8. Decision trees

Shabana K M
PhD Research Scholar
Computer Science and Engineering

IIT Palakkad

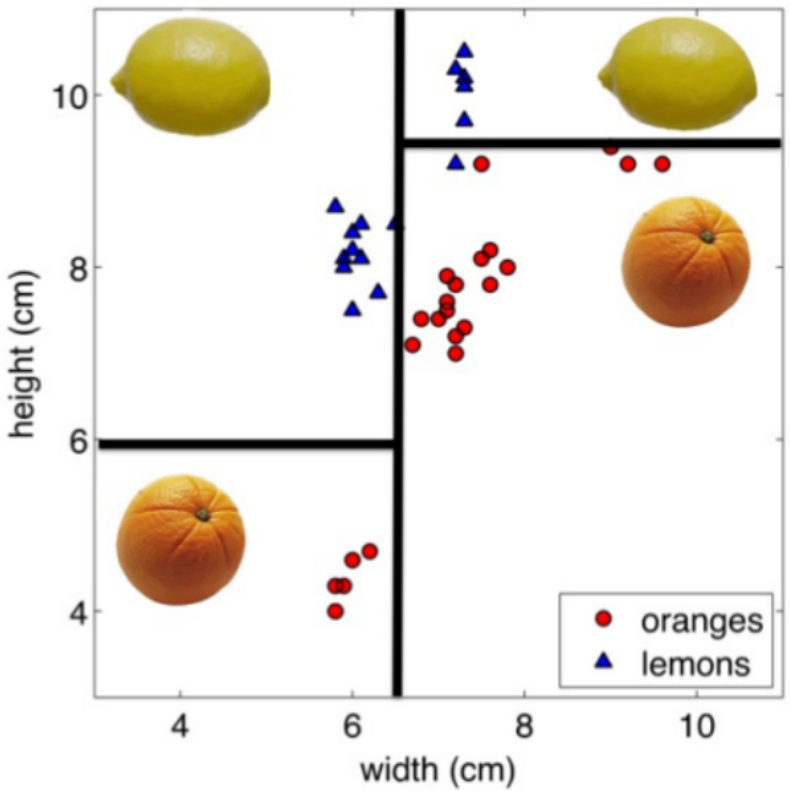
13 November 2021

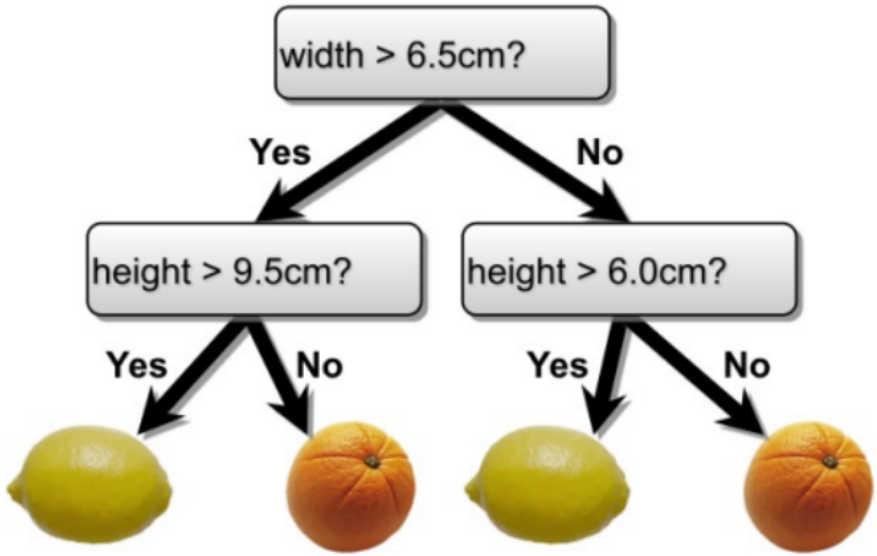
INDIAN INSTITUTE
OF TECHNOLOGY
PALAKKAD

Recap

- Evaluation metrics for classification
  - accuracy
  - confusion matrix
  - precision, recall, F1 score
  - ROC and area under curve
- Decision trees
  - how a decision tree works
  - learning a decision tree
  - selecting the splitting attribute in classification tree

