Random Forests

Objectives

- Thoroughly explain the algorithm for constructing a random forest
- Explain why Random Forests are more accurate than a single decision tree, in terms of bias and variance.
- Compute the feature importance for a random forest model
- Explain out-of-bag error

Decision Trees – Quick Review

- Advantages
  - Model nonlinear relationships.
  - Easily deal with continuous or categorical data*
  - No feature scaling necessary
  - Easily handles missing values*
  - ***Highly interpretable***
- Disadvantages
  - Expensive to train
  - Often poor predictors

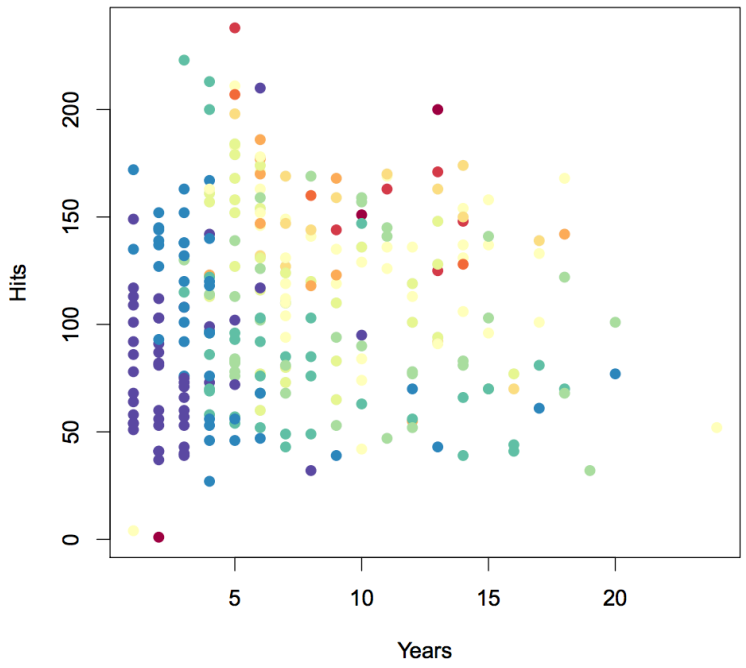
* Not in Python ☹

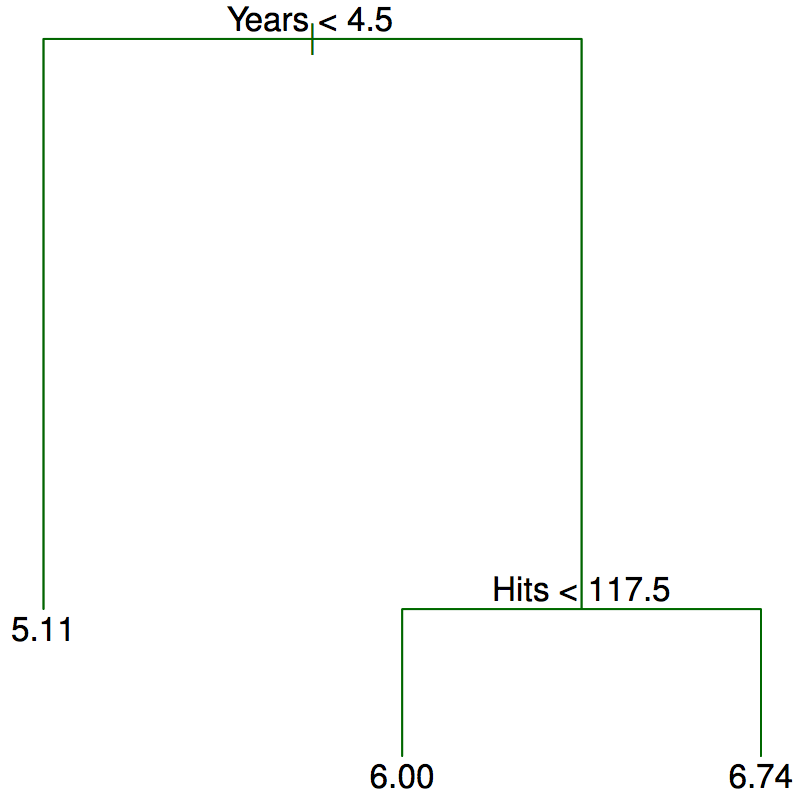
Decision Trees – Regression

Baseball salaries:

