Introduction to Time Series

THE UNIVERSITY OF CHICAGO - MASTERS IN ANALYTICS – WEEK 7

Asynchronous Agenda

Unsupervised learning

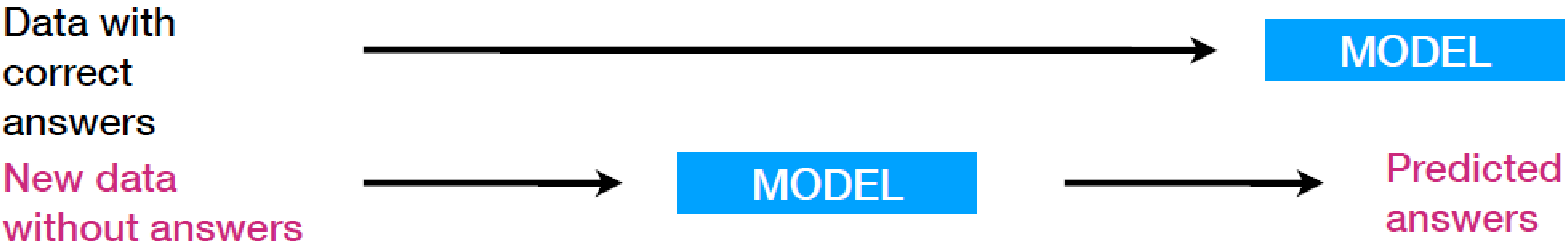
Supervised learning

- Decision trees
- Bagging
- Random forest
- Boosted trees

Unsupervised learning

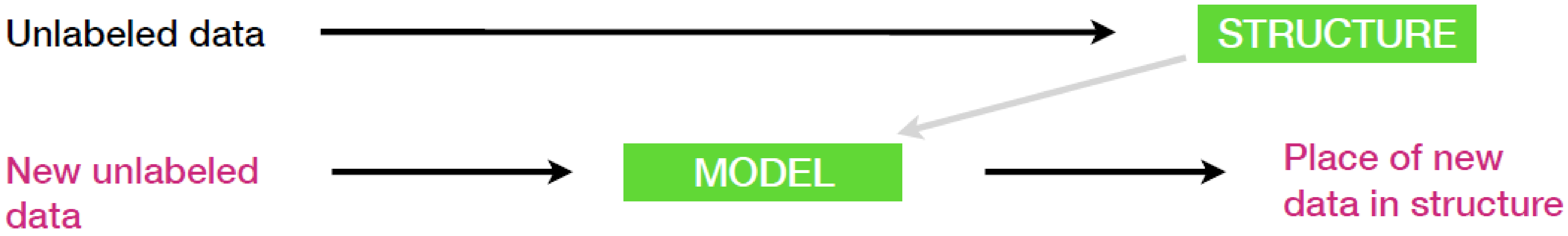
Supervised learning problems

- Involve constructing an accurate model that can predict some kind of an outcome when **past data has labels** for those outcomes



Unsupervised learning problems

- Involve constructing models where **labels on historical data are unavailable**



Unsupervised business problems

- Similarity matching – How can we identify similar individuals based on data we know about them
- Example: IBM is interested in finding customers similar to their best business customers
- K-nearest neighbors, hierarchical clustering

Unsupervised business problems

- Clustering– Not driven by any specific purpose
- Example: Do our customers form natural groups or segments?
- Preliminary exploration, may lead to questions like:
  - What products should we offer?
  - How should our customer care team be developed?
- K-means clustering, DBSCAN

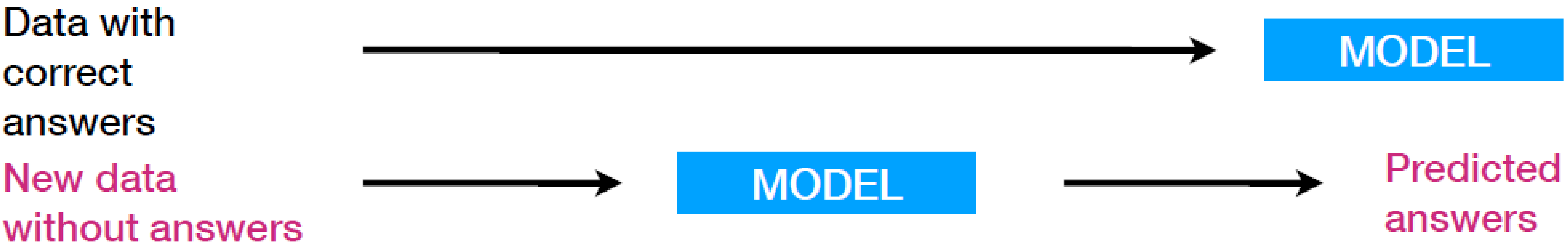
Unsupervised business problems

- Dimensionality reduction
- Reducing the number of predictors you have and determining which are the most important
- Example: Can we determine what are the most important variables that influence gallons of gasoline purchased?
- PCA, SVD

Supervised learning

Supervised learning problems

- Involve constructing an accurate model that can predict some kind of an outcome when past data has labels for those outcomes



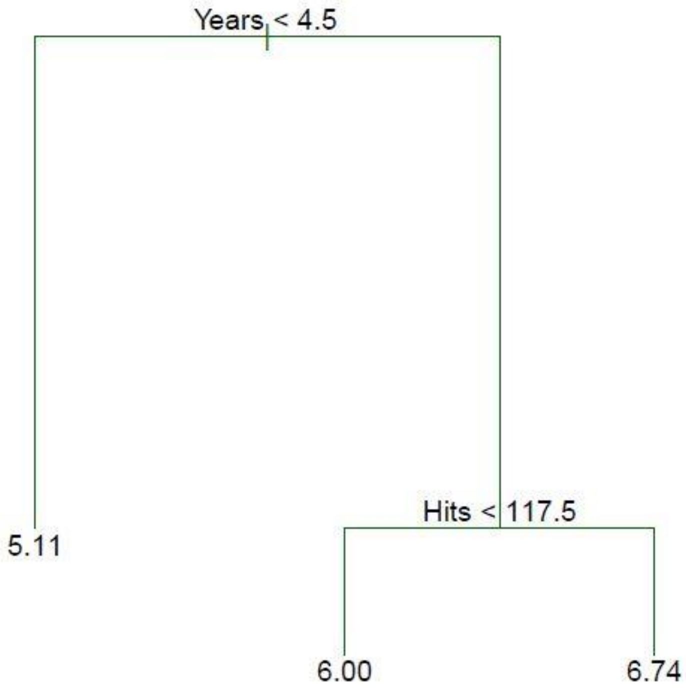
Regression vs Classification

- Regression models predict a continuous value
- Classification models predict a discrete class label
- Some algorithms can be used for both types of ML tasks
- These problems have different error metrics that are not interchangeable

Decision Trees

- Goal: Segmenting the predictor space into a number of simple region
- Benefits: Easily interpretable
- Drawbacks: Variance and accuracy
- Regression trees: Response is continuous, uses the region’s mean as the predictive value
- Classification trees: Response is categorical, uses the region’s mode as the predictive value

Hitters data example



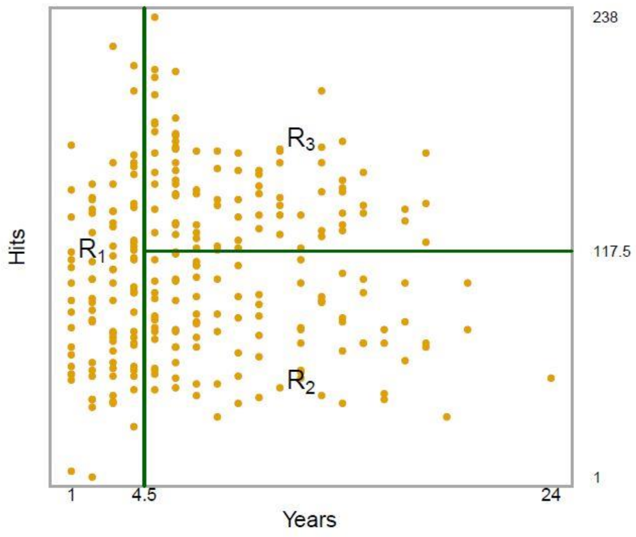
Use the Hitters data from the ISLR library for a simple example

```
data("Hitters")
# Remove incomplete cases
Hitters <- na.omit(Hitters)
kable(head(Hitters,3))
```

