ALGONQUIN COLLEGE

CST8502 MACHINE LEARNING

Week 3

Classification – Decision Trees

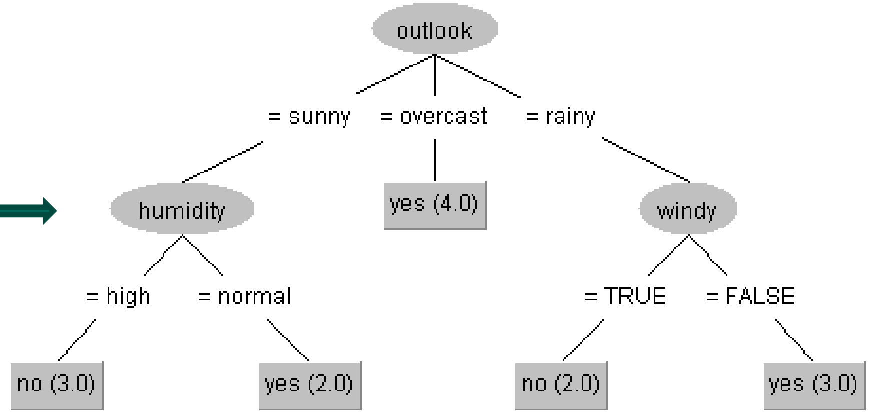
Professor : Dr. Anu Thomas

Email: thomasa@algonquincollege.com

Office: T315

Decision Trees

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
overcast	cool	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no



Decision Trees

- How to construct decision trees?
- How to avoid overfitting?

Decision Trees

- Decision tree is a tree where:
  - each node represents a feature (attribute)
  - each branch represents a decision (rule)
  - each leaf represents an outcome (categorical or continuous values)

Decision Tree

- One of the most popular ML algorithms
- Used for both classification and regression

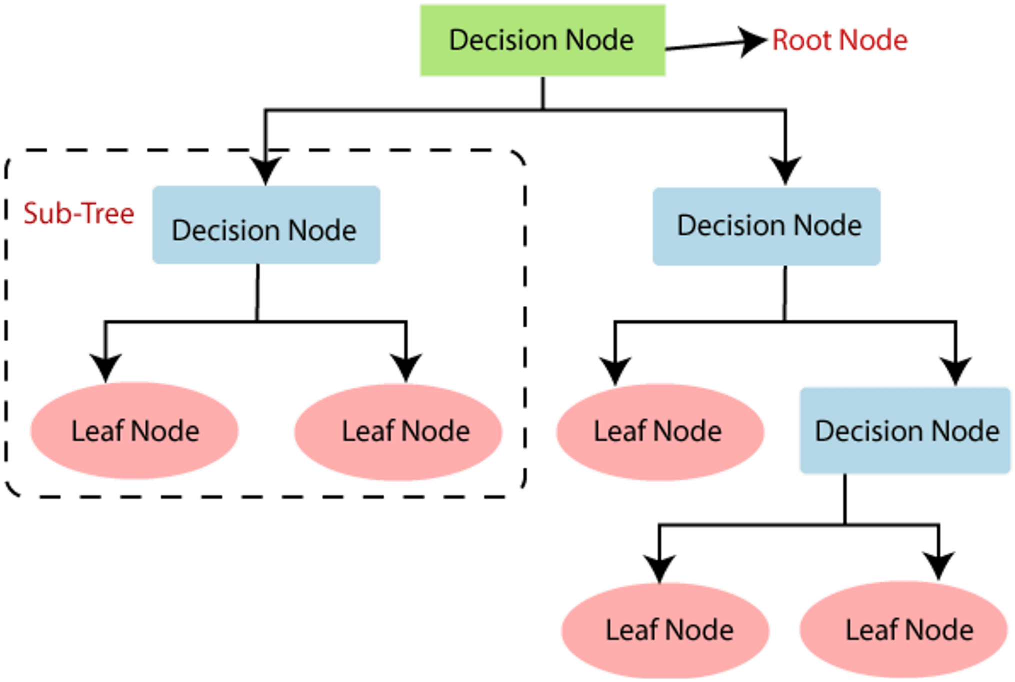


Image taken from ://www.almabetter.com/bytes/tutorials/data-science/decision-tree

Decision Tree Algorithms

- ID3 (Iterative Dichotomiser 3)
  - Uses Entropy function and Information gain as metrics
- CART (Classification and Regression Trees)
  - Uses Gini Index as metric

Classification using ID3 Algorithm

Weather Dataset

Based on weather conditions, predict Y or N for "Play".