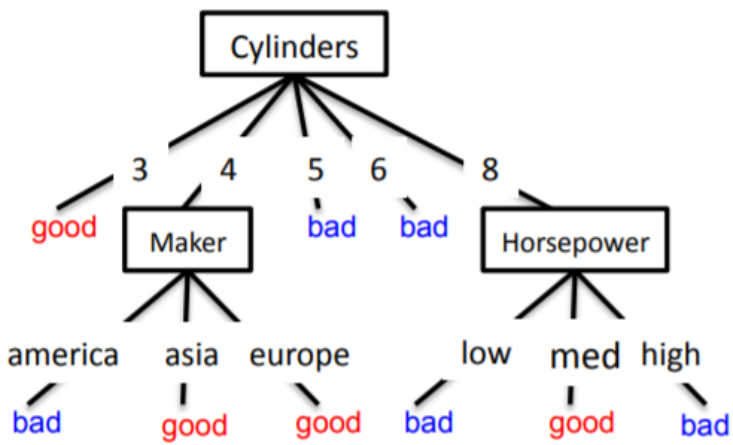
Decision trees



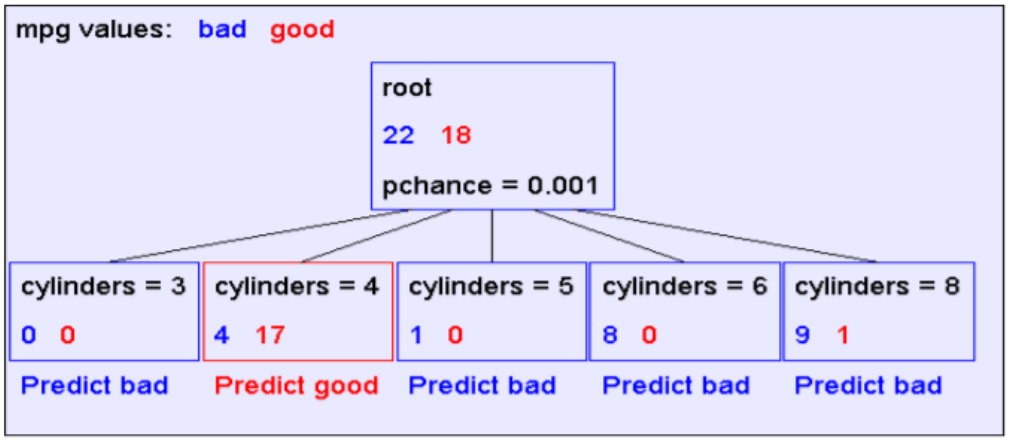


Greedily learning trees using recursion

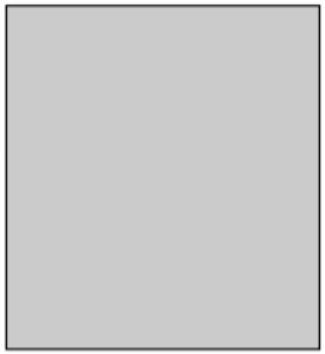
mpg	cylinders	displacement	horsepower	weight	acceleration	modelyear	maker
good	4	low	low	low	high	75to78	asia
bad	6	medium	medium	medium	medium	70to74	america
bad	4	medium	medium	medium	low	75to78	europe
bad	8	high	high	high	low	70to74	america
bad	6	medium	medium	medium	medium	70to74	america
bad	4	low	medium	low	medium	70to74	asia
bad	4	low	medium	low	low	70to74	asia
bad	8	high	high	high	low	75to78	america
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
bad	8	high	high	high	low	70to74	america
good	8	high	medium	high	high	79to83	america
bad	8	high	high	high	low	75to78	america
good	4	low	low	low	low	79to83	america
bad	6	medium	medium	medium	high	75to78	america
good	4	medium	low	low	low	79to83	america
good	4	low	low	medium	high	79to83	america
bad	8	high	high	high	low	70to74	america
good	4	low	medium	low	medium	75to78	europe
bad	5	medium	medium	medium	medium	75to78	europe



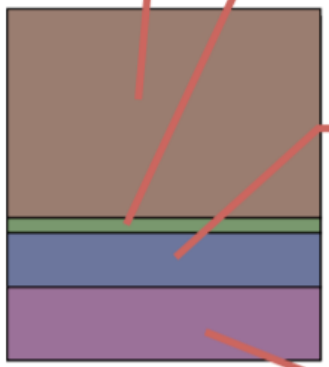
Greedily learning trees using recursion



Take the Original Dataset..



