HOCHSCHULE DER MEDIEN

Decision Trees

Part 1) A visual introduction

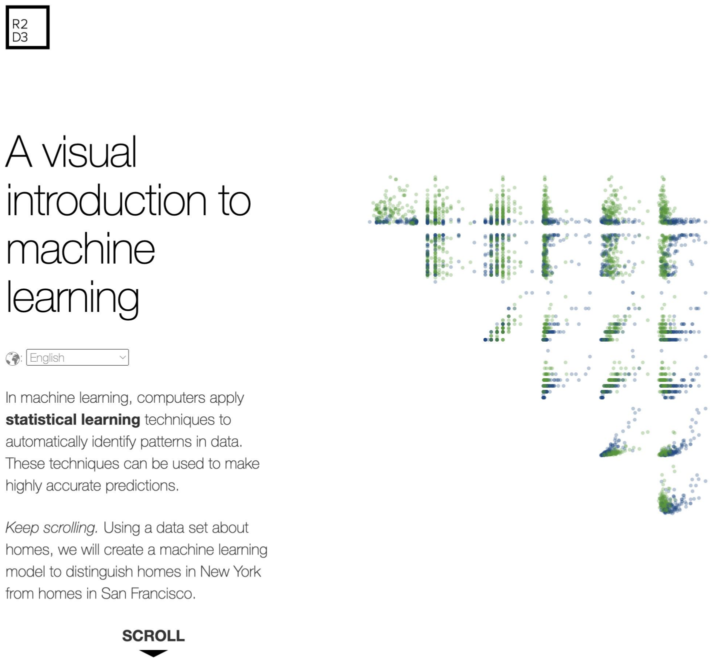
Prof. Dr. Jan Kirenz
HdM Stuttgart

Decision trees part 1

The following slides are based on the interactive tutorial

"[A visual introduction to machine learning](#)"

