CPSC 340:
Machine Learning and Data Mining

Training, Testing, and

Cross-Validation

Fall 2015

Admin

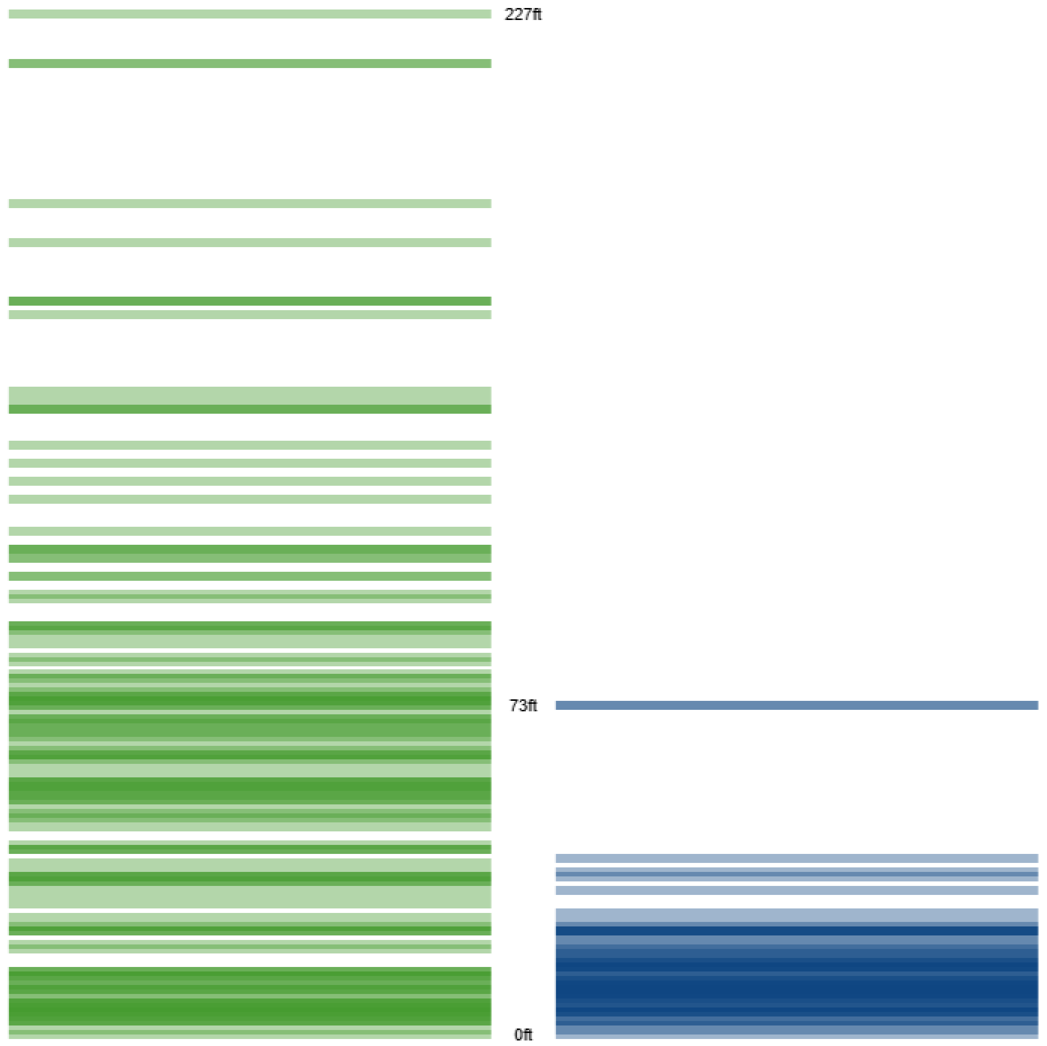
- Assignment 1 due Friday at 3pm
- Changes in submission of assignment:
  - Do Question 1 via the online survey (linked in a1.pdf).
  - Submit assignment via handin (instructions on Piazza).
    (submit via paper only if this doesn't work)
- Auditor requirements:
  - Hand in 2 of the assignments.
  - Or write a 2-page report describing one of the techniques from class.
  - Or attend more than 90% of classes.
- "Notes on Big-N" added to the course webpage:
  - Overview of what is meant by "O(nd)".
- Lecture 3 slides updated to be more clear.

Motivation: Determine Home City

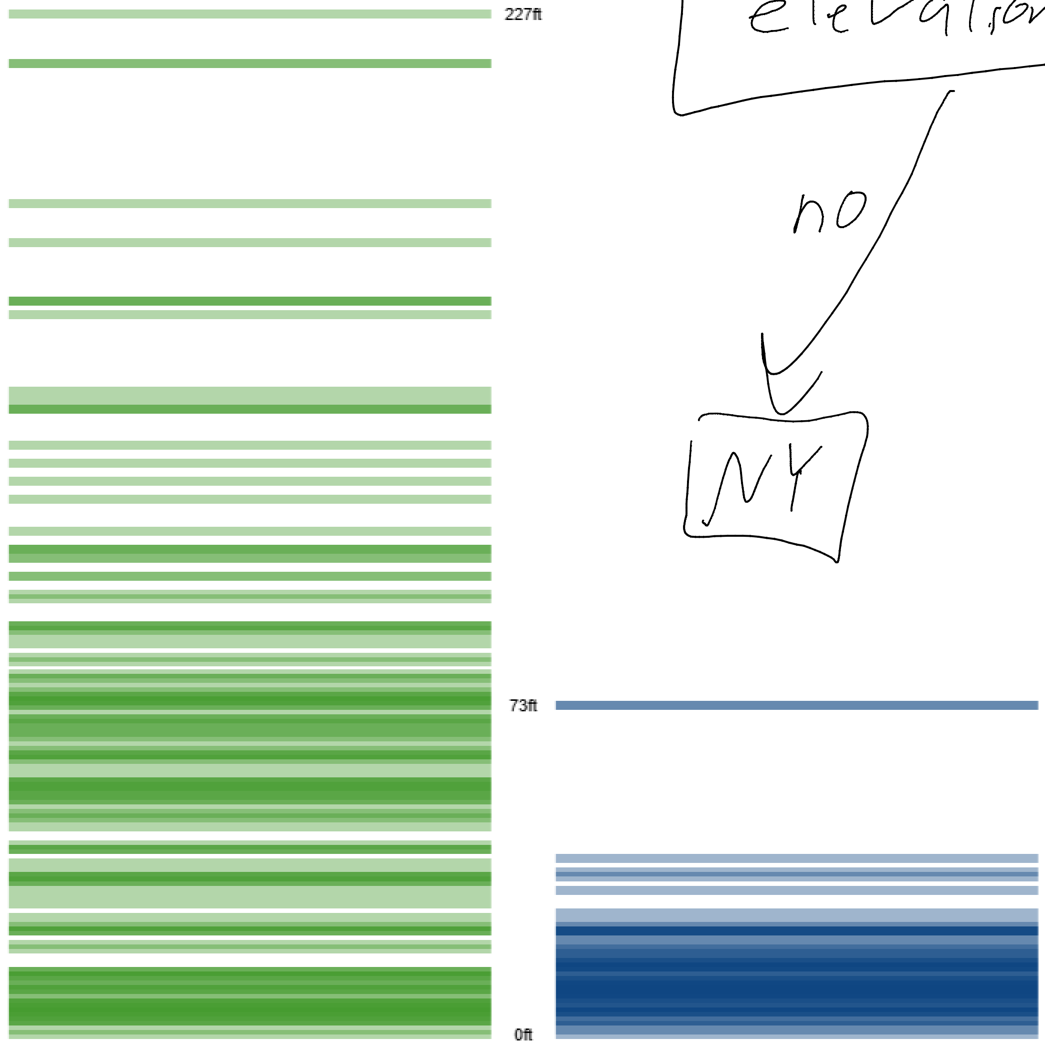
- We are given data from 248 homes.
- For each home/object, we have these features:
  - Elevation.
  - Year.
  - Bathrooms
  - Bedrooms.
  - Price.
  - Square feet.
- Goal is to build a program that predicts SF or NY.

