Selecting An Approach

- Unstructured Data (text, images, video, sound)

  DL

- Tabular Data: classical ML methods

  - do you know the functional form?

    GLM

  - do you need a simple model? GLM
    Decision trees

  - are you interested in inference? GLM

  - are you interested in prediction? tree-based ensembles

  But, always start simply.

Non Parametric Models

- More flexible, "data driven"
- Don't depend on a functional form w/ a fixed number of parameters $y = \beta_0 + \beta_1 x_1 + \dots$
- need more data
- ex: decision trees, RF, GBM

Example

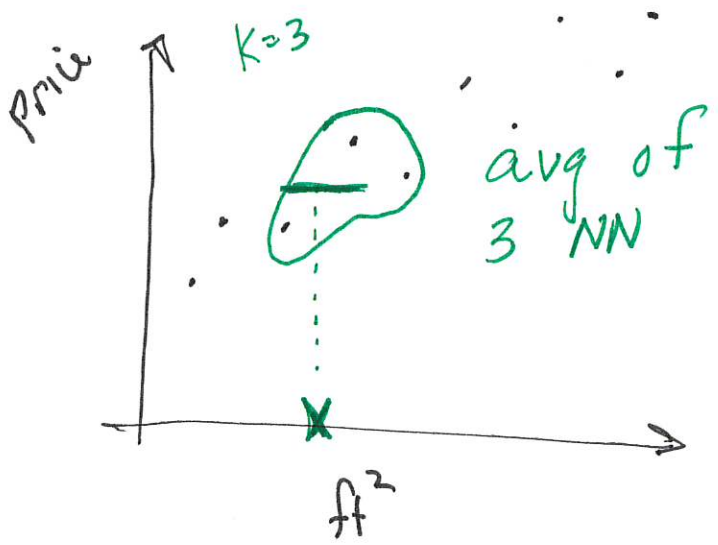
~~Sq~~ Ft^2	Price
700	$1M
850	$1.1M
1000	$1.2M
1100	$1.15M
1300	$1.25M
1325	$2M
⋮	

Ask: What if I knew the ft^2, but not the price, how can I predict it?

$ft^2 = 1100$

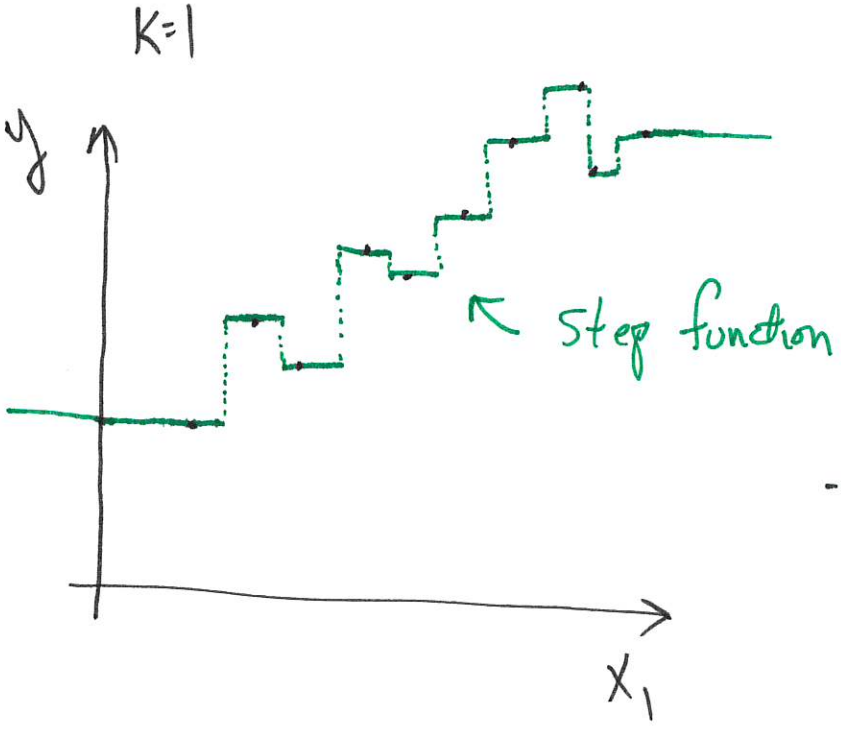
$ft^2 = 1070$

KNN



Choose K NN
using Euclidean distance.

