Fundamentals of Learning

Kaiqi Zhao
The University of Auckland

Slides are partially based on the materials from the University of British Columbia

Content

- Training Error and Test Error
- Golden Rule of Machine Learning
- IID
- Fundamental Trade-Off
- Validation Error
- Hyperparameter Tuning
- Classifier Evaluation Metrics

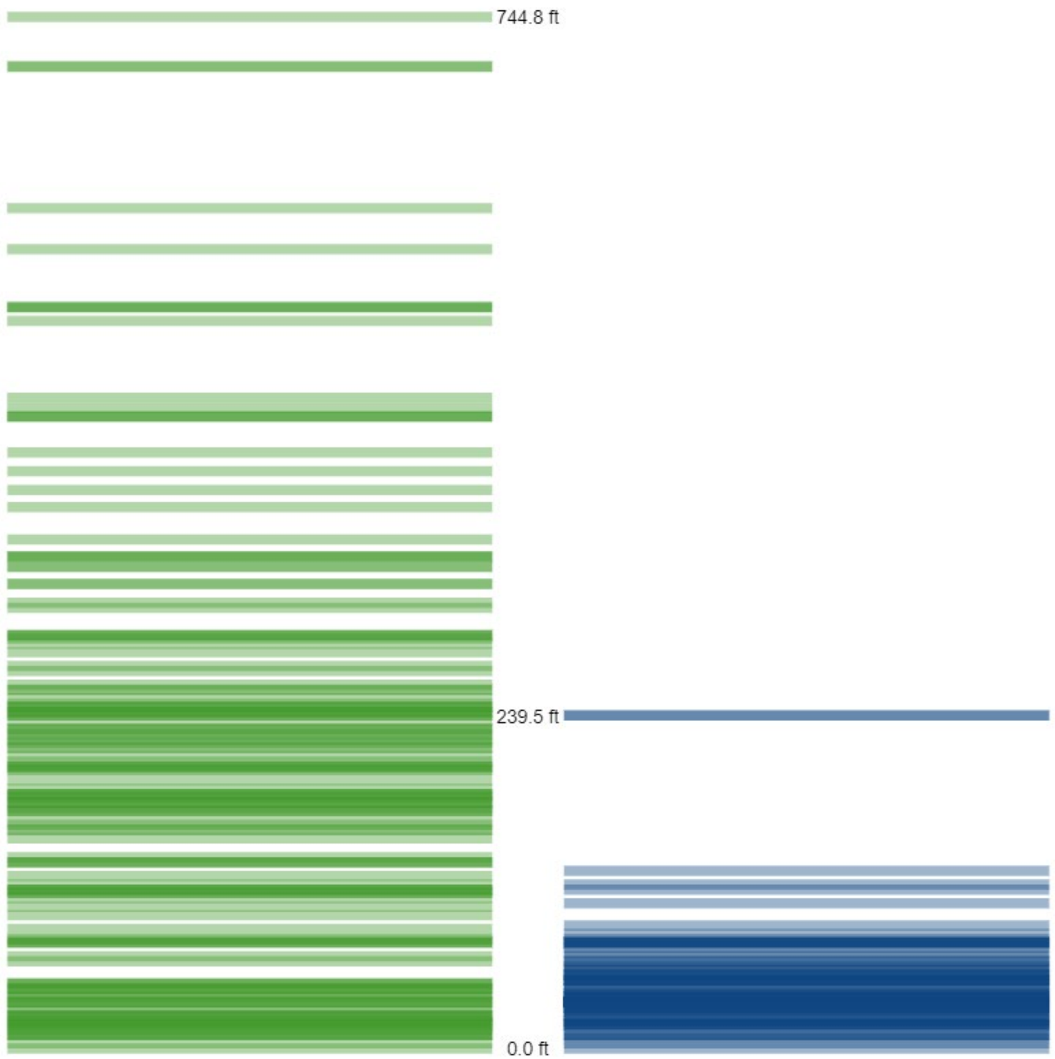
Motivative Example: Determine Home City

- We are given data from 250 homes
- For each home/example, we have these features
  - Elevation
  - Year
  - Bathrooms-Bedrooms
  - Price
  - Square feet
- Goal is to build a program that predicts location (SF or NY) for homes