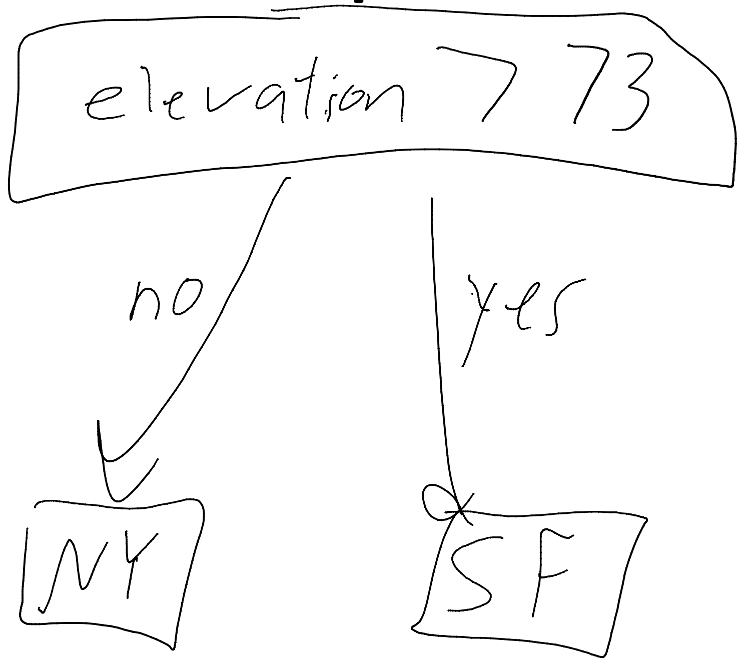
This example and images of it follow:
http://www.r2d3.us/visual-intro-to-machine-learning-part-1