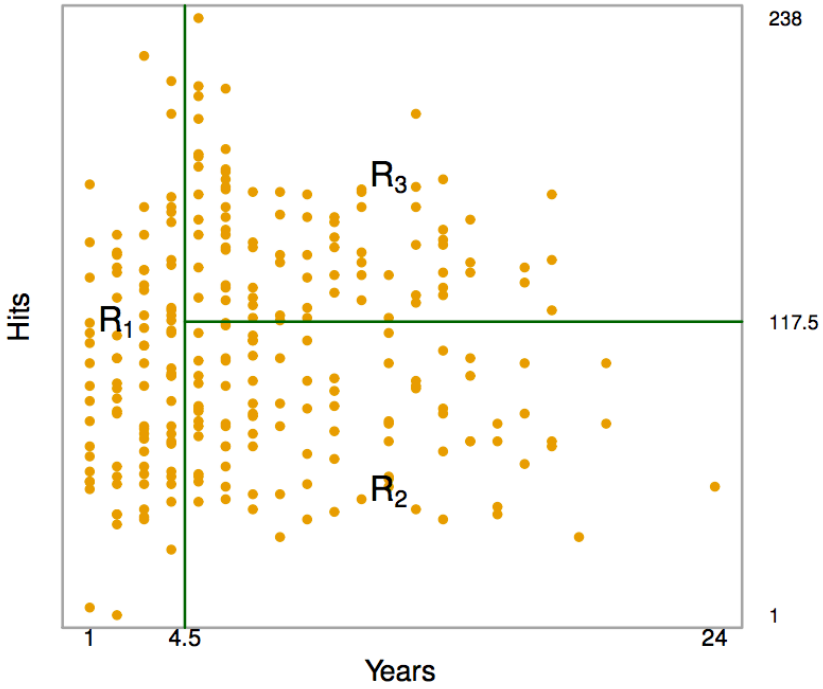
(Blue, Green) for low salaries

(Yellow, Red) for high salaries





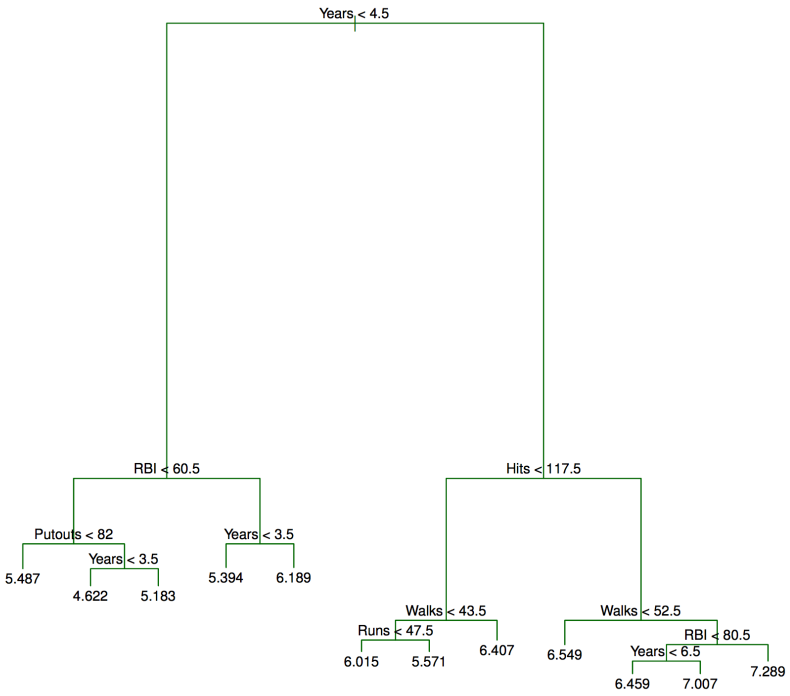
Decision Trees – Regression



At each split, we aim to minimize:

$$\sum_{m=1}^{|T|} \sum_{i:\, x_i \in R_m} (y_i - \hat{y}_{R_m})^2$$

Decision Trees – Regression



When to stop?

- You don't! Best practice to grow a very large "bushy" tree and prune backwards.

How?

- Let |T| = # of terminal nodes
- Then for any penalty term α, we have a subtree T which minimizes

$$\sum_{m=1}^{|T|} \sum_{i:\, x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|$$

- Cross-validate as usual to choose α and it's corresponding tree.

This should feel familiar!