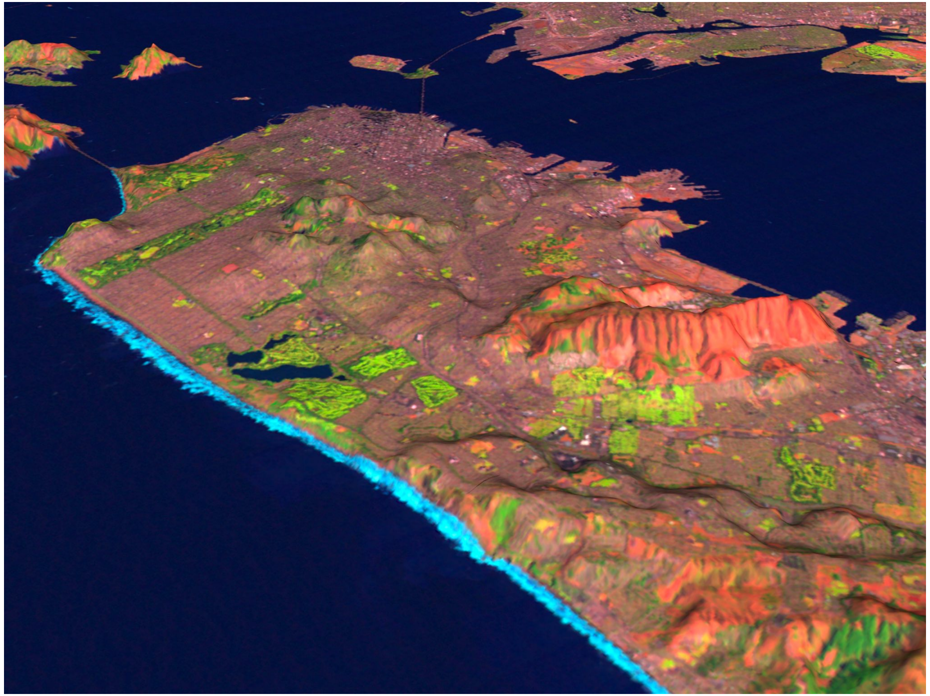
by R2D3



Homes: **San Francisco** vs **New York**

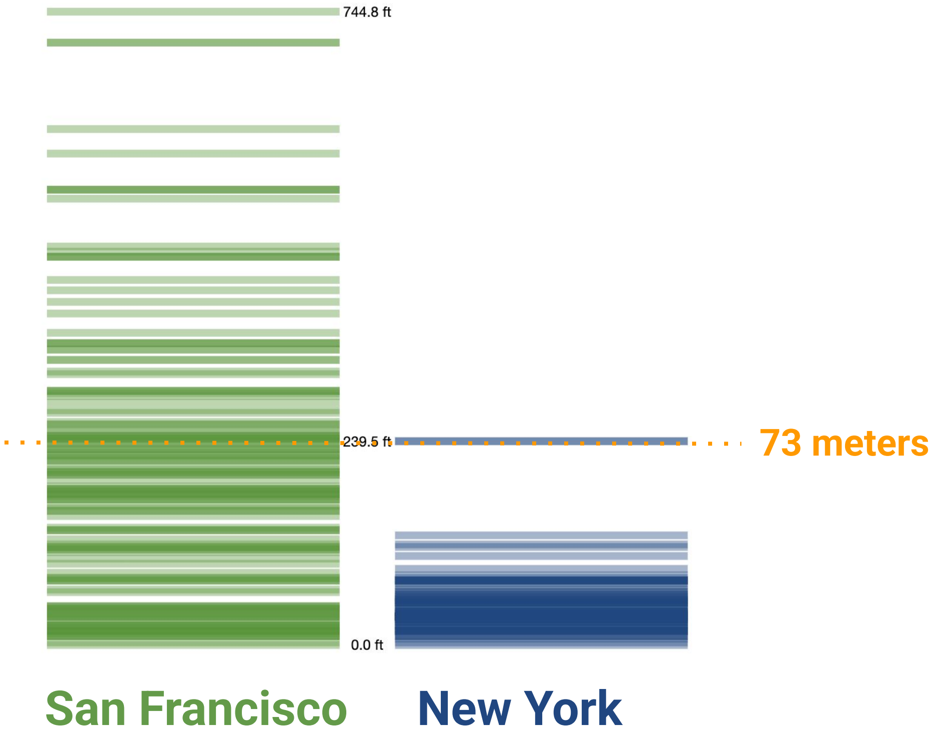
- We want to classify homes in San Francisco
- San Francisco is our “positive class”

Picture source: Unsplash



Picture Source: Wikipedia

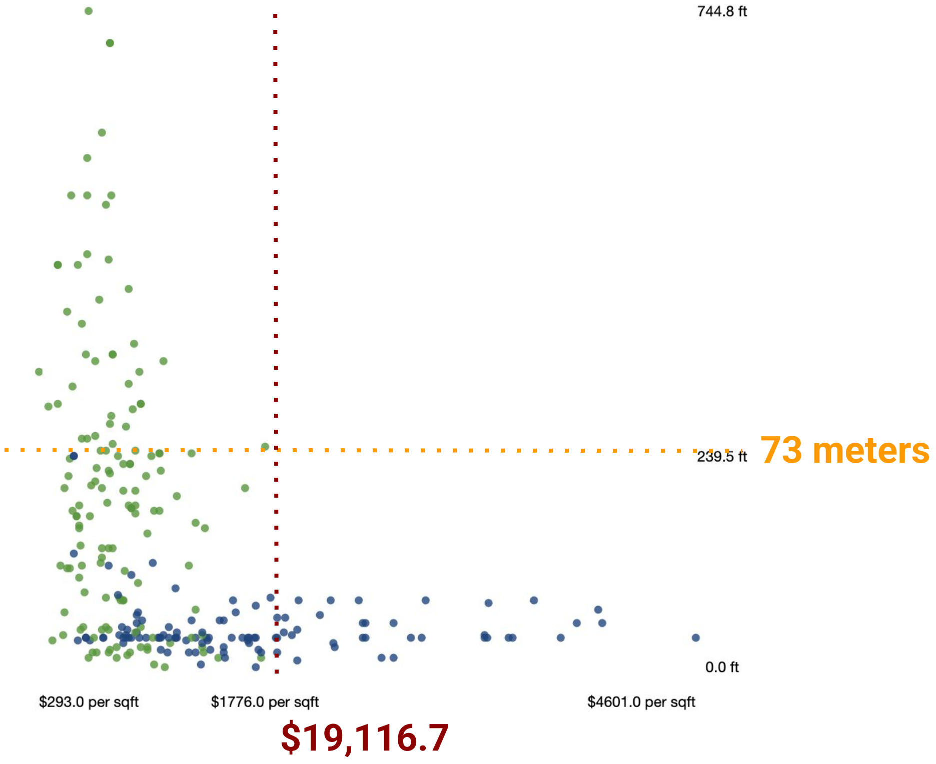
Using home-elevation to distinguish homes



- Label (dependent variable):
    - San Francisco
    - New York
- Data (feature): home-elevation
- A home above **73 meters** should be classified as one in