Plotting Elevation

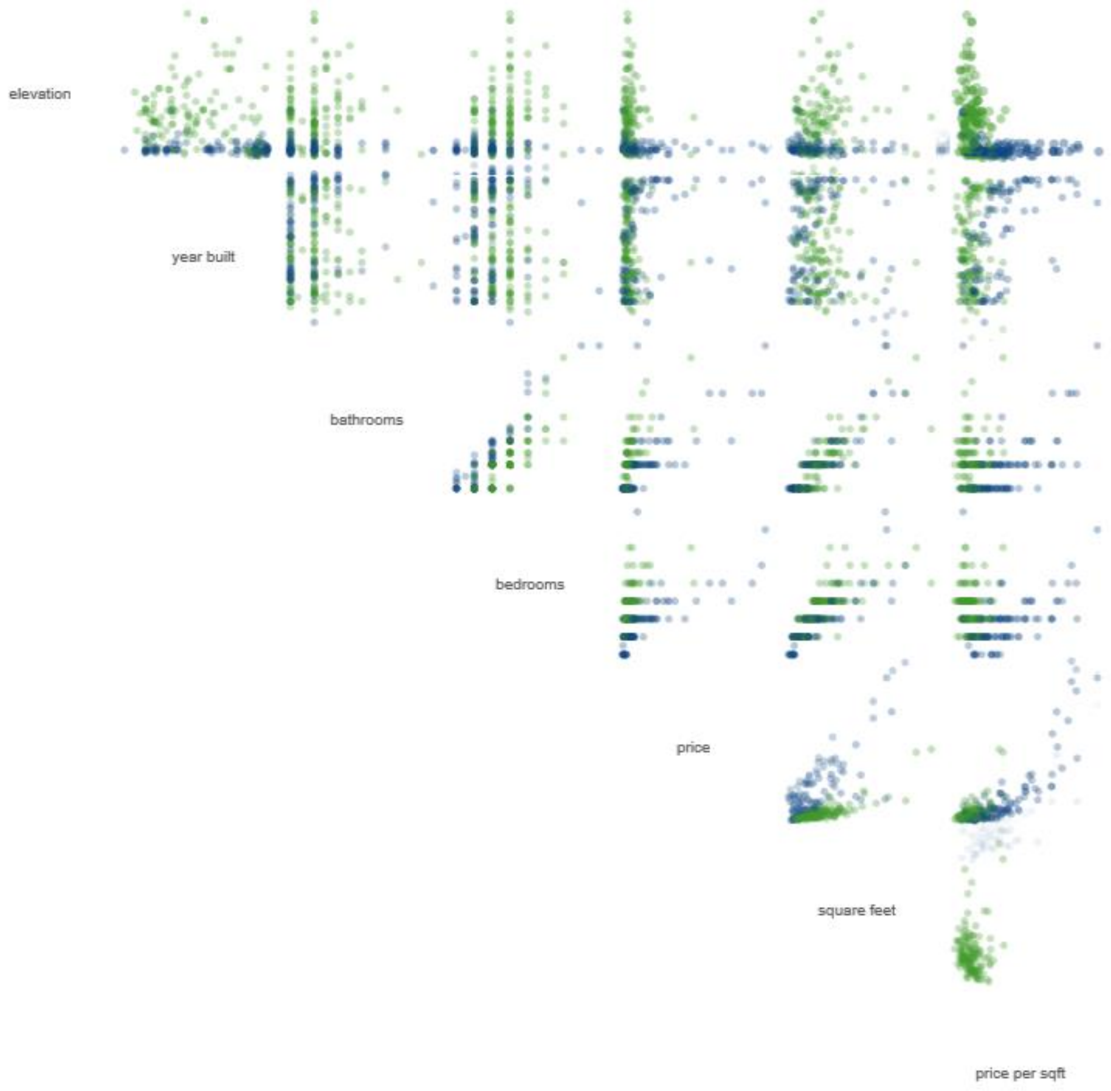


Simple Decision Stump

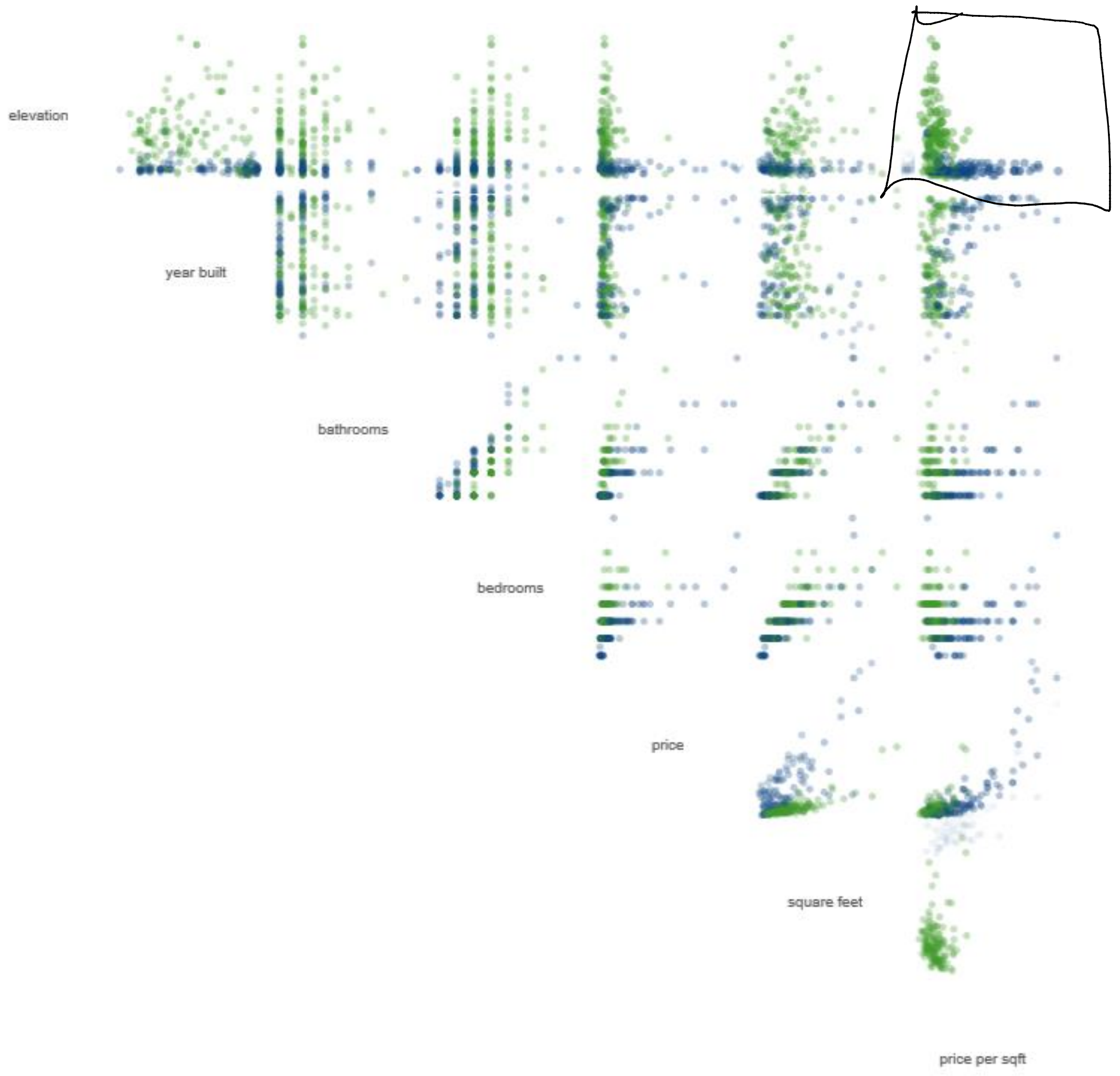




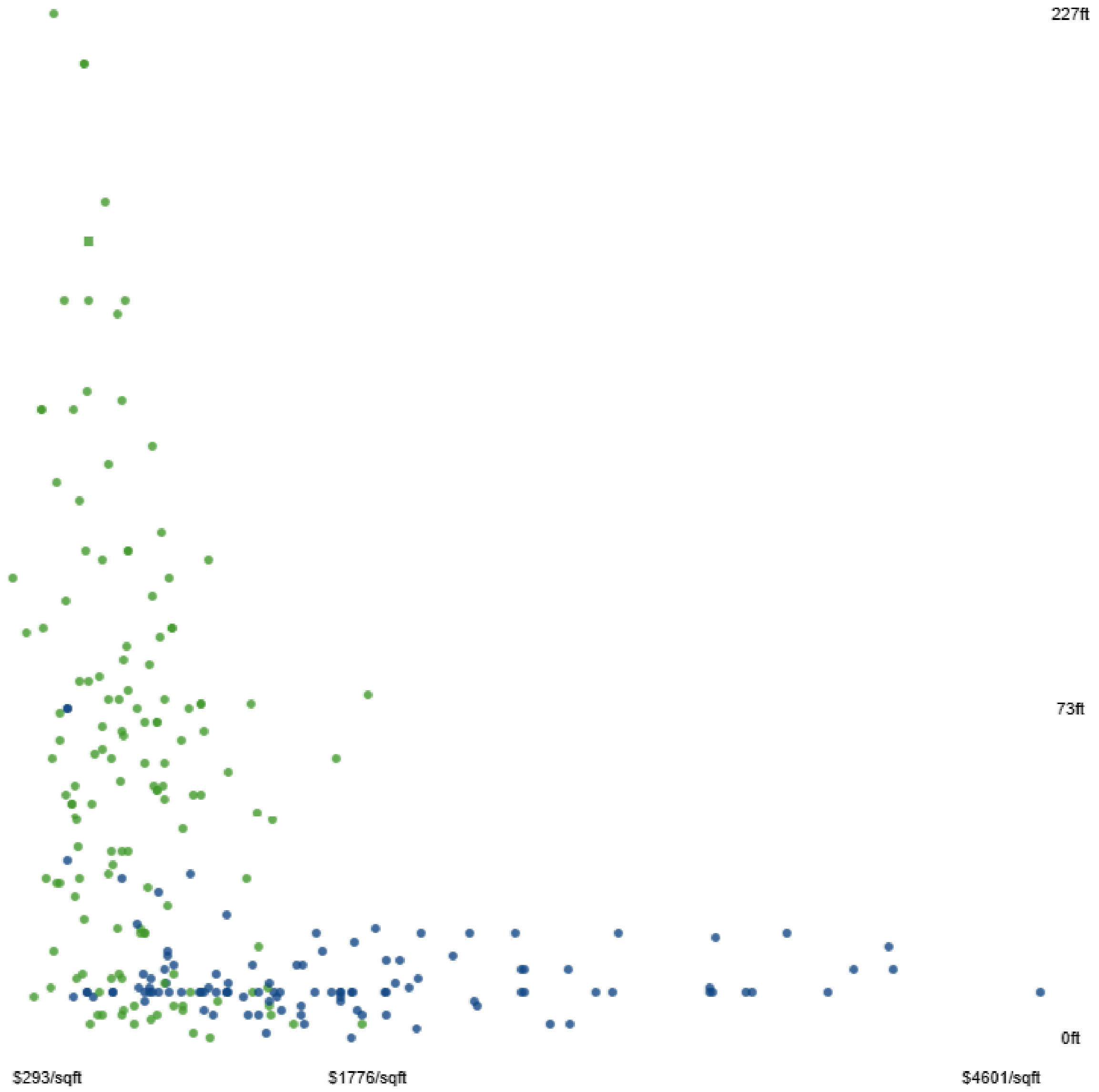
Scatterplot Array