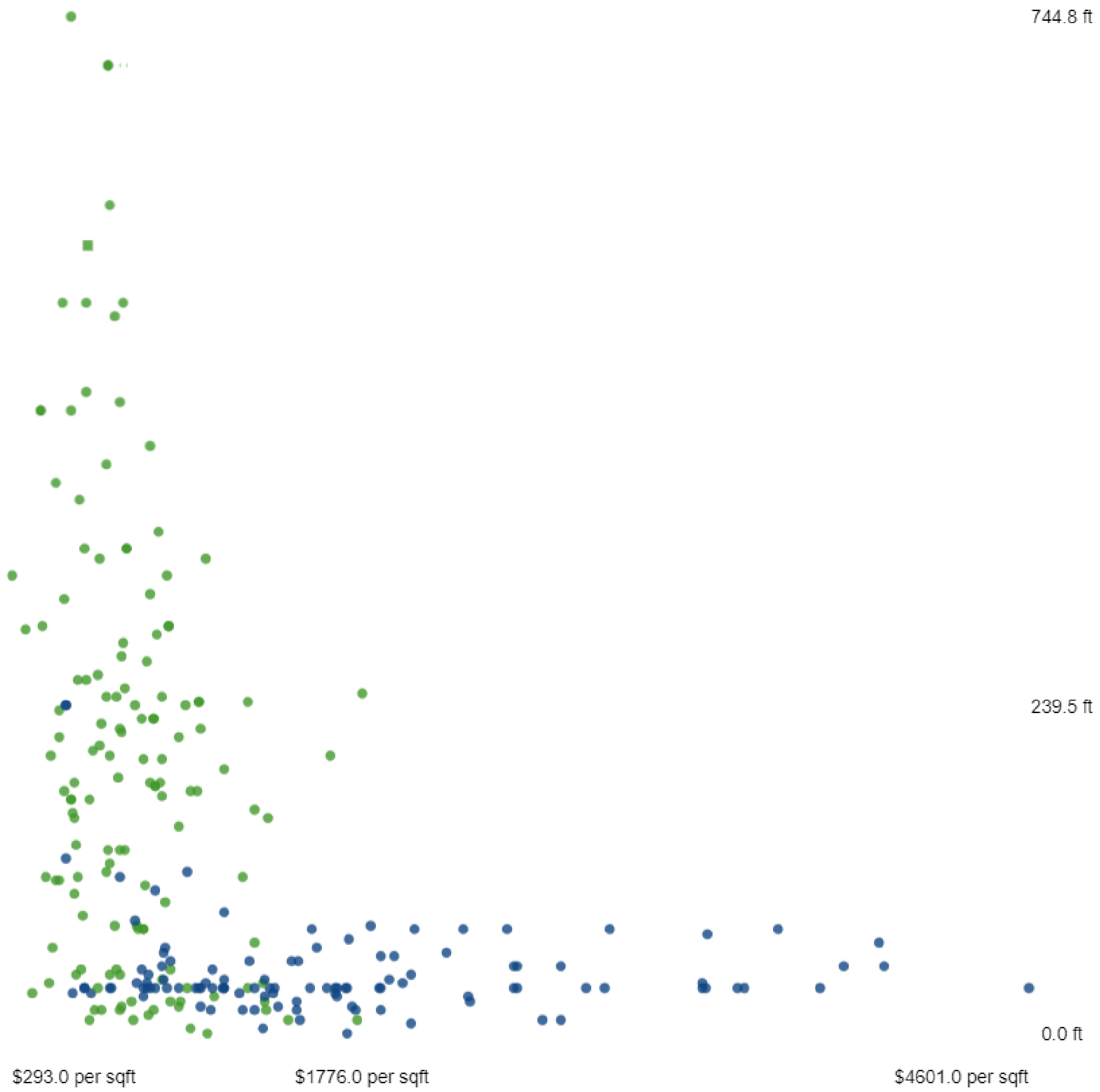
This example and images of it come from: http://www.r2d3.us/visual-intro-to-machine-learning-part-1