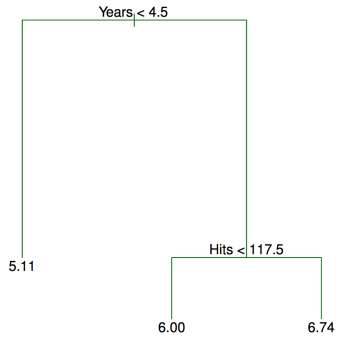
In Lasso/Ridge, attack high variance of *linear regression* with penalty λ

Here attack high variance of *decision tree* with penalty α

* In Python, pre-pruning is your only option ☹

Decision Trees – Classification

Making Predictions



At each terminal node (or rectangular region), predict

- Regression: Average
- Classification: Most commonly occurring class

How to split?

At each potential splitting node, minimize (in terms of information gain)

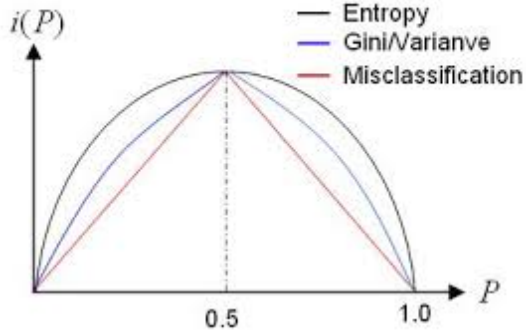
- Regression: RSS
- Classification:

Classification Error Rate $E = 1 - \max_k(\hat{p}_{mk})$

Gini index $G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$

Cross-entropy $D = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$

$\hat{p}_{mk}$ is proportion in m-th region in k-th class



Bagging

Bagging is a general method for combining many *weak* learners into a *strong* learner. That is, it is an *ensemble* method.

Intuition:

Classifier Example with Majority Vote:

- Suppose we have 5 completely independent classifiers with an accuracy of 70% for each. Then we can calculate the probability that they vote correctly:

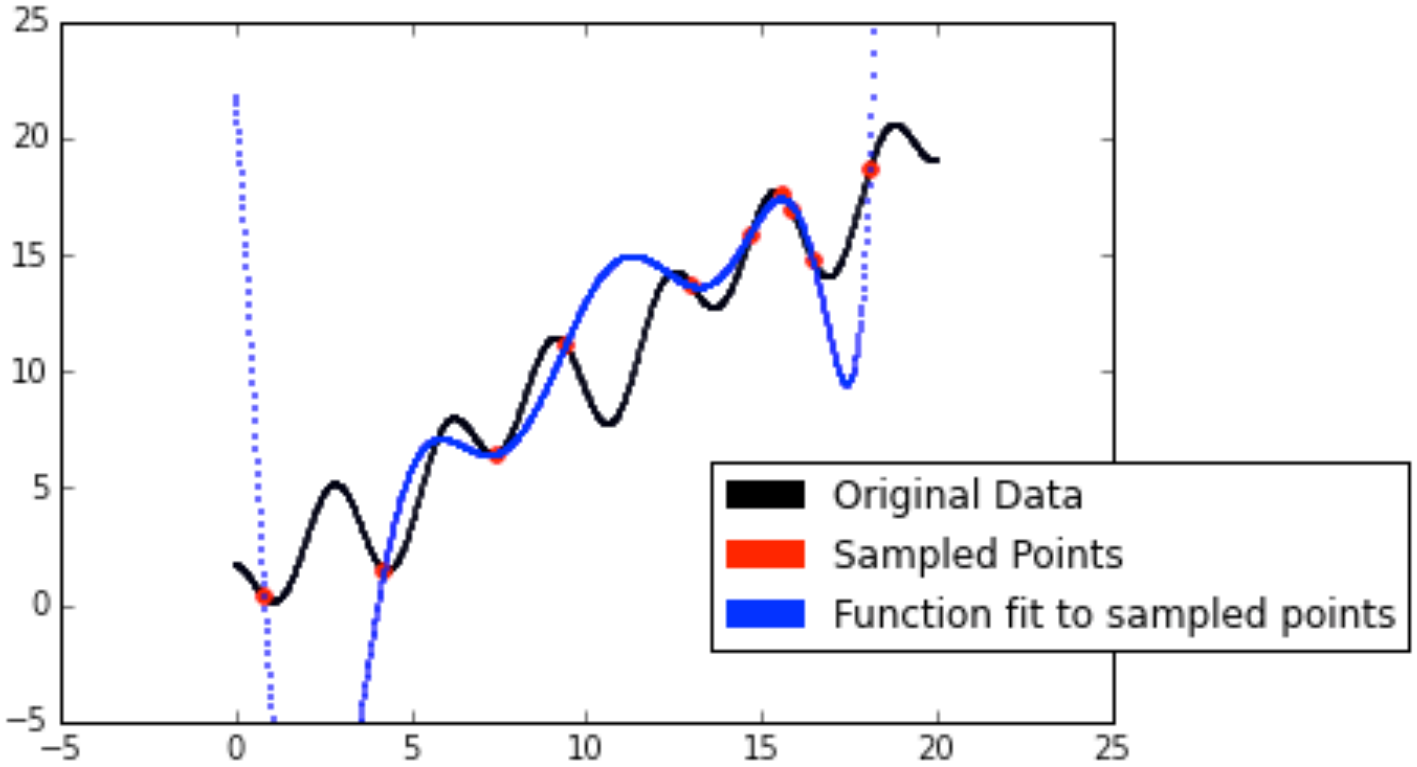
$$\binom{5}{3}(.7)^3(.3)^2 + \binom{5}{4}(.7)^4(.3)^1 + \binom{5}{5}(.7)^5(.3)^0 = .87$$