	AtBat	Hits	HmRun	Runs	RBI	Walks	Years	CAtBat	CHits	CHmRun
-Alan Ashby	315	81	7	24	38	39	14	3449	835	69
-Alvin Davis	479	130	18	66	72	76	3	1624	457	63
-Andre Dawson	496	141	20	65	78	37	11	5628	1575	225

Source

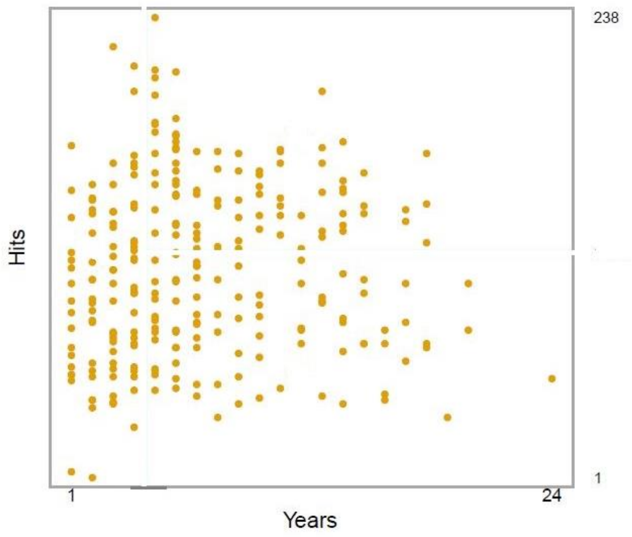
Choosing regions

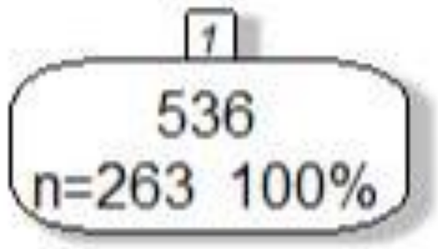


$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$$

- Divide the predictor space into J distinct and non overlapping regions
- For every observation that falls into R_j we make the same prediction. The mean of the response values from the training set

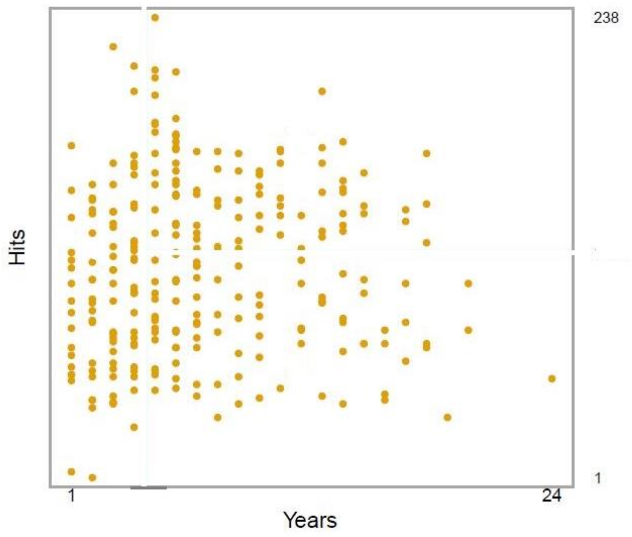
Choosing regions

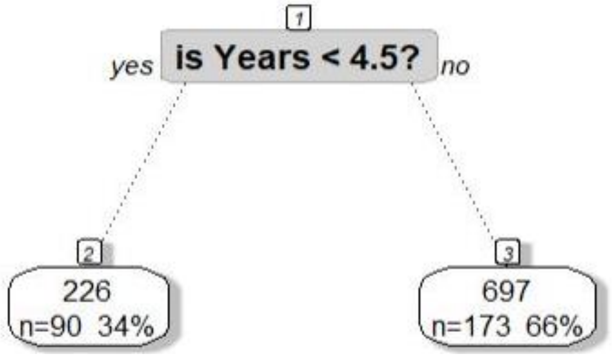




- Recursive Binary Splitting – greedy approach
- Start from the top of the tree where all the observations are in one single region
- Greedy because only the best split is made at that particular step, rather than looking ahead and picking a split that will lead to a better tree in some future step

Choosing regions





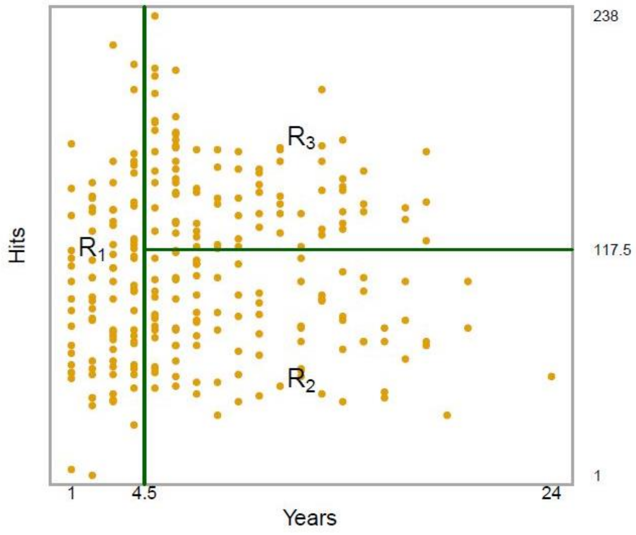
- Successively split the predictor space
- Each split will be indicated by 2 new branches down the tree

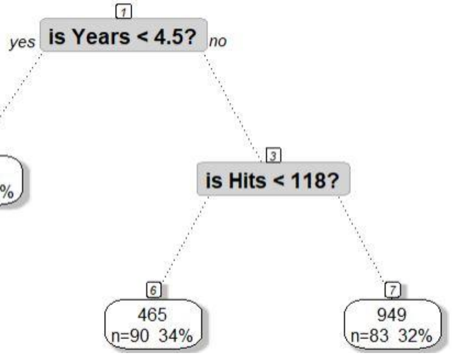
$$R_1(j,s) = \{X|X_j < s\} \text{ and } R_2(j,s) = \{X|X_j \geq s\},$$

seek the value of j and s that minimize the equation

$$\sum_{i:\, x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:\, x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2,$$

Choosing regions





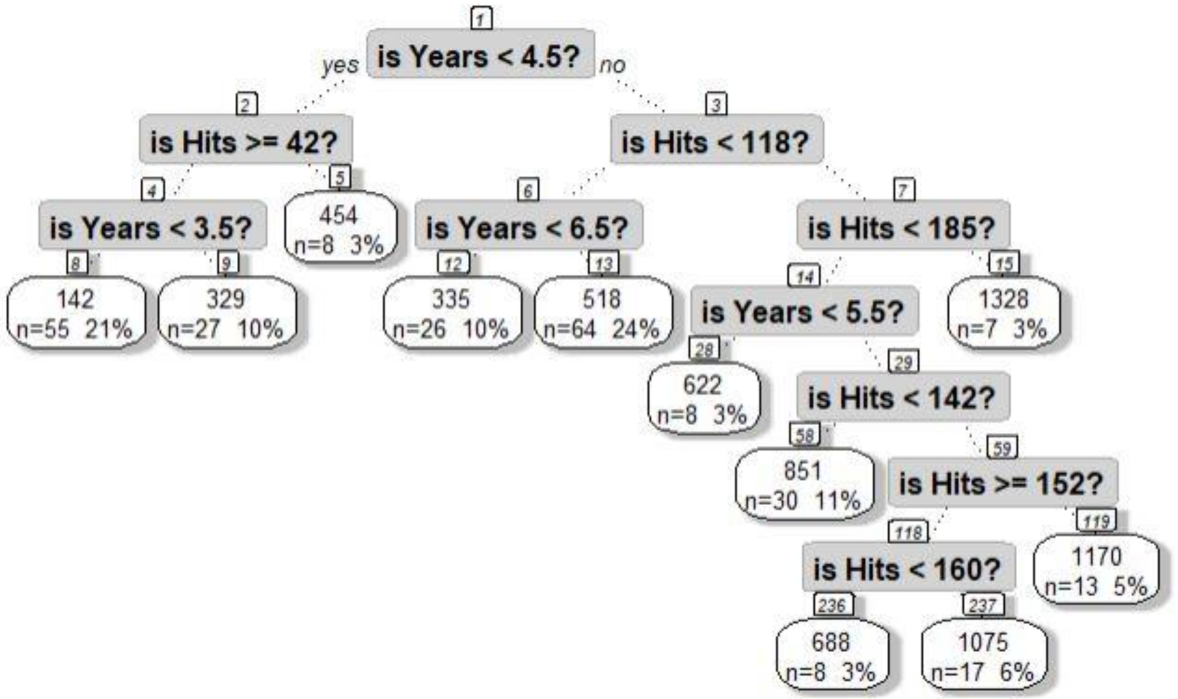
- Repeat the process looking for the best predictor and bet cut point
- Minimize the Regional Sum of Squares