Elevation

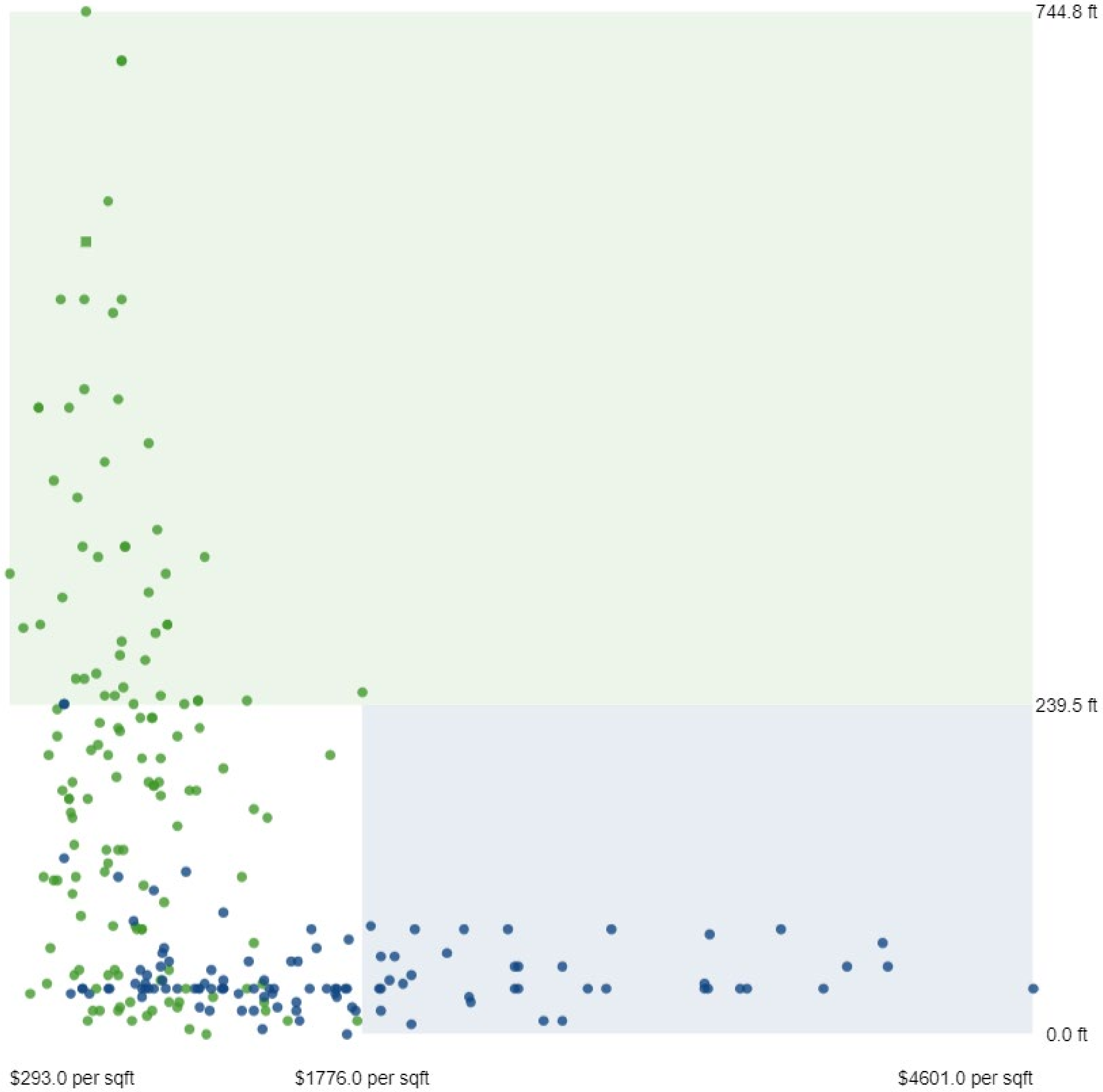
- Elevation (in feet) for San Francisco (SF) and New York (NY).