And partition it according to the value of the attribute we split on



Records in which cylinders = 4

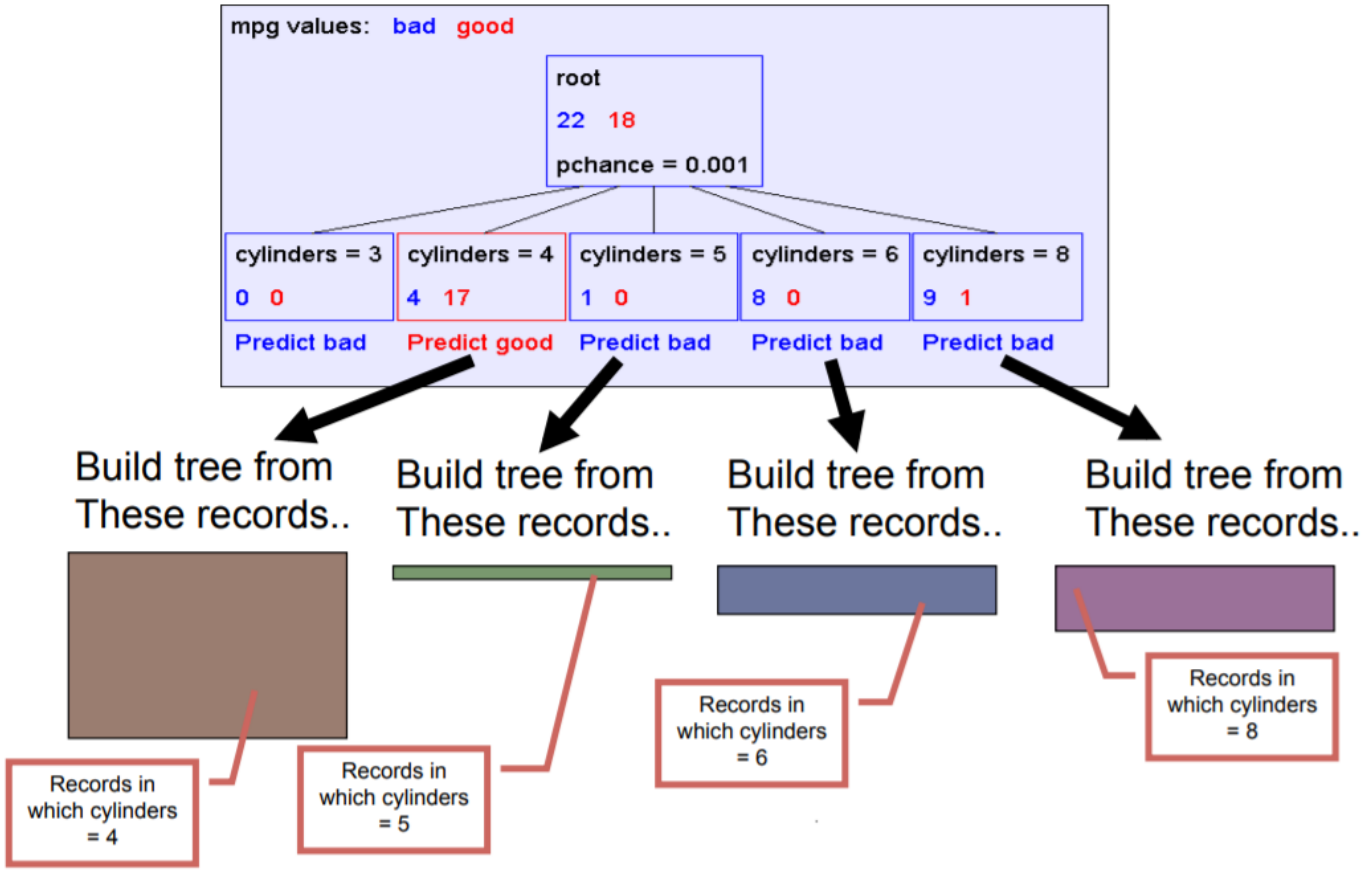
Records in which cylinders = 5

Records in which cylinders = 6

Records in which cylinders = 8

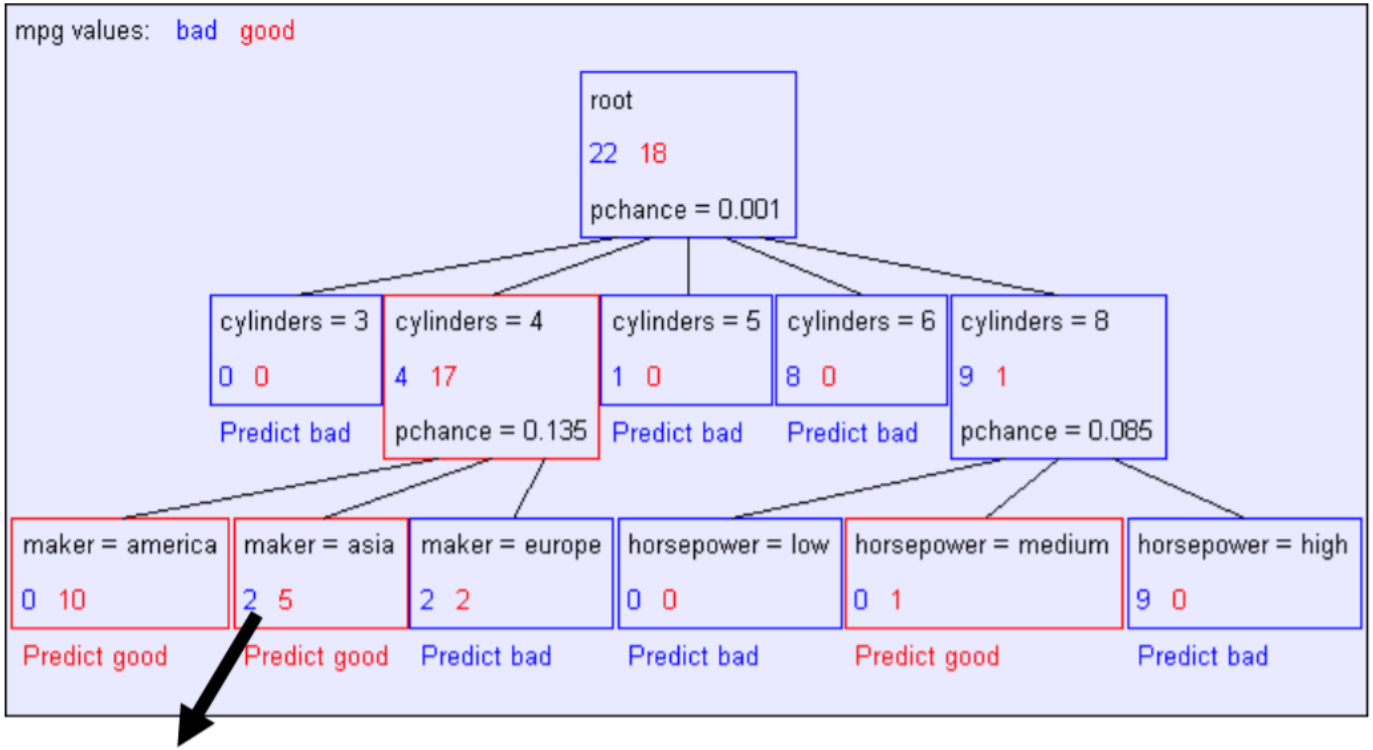
Greedily learning trees using recursion

Recursive step



Greedily learning trees using recursion

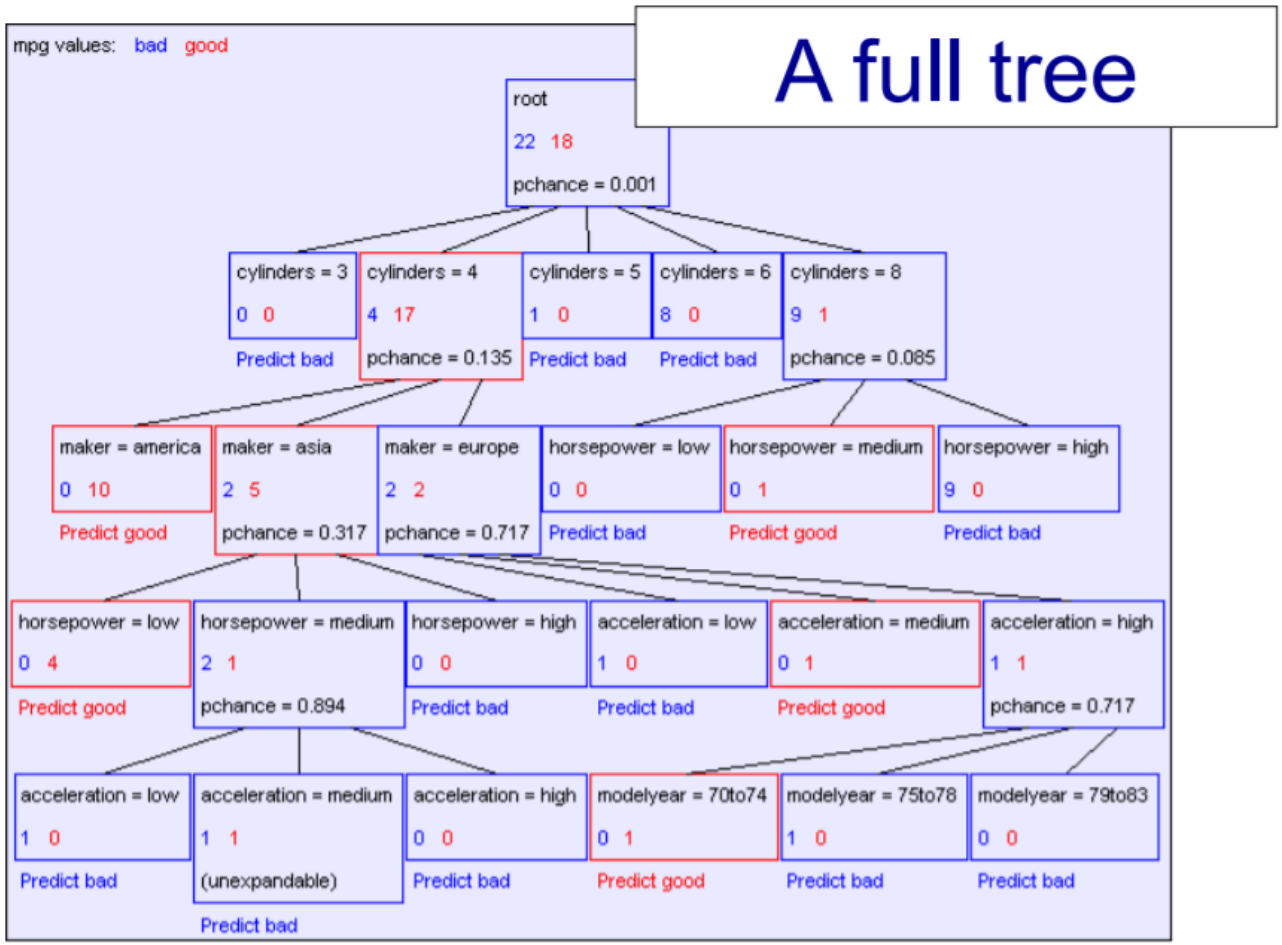
Second level of tree



Recursively build a tree from the seven records in which there are four cylinders and the maker was based in Asia

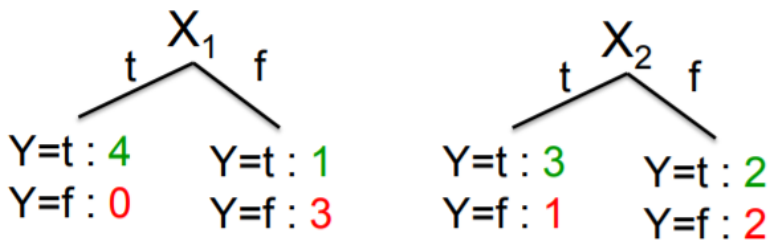
(Similar recursion in the other cases)

Greedily learning trees using recursion



Choosing the best attribute to split - discrete attributes

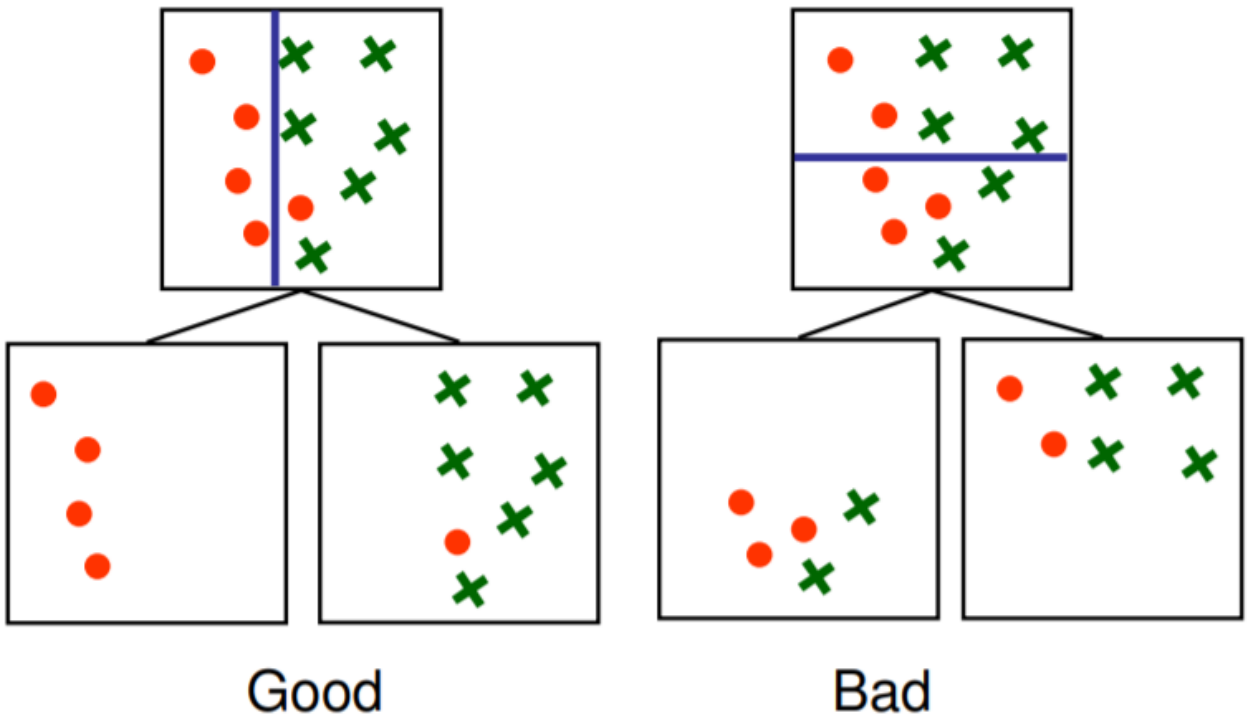
Should we split on X_1 or X_2?



X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F