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
overcast	cool	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no

Entropy

- Measure of the amount of impurity or uncertainty in the dataset

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

Where S – current dataset for which entropy is being calculated

C – set of classes in S Example: $C = \{yes, no\}$

$p(c)$ – The proportion of the number of elements in class c to the number of elements in S

In ID3, entropy is calculated for each remaining attribute. The attribute with the smallest entropy is used to split the set S on the current iteration.

Information Gain

- Measure of the difference in entropy from before to after the set S is split on an attribute A.
- Measure on how much uncertainty in S was reduced after splitting S on attribute A

Information Gain

$$IG(A,S) = H(S) - \sum_{t \in T} p(t)H(t)$$

Where $H(S)$ - *Entropy of set S*

T – Subset created by splitting S by attribute A

$p(t)$ – The proportion of the number of elements in t to the number of elements in S

$H(t)$ – Entropy of subset t

Metrics for Weather dataset

Steps

1. Compute the entropy for the dataset
2. For every attribute:
   i. Calculate entropy for all categorical values
   ii. Take weighted average for the current attribute
   iii. Calculate gain for the current attribute
3. Pick the attribute with highest information gain
4. Repeat until we get the tree we desired

Entropy for Weather dataset

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

Out of 14 instances, 9 are classified as Yes and 5 as No

$$P_{Yes} = -\frac{9}{14} * \log_2 \frac{9}{14} = 0.41$$

$$P_{No} = -\frac{5}{14} * \log_2 \frac{5}{14} = 0.53$$

$$H(S) = P_{Yes} + P_{No} = 0.94$$

Entropy of Outlook feature of Weather dataset

- $H(Outlook = Sunny) = -\frac{2}{5} * \log_2\frac{2}{5} - \frac{3}{5} * \log_2\frac{3}{5} = 0.5288 + 0.4422 = 0.971$
- $H(Outlook = Overcast) = -\frac{4}{4} * \log_2\frac{4}{4} - \frac{0}{4} * \log_2\frac{0}{4} = 0$
- $H(Outlook = Rainy) = -\frac{3}{5} * \log_2\frac{3}{5} - \frac{2}{5} * \log_2\frac{2}{5} = 0.4422 + 0.5288 = 0.971$
- $Average\ Entropy\ for\ Outlook$
- $M(Outlook) = \frac{5}{14} * 0.971 + \frac{4}{14} * 0 + \frac{5}{14} * 0.971 = 0.6936$
- Gain$(Outlook) = H(S) - M(Outlook) = 0.94 - 0.6936 = 0.2464$

Entropy of Windy feature of Weather dataset

- $H(Windy = False) = -\frac{6}{8} * \log_2\frac{6}{8} - \frac{2}{8} * \log_2\frac{2}{8} = 0.3113 + 0.5 = 0.8113$
- $H(Windy = True) = -\frac{3}{6} * \log_2\frac{3}{6} - \frac{3}{6} * \log_2\frac{3}{6} = 0.5+0.5=1$

- $Average\ Entropy\ for\ Windy$
- $M(Windy) = \frac{8}{14} * 0.8113 + \frac{6}{14} * 1 = 0.4636 + 0.4286 = 0.8922$