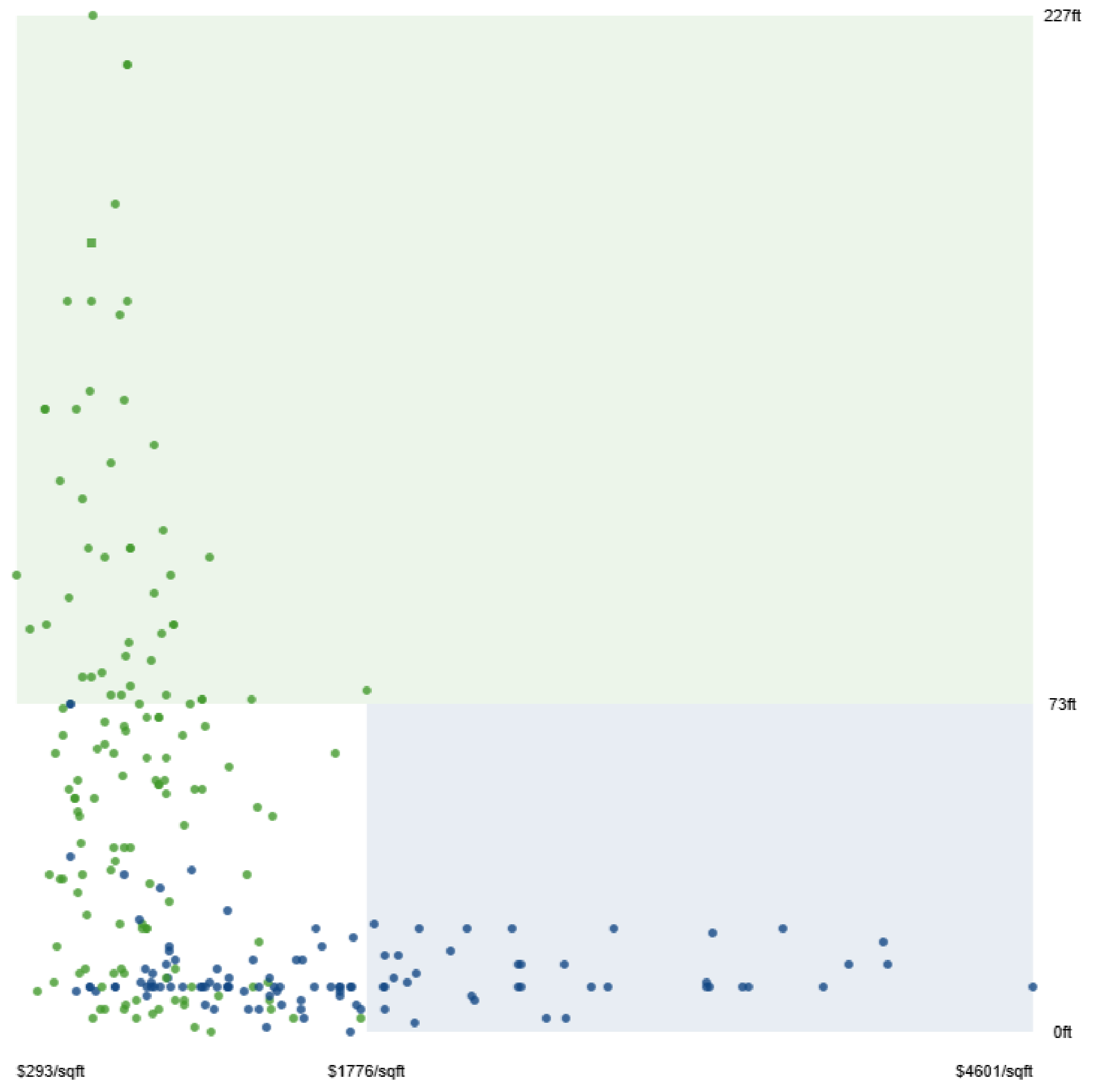
Scatterplot Array



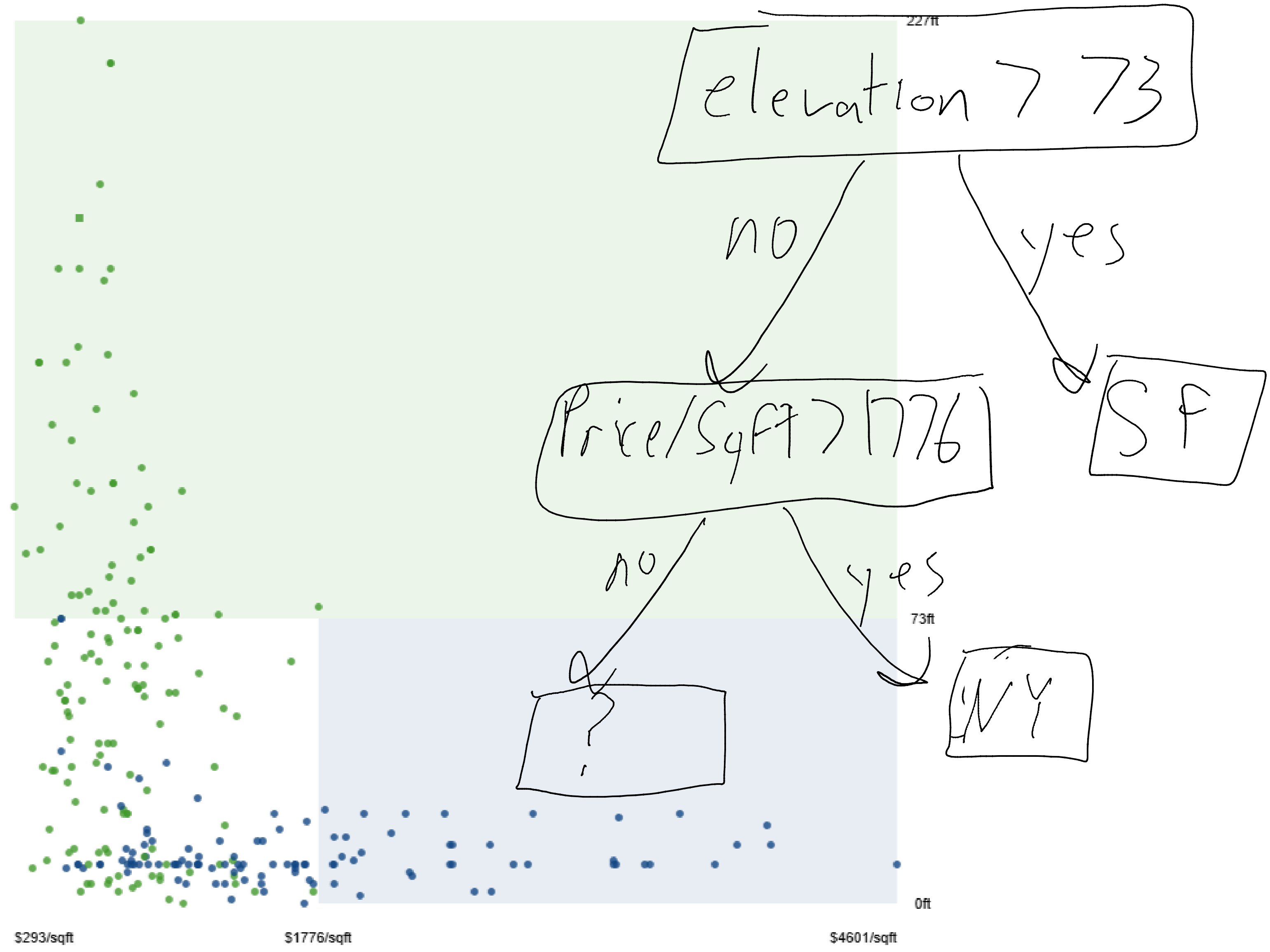
Plotting Elevation and Price/SqFt



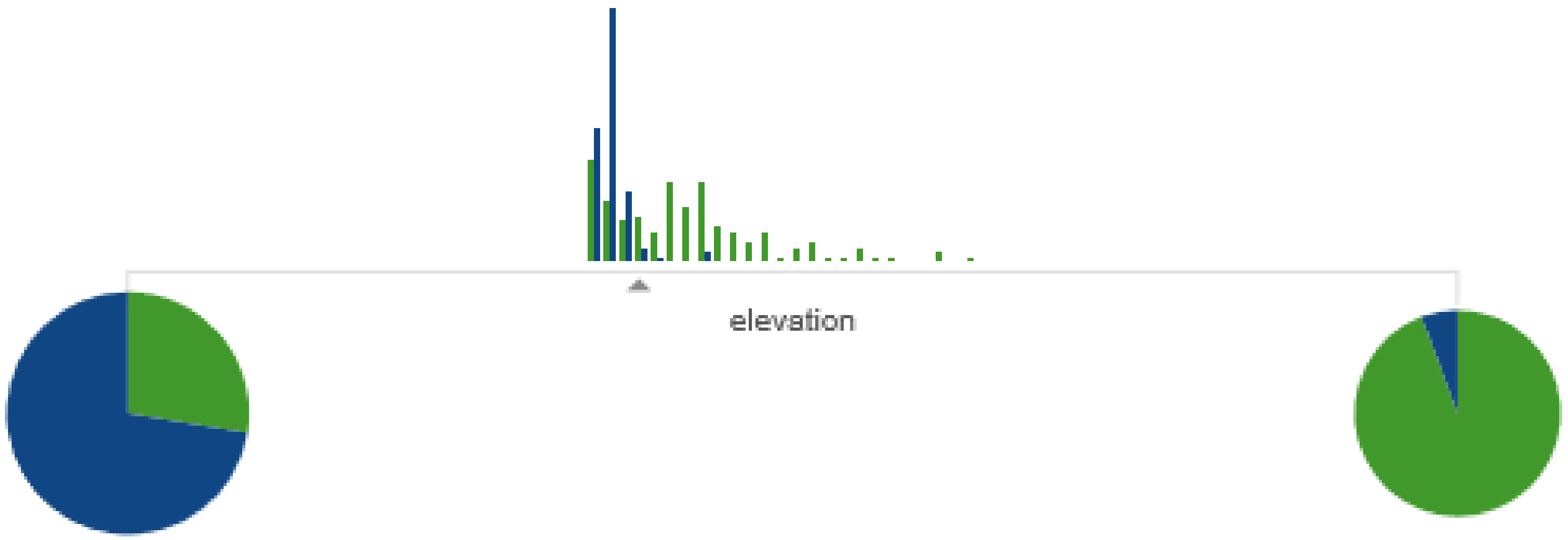
Simple Decision Tree Classification