____

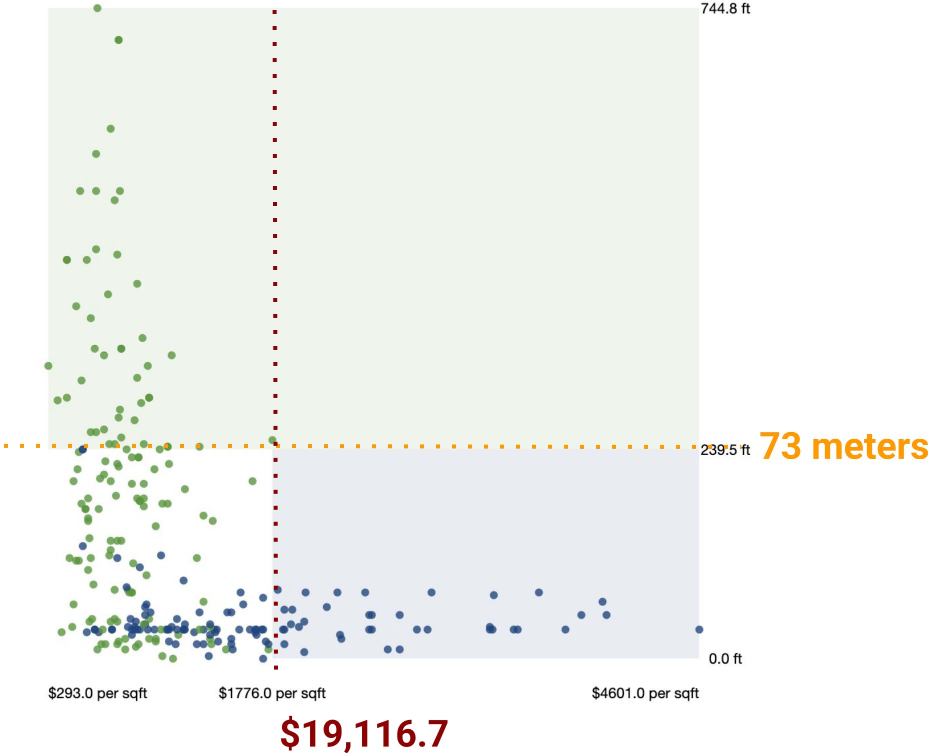
Home-elevation vs cost per square foot



- **New York** apartments can be extremely expensive per square foot.
- Homes at or below **73 meters**, those that cost more than **$19,116.7 per square meter** are in ____________.