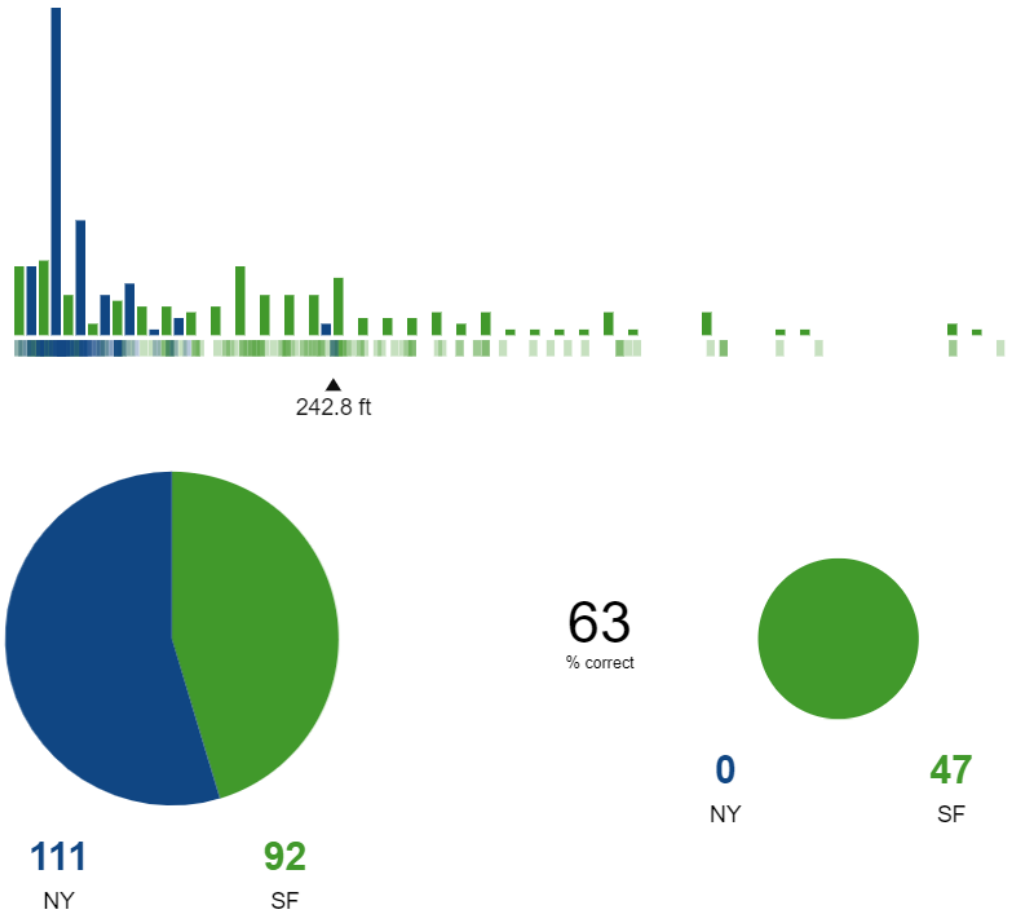
Elevation vs. Price/SqFt

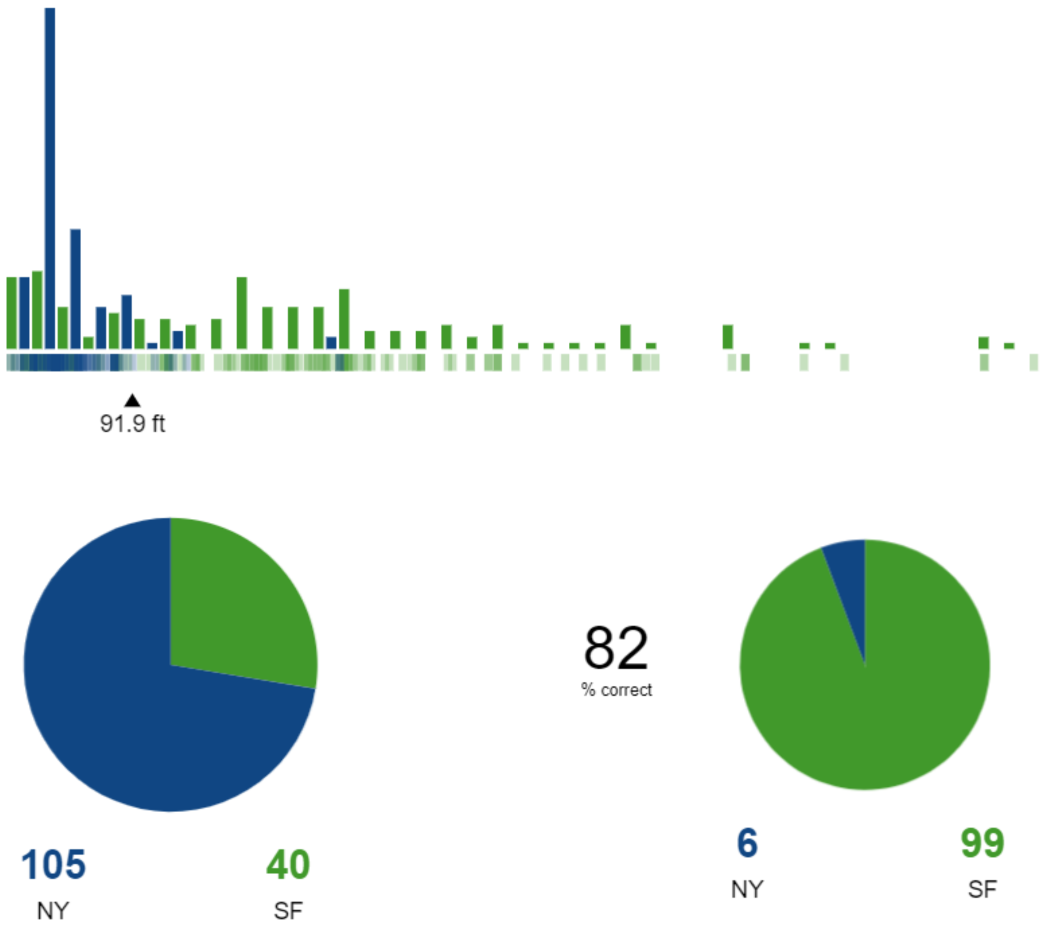


Simple Decision Tree



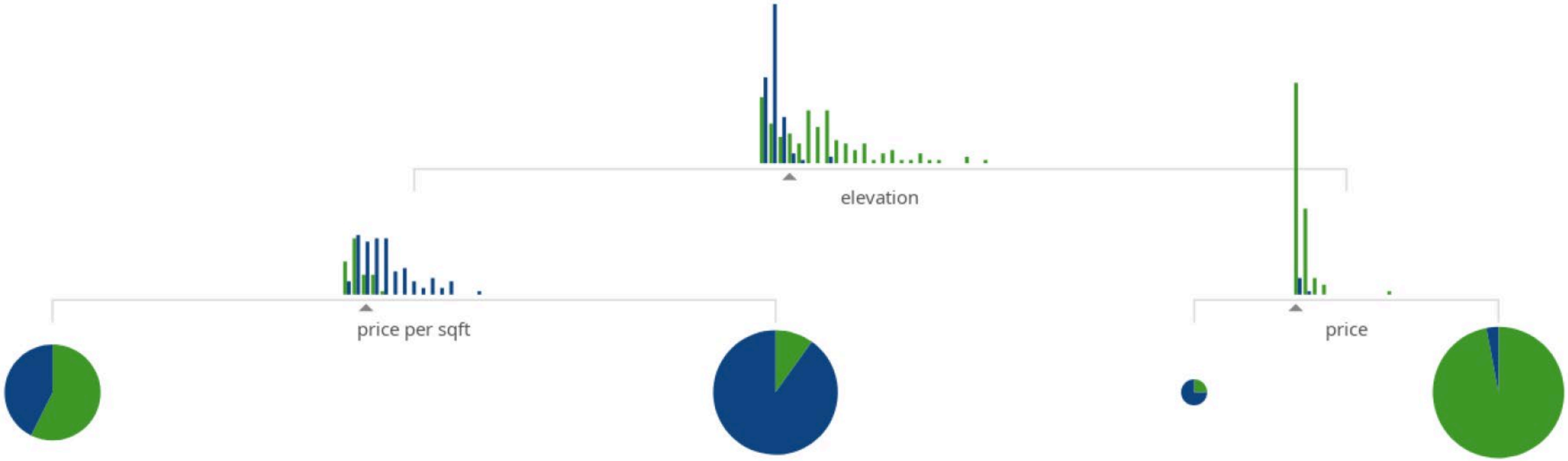
Depth vs. Accuracy



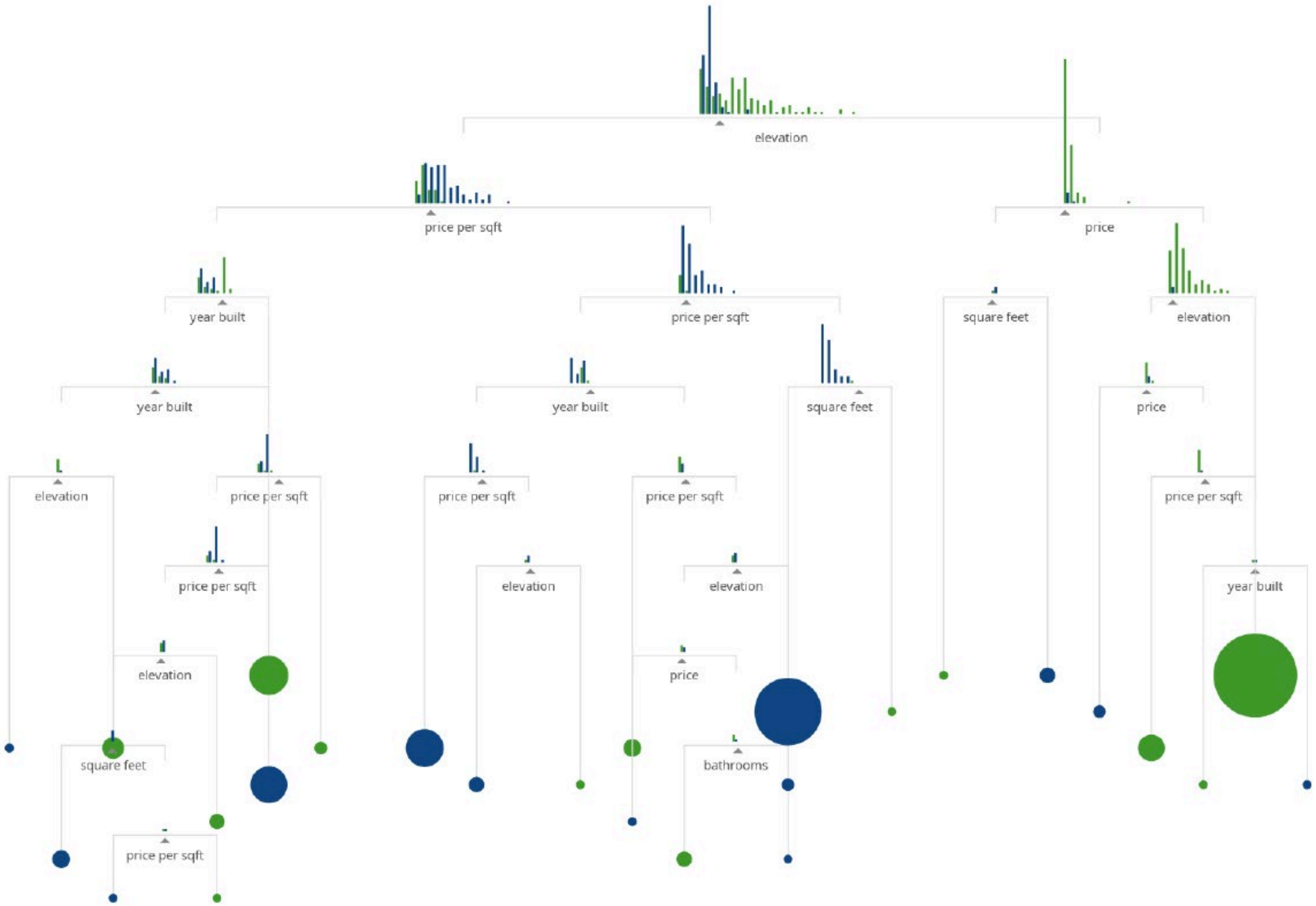


The best split

Depth vs. Accuracy



Depth vs. Accuracy



Training Error and Test Error

Error

- Eventually, we achieved a perfect classification on the data ☺
- With this decision tree, ‘training accuracy’ is 1 (training error=0)
  - It perfectly labels the data we used to make the tree

- We are now given features for 242 new homes
- What is the 'testing accuracy’ (or test error) on the new data NOT used to make the tree?

- **Overfitting**: lower accuracy on new data
  - Our rules got too specific to the training dataset
  - Some of the "deep" splits only contain a few examples (extreme case - only one example)