$$R_1(j,s) = \{X|X_j < s\} \text{ and } R_2(j,s) = \{X|X_j \geq s\},$$

seek the value of j and s that minimize the equation

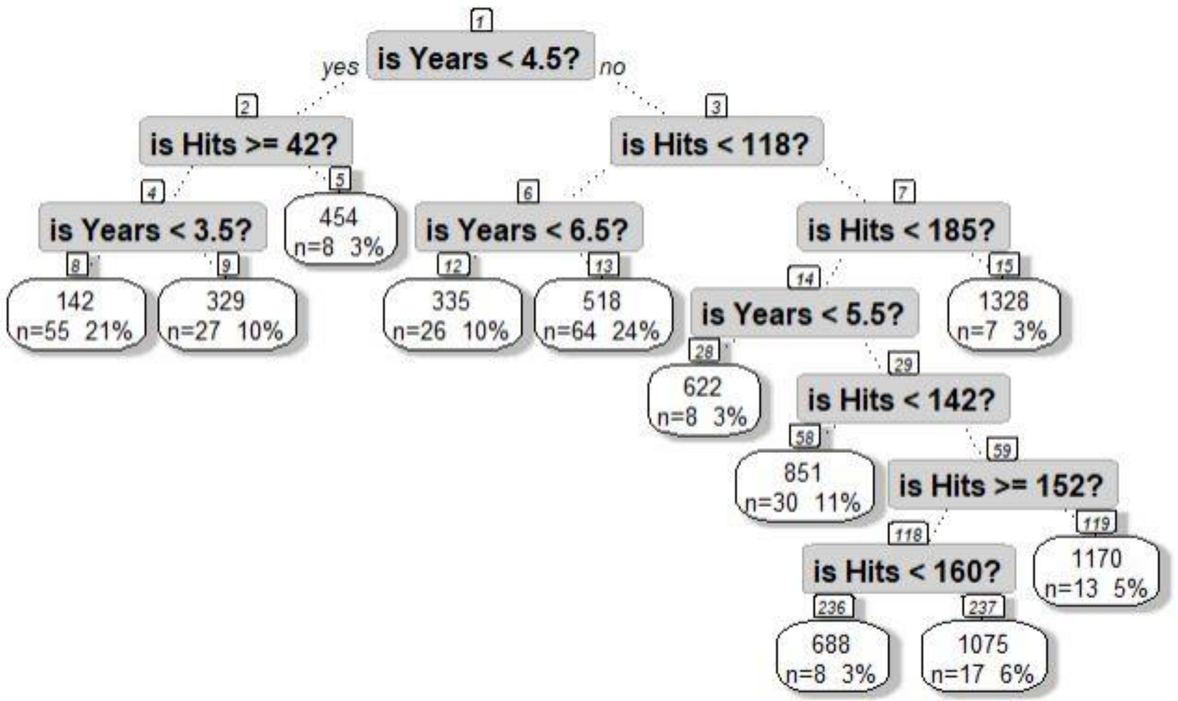
$$\sum_{i:\, x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:\, x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2,$$

Choosing regions



- Process continues until it reaches a stopping criterion
- Example: Limit the number of observations in a node to 5
- We can now predict a new test observation by returning the mean of the training observations in the given region

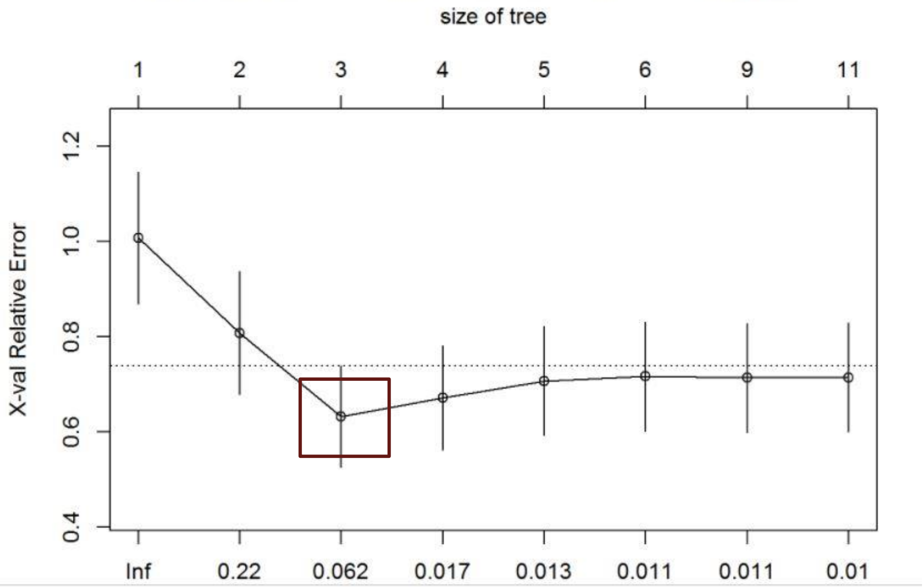
Pruning



- Full tree may overfit the training dataset, leading to poor prediction on test.
- Pruning will lower the variance and increase the bias.
- Consider sub-trees and look for the one with the lowest test error using cross-validation

Pruning

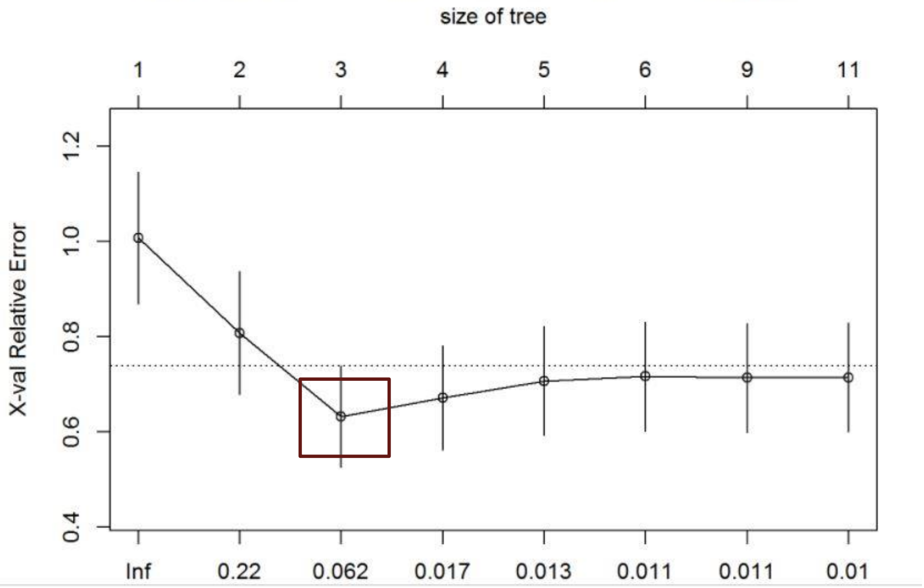
```
##          CP nsplit rel error  xerror    xstd
## 1 0.246750      0   1.00000 1.00686 0.13924
## 2 0.189906      1   0.75325 0.80766 0.12971
## 3 0.020522      2   0.56334 0.63206 0.10662
## 4 0.014281      3   0.54282 0.67086 0.10992
## 5 0.011625      4   0.52854 0.70686 0.11418
## 6 0.010870      5   0.51692 0.71573 0.11457
## 7 0.010267      8   0.48430 0.71287 0.11489
## 8 0.010000     10   0.46377 0.71403 0.11488
```