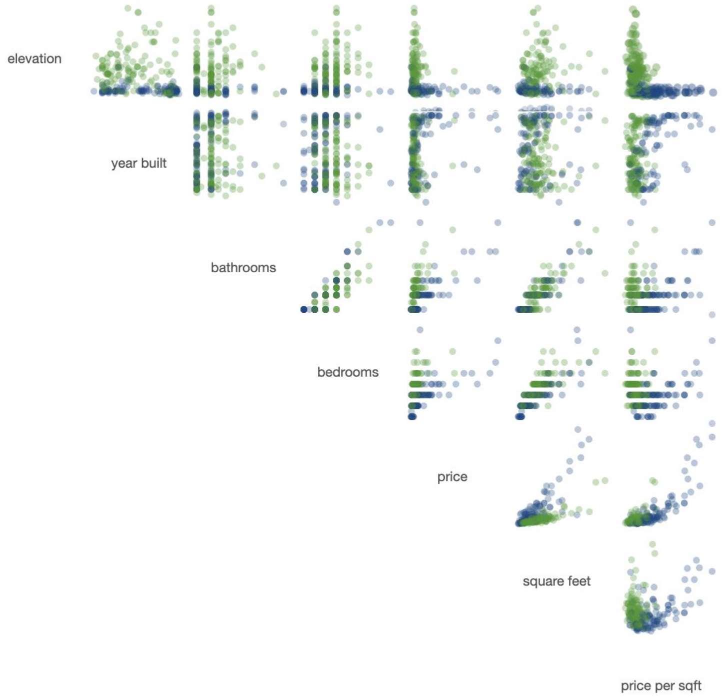
Prof. Dr. Jan Kirenz

Drawing boundaries



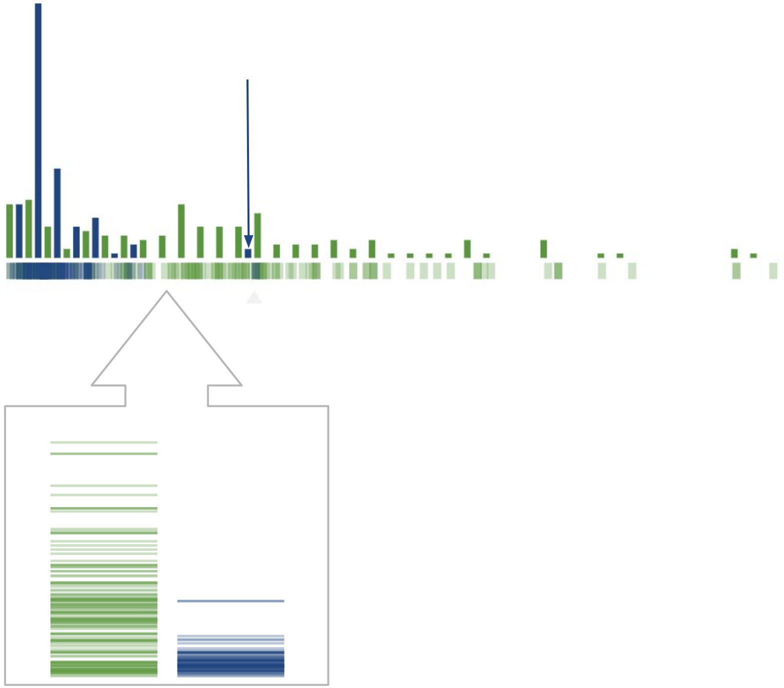
- You can visualize **boundaries** of regions at:
  - elevation (>73 m) and
  - price per square foot (>$19,116.7)
- Homes plotted in the green would be in ___________.
- *What about homes with lower elevations and lower per-square-foot prices (white region)?*

Using more data (dimensions)



- We have 7 different dimensions (also called features).
- Scatterplot matrix shows the relationships between each pair of dimensions.
- There are **patterns (rules)** in the data, but the **boundaries** between our **labels** (answers) are not obvious.

Finding better boundaries

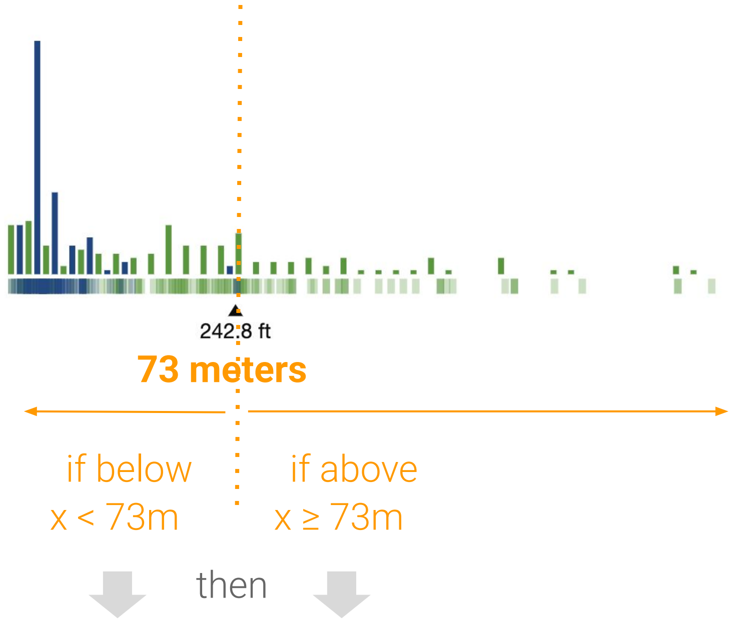


- Transform chart into a __________
- The highest home in New York is 73m
- Majority of homes in New York are far below 73m

