Decision trees

Terence Parr
MSDS program
University of San Francisco

UNIVERSITY OF SAN FRANCISCO

Basic properties of decision tree models

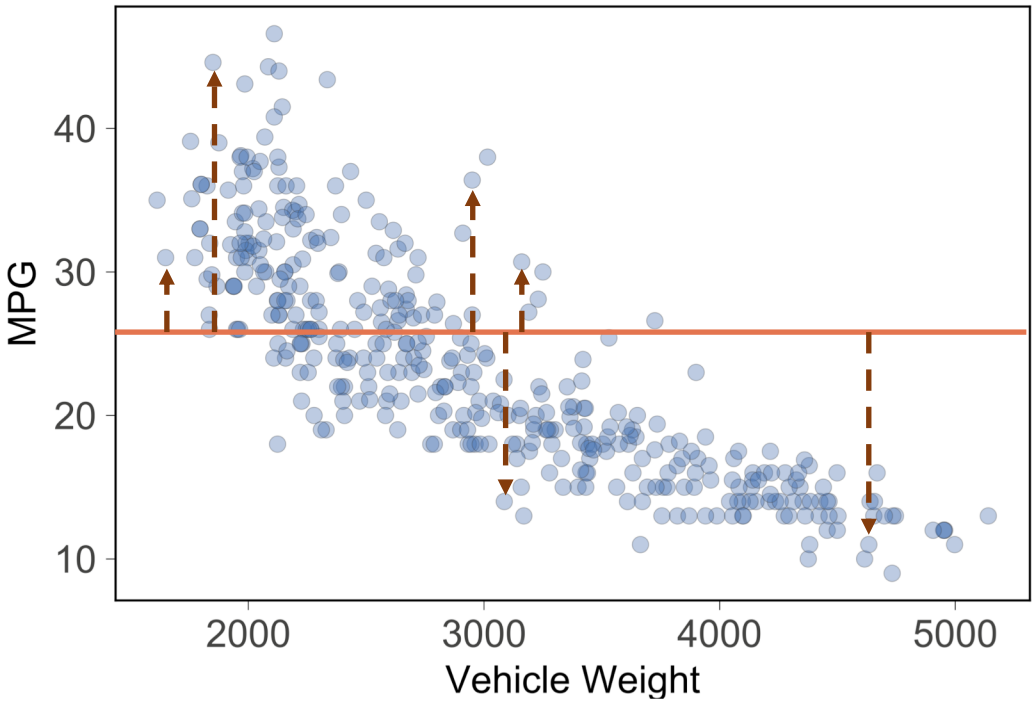
- Decision trees consist of internal decision nodes and leaf nodes that make predictions
- Each input record (feature vector) associated w/exactly one leaf
- Each leaf has 1 or more records whose y is same class for classifier or similar value for regressor
  (model hyperparameters change this property)
- Prediction proceeds from root to leaf, testing var/value combos
- Regressor: leaf predicts average of y for associated records
- Classifier: leaf predicts mode (most common) class

Let’s reinvent decision trees

UNIVERSITY OF SAN FRANCISCO

Let's invent a simple regressor in 1D

Predict MPG from weight



Predict a single constant for entire region (the mean)

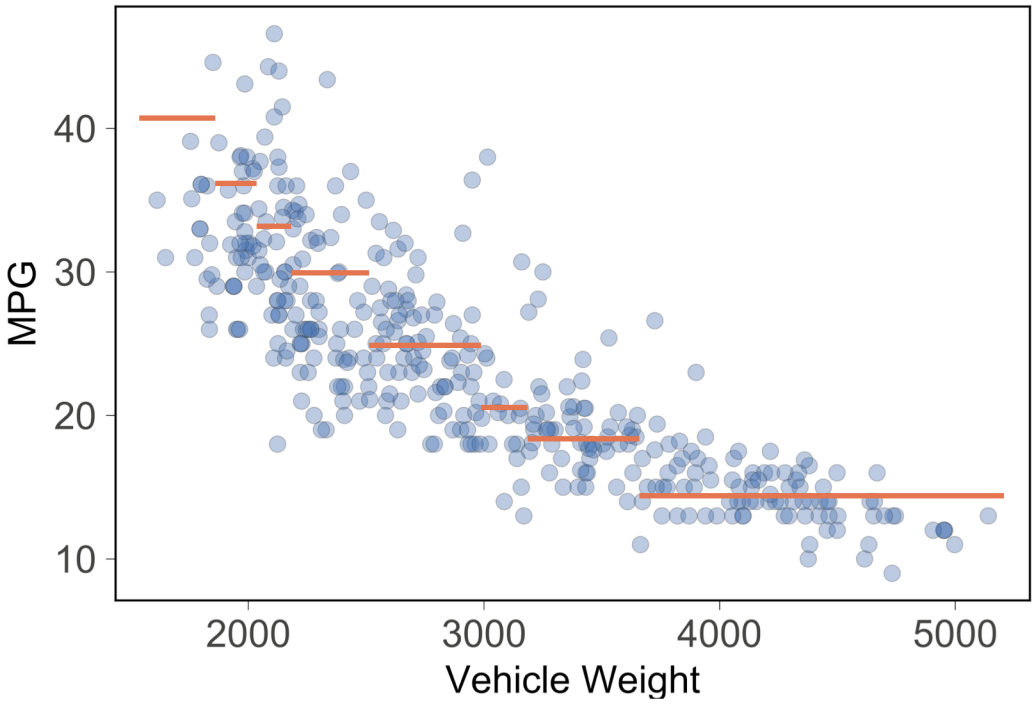
MSE is high

Residuals from point to region mean

Improve by partitioning, using multiple lines

WHERE DO WE SPLIT??