- Reg + Class.
- Need lots of data
- Slow to fit
- Normalize features
- Find suitable K



Don't extrapolate well.

- Segmented our data into regions using features and distance.

What if we partition our data using a loss function?

Decision Trees

- Partition our data into high-dimensional rectangular regions.
- If an obs. falls in a region, use training set y-value to make a prediction.

($\bar{y}$ or mode(y))

- Not using distance ⇒ features can be unnormalized.
- Handle categorical features.

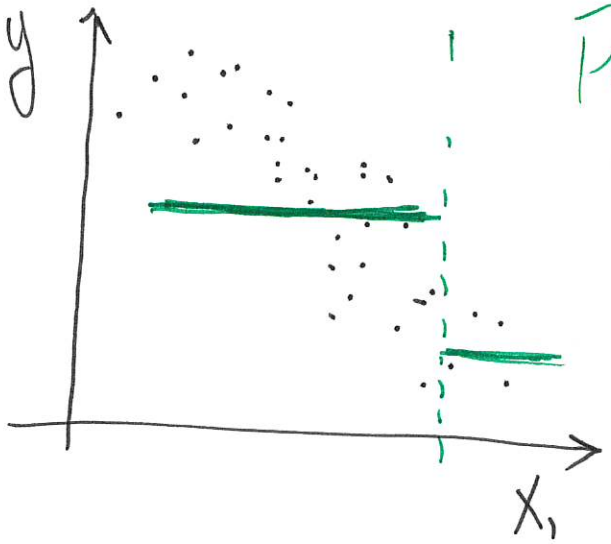
Rule-Based Systems

If bdrms < 3
  And bath < 2
  And city = Pacifica
    Then Price = $900k

If ...

- Partition data using binary splits of our features

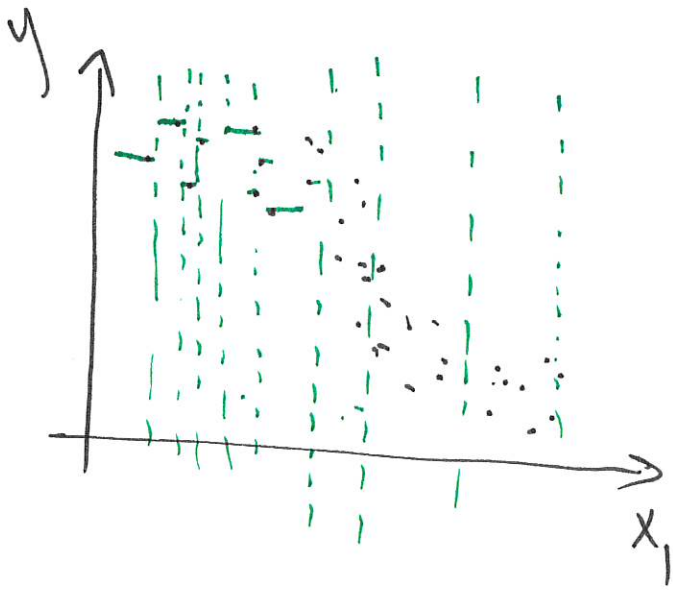
- Choose splits s.t. y-values in a region are as "alike" as possible.



Predict using $\bar{y}$

Not good split!