F	T	F
F	F	F

Idea: Use `entropy` to quantify the `purity` of child nodes obtained through split

Comparing two splits



Splitting the node - discrete attributes

- **Classification tree:** Split the node to minimize entropy
- Let S be set of data points in a node, $c = 1, ..., C$ are labels:

$$H(S) = -\sum_{c=1}^{C} p(c) log_2(p(c))$$

where $p(c)$ is the proportion of data belonging to class c

  - Entropy $= 0$ if all samples are in the same class
  - Entropy is large if $p(1) = \cdots = p(C)$

P(Y=t) = 5/6
P(Y=f) = 1/6

H(Y) = - 5/6 $\log_2$ 5/6 - 1/6 $\log_2$ 1/6
= 0.65

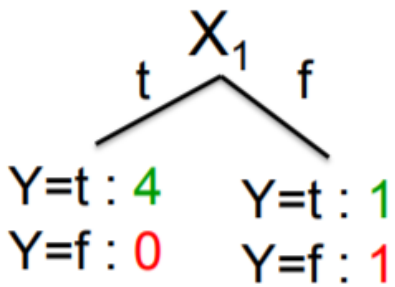
X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F

Conditional Entropy

`Conditional Entropy` $H(Y|X)$ of a random variable Y conditioned on a random variable X,

$$H(Y|X) = -\sum_{i=1}^{k} P(X = x_i) H(Y|X = x_i)$$

Example:



$P(X_1=t) = 4/6$

$P(X_1=f) = 2/6$

$H(Y|X_1) = -\ 4/6\ (1 \log_2 1 + 0 \log_2 0)$

$-\ 2/6\ (1/2 \log_2 1/2 + 1/2 \log_2 1/2)$

$= 2/6$

X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F

Information gain

- The average entropy of a split $S \to S_1, S_2$

$$\frac{|S_1|}{|S|} H(S_1) + \frac{|S_2|}{|S|} H(S_2)$$

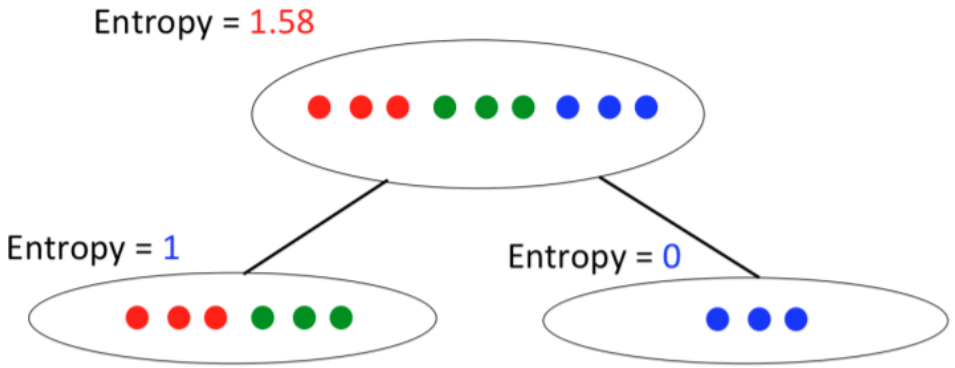
Information gain

- measures how well a given attribute separates the training examples according based on their target value
- $Gain(S, a)$ = expected reduction in entropy of Y due to splitting on attribute a

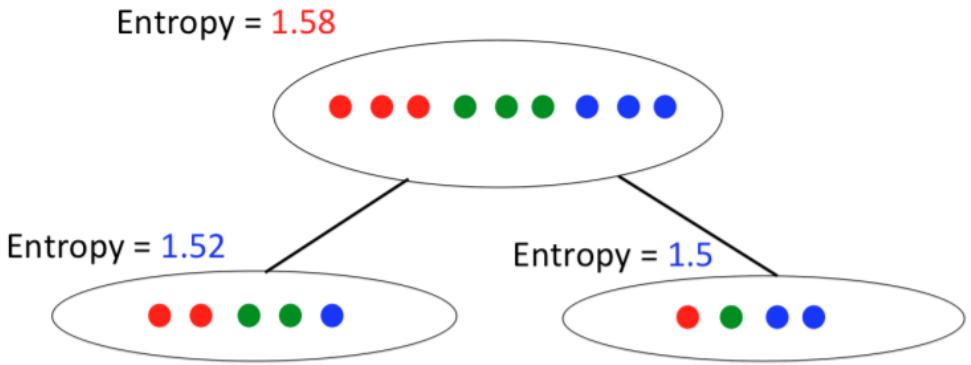
$$\begin{aligned} Gain(S, a) &= H(S) - H(S|a) \\ &= H(S) - \left(\frac{|S_1|}{|S|} H(S_1) + \frac{|S_2|}{|S|} H(S_2) \right) \end{aligned}$$

- the attribute with the maximal information gain is chosen for split

Deciding a good split based on information gain



Averaged entropy: 2/3*1 + 1/3*0 = 0.67
Information gain: 1.58 – 0.67 = 0.91



Averaged entropy: 1.51
Information gain: 1.58 – 1.51 = 0.07

Choosing the splitting attribute - example

Outlook	Temperature	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes

$$\begin{aligned}\text{Entropy(Play Golf)} &= -p(Yes) * log_2(p(Yes)) - p(No) * log_2(p(No)) \\ &= -0.6 * log_2(0.6) - 0.4 * log_2(0.4) \\ &= 0.971\end{aligned}$$

Choosing the splitting attribute - example

Outlook	Temperature	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes

Windy

$$Entropy(\text{Play Golf}|Windy = True) = -p(No|Windy = True) * log_2 p(No|Windy = True) - p(Yes|Windy = True) * log_2 p(Yes|Windy = True)$$
$$= -(2/3) * log_2(2/3) - (1/3) * log_2(1/3) = 0.918$$

$$Entropy(\text{Play Golf}|Windy = False) = -(2/7) * log_2(2/7) - (5/7) * log_2(5/7) = 0.863$$

$$\text{Gain(Windy)} = 0.971 - \Big((3/10) * 0.918 + (7/10) * 0.863\Big) = 0.091$$

Choosing the splitting attribute - example

Outlook	Temperature	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes

Humidity

$$Entropy(\text{Play Golf}|Humidity = High) = -(3/5) * log_2(3/5) - (2/5) * log_2(2/5) = 0.971$$

$$Entropy(\text{Play Golf}|Humidity = Normal) = -(1/5) * log_2(1/5) - (4/5) * log_2(4/5) = 0.722$$

$$\text{Gain(Humidity)} = 0.971 - \Big((5/10) * 0.971 + (5/10) * 0.722\Big) = 0.125$$

Choosing the splitting attribute - example

Outlook	Temperature	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes

Temperature

$$Entropy(\text{Play Golf}|Temperature = Hot) = -(1/3) * log_2(1/3) - (2/3) * log_2(2/3) = 0.918$$

$$Entropy(\text{Play Golf}|Temperature = Mild) = -(1/3) * log_2(1/3) - (2/3) * log_2(2/3) = 0.918$$

$$Entropy(\text{Play Golf}|Temperature = Cool) = -(1/4) * log_2(1/4) - (3/4) * log_2(3/4) = 0.811$$

$$\text{Gain}(\text{Temperature}) = 0.971 - \Big((3/10)*0.918 + (3/10)*0.918 + (4/10)*0.811\Big) = 0.096$$

