IFN509

Data Exploration and Mining

Week 9

Algorithms of Predictive Data Mining
Decision Tree

Prof Richi Nayak
r.nayak@qut.edu.au

School of Computer Science
Centre for Data Science
Faculty of Science and Engineering
https://research.qut.edu.au/adm

Recap: Predictive Modelling

- It is a **supervised learning approach**.
  - Due to the presence of the target attribute (or class labels) in past observations.
  - Analyses a dataset to determine essential characteristics in the presence of a target.
- Like human learning
  - Uses observations to form a model of the important characteristics of a phenomenon.
  - Uses generalizations to fit new data into a general framework.
- Several algorithms exist.
  - Decision tree, Neural Networks, Regression modelling, etc.

Learning Objectives: Week 9

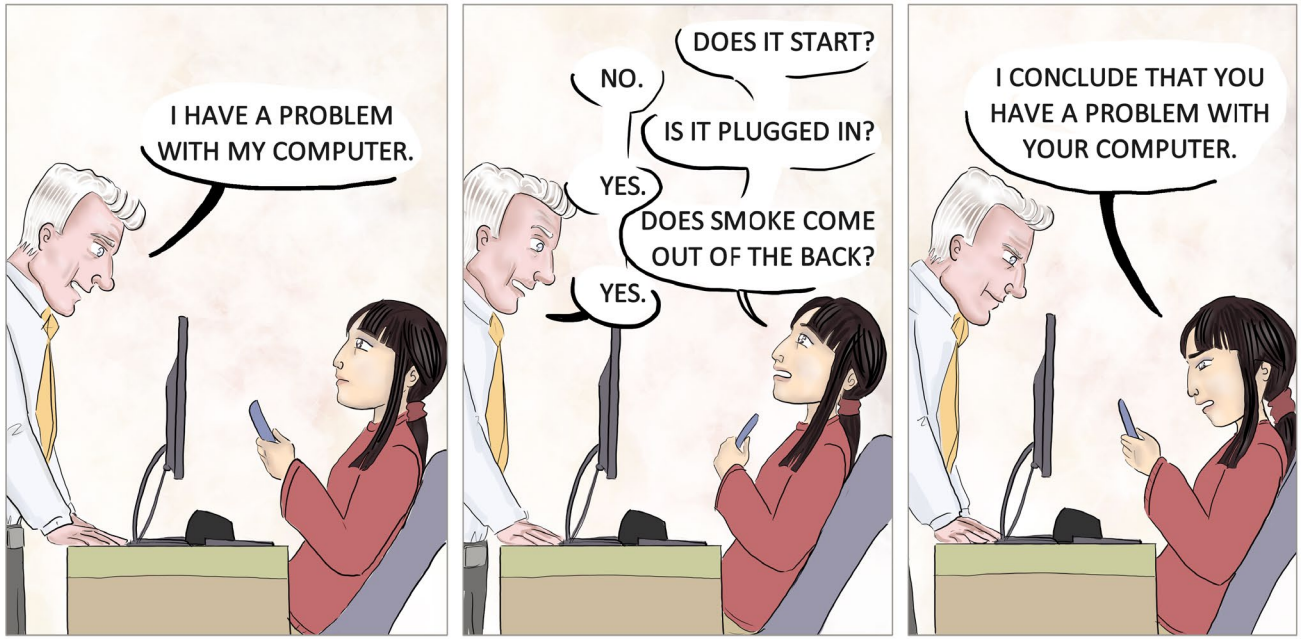
- Predictive Modelling Algorithm: Decision Tree for Classification
  - The process of building a tree
  - Bias vs Variance
  - Maximal vs Optimal trees
  - Benefits and Drawbacks
  - Tree Ensembles (Random Forest & XGBoost) – A quick tour
- Practical & Tutorial Sessions
  - Tutorial - Reflective Pen-and-Paper exercises
    - Clustering: proximity measures and finding clustering solutions
  - Practical Exercises
    - Data processing for clustering
    - Finding clustering solutions
      - With k-means
      - With agglomerative clustering
      - With k-prototypes
    - Profiling clustering solutions

What Should You Do in Week 9?

- Listen to the lecture recording and review the lecture slides (Decision Trees)
- Tutorial: Attempt the exercise questions related to the lecture on Clustering
- Practical: Complete practical tasks on Clustering Consult the Lecturer if you have any questions related to the subject.
- Assessment Item 2
  - Association mining: Should have finished
  - Clustering: Should start attempting
  - Register your (new) team on Blackboard