We use machine learning to find **rules** in data

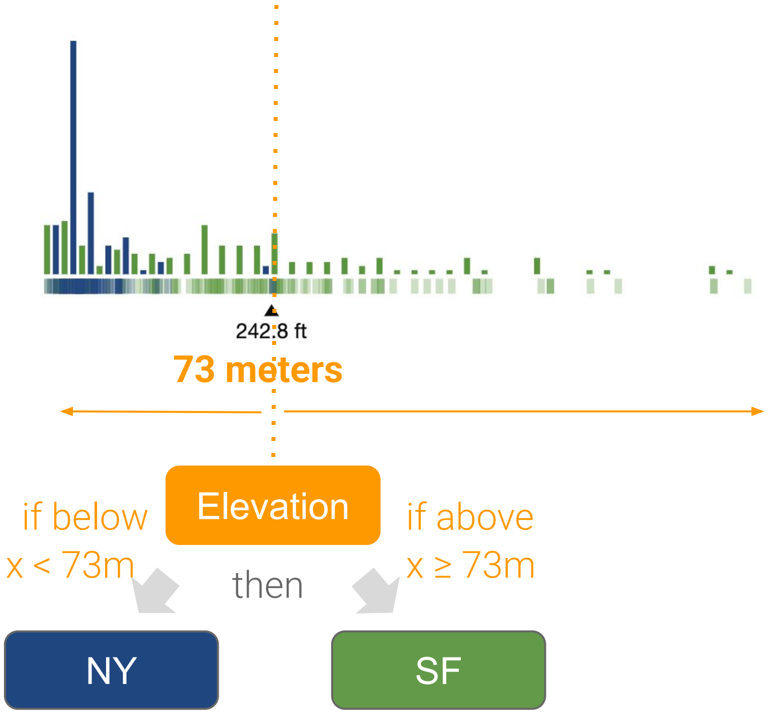
- Machine learning methods use labels (**answers**) and features (**data**) to identify **rules** (in our case boundaries).
- Creating a machine learning **model** is also known as **training** a model (with data and answers).

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



- For example, if a home's elevation is
  - *above* some number (x > boundary), then the home is probably in ________.
  - *below* (x < boundary), then the home is probably in ________

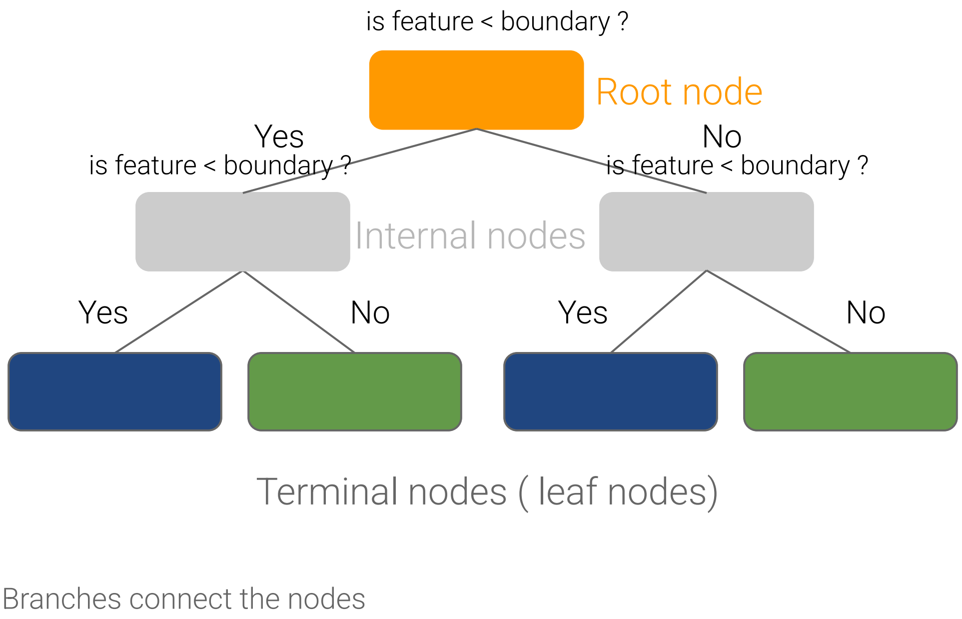
Your first fork



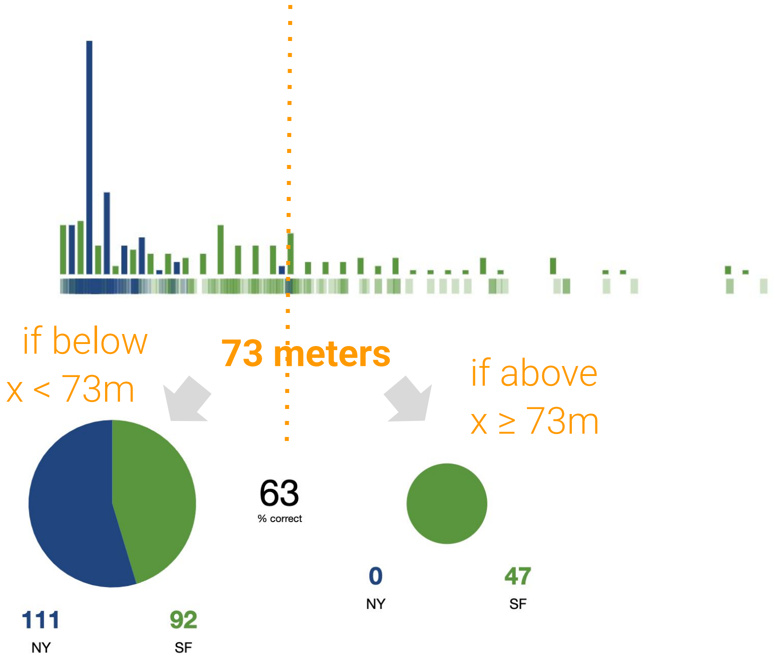
- The if-then statements are called **forks**
- They split the data into two **branches** based on some value.
- That value between the branches is called a **split point**.
- Homes to the left of that point get categorized in one way, while those to the right are categorized in another.
- A split point is the decision tree's version of a **boundary**.