Choosing the splitting attribute - example

Outlook	Temperature	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes

Outlook

$$Entropy(\text{Play Golf}|Outlook = Rainy) = -(3/4) * log_2(3/4) - (1/4) * log_2(1/4) = 0.811$$

$$Entropy(\text{Play Golf}|Outlook = Overcast) = - * log_2(1) = 0$$

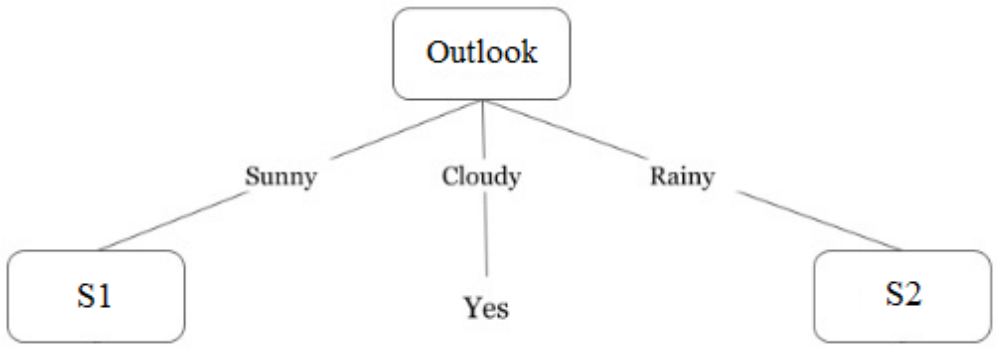
$$Entropy(\text{Play Golf}|Outlook = Sunny) = -(1/4) * log_2(1/4) - (3/4) * log_2(3/4)$$

$$= 0.811$$

$$\text{Gain(Outlook)} = 0.971 - \Big((4/10) * 0.811 + (4/10) * 0.811\Big) = 0.322$$

Choosing the splitting attribute - example

Outlook has the highest information gain



- The first split is performed on `Outlook`
- The node corresponding to `Outlook=Cloudy` is a leaf node as all the observations with `Outlook=Cloudy` have `Play Golf = Yes`
- $S1$ contains the observations for which `Outlook = Sunny`
- $S2$ contains the observations for which `Outlook = Rainy`
- Now build the tree recursively for $S1$ and $S2$

Gini impurity

- another measure used to select the splitting attribute
- measures the probability of a random sample being wrongly classified

$$\text{Gini impurity} = 1 - \sum_{i=1}^{n} (p_i)^2$$

where p_i is the probability of an observation belonging to a particular class

- Gini impurity takes a value between 0 and 1
- the Gini impurity of a pure node is zero
- lower the Gini Impurity, higher the homogeneity of the node

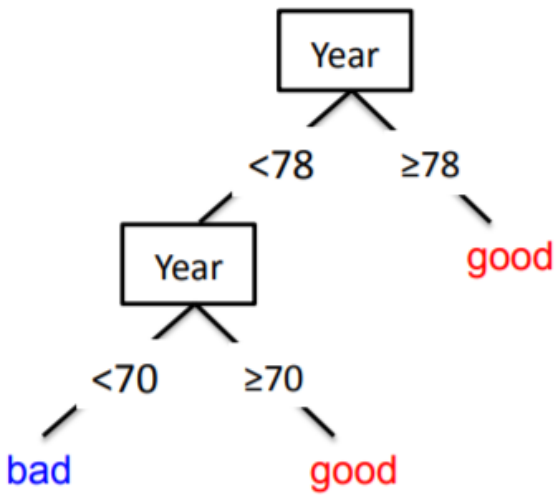
Splitting the node - continuous attributes

What should we do if some of the attributes are real-valued?

mpg	cylinders	displacemen	horsepower	weight	acceleration	modelyear	maker
good	4	97	75	2265	18.2	77	asia
bad	6	199	90	2648	15	70	america
bad	4	121	110	2600	12.8	77	europe
bad	8	350	175	4100	13	73	america
bad	6	198	95	3102	16.5	74	america
bad	4	108	94	2379	16.5	73	asia
bad	4	113	95	2228	14	71	asia
bad	8	302	139	3570	12.8	78	america
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
good	4	120	79	2625	18.6	82	america
bad	8	455	225	4425	10	70	america
good	4	107	86	2464	15.5	76	europe
bad	5	131	103	2830	15.9	78	europe

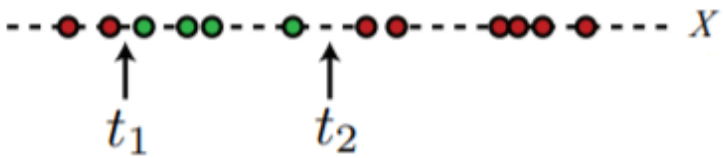
Threshold splits

- split on attribute a at value t
  - one branch: $X < t$
  - other branch: $X \geq t$
- allow repeated splits on the same variable along a path



Threshold splits

• Sort the data based on attribute a into $a_1, ..., a_m$ and consider means of each pair of neighbouring values as thresholds



Let S be the set of data points in a node and S_1 and S_2 be the splits obtained with $a < t$ and $a \geq t$ respectively

$$H(S|a,t) = \frac{|S_1|}{|S|} H(S_1) + \frac{|S_2|}{|S|} H(S_2)$$

$$Gain(S,a,t) = H(S) - H(S|a,t)$$

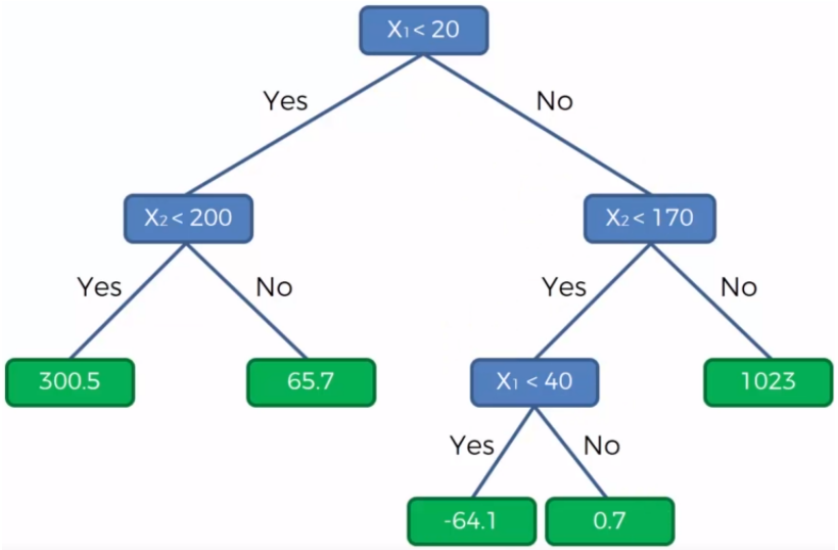
$$Gain^*(S,a) = \max_t Gain(S,a,t)$$

When to stop splitting?