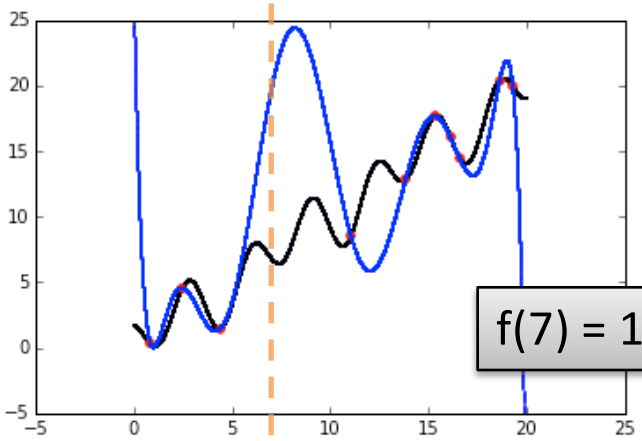
  - yields 83.7% chance that majority vote is accurate
- 101 such classifiers would yield 99.9% majority vote accuracy

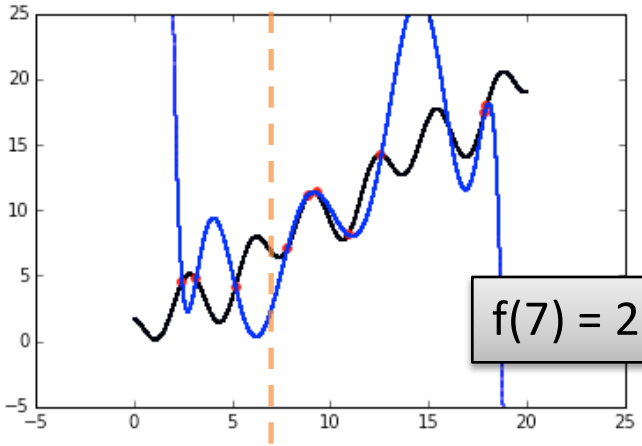
More Intuition

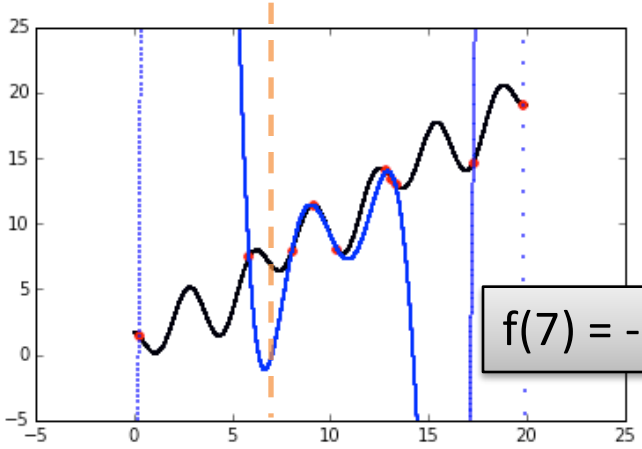
- Generate some sample data from a complex function (black points)
  - $f(x) = \sin(x) - 2\sin(2x - 1) + x$
- Pick 10 points at random from the sample and fit a degree 8 polynomial to the sample.

- **High Variance:** This model is highly sensitive---a slightly different dataset (choice of 10 points) will yield a very different model
- **Low Bias:** The model is complex enough---it just needs to see more data to make better predictions.