Decision tree terminology

- Decision trees can be applied to both **classification** and **regression** problems.
- Decision trees use one variable at a time (to make a split)
- They use **recursive** binary splitting to grow a tree on the training data
- Our decision tree models will classify the homes in each leaf node according to which class of homes is in the **majority**



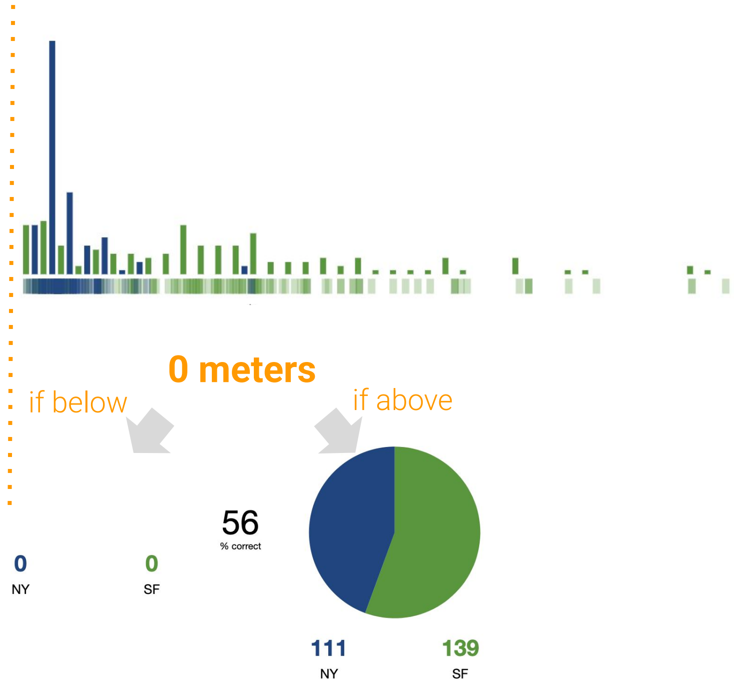
Split point tradeoffs & misclassifications



- Our initial split (**73 m**)
  - above: correctly classifies all SF homes
  - below: some misclassifications
- San Francisco homes that are **misclassified** are called

  ____

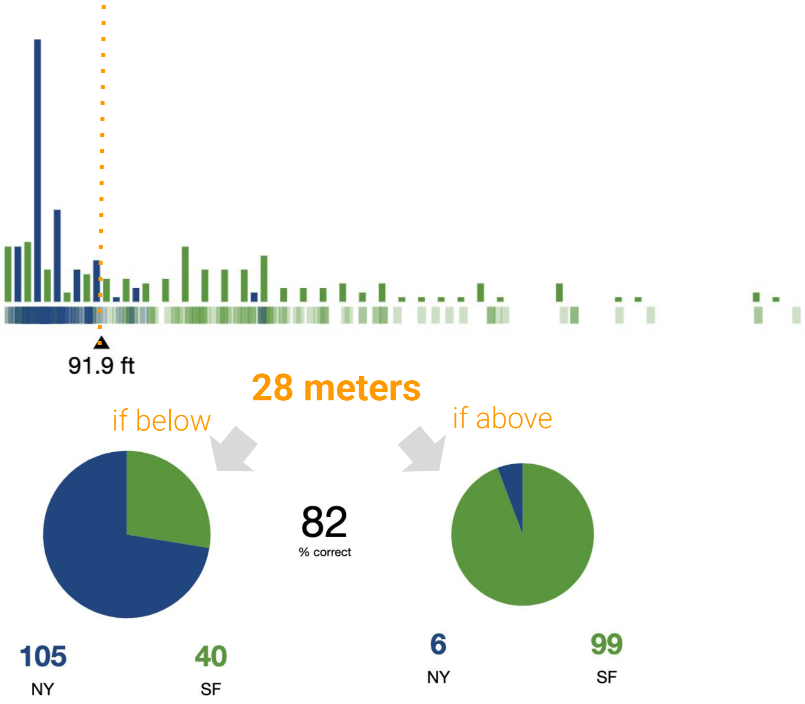
Split point tradeoffs & misclassifications