Find groups of similar MPG values, which yields a lower MSE

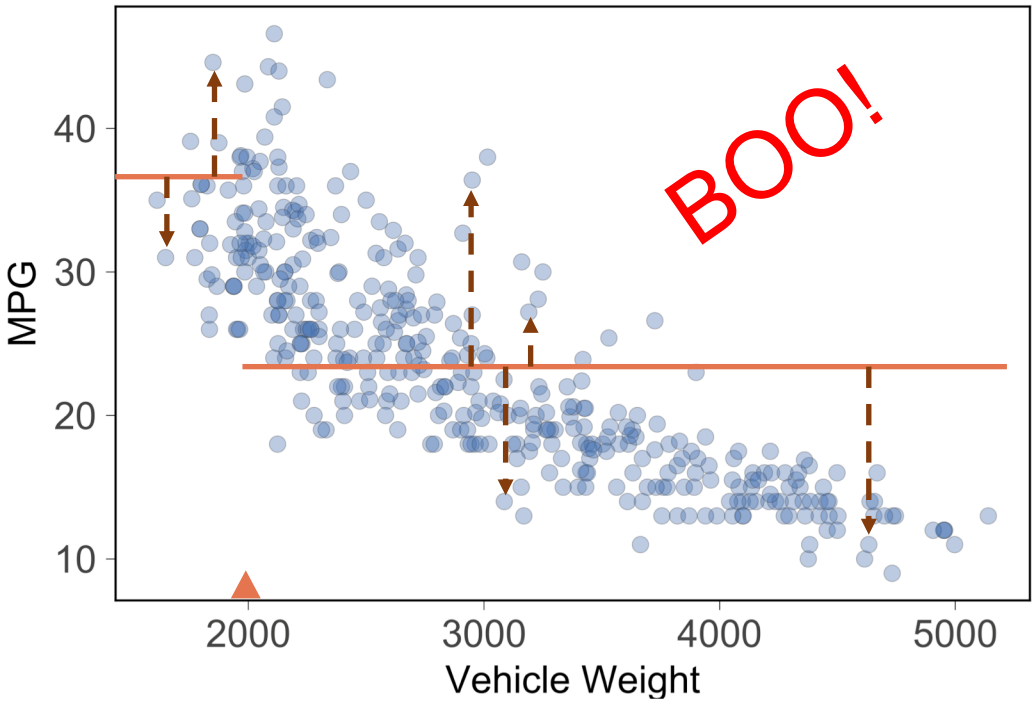


Can only use horizontal lines, but can use lots

Each region predicts mean in piecewise fashion

Strategy: find split point giving least MSE

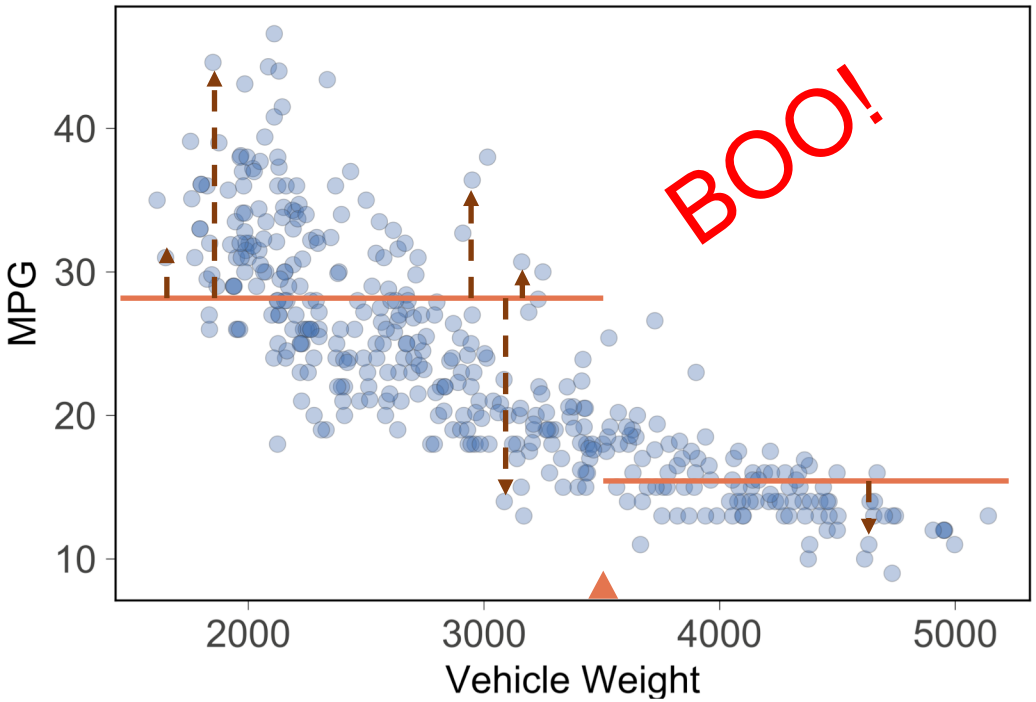
Split region into two subregions, each predicting mean



X=2000 is BAD CHOICE: MSE very high

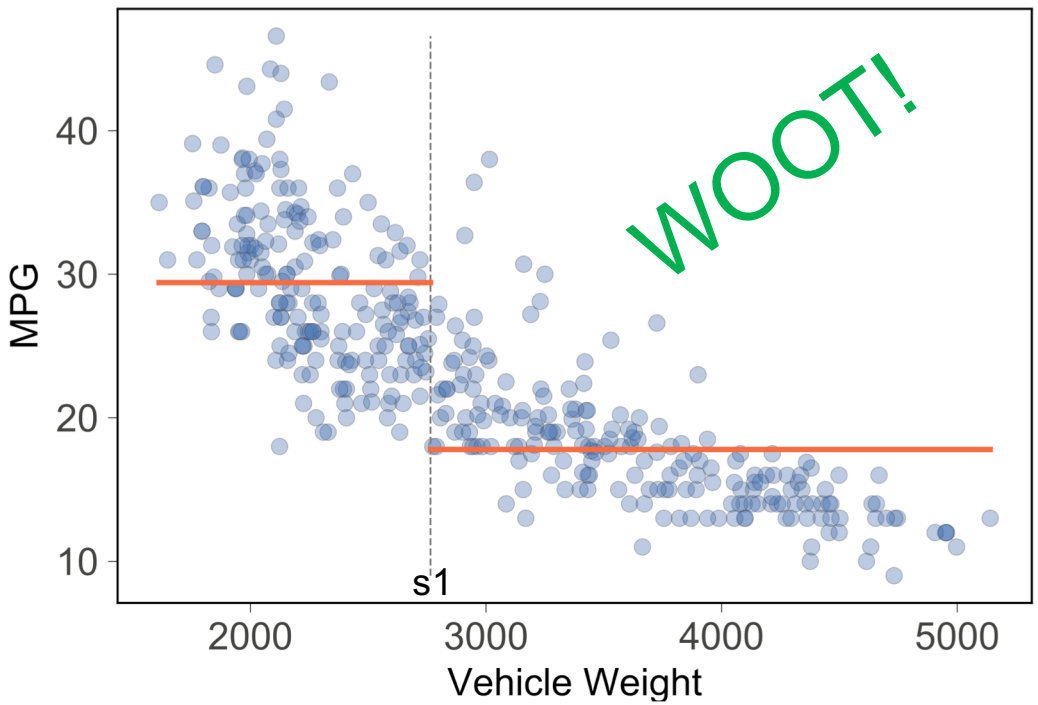
Strategy: find split point giving least MSE

Split region into two subregions, each predicting mean



X=3500 is ANOTHER BAD CHOICE: MSE very high

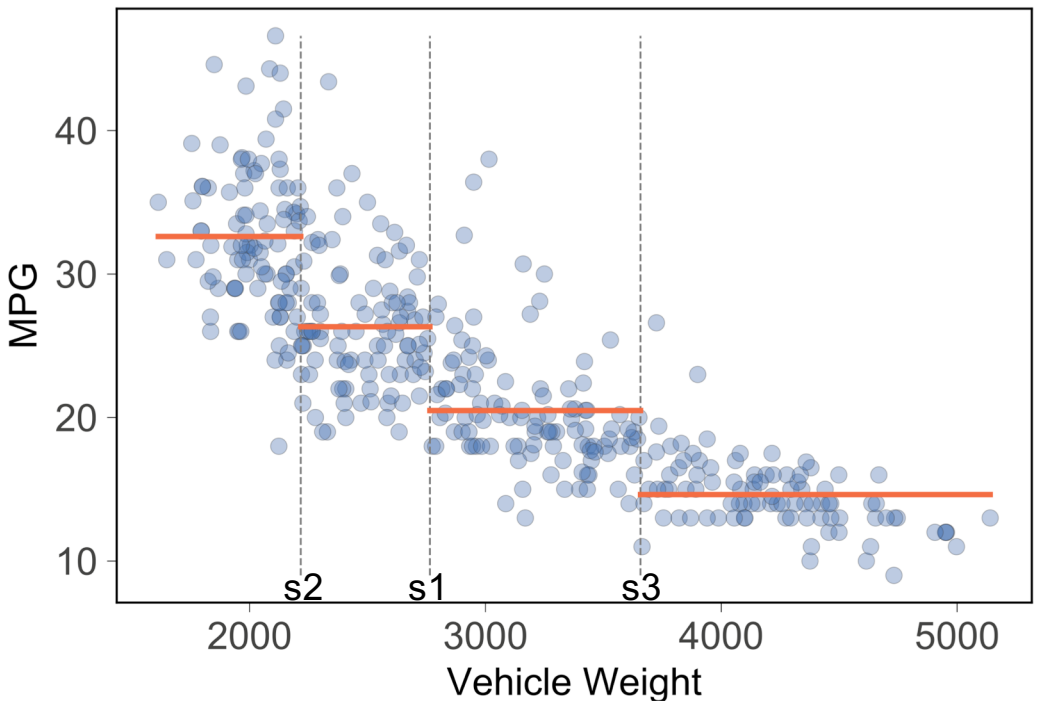
A split exists that gives min MSE for regions



Technique: Exhaustively check all feature values, computing MSE or variance for each split

Track split point giving min MSE

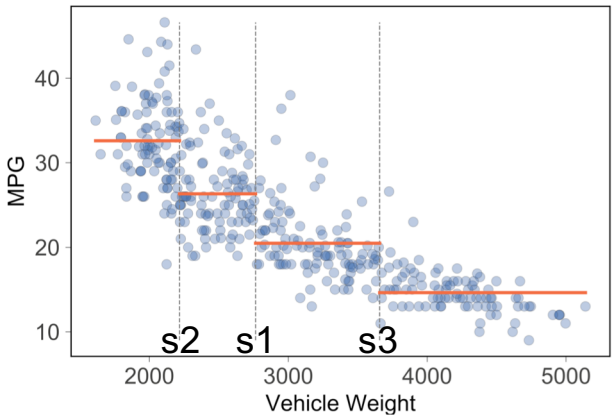
Now split those 2 regions into 4 regions