Simple Decision Tree Classification

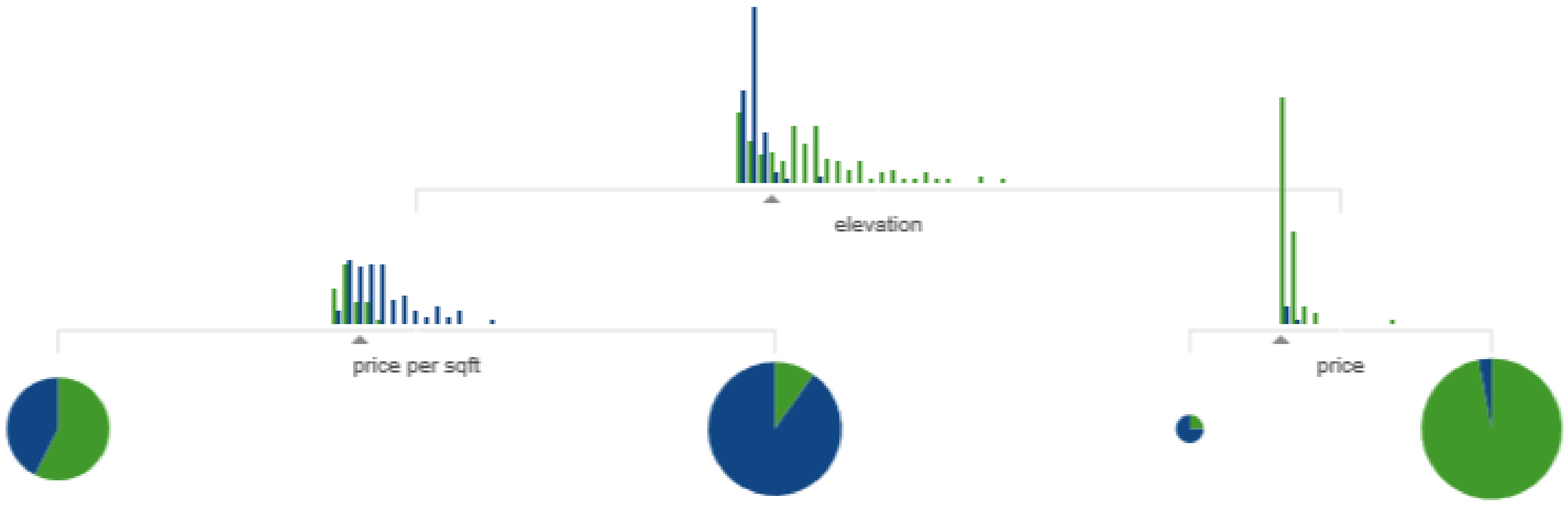


How does the depth affect accuracy?



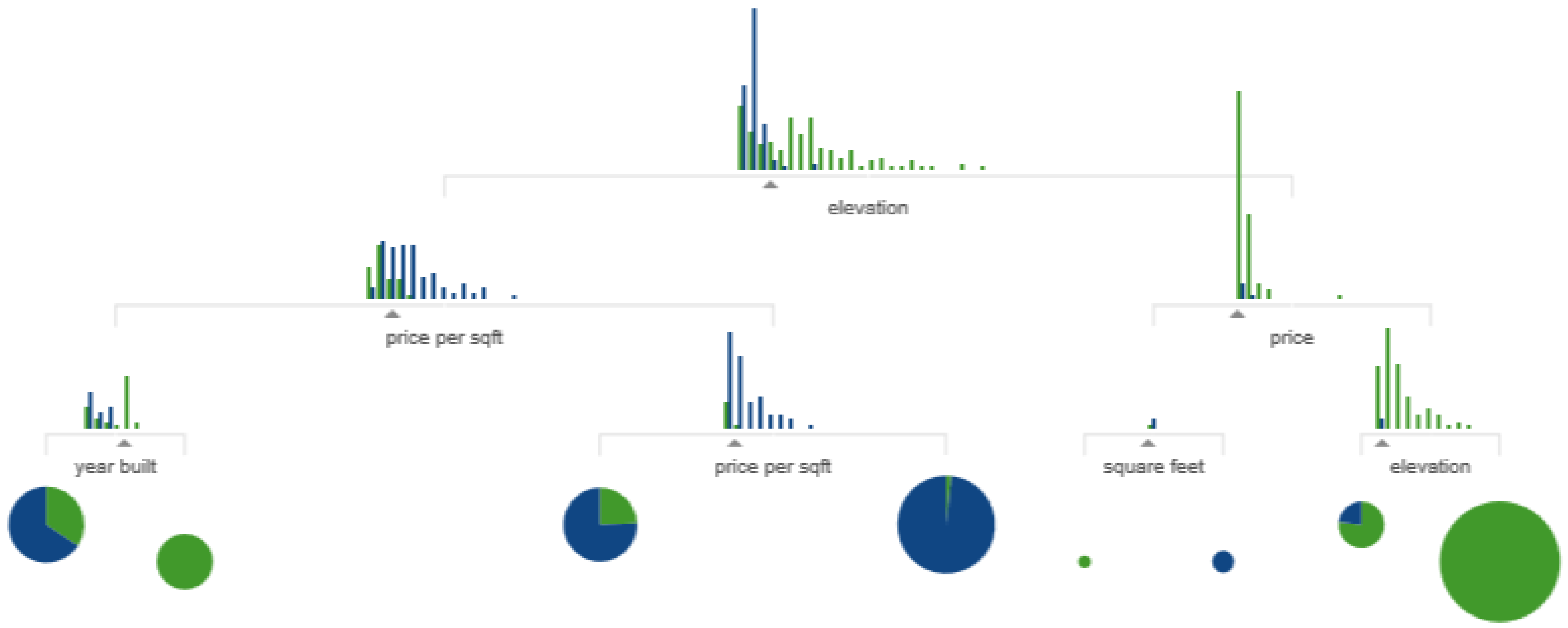
This is a good start (> 75% accuracy).