Supervised Learning

- We are given training data where we know the labels

$X =$

Egg	Milk	Fish	Wheat	Shellfish	Peanuts	…
0.0	0.7	0.0	0.3	0.0	0.00	…
0.3	0.7	0.0	0.6	0.0	0.01	…
0.0	0.0	0.0	0.8	0.0	0.00	…
0.3	0.7	1.2	0.0	0.1	0.01	…
0.3	0.0	1.2	0.3	0.1	0.01	…

$y =$

Sick?
1
1
0
1
1

- But there is also testing data we want to label

$\tilde{X} =$

Egg	Milk	Fish	Wheat	Shellfish	Peanuts	…
0.5	0.0	1.0	0.6	2.0	1.0	…
0.0	0.7	0.0	1.0	0.0	0.0	…
3.0	1.0	0.0	0.5	0.0	0.0	…

$\tilde{y} =$

Sick?
?
?
?

Supervised Learning

- Typical supervised learning steps
  1. Build model based on training data X and y (training phase)
  2. Model makes predictions $\hat{y}$ on test data $\tilde{X}$ (testing phase)

- Instead of training error, consider test error:
  - Are predictions $\hat{y}$ similar to true unseen labels $\tilde{y}$?

Goal of Machine Learning

- In machine learning:
  - We care more about the test error
- Course analogy:
  - The training error is practice on the exam
  - The test error is the actual exam
  - Goal: do well on actual exam, not only the practice
- Memorization vs learning:
  - Can do well on training data by memorizing it?
  - You have only learned if you can do well in new situations

Golden Rule of Machine Learning

Golden Rule of Machine Learning

- Even though what we care about is test error:
  - The test data cannot influence the training phase in any way
- We're measuring test error to see how well we do on new data:
  - If used during training, you are not actually measuring the error on **unseen** data, because the test information leaks to your model during training.
  - You can start to overfit if you use it during training
  - Course analogy: You see the exam questions before you attend the exam!