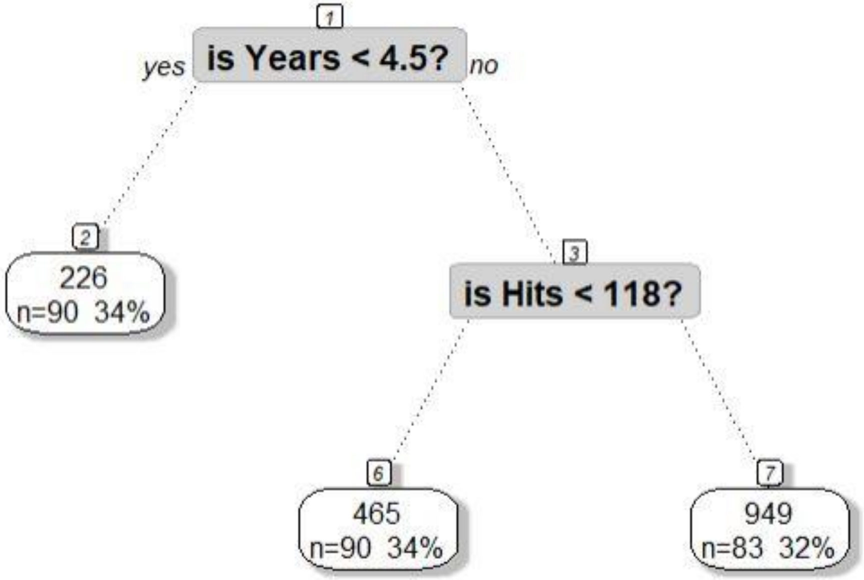


- For each additional node (sub tree)
  - Run cross validation
  - Return the error
  - Select the tree with the lowest error

Pruning

```
##          CP nsplit rel error  xerror    xstd
## 1 0.246750      0   1.00000 1.00686 0.13924
## 2 0.189906      1   0.75325 0.80766 0.12971
## 3 0.020522      2   0.56334 0.63206 0.10662
## 4 0.014281      3   0.54282 0.67086 0.10992
## 5 0.011625      4   0.52854 0.70686 0.11418
## 6 0.010870      5   0.51692 0.71573 0.11457
## 7 0.010267      8   0.48430 0.71287 0.11489
## 8 0.010000     10   0.46377 0.71403 0.11488
```





Classification

- Used to predict a qualitative response vs a quantitative response
- Using the mode of the region's observations instead of the mean
- Also interested in the class proportions of observations per region

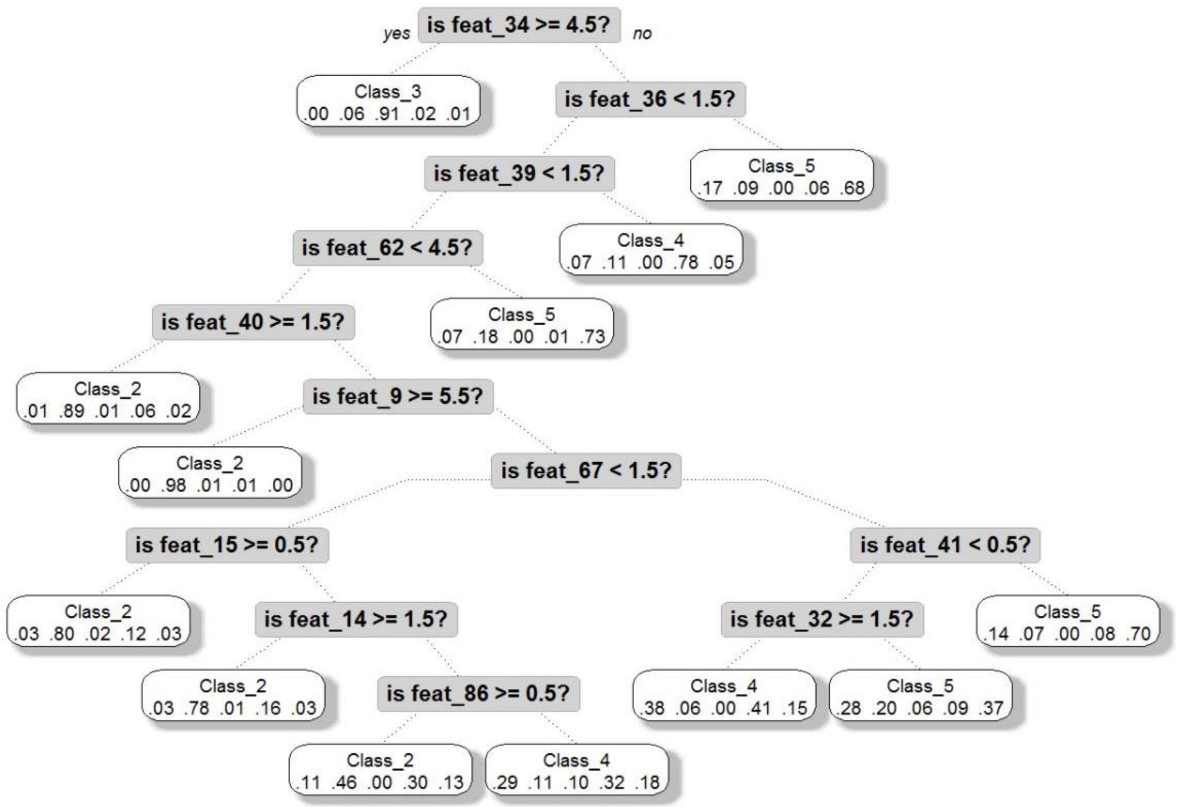
Classification

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

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

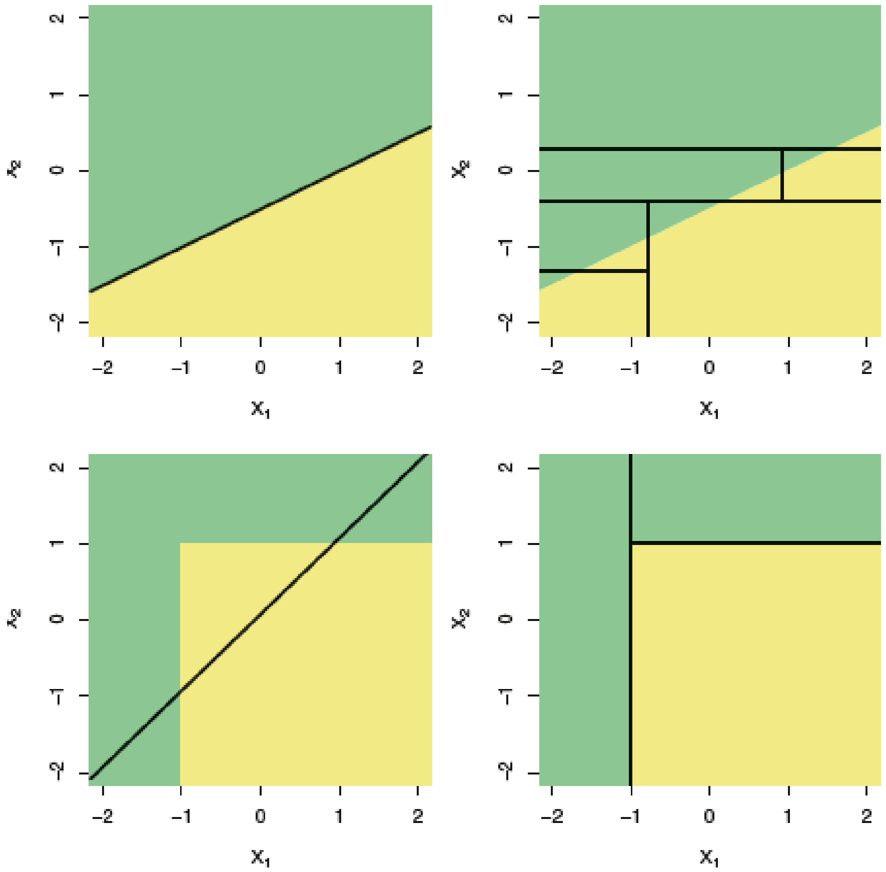
- Process is the same as regression trees, except the splitting criterion
- Gini Index – measure of node purity – how often an element is labeled correctly
  - P_{mk} represents the proportion of observations in the *m*th region from the *k*th class
  - Small value indicates that a node predominantly contains observations from a single class
  - Cross Entropy – alternative measure of purity