How does the depth affect accuracy?



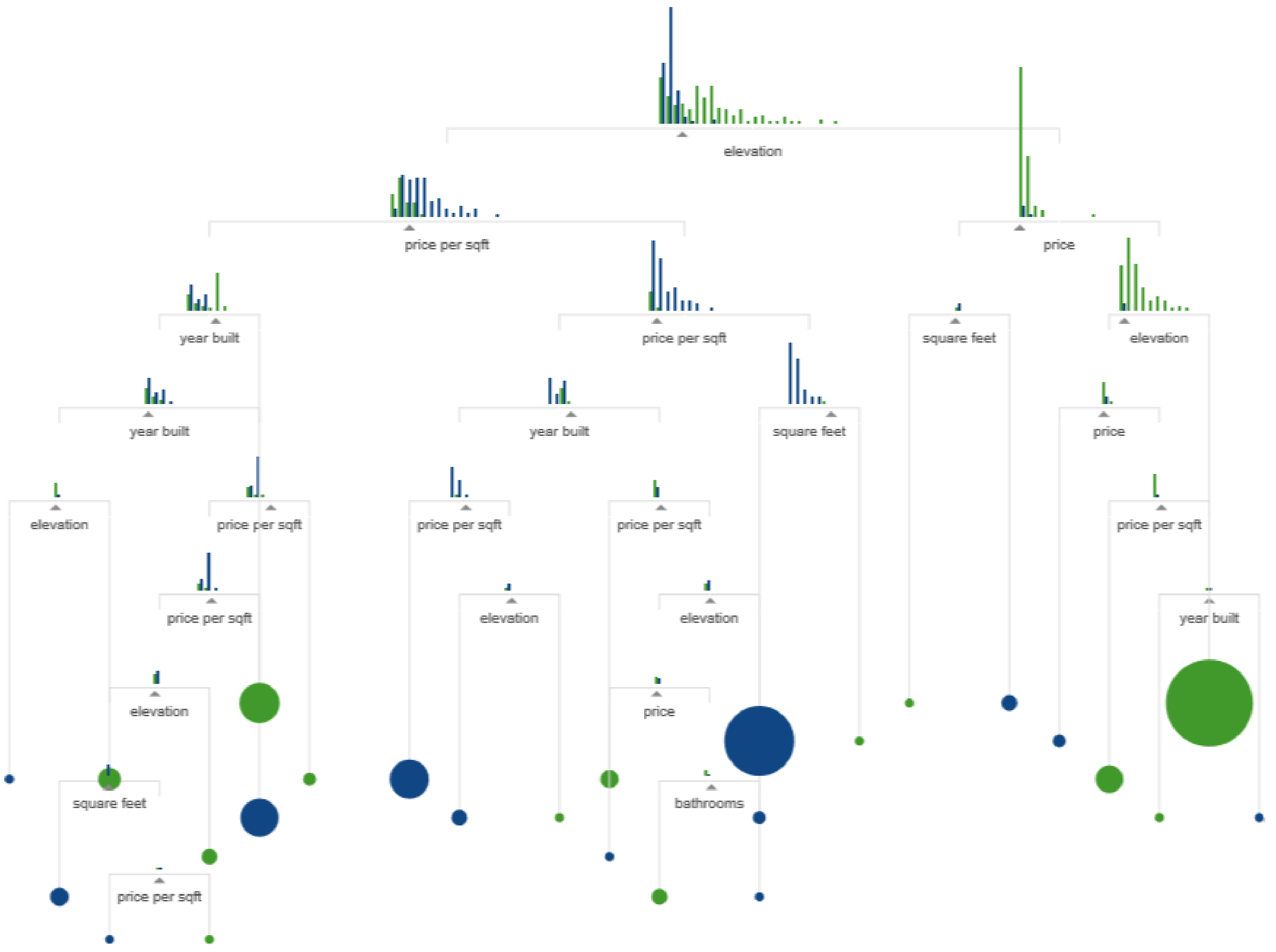
Start splitting the data recursively…

How does the depth affect accuracy?



Accuracy keeps increasing as we add depth.

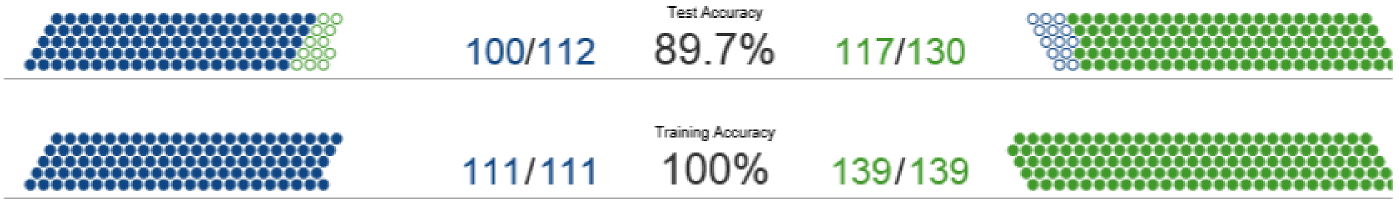
How does the depth affect accuracy?



Eventually, we can perfectly classify all of our data.

Training vs. Testing Error

- With this decision tree, ‘training accuracy’ is 1.
  - It perfectly labels the data we used to make the tree.
- We are now given features for 217 new homes.
- What is the ‘testing accuracy’ on the new data?
  - How does it do on data not used to make the tree?



- Overfitting: lower accuracy on new data.
- Our rules got too specific to our exact dataset.