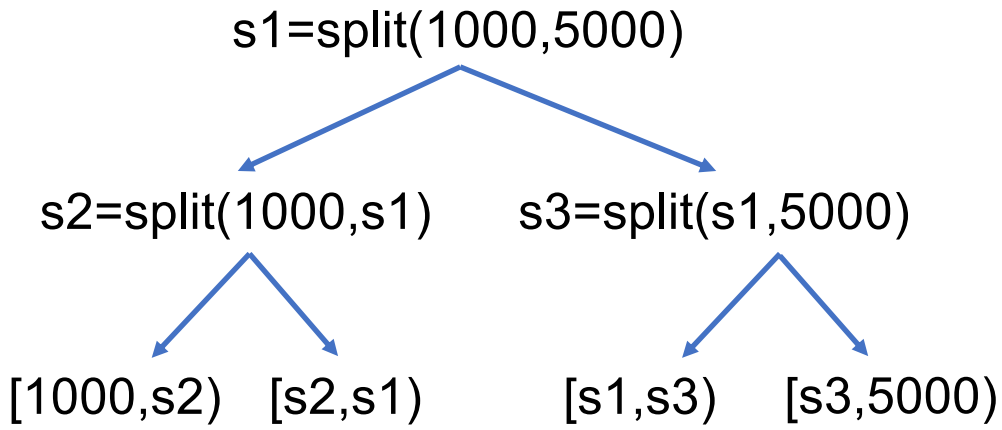


Split s1 stays, *recursively* split left/right regions to get splits s2, s3

Kinda like binary search or other divide-and-conquer strategy

Model training recursion tree gives regions defined by splits s1,s2,s3



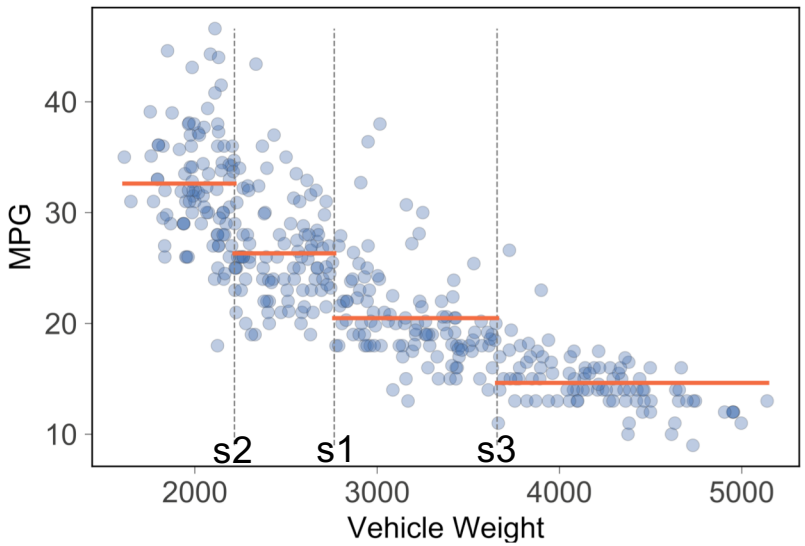


Split (recurse) until:

- All potential splits do not reduce MSE
- All nodes have min num samples
- Max number of splits reached
- Etc…

Predictions are avg of MPG (target) values in regions

Hardcoded model implementation

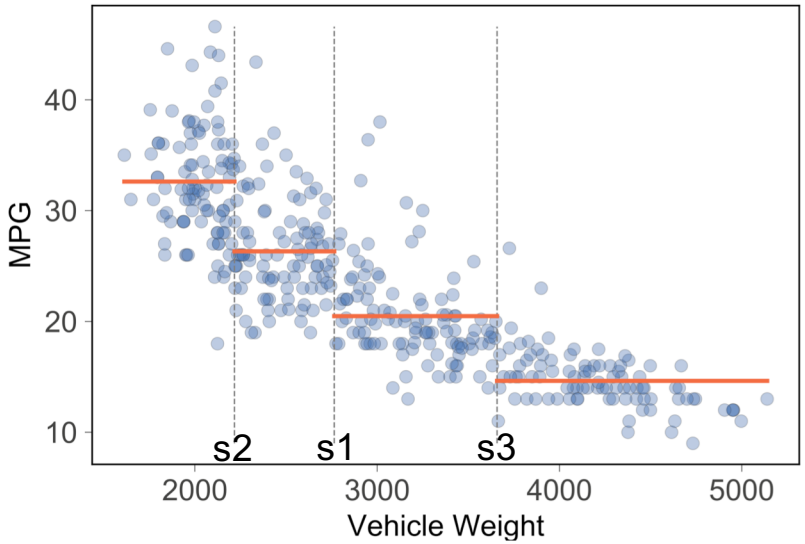


To partition space, test in recursion/split order

```
if x<s1 and x<s2: predict 32.6
if x<s1 and x>=s2: predict 26.3
if x>=s1 and x<s3: predict 20.5
if x>=s1 and x>=s3: predict 14.6
```

Note repeated comparisons!

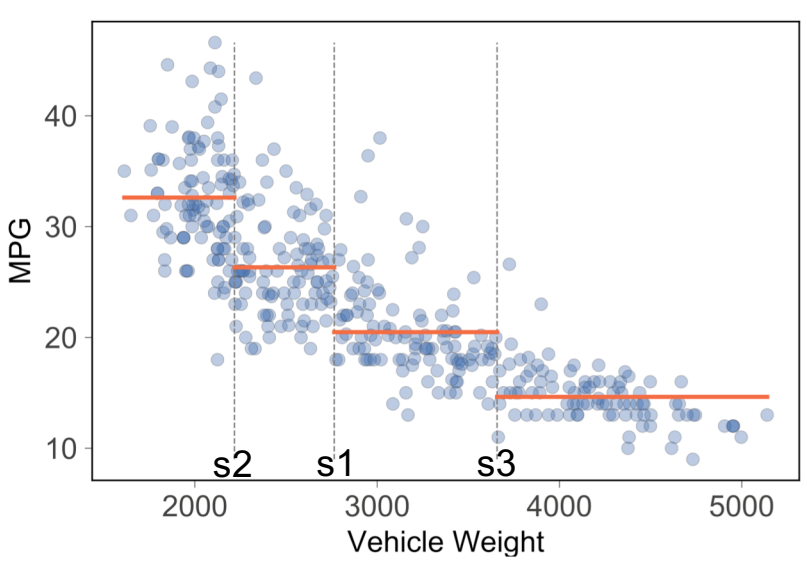
Factor the split comparisons for efficiency

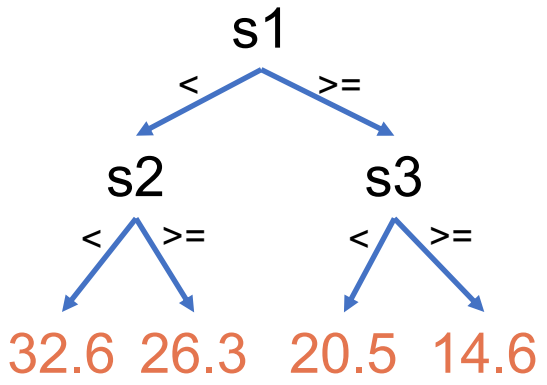



```
if x<s1:
  if x<s2: predict 32.6
  else: predict 26.3
else:
  if x<s3: predict 20.5
  else: predict 14.6
```

But, don't want to hardcode model!