Classification



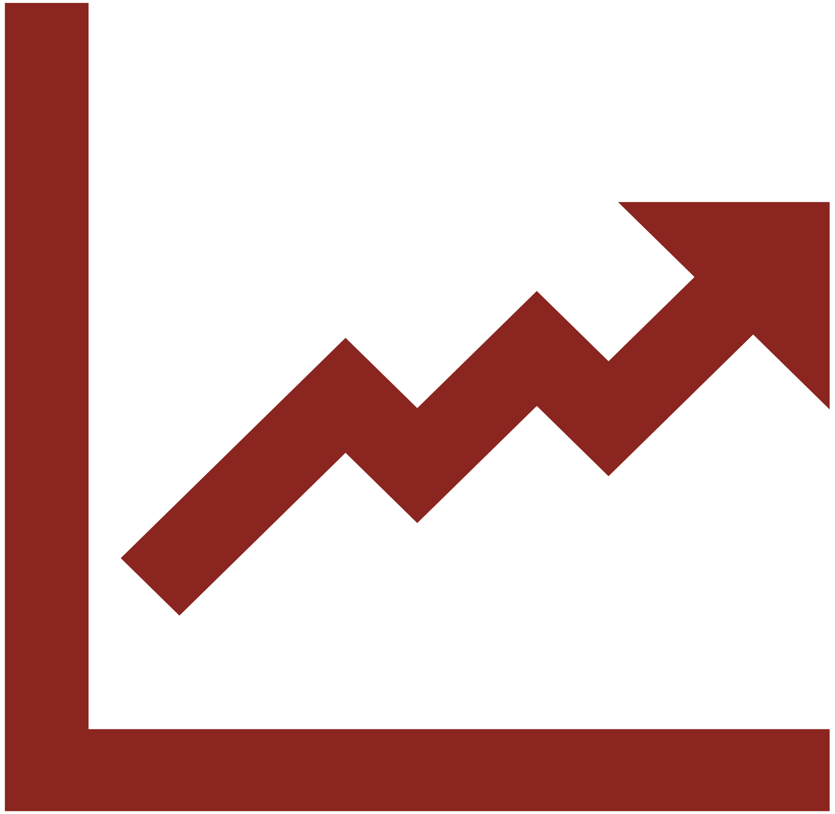
- Fully grown classification tree
- Nodes show
  - Proportion of classes
  - Mode/prediction of node

Tree vs Linear model



- Top row: Linear classifier provides a better fit than trees for a linear space
- Bottom row: Trees provide a better fit for non-linear space

Ensembles – Improving trees



- Biggest problem with building a decision tree is high variance
- Solution to this is ensembling
- Use multiple trees to get more accurate predictions and lower the variance

Bagging – Bootstrap aggregation

Step 1

- Bootstrap the data and create dataset 1, build dec tree 1

Step 2

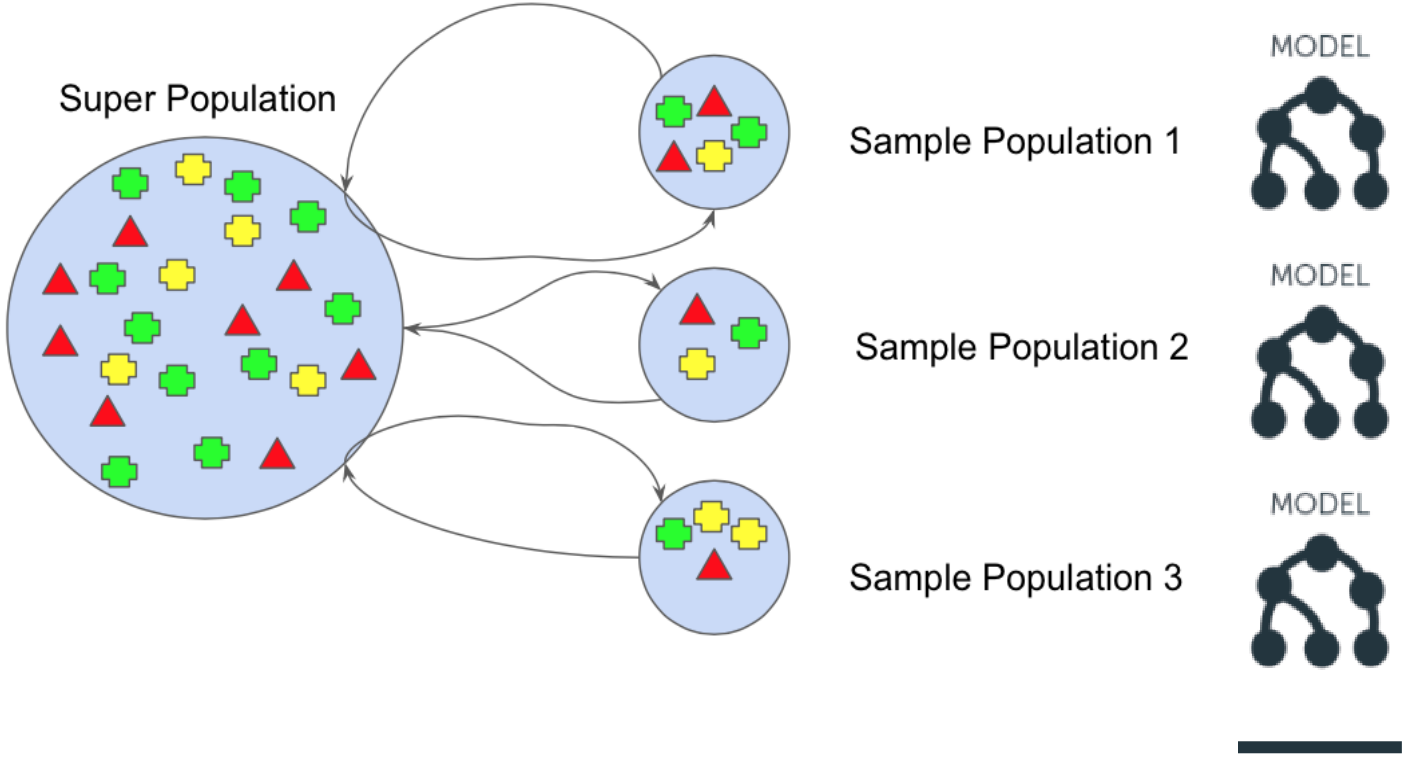
- Bootstrap the data and create dataset 2, build dec tree 2

Step n

- Bootstrap the data and create dataset n, build dec tree n

Final step

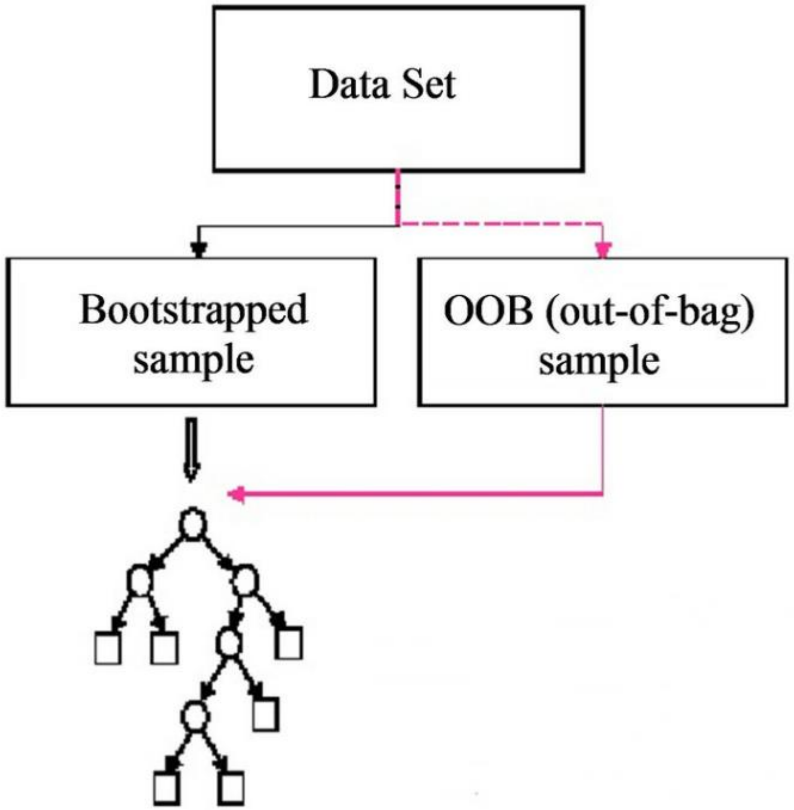
- Aggregate predictions
- Regression (mean) , Classification (mode