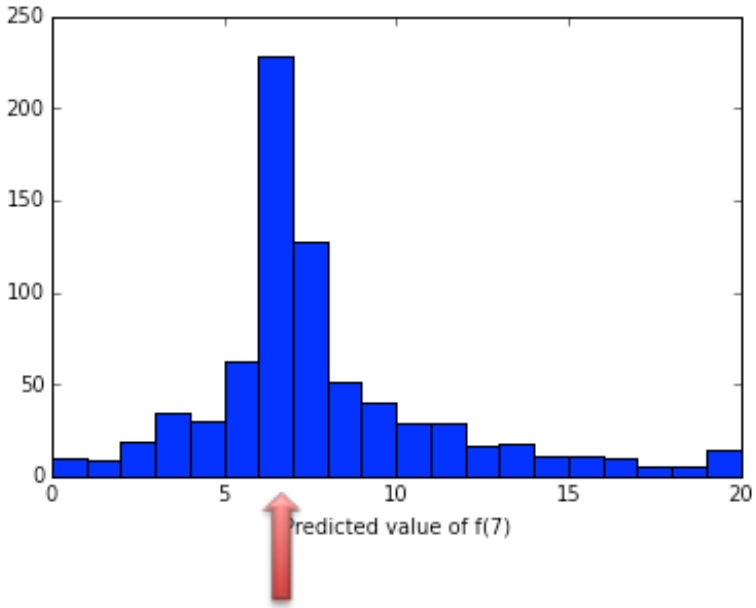






Repeat this process three times to get 3 very different functions (think *high variance*) with 3 very different predictions for f(7)

Repeat the process 1000 times and *on average* the prediction for f(7) is good (think *low bias)*.



Actual value: f(7) = 6.8

Bagging

- Previously we looked at post-pruning our single decision tree to attack the variance
- Instead, we can just grow many large "bushy" trees and average away the variance (*central limit theorem*) by growing lots of trees (*bootstrapping*)!
- Should work as long as the individual trees give the correct prediction *on average* (low bias).

Bagging

Training

- Take B bootstrap samples from your data
- Build a decision tree on each sample (technically, any weak classifier)

Prediction

- Regression: Average prediction of B trees

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x)$$

- Classification: Majority vote among B trees

Error Estimation

- Since bootstrapped, each tree only uses about 2/3 of observations → remaining 1/3 can be used to estimate OOB (out-of-bag) error. Like Test-error!

Bias-Variance "Tradeoff"

Bias

- Deep tree → Relatively low bias
- Expectation of average of B trees same as expectation of any one of the trees

Variance

- Where we really win!
- Average of *B* i.d. (identically distributed) random variables, with pairwise correlation ρ, has variance…

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

What happens as *B* increases?
What does *ρ* depend on?

Random Forests

- Bagging works well, but it can fail when the individual predictors are biased, or when they are all wrong *in the same way.* That is, if they are correlated.
- Random Forests improve on bagging by *de-correlating* the trees.
- At each decision tree split, only m (usually m $= \sqrt{nfeatures}$) features are considered for the split.

Random Forest Parameters

- Total number of trees
- Number of features to use at each split
- Number of points to grab for each sample
- Individual decision tree parameters
  - Usually the individual estimators are grown to maximum depth. Remember, each estimator should have low bias, and the RF will deal with the variance.

In general, RF are fairly robust to the choice of parameters and over-fitting.