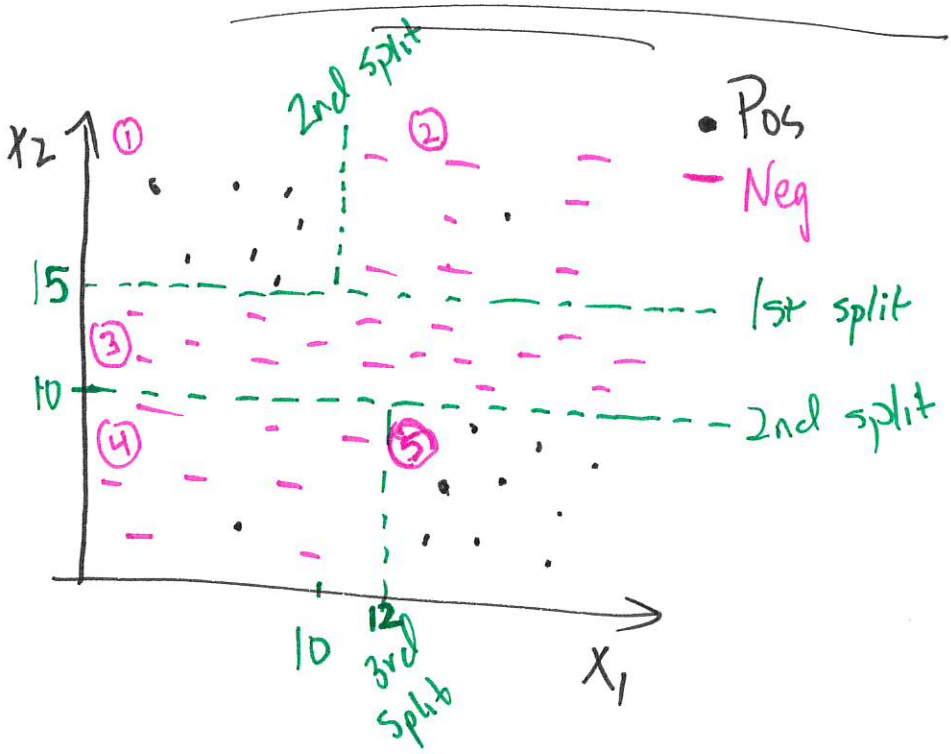
High MSE



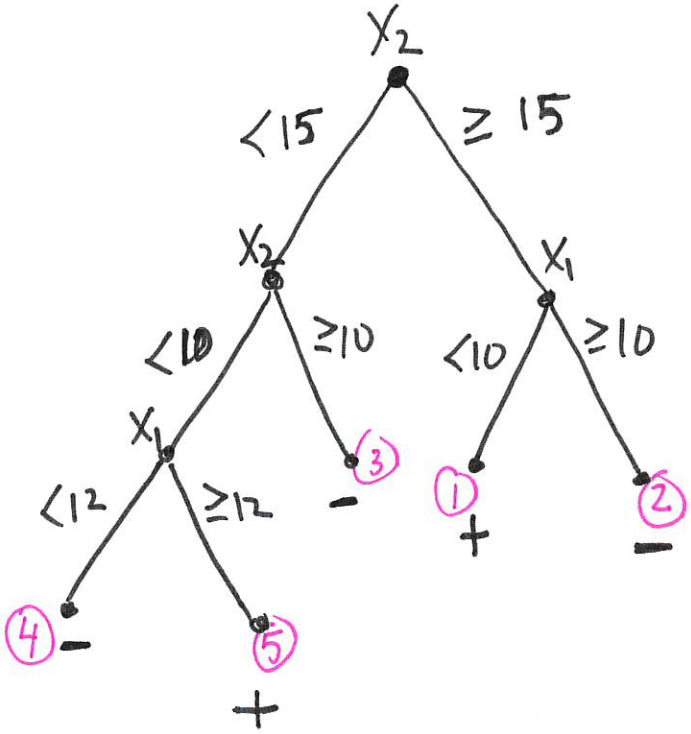
If you keep going you might overfit.

High variance.

2D Example (Classification)



Tree Structure



- Top-down greedy algorithm

- Nodes
  Parent + child
- Decision nodes - nodes w/a split
- Leaf nodes: prediction happens

Finding The Best Splits

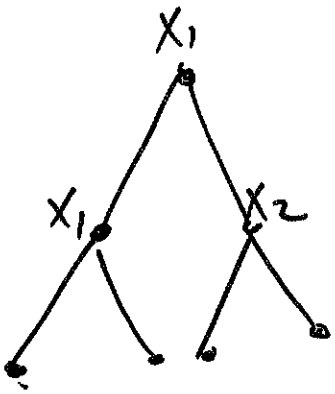
For each feature:

  For each value in the region:

    Split the data & calculate the loss

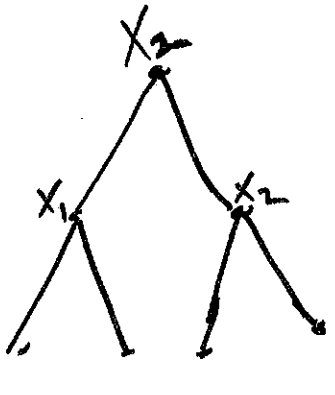
Return feature and split point w/ smallest loss.

Do this recursively until some stopping criteria.



MSE = 2

Greedy



MSE = 1.8

Non-greedy