Aggregate predictions

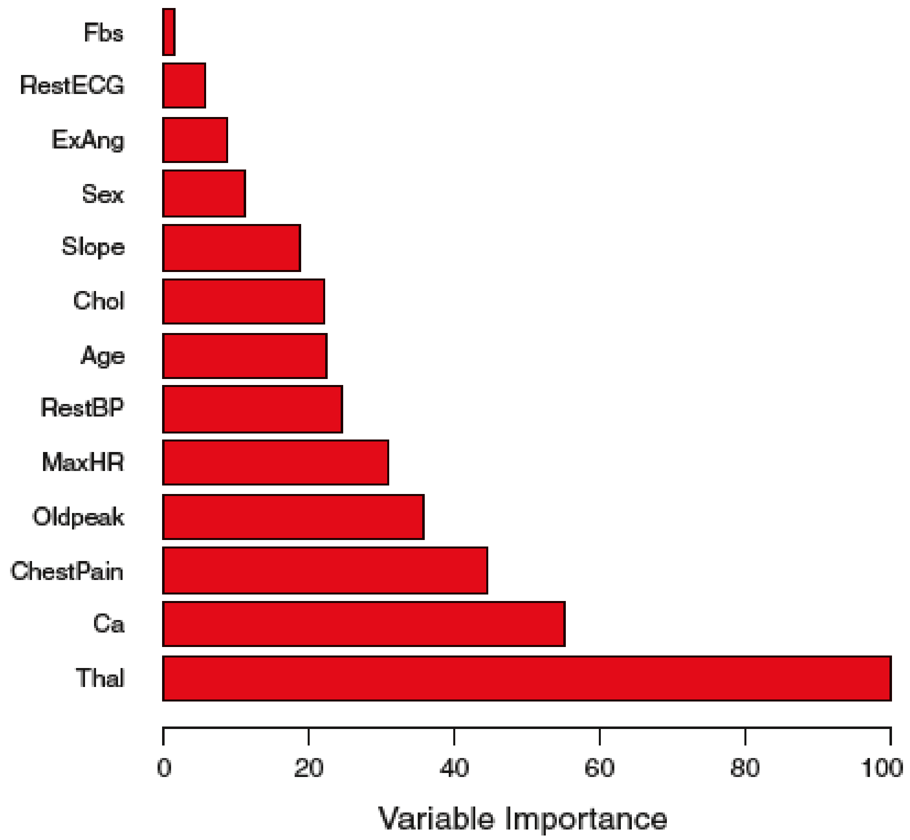
Ensembles – Improving trees

- OOB – Out of Bag observations
- Bootstrap method uses 2/3 of the data
- Keep 1/3 for the test set



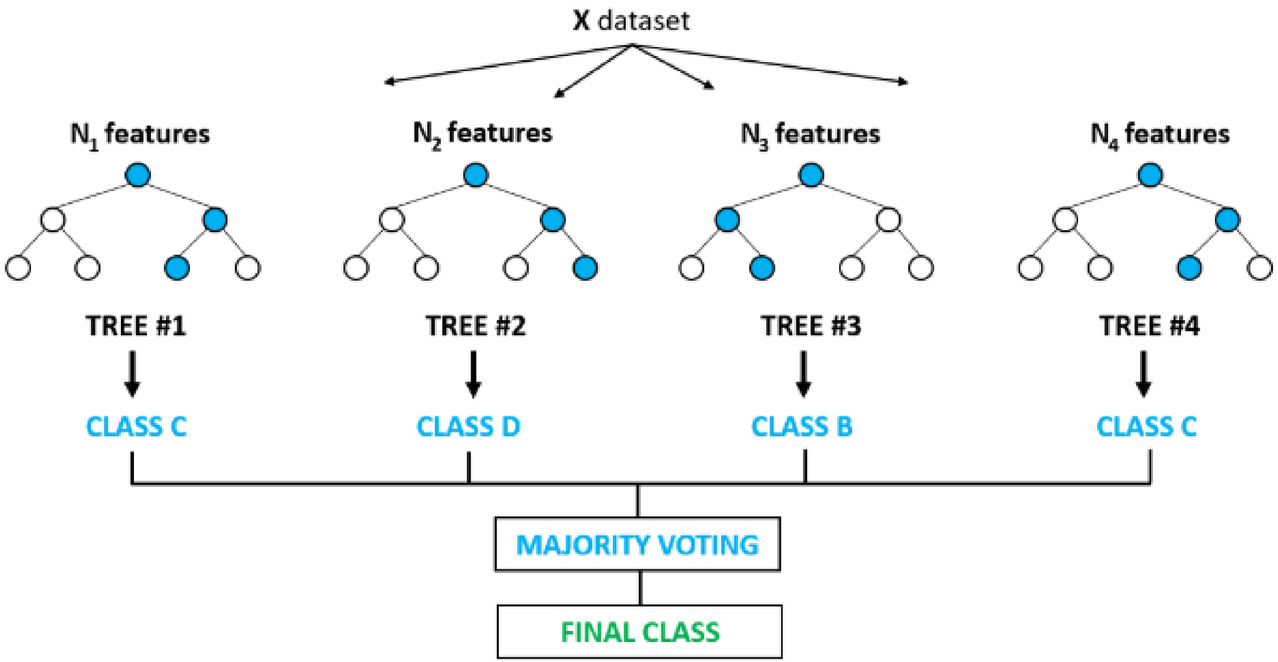
Ensemble interpretation

- Interpretation of the features is lost because we have many trees
- Different trees and different features combine to give the aggregated prediction
- Remove one feature and measure how much error changes
- Importance is relative to the most important predictor



Random Forest

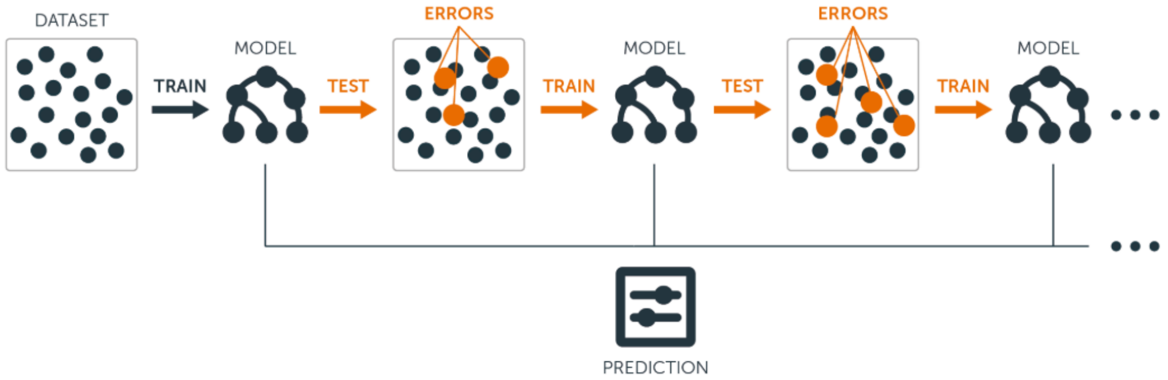
- Similar to Bagging
- Difference is in how we make our splits (which features to consider)
- At each split, we take a new subset of N features to choose from
- Regression subset: $^{N}/_{3}$
- Classification subset: $\sqrt{N}$



Aggregate predictions

Gradient boosted trees

- Difference is trees are sequential and dependent
- Residual output of the first tree is the input to the next tree
- Typically use short trees (stumps)
- Slow learner progresses to become powerful
- Learning rate parameter
- Regression or classification tasks