Golden Rule of Machine Learning

- You also shouldn't change the test set to get the result you want
- Note the golden rule applies to hypothesis testing in scientific studies
  - Data that you collect can't influence the hypotheses that you test
- **Extremely common and a major problem**, coming in many forms
  - Collect more data until you coincidentally improve your hypothesis
  - Try different ways to measure the test performance, choose the one that looks best
  - Choose a different type of model / hypothesis after looking at the test data
- If you want to modify your hypotheses, you need to test on new data
- Or at least be aware and honest about this issue when reporting results

Is Learning Possible?

- Does training error say anything about test error?
  - In general, NO: Test data might have nothing to do with training data
  - E.g., "adversary" takes training data and flips all labels.

- In order to learn, we need assumptions:
  - The training and test data need to be related in some way
  - Most common assumption: independent and identically distributed (IID)

IID

IID Assumption

- Training/test data is independent and identically distributed (IID) if:
  - All examples come from the same distribution (identically distributed)
  - The example are sampled independently (order does not matter)

Age	Job?	City	Rating	Income
23	Yes	Ham	A	22,000.00
23	Yes	Wel	BBB	21,000.00
22	No	Ham	CC	0.00
25	Yes	Akl	AAA	57,000.00

- Examples in terms of cards - which is IID?
  - Pick a card, put it back in the deck, re-shuffle, repeat
  - Pick a card, put it back in the deck, repeat
  - Pick a card, don't put it back, re-shuffle, repeat.

IID in the Food Allergy Example

- Is the food allergy data IID?
  - Do all the examples come from the same distribution?
  - Does the order of the examples matter?
- No!
  - Being sick might depend on what you ate yesterday (not independent)
  - Your eating habits might changed over time (not identically distributed)
- What can we do about this?
  - Just ignore that data isn't IID and hope for the best?
  - For each day, maybe add the features from the previous day?
  - Maybe add time as an extra feature?

Learning Theory

- Why does the IID assumption make learning possible?
  - Patterns in training examples are likely to be the same in test examples
- The IID assumption is rarely true:
  - But it is often a good approximation
  - There are other possible assumptions
- Also, we're assuming IID across examples but not across features
- Learning theory explores how training error is related to test error
- We'll look at a simple example, using this notation:
  - E_{train} is the error on training data
  - E_{test} is the error on testing data

Fundamental Trade-Off

Fundamental Trade-Off

- The test error E_{test} could be decomposed to training error E_{train} and the error of using E_{train} to approximate E_{test}
- Start with $E_{test} = E_{test}$, then add and subtract E_{train} on the right:

$$\underbrace{E_{test}}_{\text{test error}} = \underbrace{(E_{test} - E_{train})}_{\text{approximation error}} + \underbrace{E_{train}}_{\text{training error}}$$

