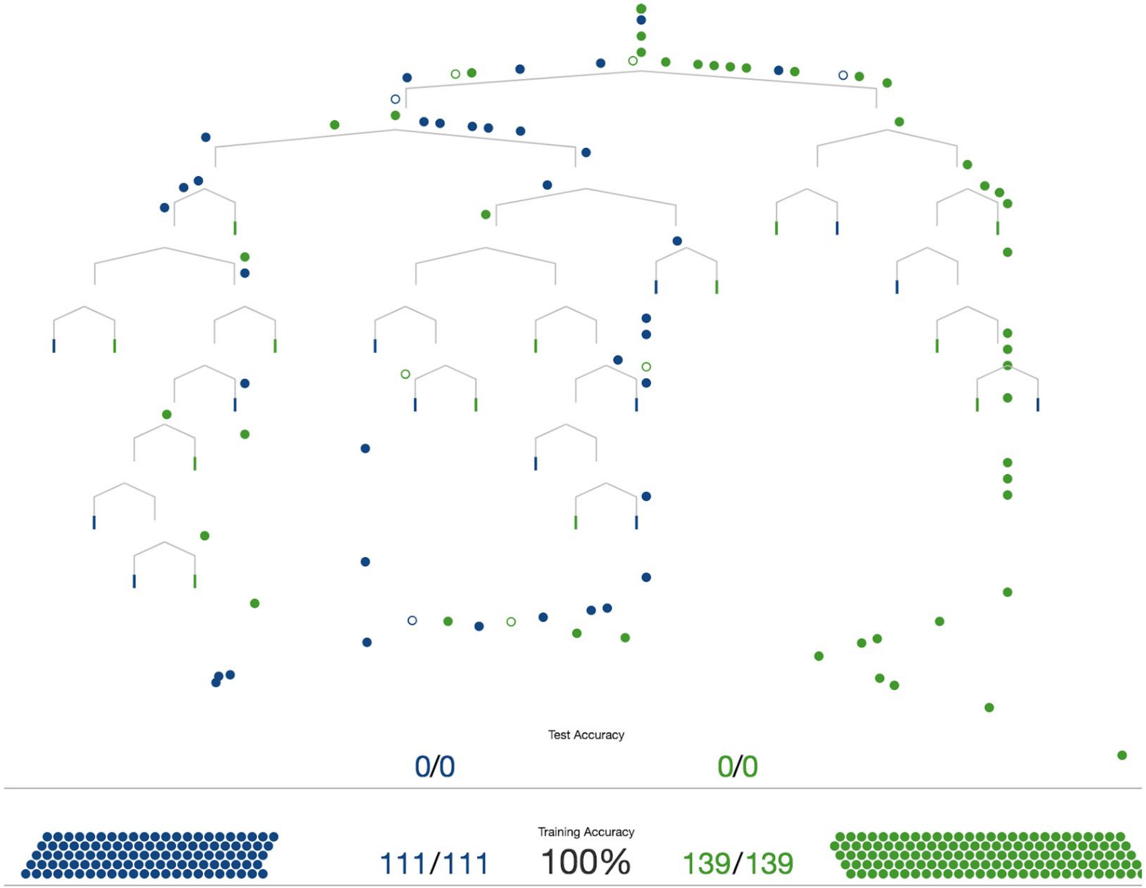


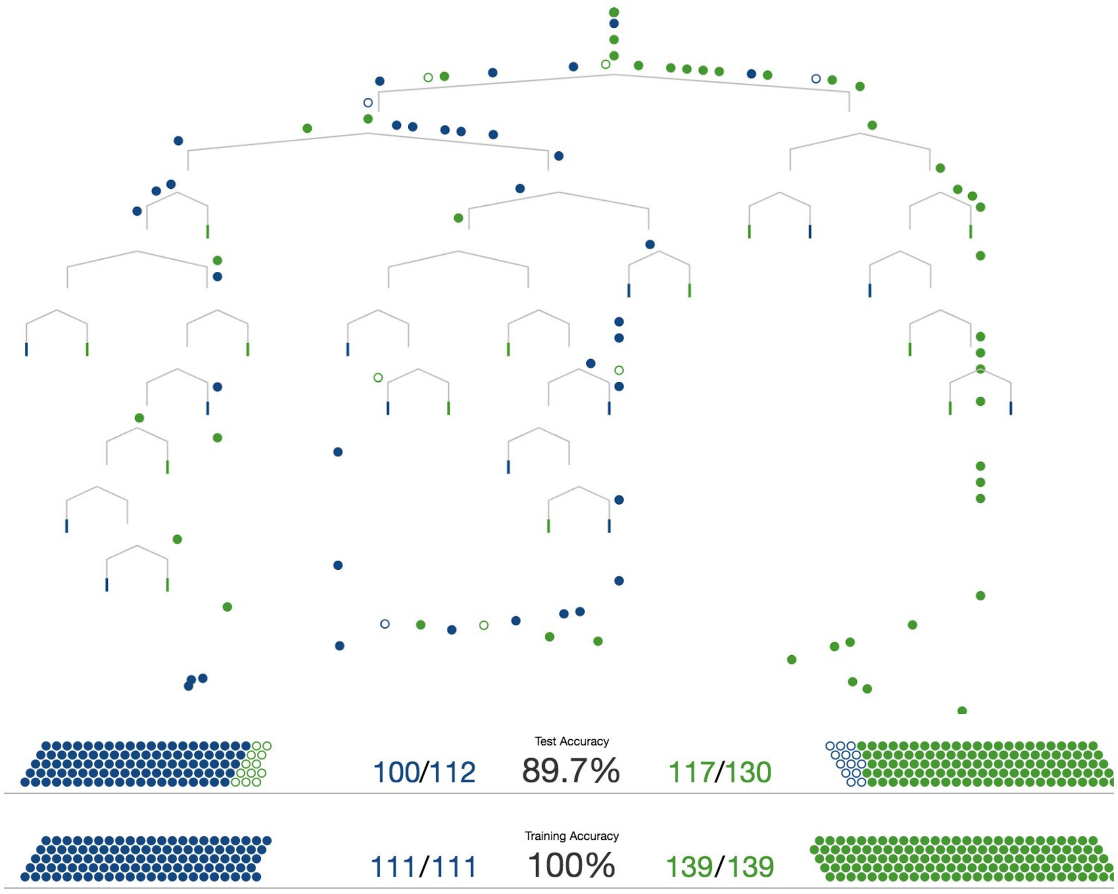
http://www.r2d3.us/visual-intro-to-machine-learning-part-1/?from=@

Reality check

But…how does this tree on data that the model hasn't seen before?

The **test set** then makes it way through the decision tree.

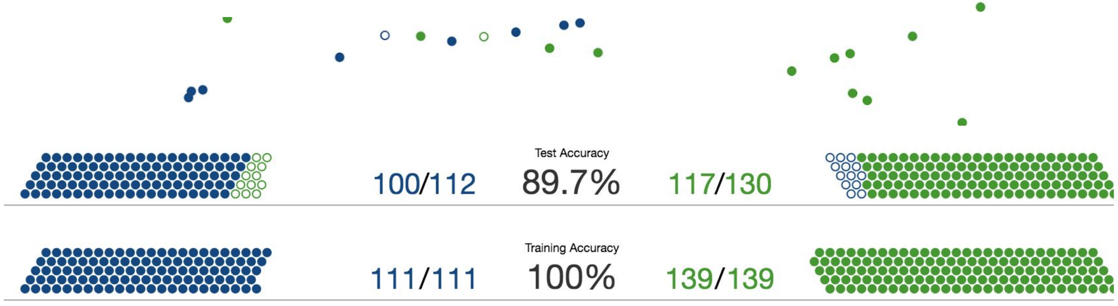




Ideally the tree should perform similarly on both known and unknown data

http://www.r2d3.us/visual-intro-to-machine-learning-part-1/?from=@

These errors are due to **overfitting**. Fitting every single detail in the training data led to a tree that modeled unimportant features, that did not allow for similar accuracy in new data.



http://www.r2d3.us/visual-intro-to-machine-learning-part-1/?from=@

Recap

1. Machine learning identifies patterns using **statistical learning** and computers by unearthing **boundaries** in data sets. You can use it to make predictions.

2. One method for making predictions is called a decision trees, which uses a series of if-then statements to identify boundaries and define patterns in the data.

3. **Overfitting** happens when some boundaries are based on on *distinctions that don't make a difference*. You can see if a model overfits by having test data flow through the model.