Supervised Learning Notation

- We are given training data where we know labels:

X =

Egg	Milk	Fish	Wheat	Shellfish	Peanuts	…
0	0.7	0	0.3	0	0	
0.3	0.7	0	0.6	0	0.01	
0	0	0	0.8	0	0	
0.3	0.7	1.2	0	0.10	0.01	
0.3	0	1.2	0.3	0.10	0.01	

y =

Sick?
1
1
0
1
1

- But there is also testing data we want to label:

Xtest =

Egg	Milk	Fish	Wheat	Shellfish	Peanuts	…
0.5	0	1	0.6	2	1	
0	0.7	0	1	0	0	
3	1	0	0.5	0	0	

ytest =

Sick?
?
?
?

Supervised Learning Notation

- Typical supervised learning steps:
  1. Build model based on training data X and y.
  2. Model makes predictions 'yhat' on new data Xtest.

     $\hookrightarrow \hat{y}$

- Evaluation: is yhat similar to true unseen ytest?

Goal of Machine Learning

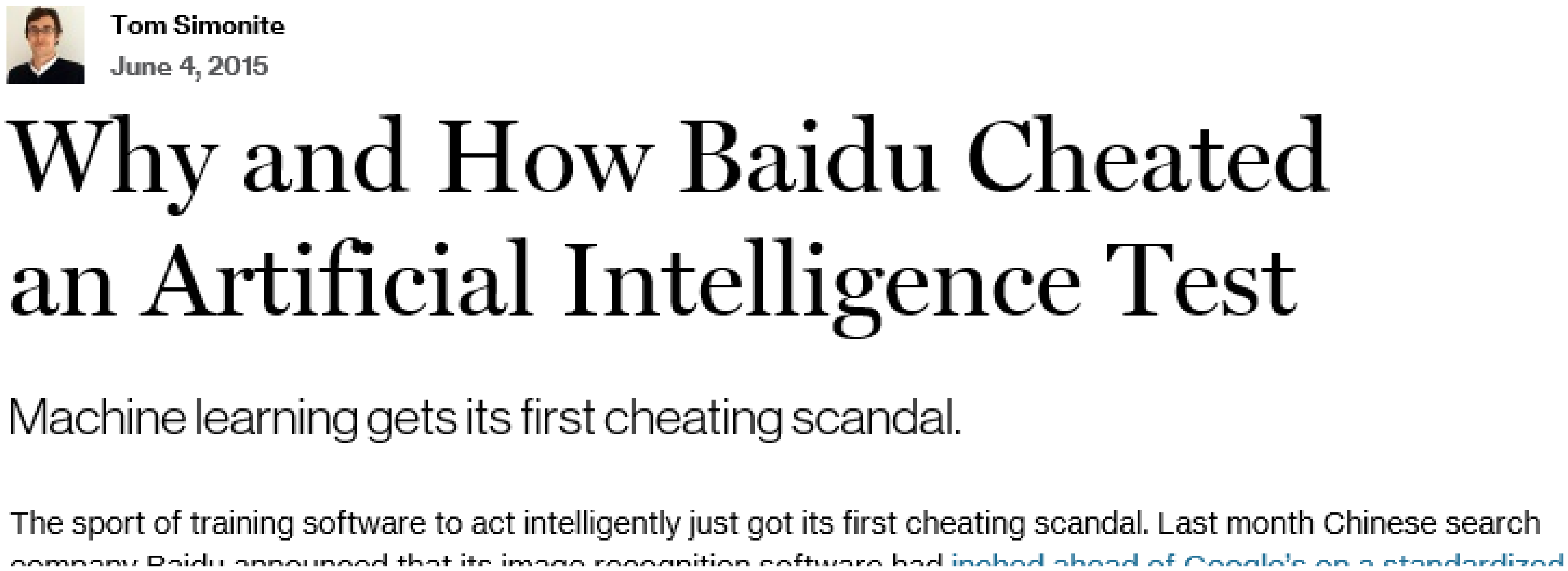
- In machine learning:
  - What we care about is the testing error!
- Midterm analogy:
  - The training error is the practice midterm.
  - The testing error is the actual midterm.
  - Goal: do well on actual midterm, not the practice one.
- Memorization vs learning:
  - Can always do well on training data by memorizing it.
  - Only learned if you can do well in new situations.

Golden Rule of Machine Learning

- Even though what we care about is test error:
  - YOU CANNOT USE THE TEST DATA TO BUILD MODEL.
- Minimizing test error is not the goal, but is a measure of how well we do on new data:
  - If used during training, doesn't measure this.
  - You can start to overfit if you use it during training.
  - Midterm analogy: you are cheating on the test.

Golden Rule of Machine Learning

- Even though what we care about is test error:
  - YOU CANNOT USE THE TEST DATA TO BUILD MODEL.



Tom Simonite

June 4, 2015

Why and How Baidu Cheated an Artificial Intelligence Test

Machine learning gets its first cheating scandal.

The sport of training software to act intelligently just got its first cheating scandal. Last month Chinese search company Baidu announced that its image recognition software had inched ahead of Google's on a standardized

http://www.technologyreview.com/view/538111/why-and-how-baidu-cheated-an-artificial-intelligence-test/

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.
- For IID data, probabilistic learning is possible.
- Field of learning theory explores this.
- Some keywords in learning theory:
  - Bias-variance decomposition, Hoeffding’s inequality and union bounds, sample complexity, probably approximately correct (PAC) learning, Vapnik-Chernovenkis (VC) dimension.

Fundamental Trade-Off

- Learning theory leads to fundamental trade-off:
  1. How small you can make the training error.

     vs.
  2. How well training error approximates the test error.

- Different models make different trade-offs.
- Simple models (like decision stumps):
  - Training error is good approximation of test error:
    - Not very sensitive to the particular training set you have.
  - But don't fit training data well.
- Complex models (like deep decision trees):
  - Fit training data well.
  - Training error is poor approximation of test error:
    - Very sensitive to the particular training set you have.

Validation Error

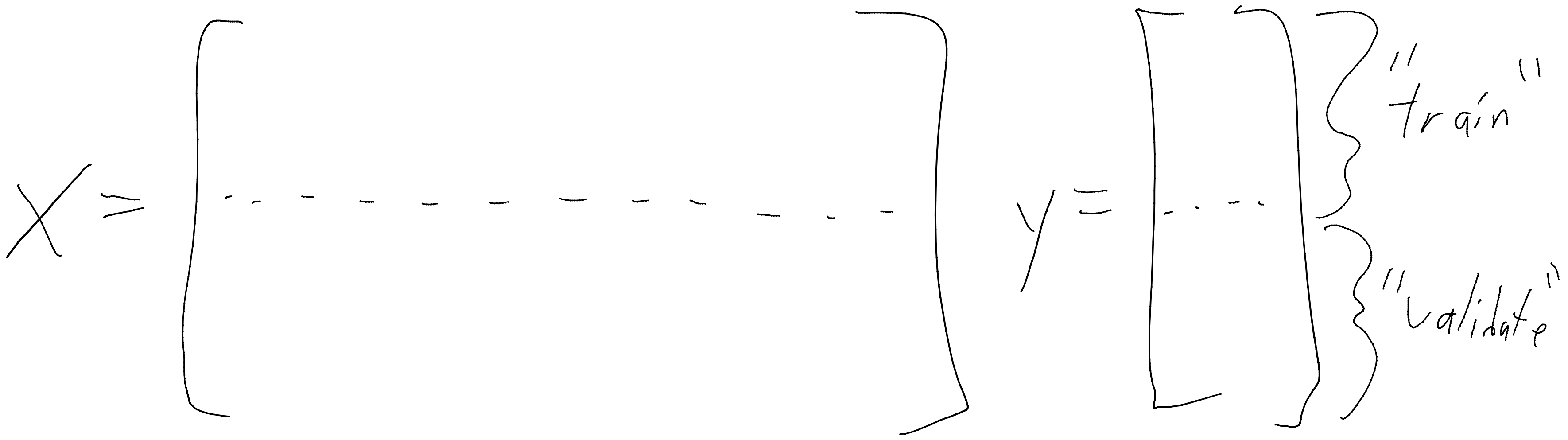
- How do we decide decision tree depth?
- We care about test error.
- But we can’t look at test data.
- So what do we do?????