Represent nested conditionals as regression decision tree

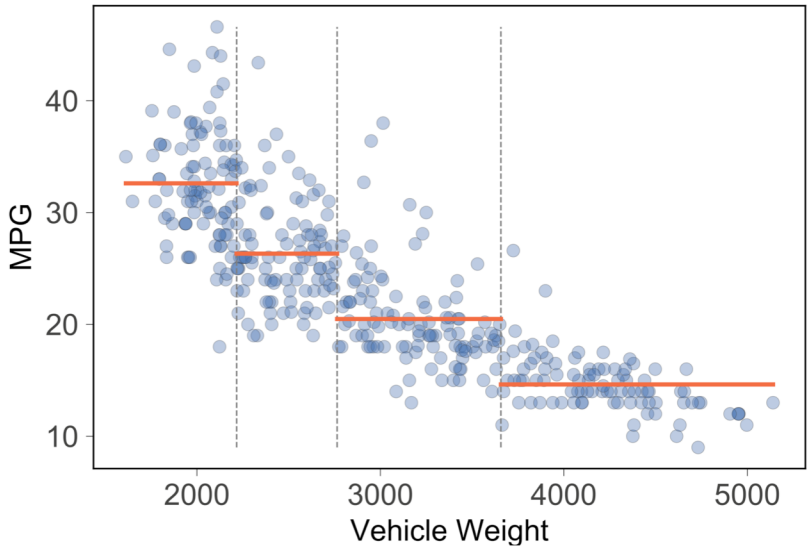


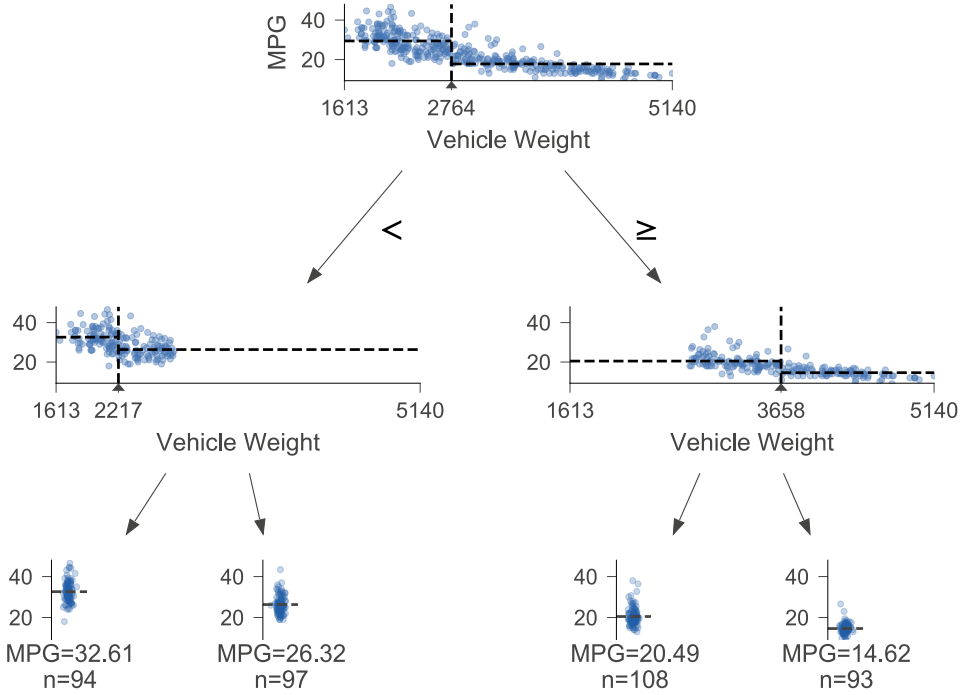


Internal nodes test features
Leaves predict region mean

Morph tree of recursion from training into decision tree!

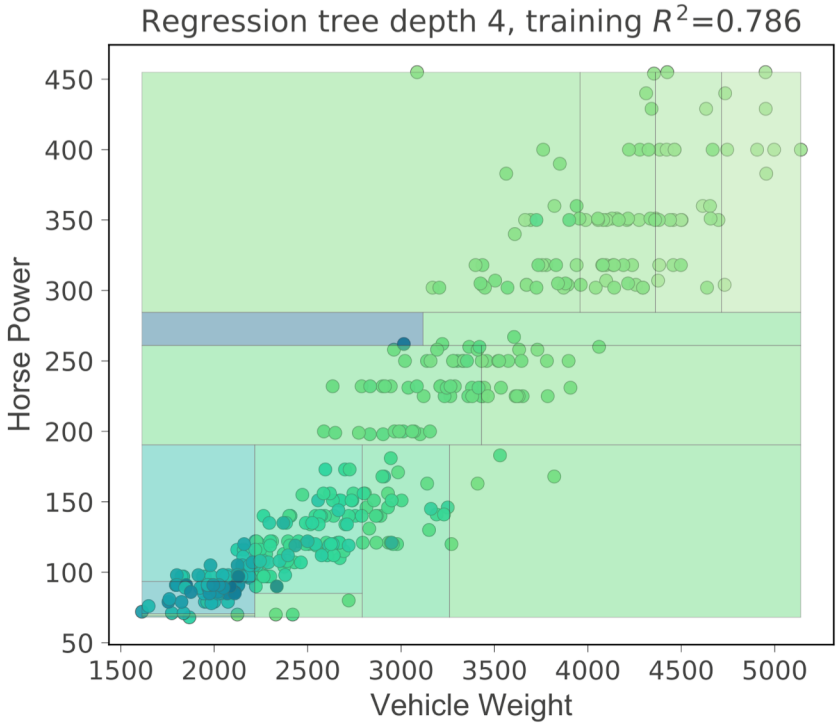
1D feature space vs `dtreeviz` decision tree