Afternoon

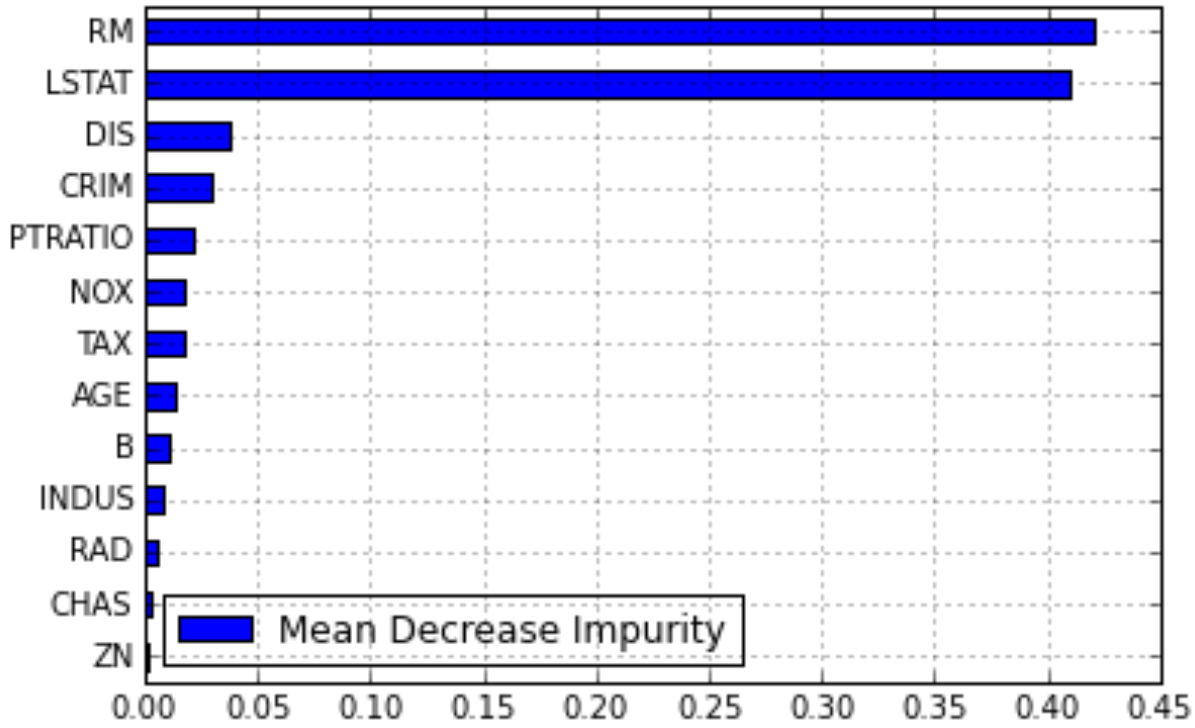
Model Interpretation

- Model Interpretability is the most commonly cited drawback of using Random Forests.
- The standard approach to interpretation is to measure the importance of each variable. Two methods exist:
  1. Mean Decrease Impurity
  2. Mean Decrease Accuracy
- Another approach is to analyze decision paths for individual data points.

Mean Decrease Impurity

- For each tree, each split is made in order to reduce the total impurity of the tree (Gini Impurity for classification, RSS for regression); we can record the magnitude of the reduction.
- Then the importance of a feature is the average decrease in impurity across trees in the forest, as a result of splits defined by that feature.
- **GOAL:** To determine which features have the largest impact on the model.

MDI - Boston Housing data



The biggest factors in predicting the value of a neighborhood are the average number of rooms and the class status of the neighborhood

Mean Decrease Accuracy

Alternative way to calculate variable importance

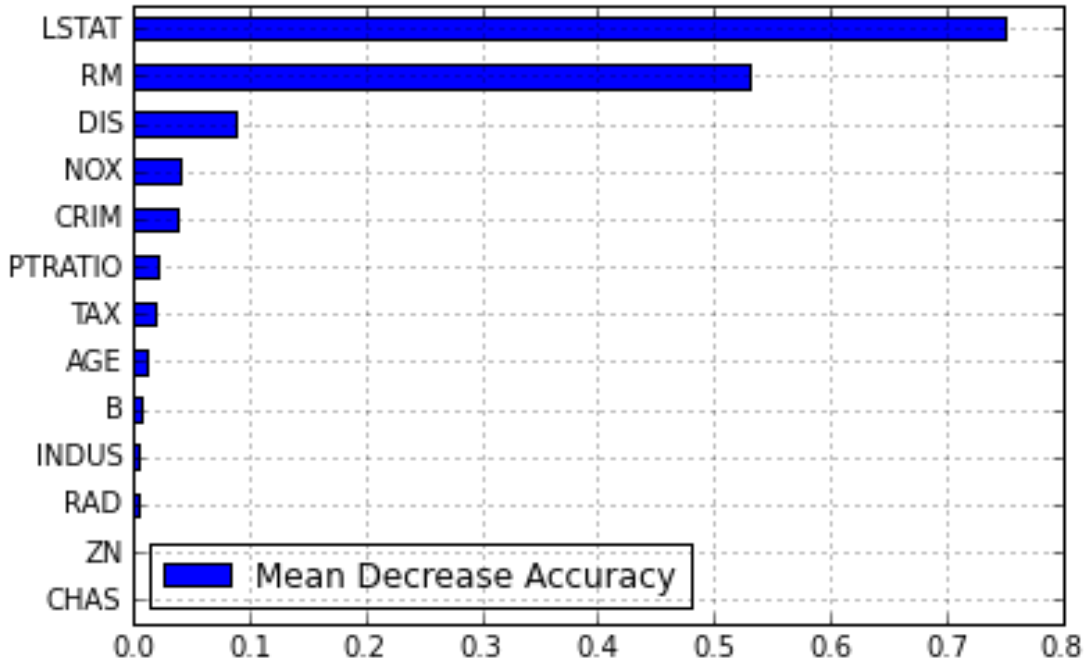
To evaluate importance of *j*th variable…