- One answer:
  - Use part of your dataset to approximate test error.
- Split training objects into ‘train’ and ‘validation’:
  - Build model based on the training data.
  - Test model based on the validation data.

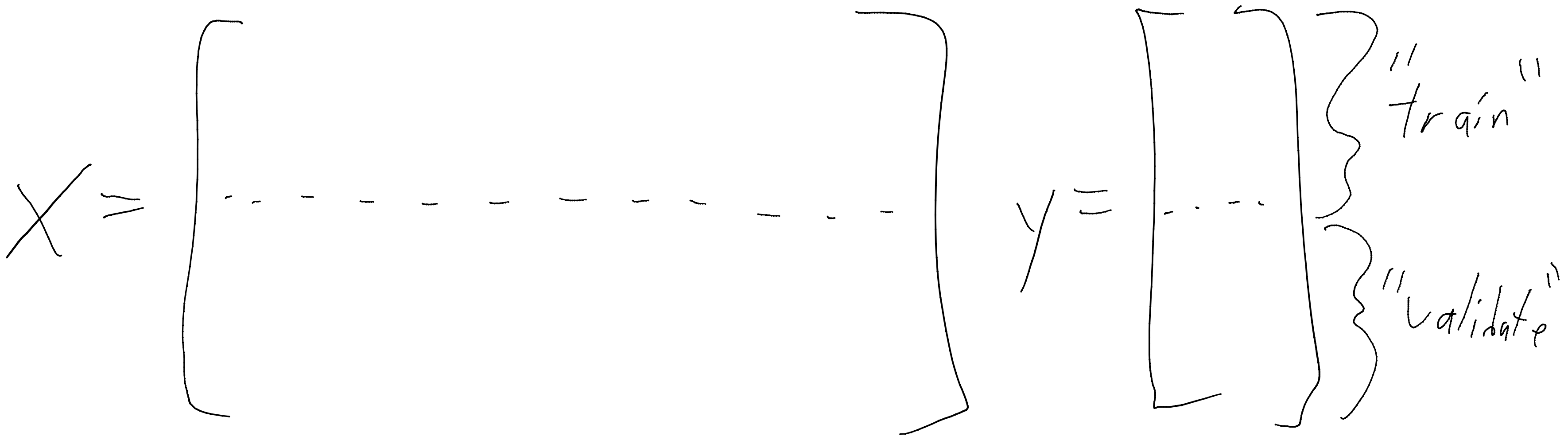
Validation Error

$$X = \begin{bmatrix} \quad & \quad & \quad \\ \quad & \quad & \quad \end{bmatrix} \qquad y = \begin{bmatrix} \quad \\ \quad \end{bmatrix}$$

Validation Error



Validation Error



1. $\text{model} = \text{fit}(X_{train}, y_{train})$
2. $\hat{y} = \text{predict}(\text{model}, X_{validate})$
3. $\text{error} = \text{sum}(\hat{y} \neq y_{validate})$

Validation Error

- Think of validation error as practice midterm.
- Validation error if used once:
  - unbiased approximation of test error.
- Not unbiased if used to pick between models:
  - Will approximate test error if you use it a few times.
  - Can overfit if you use validation error too many times.
  - It lets us “reset” our overfitting.

Fundamental Trade-Off with Validation Error

- Validation error for choosing decision tree depth:
  - At some depths, tree could have good validation error.
  - You only evaluate a few depths, so validation error likely approximates test error.
  - (Assuming big enough validation set.)

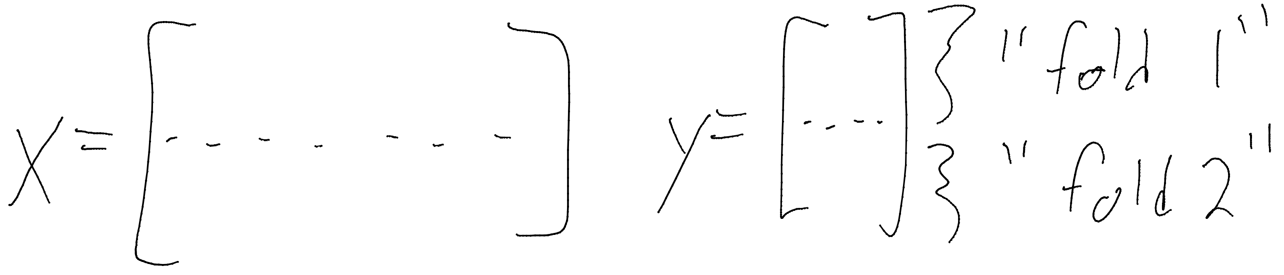
- Compare to training error:
  - At deep depths, you always have good training error.
  - But deep decision trees search over so many rules that you are likely to achieve low training error by chance.
  - This means model with best training error doesn’t say much about test error.

Cross-Validation

- Isn’t it wasteful to only use part of your data?
- 2-fold cross-validation:
  - Train on half of the data, validate on other half.
  - Switch halves and repeat.
  - Average score.

Cross-Validation

- Isn't it wasteful to only use part of your data?
- 2-fold cross-validation:
  - Train on half of the data, validate on other half.
  - Switch halves and repeat.
  - Average score.



- train on on fold 1, test on fold 2.
- train on on fold 2, test on fold 1.
- average errors.

Cross-Validation

- Isn’t it wasteful to only use part of your data?
- 2-fold cross-validation:
  - Train on half of the data, validate on other half.
  - Switch halves and repeat.
  - Average score.
- 10-fold cross-validation:
  - Train on 9/10 of the data, validate on other 1/10.
  - Repeat 10 times with different validation part.
- Leave-one-out cross-validation:
  - Traing on (n-1) objects, validate on the other one.
  - Repeat n times.
- Once you pick depth, could re-train on full data.
- If data is not independent, use random folds.

Back to Decision Trees

- You can use CV to select decision tree depth.
- You can also use to decide whether to split:
  - Don’t split if CV error doesn’t improve.
- Or fit deep decision tree and use CV to prune:
  - Remove leaf nodes that don’t improve CV error.

- Popular implementations that have all of these:
  - C4.5, CART.

Cross-Validation Theory

- Cross-validation allows using more of the dataset to estimate validation error.
- Does CV give unbiased estimate of test error?
  - Yes: each data point is only used once in validation.
  - But again, that’s assuming you only do CV once.
- What about variance of CV?
  - Hard to characterize.
  - Variance of CV on ‘n’ data points is worse than if we had a validation set of size ‘n’.
  - But we believe it close!

Summary

- Training vs. testing.
- Golden rule and fundamental trade-off.
- [Cross-]validation to improve the trade-off a bit.

- Next time:
  - No free lunch theorem.
  - E-mail spam filtering.
  - Probabilistic models.