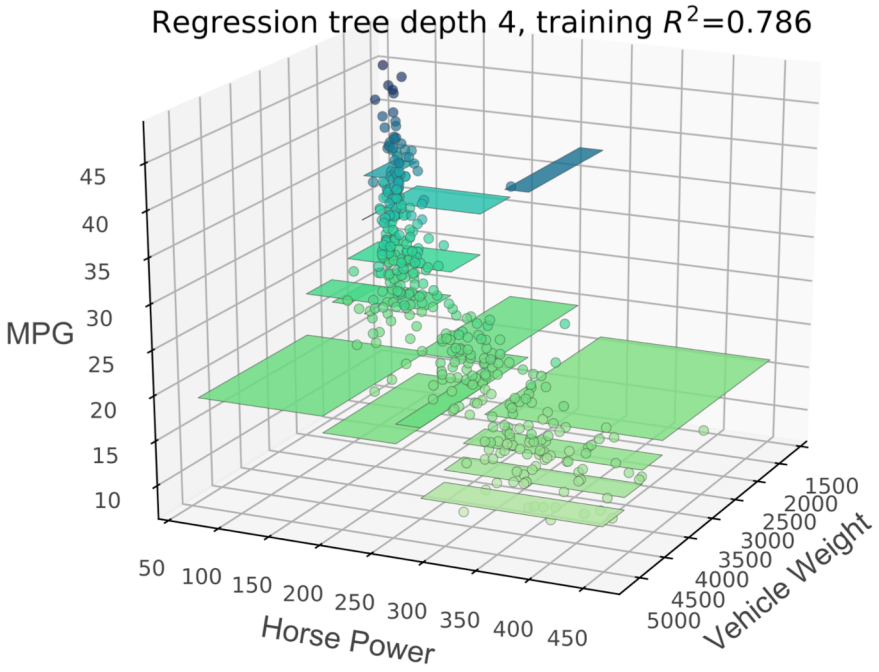


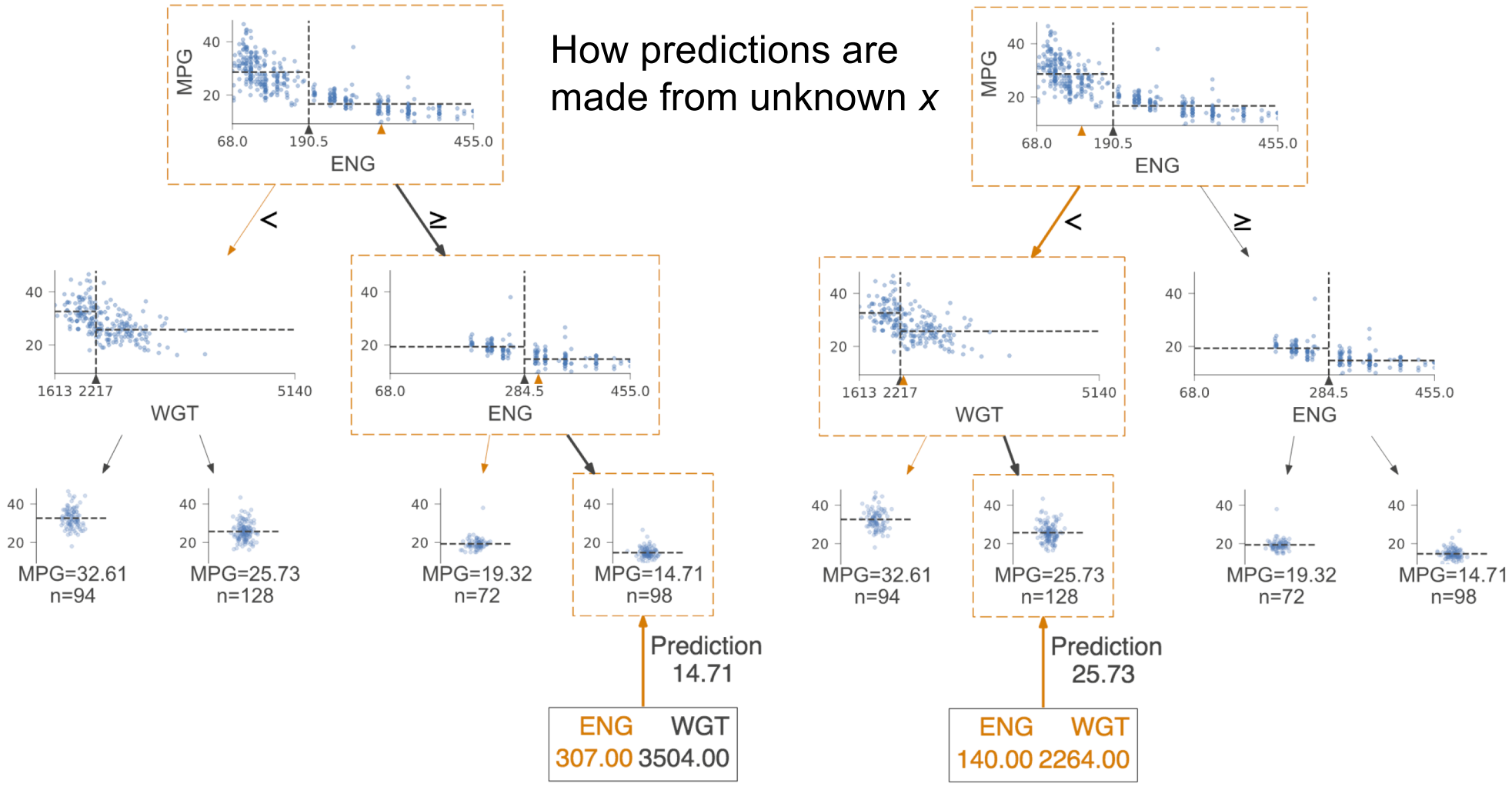


https://github.com/parrt/msds621/blob/master/notebooks/trees/partitioning.ipynb

2D regressor feature space (heatmap, 3D)






How predictions are made from unknown *x*
MPG
ENG
68.0
190.5
455.0
<
≥
WGT
1613 2217
5140
284.5
MPG=32.61
n=94
MPG=25.73
n=128
MPG=19.32
n=72
MPG=14.71
n=98
Prediction
14.71
ENG WGT
307.00 3504.00
Prediction
25.73
ENG WGT
140.00 2264.00

An aside: partitioning, tree viz done with custom library

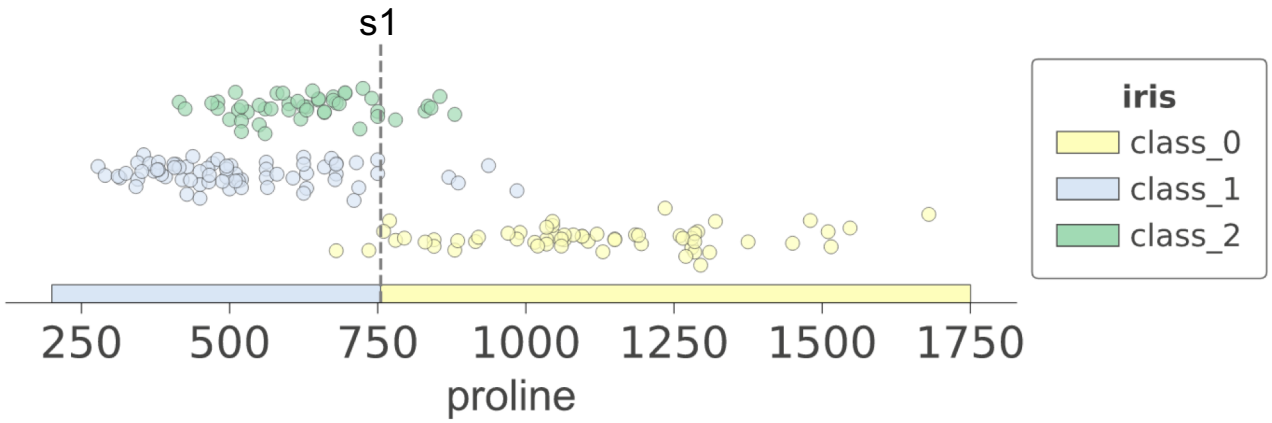
- Do "`pip install dtreeviz`"
- I built with Prince Grover, previous MSDS student
- See https://github.com/parrt/dtreeviz and the article for more detail: https://explained.ai/decision-tree-viz/index.html
- Advice: never accept status quo; always strive for more / better
- See "*How to lead a fulfilling life by being dissatisfied*" buried in my talk on decision tree viz https://twitter.com/the_antlr_guy/status/1120359898062000128

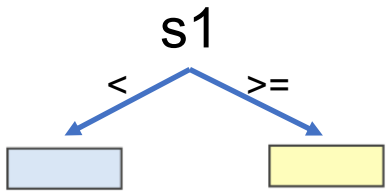
Classifiers

Classifiers split feature space too

Predict wine from proline

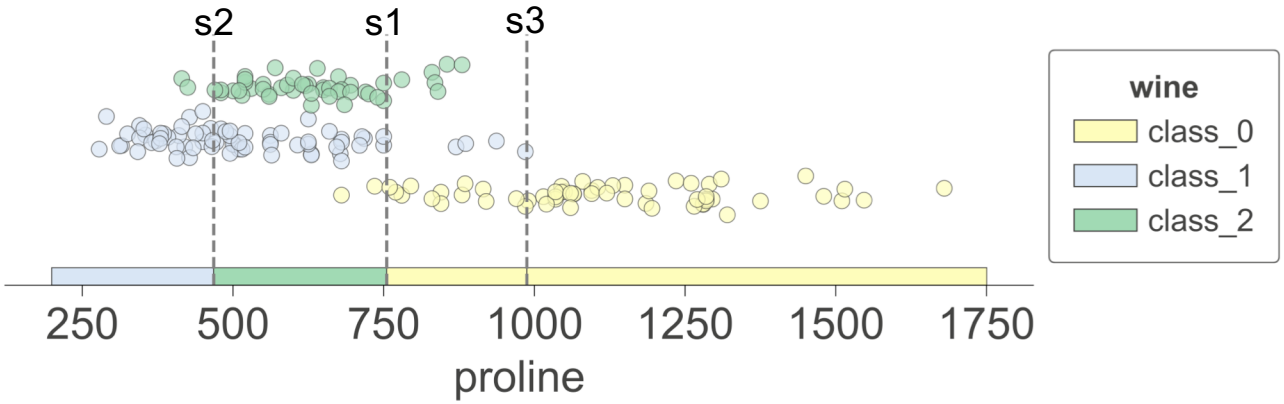
- Internal decision nodes test features just like regressor trees
- Leaves predict most common target category (mode) not mean
- Find split that decreases average impurity of left/right subregions

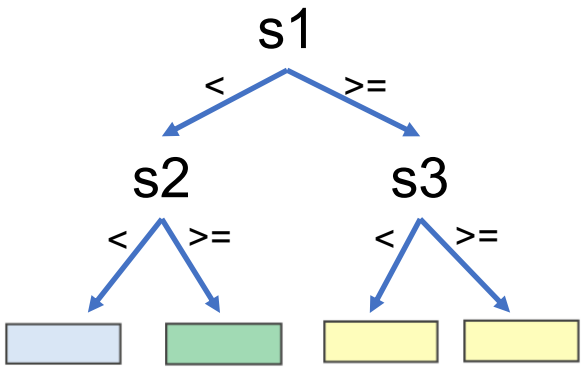




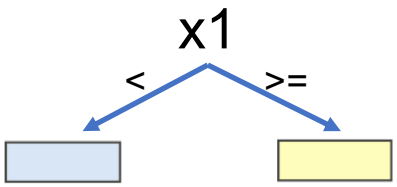
https://github.com/parrt/msds621/blob/master/notebooks/trees/partitioning.ipynb