- Split point to capture every San Francisco home: **0m**
- This split point will also include all New York homes as well.
- The misclassified New York homes are called

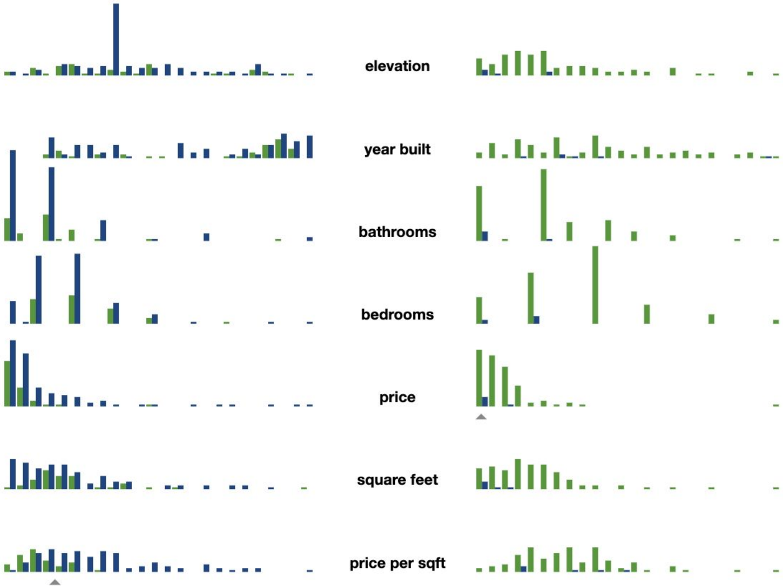
____

The best split



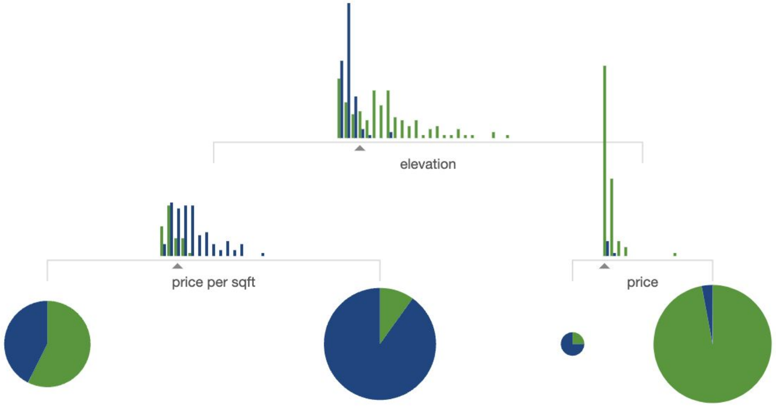
- At the best split, the results of each branch should be as homogeneous (**pure**) as possible.
- There are several mathematical methods you can choose between to calculate the best split

Recursion



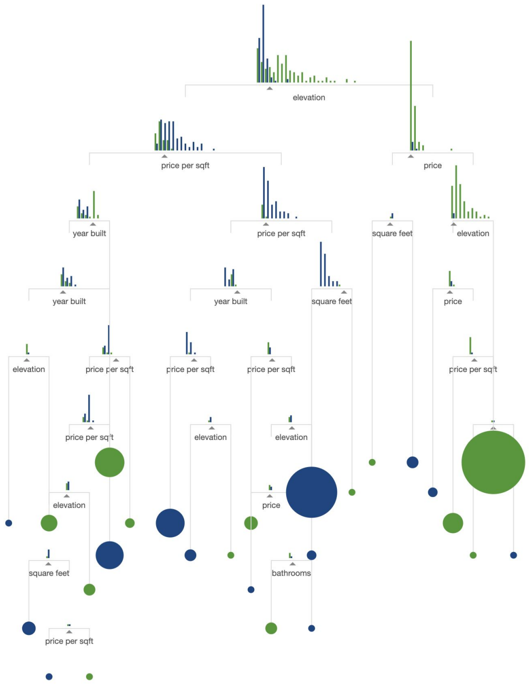
- To add another split point, the algorithm repeats the process above on the subsets of data.
- This repetition is called **recursion**