(1) When *b*th tree is grown, OOB samples passed down through tree → record accuracy

(2) Values of *j*th variable randomly permuted in OOB samples → compute new (lower) accuracy

→ Average decrease in accuracy over all trees

MDA - Boston Housing Data

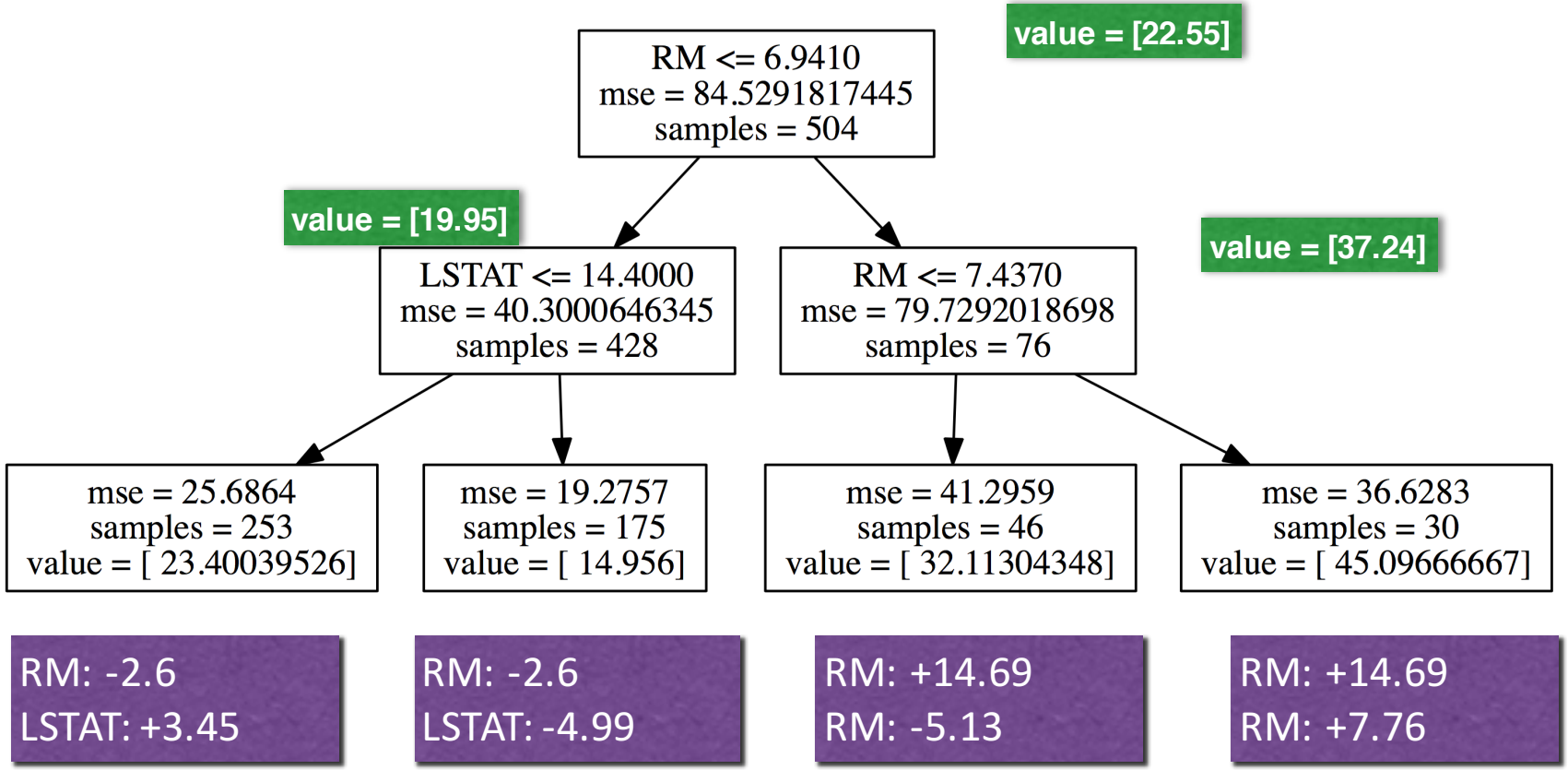


Results are similar to MDI, but some of the relative magnitudes are different

Decision Tree Paths

- There is another choice for interpreting Random Forests (or other tree ensembles)
- When a tree makes a prediction about a new observation, we can think of that observation following a path down the tree.
- Keeping track of these paths gives additional information about how the prediction is actually made.