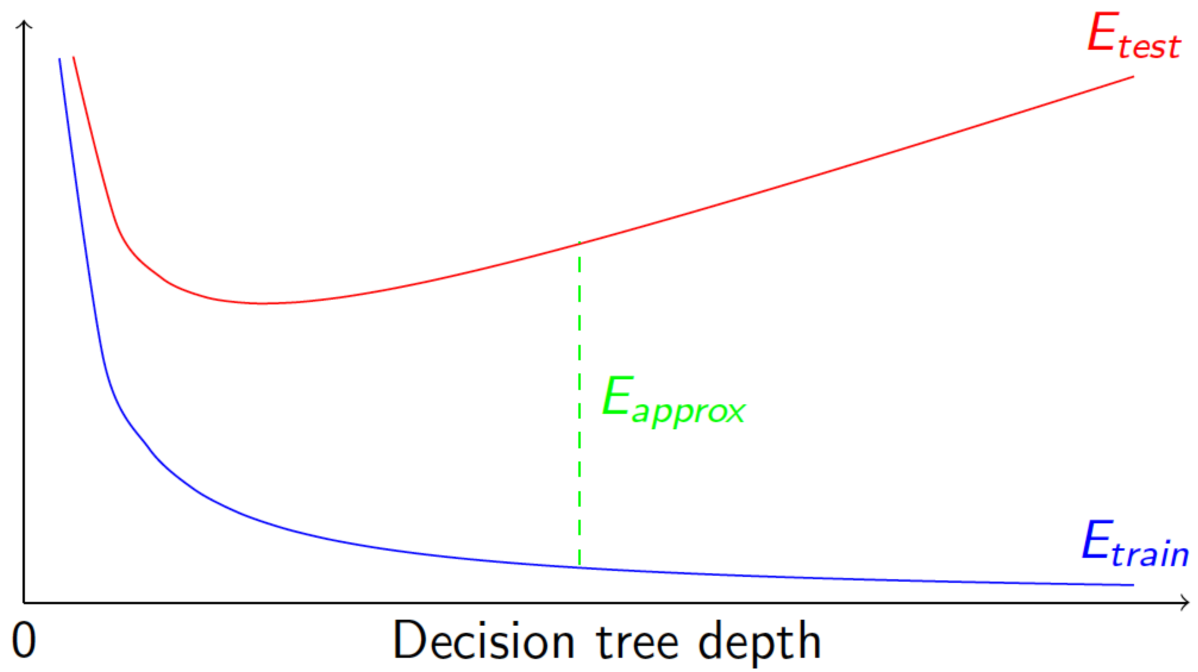
- How does this help?
  - If $E_{approx} = E_{test} - E_{train}$ is small, then E_{train} is a good approximation to E_{test}
- What does E_{approx} ("amount of overfitting") depend on?
  - It tends to get smaller as the training data set gets larger
  - It tends to grow as model get more "complicated"

Fundamental Trade-Off

- This leads to a **fundamental trade-off**:
  1. E_{train}: how small you can make the training error vs.
  2. E_{approx}: how well training error approximates the test error
- Simple models (like decision stumps):
  - E_{approx} is low (not very sensitive to training set)
  - But E_{train} might be high
- Complex models (like deep decision trees):
  - E_{train} can be low
  - But E_{approx} might be high (very sensitive to training set).

Fundamental Trade-Off

- Training error vs. test error for choosing tree depth:
  - Training error is high for low depth (underfitting)
  - Training error gets better with depth
  - Test error initially goes down, but eventually increases (overfitting)



Refined Fundamental Trade-Off

- Let E_{best} be the irreducible error (lowest possible error for any model)
  - For example, irreducible error for predicting coin flips is 0.5
- Some learning theory results use E_{best} to further decompose E_{test}

$$E_{test} = \underbrace{(E_{test} - E_{train})}_{\text{variance}} + \underbrace{(E_{train} - E_{best})}_{\text{bias}} + \underbrace{E_{best}}_{\text{noise}}$$

- This is similar to the bias-variance decomposition
  - Variance: how sensitive we are to training data
  - Bias: how low can we make the training error
  - Noise: how low can any model make test error

Refined Fundamental Trade-Off

- Decision tree with high depth
  - Very likely to fit data well, so bias is low
  - But model changes a lot if you change the data, so variance is high
- Decision tree with low depth
  - Less likely to fit data well, so bias is high
  - But model does not change much you change data, so variance is low
- And degree does not affect irreducible error
  - Irreducible error comes from the best possible model

Bias-Variance Decomposition

- You may have seen bias-variance decomposition
  - Assumes $\tilde{y}_i = \bar{y}_i + \epsilon$, where ϵ has mean 0 and variance σ^2
  - Assumes we have a learner that can take n training examples and use these to make predictions $\hat{y}_i$
- Expected squared test error in this setting is

$$\underbrace{\mathbb{E}[(\tilde{y}_i - \hat{y}_i)^2]}_{\text{test squared error}} = \underbrace{\mathbb{E}[(\hat{y}_i - \bar{y}_i)^2]}_{\text{bias}} + \underbrace{\left(\mathbb{E}\left[\hat{y}_i^2\right] - \mathbb{E}[\hat{y}_i]^2\right)}_{\text{variance}} + \underbrace{\sigma^2}_{\text{noise}}$$

  - Where expectations are taken over possible training sets of n examples
  - Bias is expected error due to having wrong model
  - Variance is expected error due to sensitivity to the training set
  - Noise (irreducible error) is the best can hope for given the noise (E_{best})

Bias-Variance vs. Fundamental Trade-Off

- Both decompositions serve the same purpose
  - Trying to evaluate how different factors affect test error

- They both lead to the same 3 conclusions
  1. Simple models can have high E_{train} / bias, low E_{approx} / variance
  2. Complex models can have low E_{train} / bias, high E_{approx} / variance
  3. As you increase n, the number of training examples, E_{approx} / variance goes down

Another Perspective