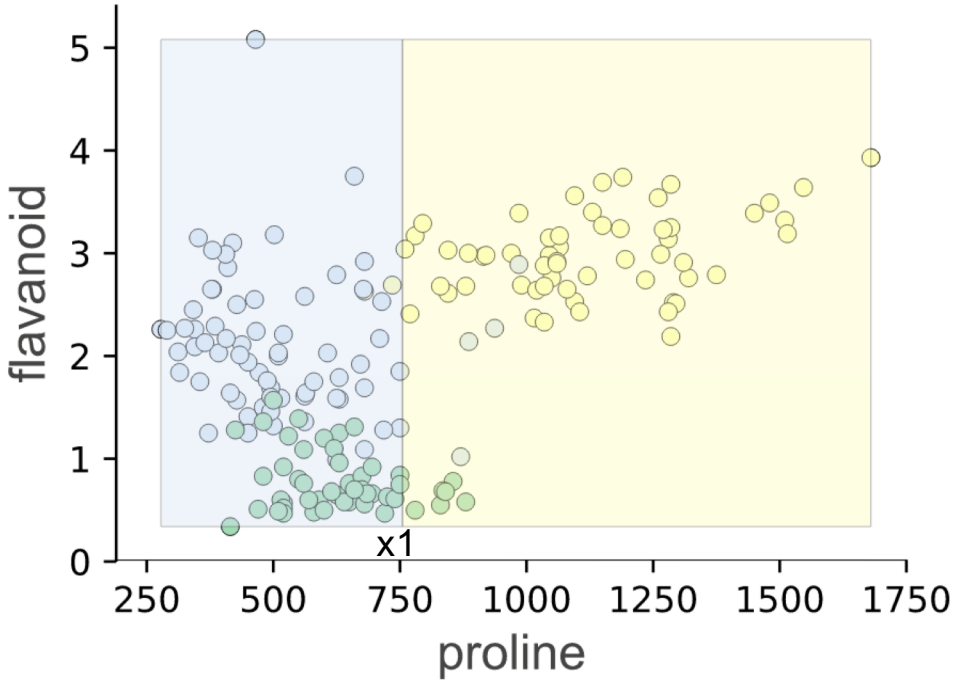
Split s1 subregions into more subregions





To improve predictions:
Use 2 features and
split 2D feature
space into regions

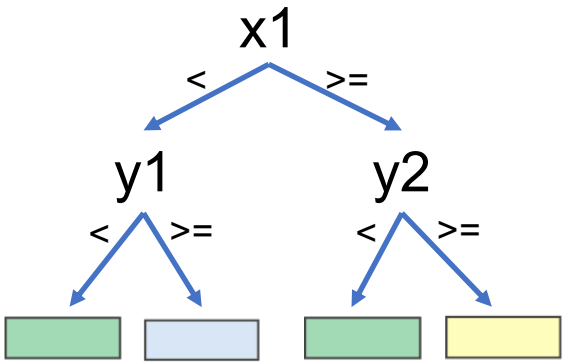


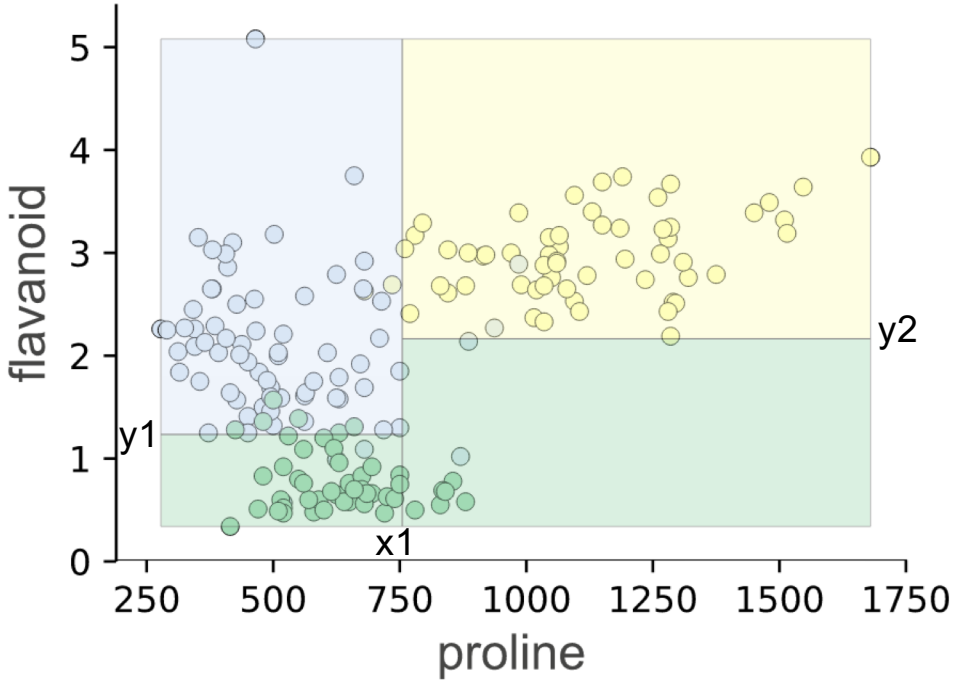


Training looks for (feature, split point) combos giving more pure subregions.

Decision nodes compare single feature value to split point

Improve predictions: Use 2 features and split 2D feature space into regions



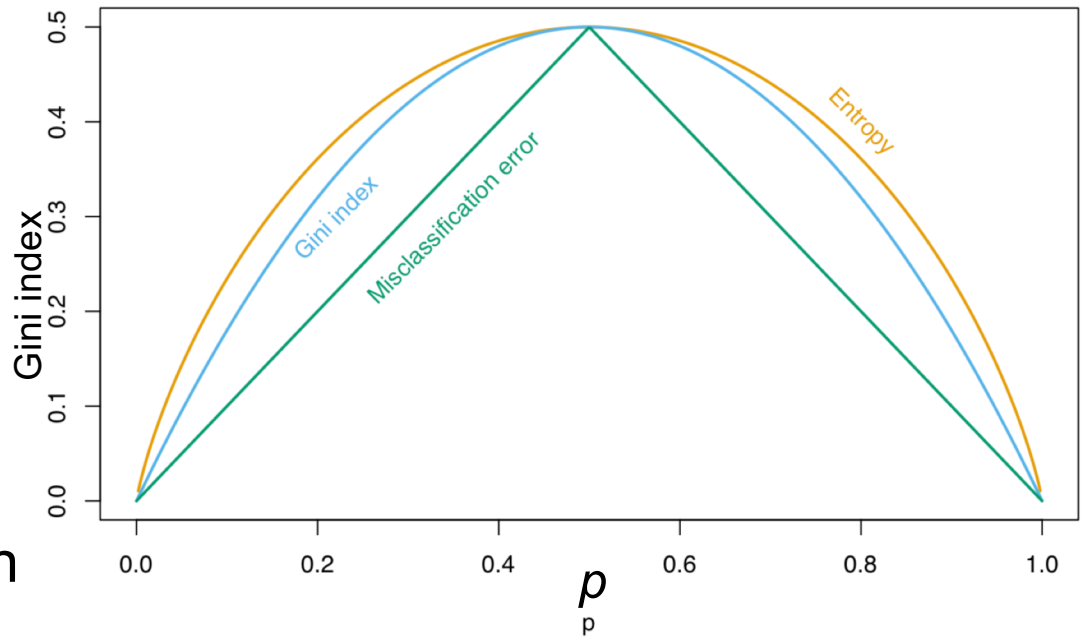


Training looks for (feature, split point) combos giving more pure subregions.