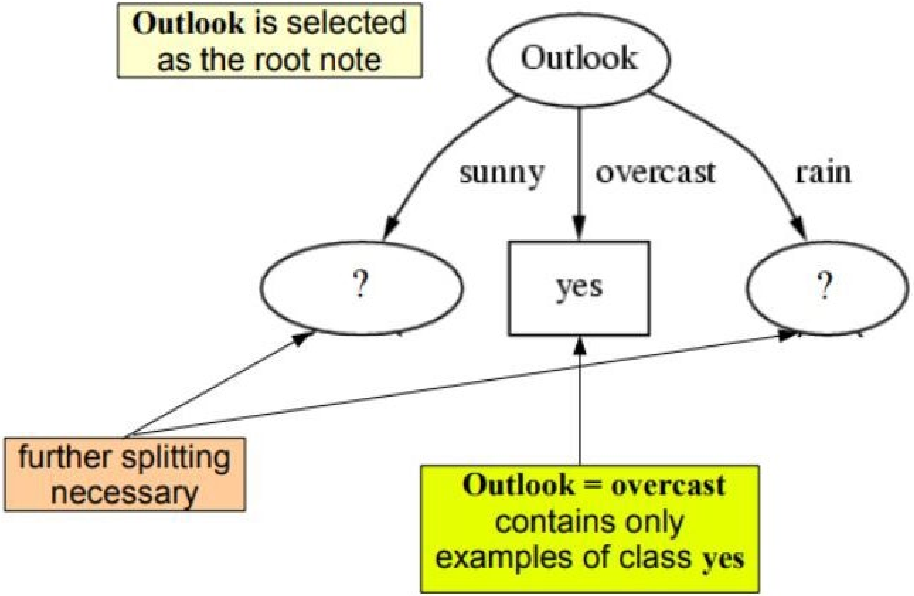
- Gain$(Windy) = H(S) - M(Windy) = 0.94 - 0.8922 = 0.0478$

Metrics Summary

Outlook		**Temperature**	
Average Entropy:	0.693	Average Entropy:	0.911
Information Gain:	**0.247**	Information Gain:	0.029
Humidity		Windy	
Average Entropy:	0.788	Average Entropy:	0.892
Information Gain:	0.152	Information Gain:	0.048