Growing a tree



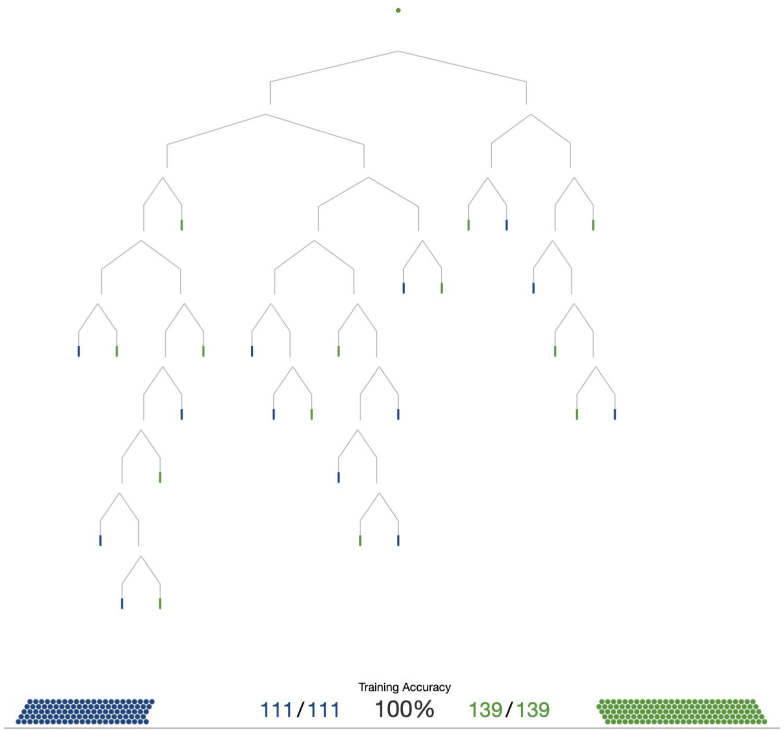
- Additional forks will add new information
- This can increase a tree's prediction accuracy.

Growing a large tree



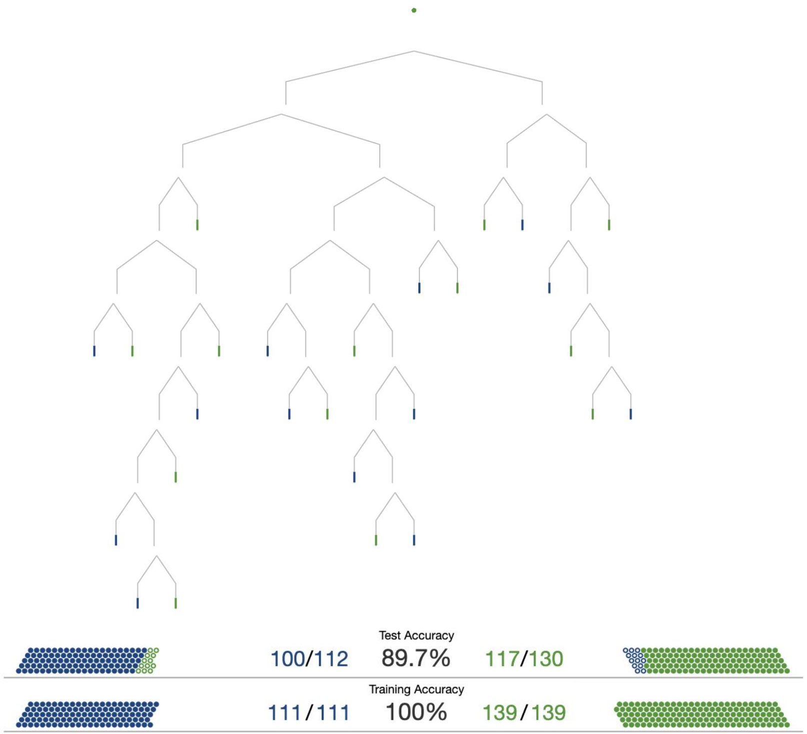
- We could add branches until the tree's predictions are 100% accurate,
- This means that there are no misclassifications at the terminal nodes of every branch

Making predictions on training data



- The newly-trained decision tree model determines whether a home is in San Francisco or New York by running each data point through the branches.
- The data is called **training data** because it was used to train the model.

Reality check with test data



- To test the tree's performance on new data, we need to apply it to data points that it has never seen before (**test data**).
- The resulting errors are due to **overfitting**.
- Our model has learned to treat every detail in the training data as important, even details that turned out to be irrelevant.

Next: Decision trees part 2

The following slides are based on the interactive tutorial

“Model Tuning and

the Bias-Variance Tradeoff”

by R2D3