1. All the data in the node belongs to a single class
   - declare the node to be a leaf and stop splitting
   - leaf will output the class of the data it contains

2. Several data points have exactly the same attributes even though they belong to different classes
   - cannot split any further
   - declare the node to be a leaf that outputs the majority class of the data points in that node

Regression trees



- target variable is continuous
- leaf node predicts the average response values for all training observations that fall in that group - `a constant`!
- `tree building:` select the split that minimizes the sum of the squared deviations from the mean in the two separate partitions

Learning decision trees

1. start with an empty tree
2. pick an attribute to split at a non-terminal node
   – use measures such as information gain, Gini impurity, etc. to select the attribute
3. split examples into groups based on attribute value
4. for each group
   - if no examples – return majority class from parent
   - else if all examples belong to the same class – return class
   - else if all examples have exactly the same attributes - return majority class from node
   - else loop to step 2

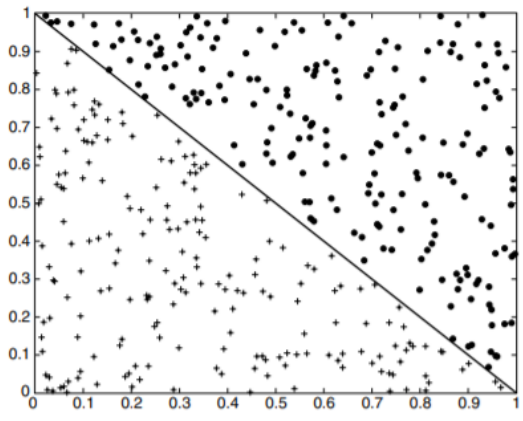
Decision trees

Strength

- nonlinear classifier
- better interpretability
- can naturally handle categorical features

Weakness

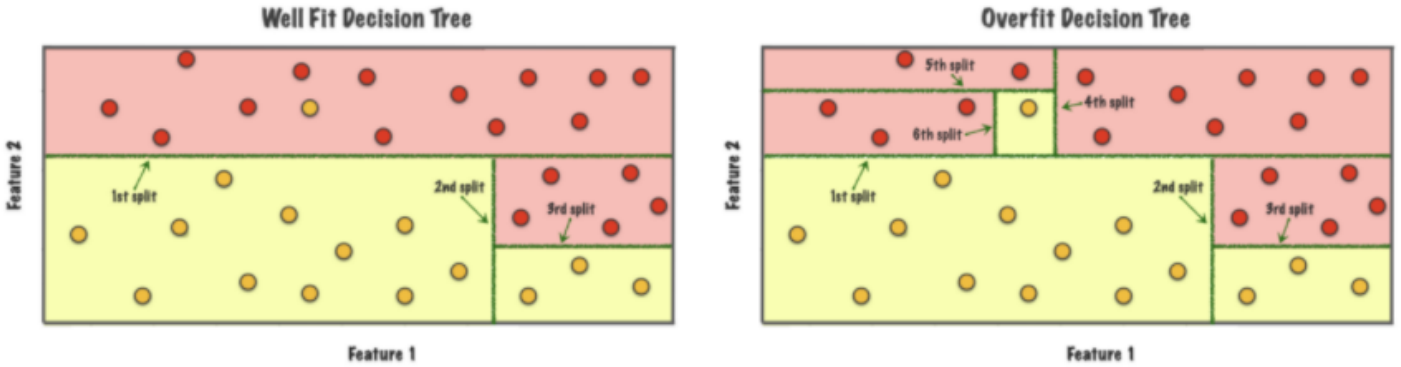
- overfitting
- cannot model complex relationships between continuous attributes
  - test conditions involve only a single attribute at a time



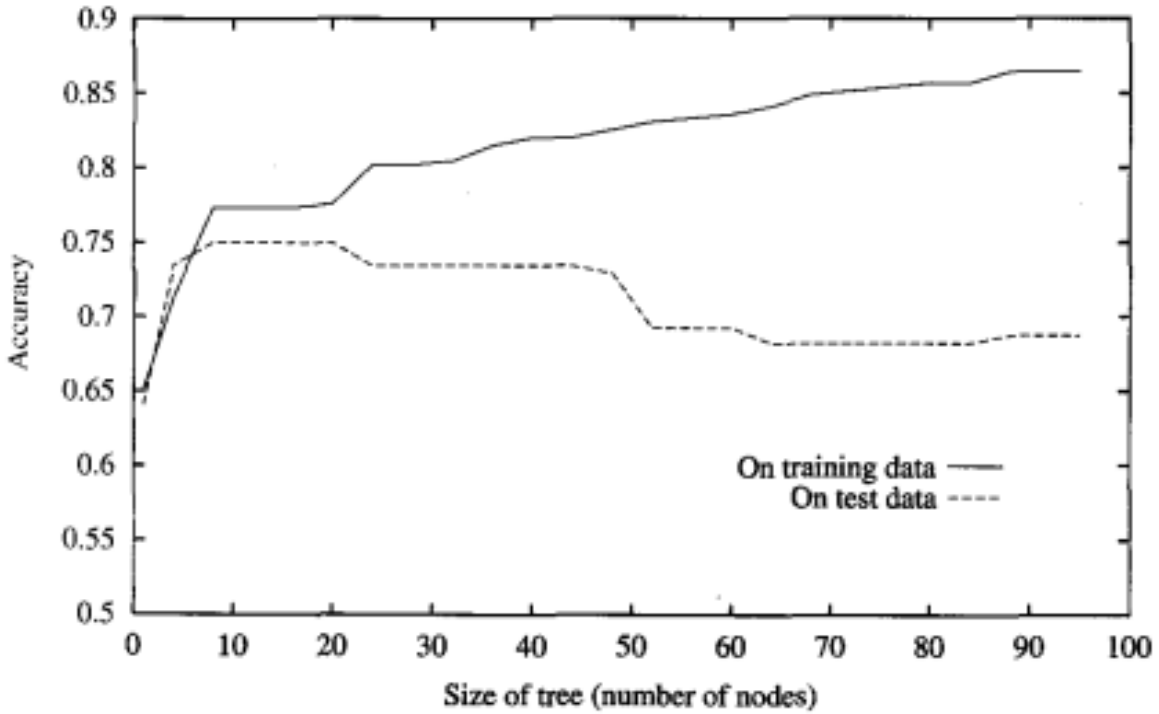
Decision trees: overfitting

Decision trees are prone to overfitting

- occurs when the splits in the tree are too specifically defined for the training data
- the leaf nodes of a tree can be expanded until it perfectly fits the training data - zero training error
- the decision tree may contain nodes that accidentally fit some of the noise points in the dataset - test error can be large