As Outlook has the highest Information Gain, our root node is **Outlook**

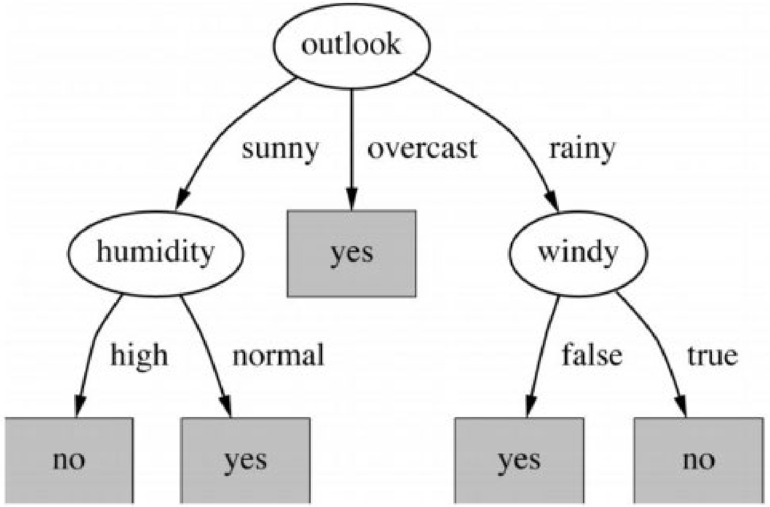
Initial Tree for Weather Dataset



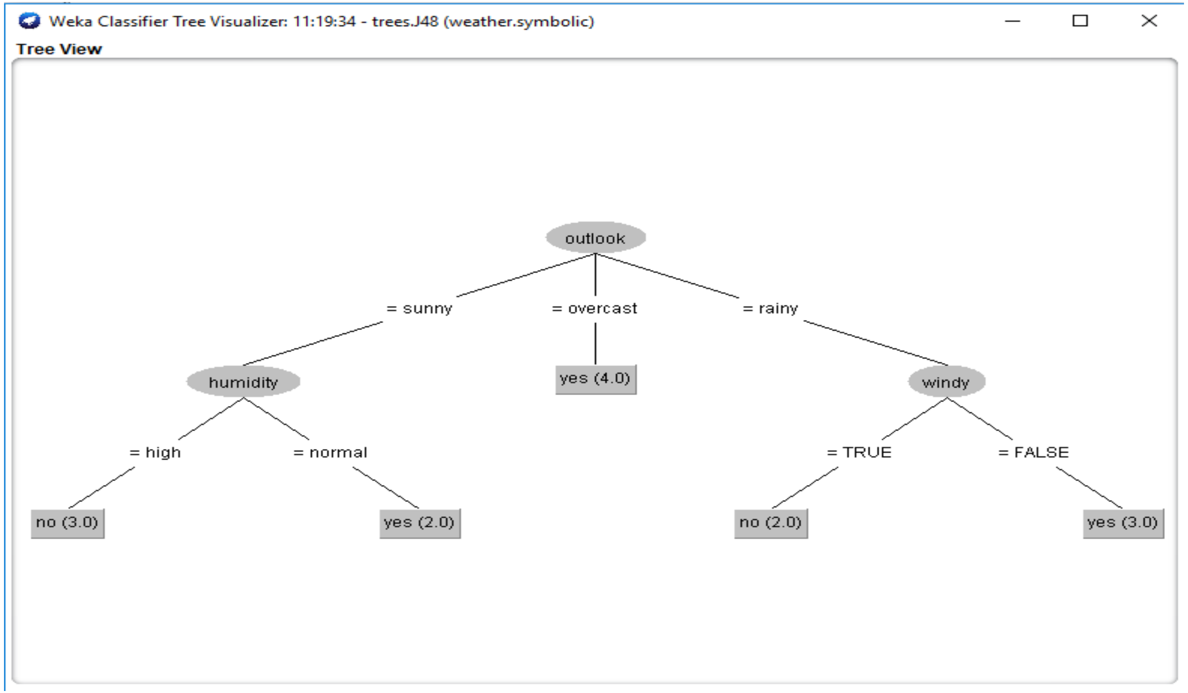
Developing Tree

- Repeat the same step for subtrees

Final decision tree



Weka Demo


Weka Classifier Tree Visualizer: 11:19:34 - trees.J48 (weather.symbolic)
Tree View
outlook
= sunny
= overcast
= rainy
humidity
yes (4.0)
windy
= high
= normal
= TRUE
= FALSE
no (3.0)
yes (2.0)
no (2.0)
yes (3.0)

Pruning

- A technique used to avoid overfitting
- Insignificant parts of the tree will be removed
- Gives a generalized tree by removing very specific, but insignificant nodes
- Two types
  - pre-pruning (early stopping)
  - post-pruning (cut back later)

Pre-pruning (Early stopping)

- Stops the decision tree growth before it becomes too complex
- Common techniques:
  - Limit maximum depth of tree
  - By setting minimum samples per split/leaf
  - Stop splitting if improvement in purity is too small
- Advantages
  - Prevent overfitting early
  - Faster training
  - Simpler trees

Post-pruning (pruning after full growth)

- Allows the tree to grow fully, then removes unnecessary branches
- How it works:
  - Use validation test to cut weak branches
  - Common techniques:
    - Reduced error pruning: Removes branches that do not significantly affect the overall accuracy
    - minimum leaf size: removes leaf nodes with fewer samples than a specified threshold
- Advantages
  - More accurate than pre-pruning
  - Reduces overfitting while keeping useful splits

Random Forest

- Ensemble learning method that builds multiple decision trees during training and combines their predictions by majority vote
- How it works – by Bagging – **B**ootstrap **Agg**regat**ing**
  - Bootstrap sampling – multiple random samples are taken with replacement
  - Random feature selection – at each split, a random subset of features is considered
  - Aggregation: Prediction by majority vote

References

- http://www.saedsayad.com/decision_tree.htm
- https://www.geeksforgeeks.org/machine-learning/pruning-decision-trees/