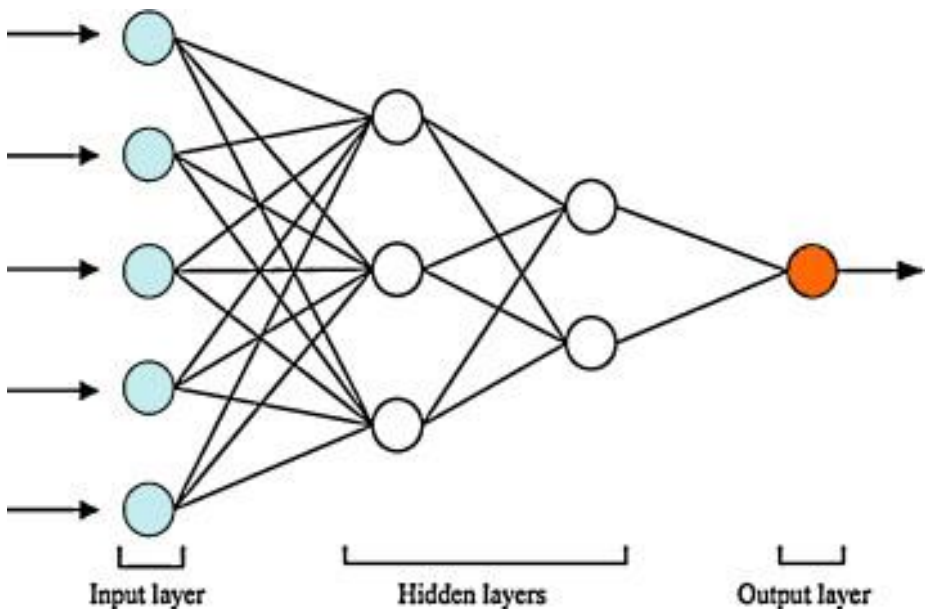


Learning rate parameter
$\lambda = 0.01 \; or \; 0.001$

Neural networks

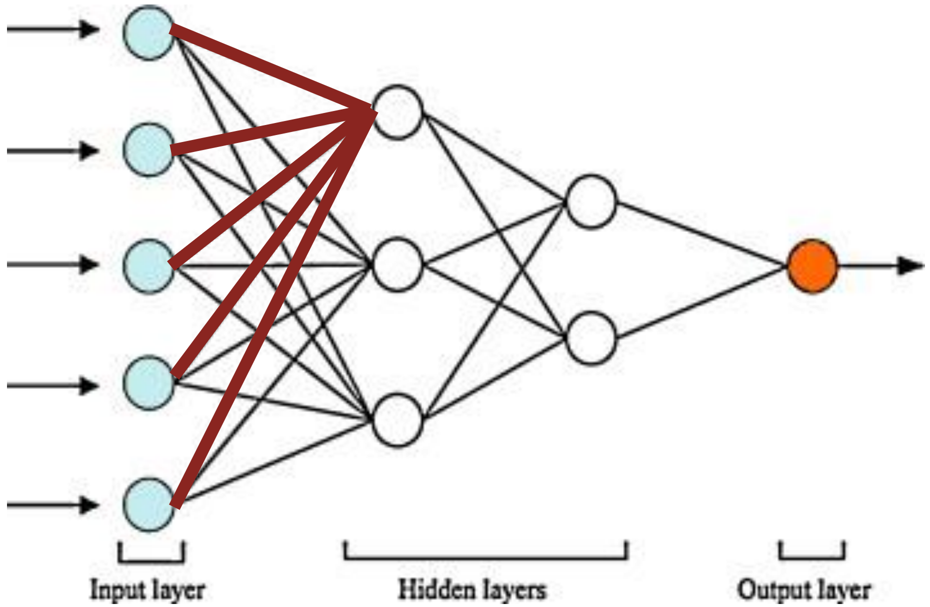
Neural net architecture

- Biological motivation for this architecture
- Each unit or node transmits information from a previous node to the next node