Decision trees: Overfitting



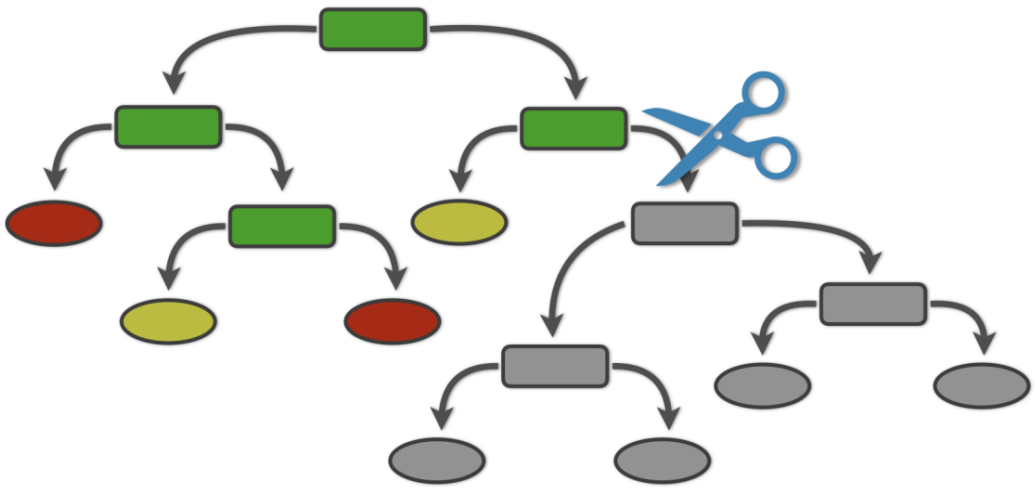
Avoiding overfitting in decision trees

1. **Early stopping**
   - stop the growth of a decision tree early before it overfits to the training data
   - a limit to the decision tree's growth can be specified in terms of
     1. Maximum tree depth
     2. Minimum number in node — minimum number of observations to appear in any child node for a split to be valid
     3. Minimum decrease in impurity

Avoiding overfitting in decision trees

2. Pruning

- reduces the size of decision trees by removing sections of the tree that provide little power to classify instances
- the original tree tested against pruned versions of it on an independent test set
- leaf nodes removed from the tree as long as the pruned tree performs better on the test data than the larger tree



Bias and variance

The error for any supervised learning algorithm comprises of three parts:

1. Bias
2. Variance
3. Noise

Bias

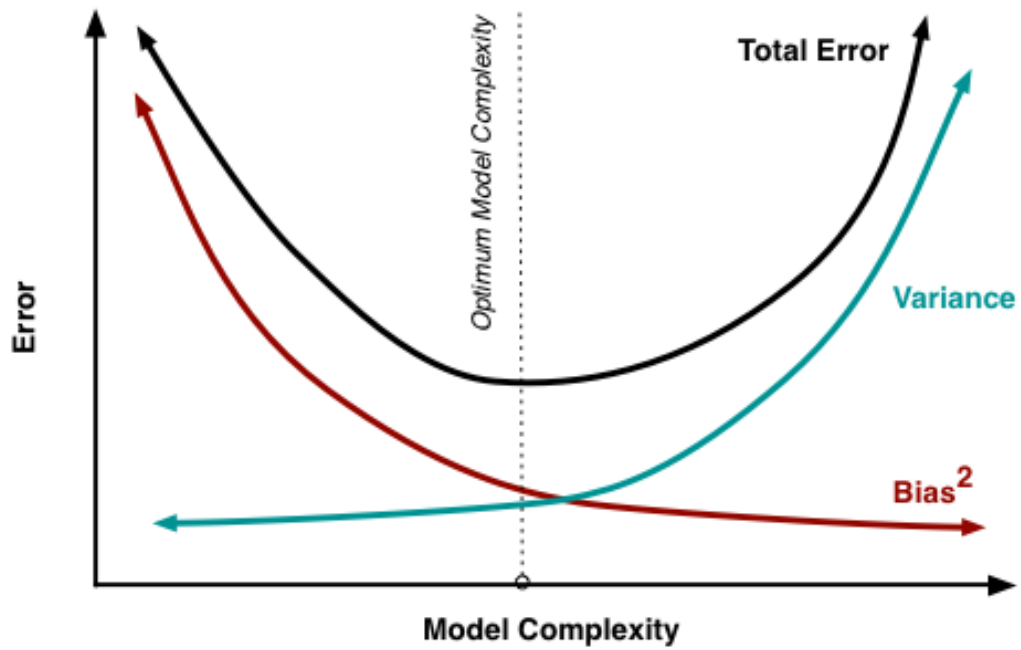
- error due to incorrect assumptions in the learning algorithm
- high bias can cause an algorithm to miss the relevant relations between features and target output - underfitting!

Variance

- error from sensitivity to small fluctuations in the training set
- high variance may result from an algorithm modeling the random noise in the training data - overfitting!

Bias-Variance tradeoff

- central problem in supervised learning
- need a model that accurately captures the patterns in its training data and also generalizes well to unseen data
- typically impossible to do both simultaneously
- increasing the bias will decrease the variance, and vice versa



References

1. https://www.cs.toronto.edu/~urtasun/courses/CSC411_Fall16/06_trees_handout.pdf
2. https://people.csail.mit.edu/dsontag/courses/ml16/slides/lecture11.pdf
3. http://www.stat.ucdavis.edu/~chohsieh/teaching/ECS171_Winter2018/lecture15.pdf
4. https://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/15381-s06/www/DTs.pdf
5. https://towardsdatascience.com/decision-trees-60707f06e836
6. https://www-users.cse.umn.edu/~kumar001/dmbook/ch4.pdf
7. https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff

Thanks Google for the pictures!