Decision nodes compare single feature value to split point

Gini index

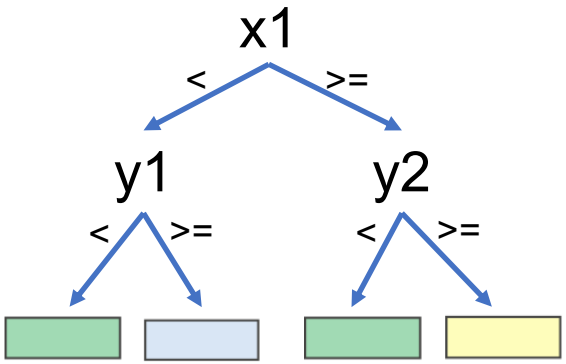
- A measure of y's impurity, like entropy
- Let p_i be the fraction of y values with class i (likelihood of i), n' = number of unique classes in y
- If likelihood is p=.5, uncertain

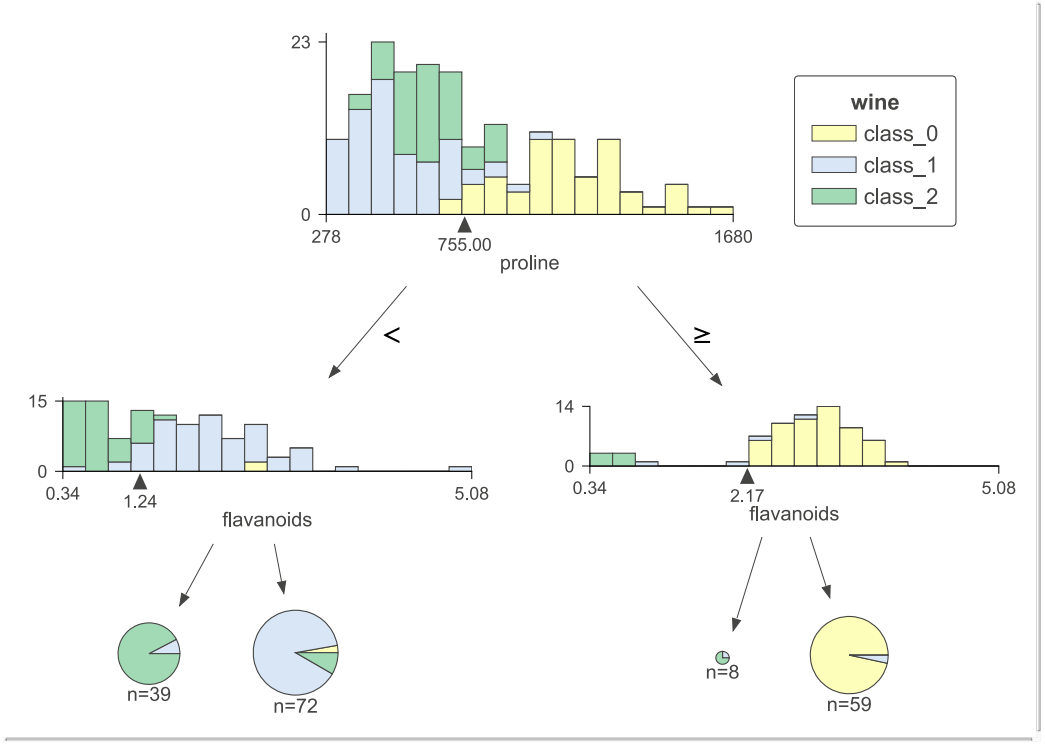


$$Gini(y) = \sum_{i=1}^{n'} p_i \sum_{j \neq i} p_j = \sum_{i=1}^{n'} p_i(1 - p_i) = 1 - \sum_{i=1}^{n'} p_i^2$$

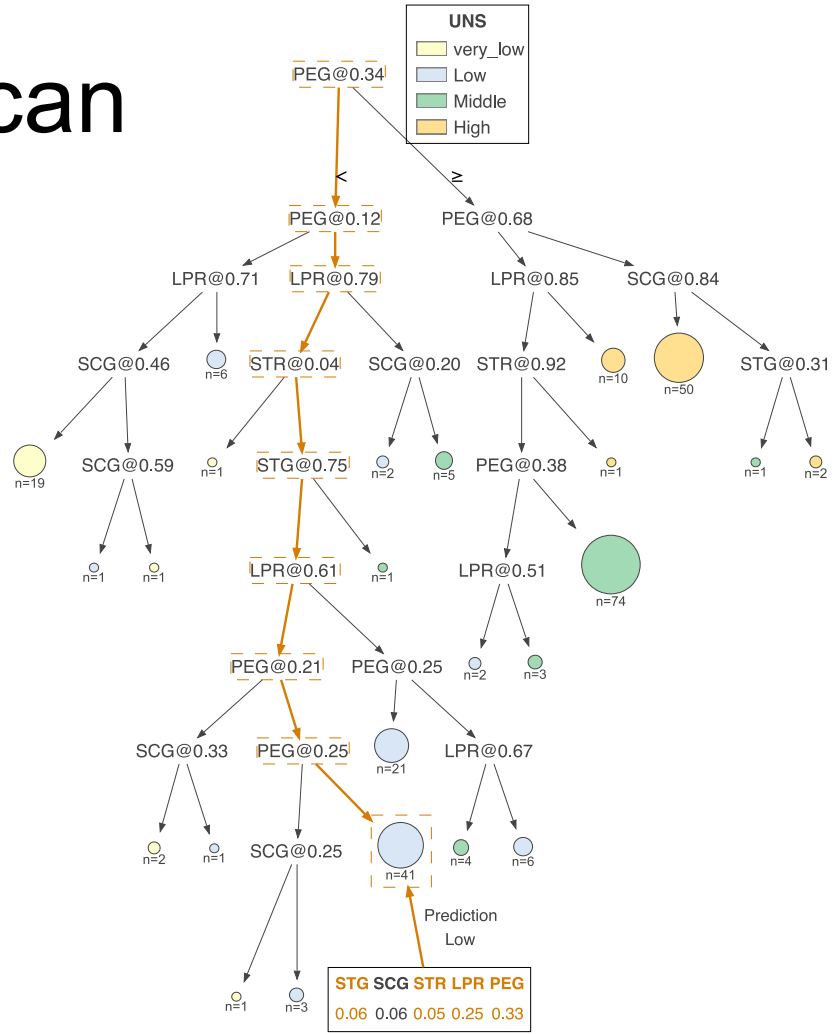
https://en.wikipedia.org/wiki/Decision_tree_learning#Gini_impurity

`dtreeviz` visualization of tree structure





For bigger trees, can do nonfancy plot



Tree structure's effect on prediction error

Hyperparameter num_samples_leaf

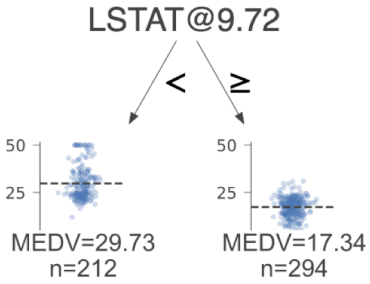
- Idea: don't split regions w/less than num_samples_leaf records
- Similar to limiting height of tree but finer granularity of control
- Degenerate case where num_samples_leaf=n (sample size)
  - What does such a regressor predict?
  - What does such a classifier predict?
  - Describe accuracy of this extreme model
  - If we trained on many different training sets pulled from same data distribution, how stable would the test set prediction error be?