Neural net architecture

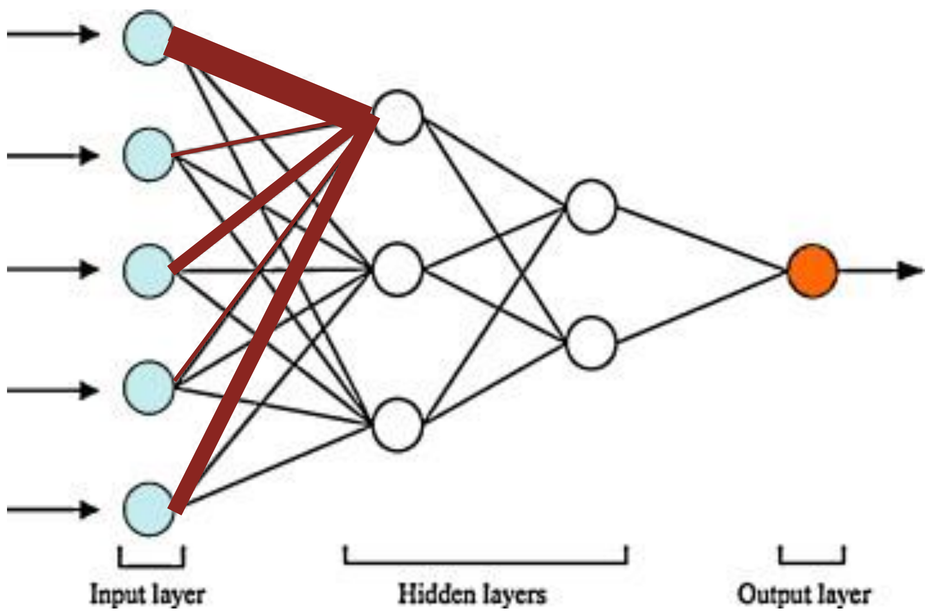
Explanatory
Variables



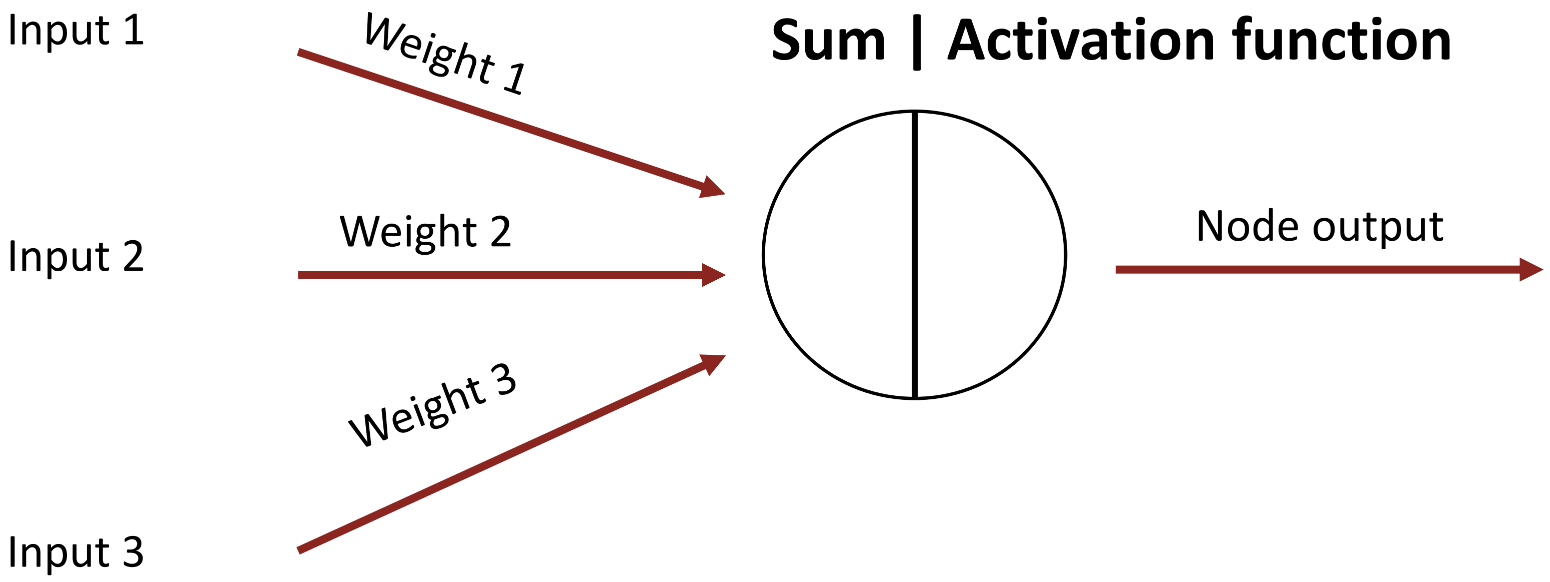
Target
Variable

Neural net architecture

- Input is a weighted sum of the previous node outputs

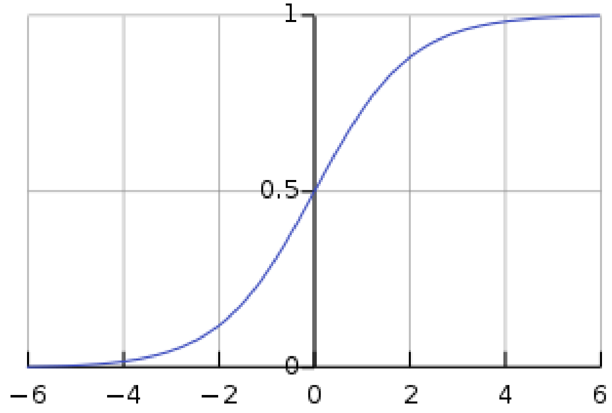


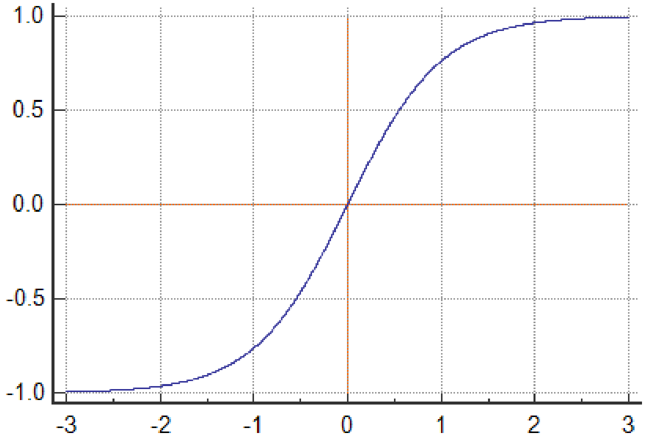
Neuron details



Activation function

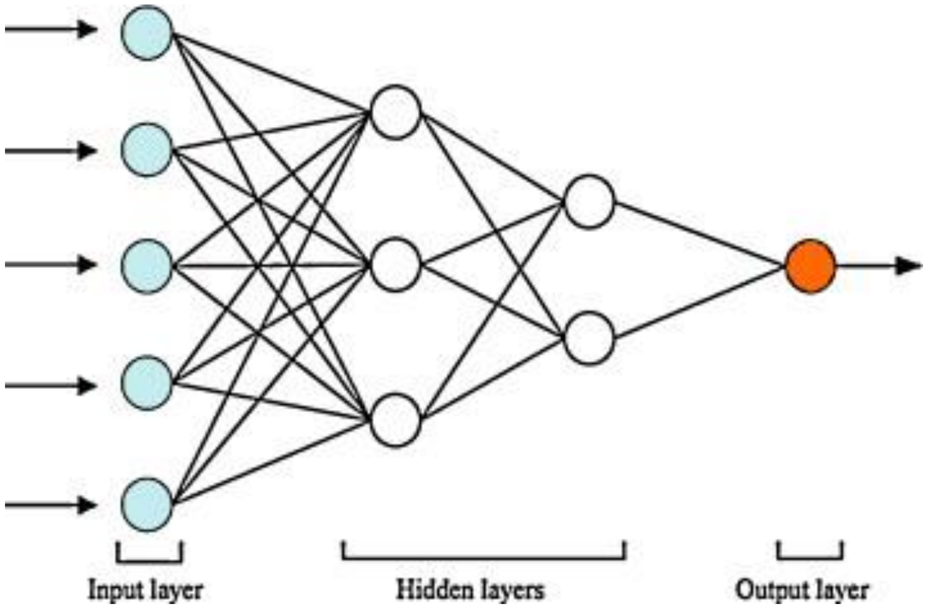
- Introduces non linearity into the model
- Sigmoid: output has range 0 to 1
- Tanh: output has range -1 to +1





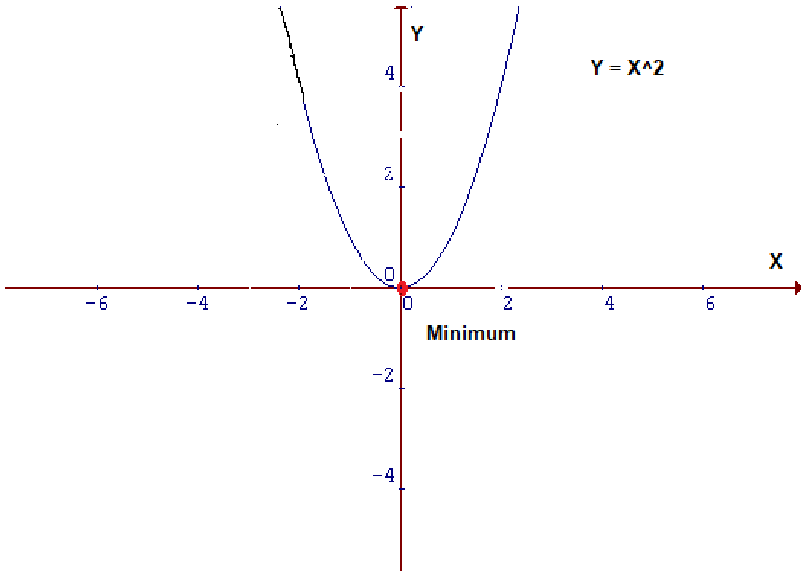
Cost function

- Cost function or loss function evaluates the performance of our network
- Goal is to find the weights that lead to the lowest error



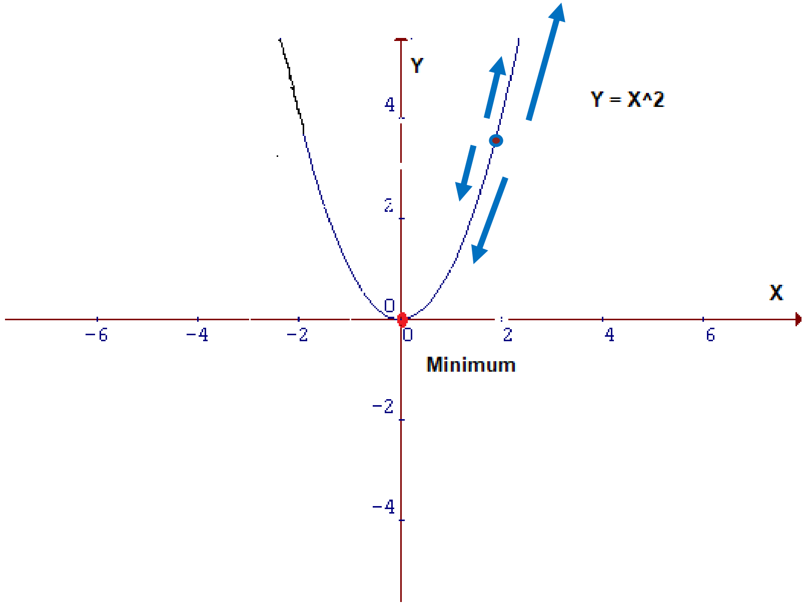
Minimizing a function

- To minimize the function, we need to find the value of x that produces the lowest value of y
- In higher dimensions we can use Gradient Descent



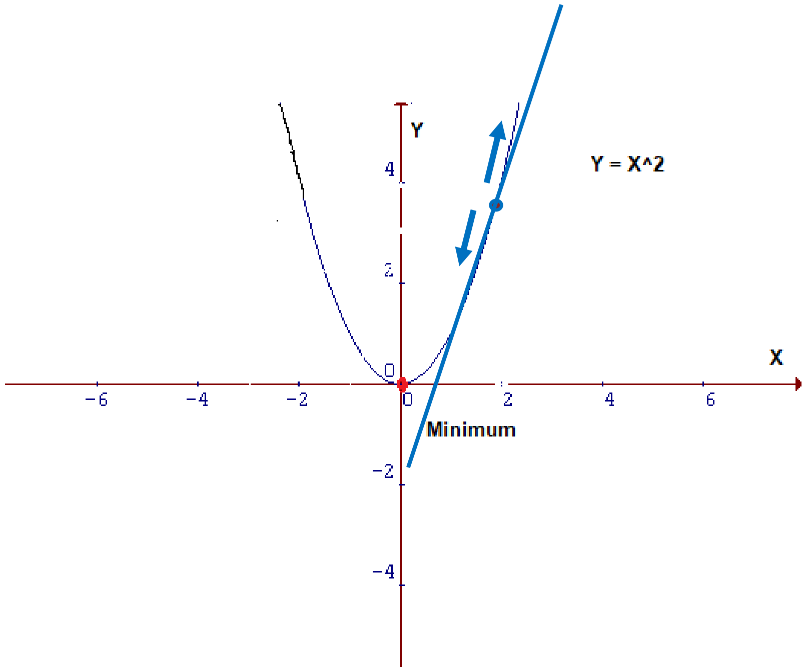
Gradient Descent

- How to find the minimum
- Decide which direction
- Decide how big of a step



Gradient Descent

- How to find the minimum
- Decide which direction
- Decide how big of a step



Neural net concepts

- Matrix/vector multiplication

$$\begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ z_1 & z_2 & z_3 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} a_x & a_y & a_z \end{bmatrix}$$

$$a_x = a_1 x_1 + a_2 x_2 + a_3 x_3$$

$$a_y = a_1 y_1 + a_2 y_2 + a_3 y_3$$

$$a_z = a_1 z_1 + a_2 z_2 + a_3 z_3$$

Appendix