Path Contributions Example



Any prediction can be decomposed into its gain/loss because of RM + its gain/loss due to LSTAT

Decision paths from one tree

Training set mean: 22.5522862823

Prediction 1: 32.97
Actual 1: 23.9
Top five contributing features:
RM 9.74326927325
NOX 1.27825396825
DIS -0.60380952381

Prediction 2: 27.5277777778
Actual 2: 22.0
Top five contributing features:
LSTAT 3.45273170552
RM 1.96992924042
DIS -0.447169450465

Prediction 3: 21.6798969072
Actual 3: 11.9
Top five contributing features:
RM -3.87795163014
LSTAT 3.45273170552
DIS -0.447169450465

Random Forest Decision Paths

- Decision paths can be naturally extended to Random Forests
- Whenever a prediction is made, the contribution paths for each tree are averaged across the forest.

Random Forest Decision Path Example

1. Split Boston data into train/test and build a Random Forest Regression model on the training data.
2. Pick 6 observations in the test set that have similar predictions (all ~ $30K).
3. Calculate the average contribution, from the model, across all trees, for each observation.

Raw Data

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
Observation 1	9.23230	0	18.10	0	0.631	6.216	100.0	1.1691	24	666	20.2	366.15	9.53
Observation 2	8.26725	0	18.10	1	0.668	5.875	89.6	1.1296	24	666	20.2	347.88	8.88
Observation 3	0.03049	55	3.78	0	0.484	6.874	28.1	6.4654	5	370	17.6	387.97	4.61
Observation 4	0.04417	70	2.24	0	0.400	6.871	47.4	7.8278	5	358	14.8	390.86	6.07
Observation 5	0.06911	45	3.44	0	0.437	6.739	30.8	6.4798	5	398	15.2	389.71	4.69
Observation 6	0.49298	0	9.90	0	0.544	6.635	82.5	3.3175	4	304	18.4	396.90	4.54

Looking at the raw data can tell you the difference between the different observations, but it doesn't tell you *how* and *why* they are all predicted to have the same value.

Average of contribution paths

Average thousands of dollars added to the estimate, by the model, because of the value of AGE.

Distance from city-center

Fraction of lower-class residents

Average number of rooms

	AGE	B	CHAS	CRIM	DIS	INDUS	LSTAT	NOX	PTRATIO	RAD	RM	TAX	ZN
Prediction 1	-0.09	0.12	-0.02	0.53	6.11	0.10	2.67	-0.02	-0.19	0.06	-2.63	-0.20	0.03
Prediction 2	-0.07	0.04	0.05	0.63	6.22	0.16	2.56	-0.01	-0.19	0.06	-3.15	-0.16	0.03
Prediction 3	-0.01	-0.12	-0.05	0.05	-0.70	0.04	7.42	-0.11	0.42	-0.02	1.10	-0.14	-0.05
Prediction 4	0.04	-0.14	-0.02	-0.02	-0.53	0.25	3.50	0.16	1.46	0.13	2.21	-0.24	0.11
Prediction 5	0.17	-0.13	-0.02	-0.03	-0.68	0.15	7.86	0.03	0.85	0.01	-1.14	-0.16	0.18
Prediction 6	0.60	-0.05	-0.03	0.03	0.18	-0.26	8.62	-0.19	-0.02	-0.07	-1.83	-0.34	-0.05

All 6 houses are predicted to have a value of $30K

- Predictions 1 and 2 are based primarily on those houses being close to the city center
- Prediction 4 is a relatively large house in a relatively good neighborhood.
- Predictions 5 and 6 are smaller houses, but in really nice neighborhoods.

Decision Trees – Quick Review

- Advantages
  - Model nonlinear relationships.
  - Easily deal with continuous or categorical data*
  - No feature scaling necessary
  - Easily handles missing values*
  - ***Highly interpretable***
- Disadvantages
  - Expensive to train
  - Often poor predictors

* Not in Python ☹

Random Forest – Quick Review

- Advantages
  - Model nonlinear relationships.
  - Easily deal with continuous or categorical data*
  - No feature scaling necessary
  - Easily handles missing values*
  - ***Somewhat interpretable***
  - Prediction accuracy on par with cutting edge algorithms (Kinect uses RF for face detection)
- Disadvantages
  - Expensive to train

* Not in Python ☹