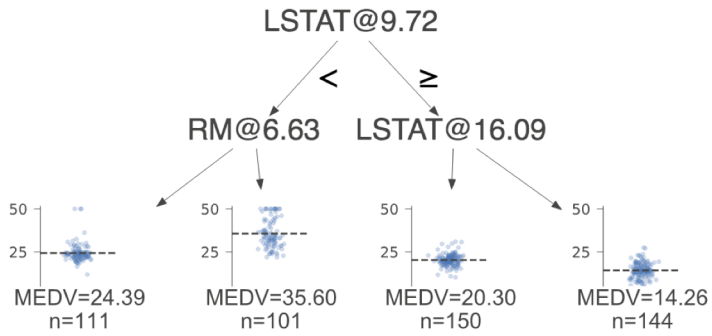
https://github.com/parrt/msds621/blob/master/notebooks/trees/decision-trees.ipynb

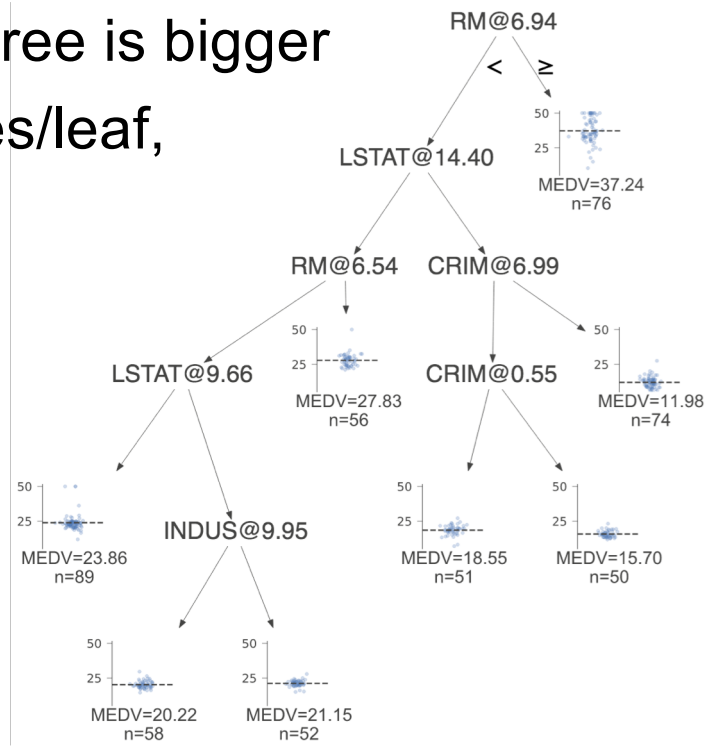
Trees for 200,100,50 samples per leaf

(Boston housing data, n=506 records)

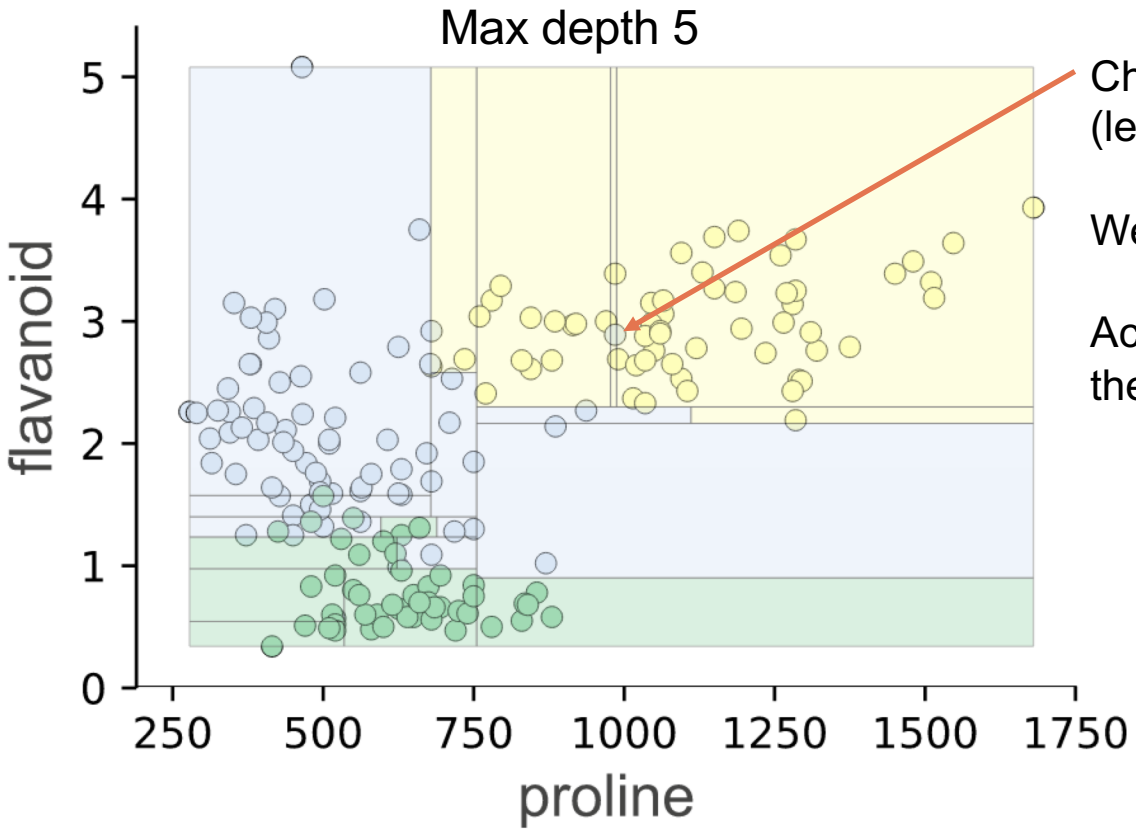
- Leaves have fewer & fewer elements, but tree is bigger
- Variance of y in leaves shrinks with samples/leaf, which is training goal (increases accuracy)







What happens with very small leaves?



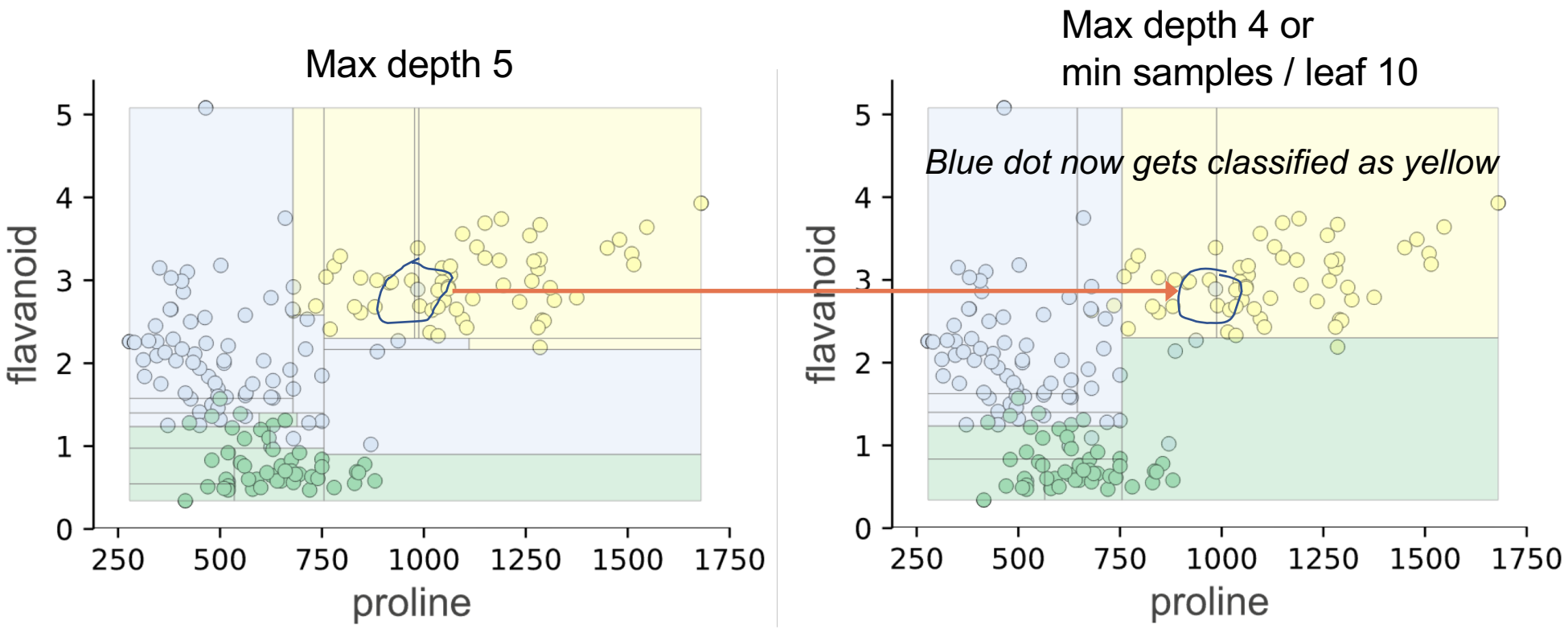
(Wine data set)

Check out that lonely blue dot in sea of yellow! (let's assume that tiny region is blue)

We let model get overly specific; it's overfit.

Accuracy on training set is very high, but at the cost of generality; test set error is high.

How could we (likely) improve generality?



(Wine data set)

Key takeaways