Decision Tree: Classification

Decision Tree Construction



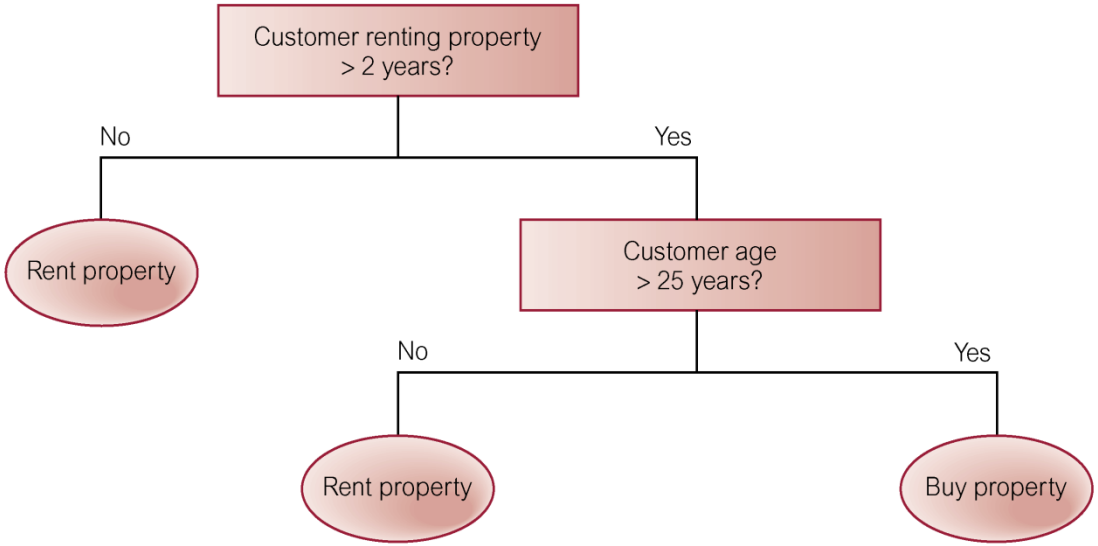
@https://livebook.manning.com/book/grokking-machine-learning/chapter-9/6

Classification by Decision Tree Induction

- Decision tree
  - A flow-chart-like tree structure
  - Root and internal nodes denote a test on an attribute
  - Branch represents an outcome of the test
  - Leaf nodes represent class labels or class distribution

Inputs = Duration of renting,
Age and many others
Target = Rent or Buy

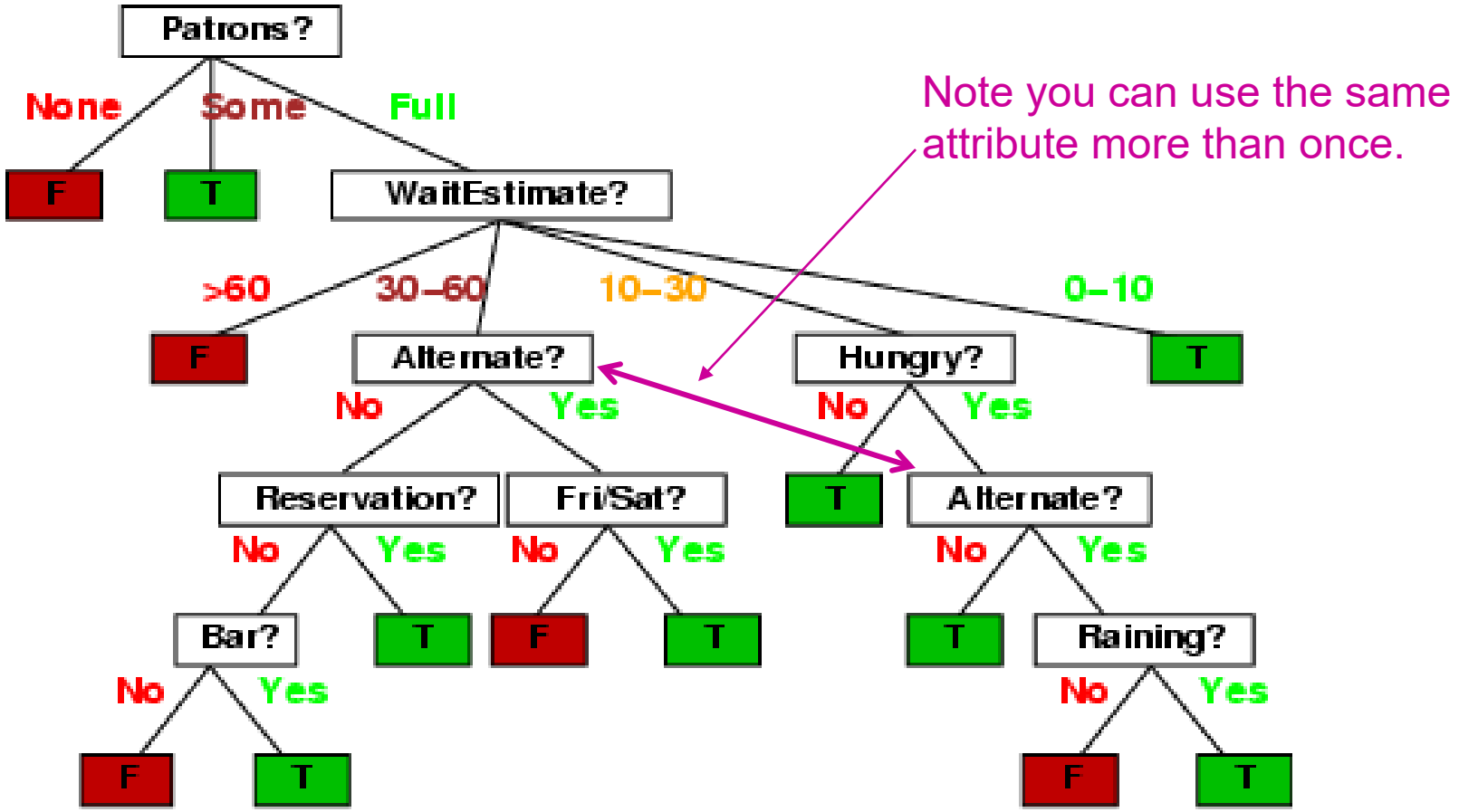
Root Node: Rent Property
Internal Node: Age
Decision Nodes: Rent or Buy



Decision trees

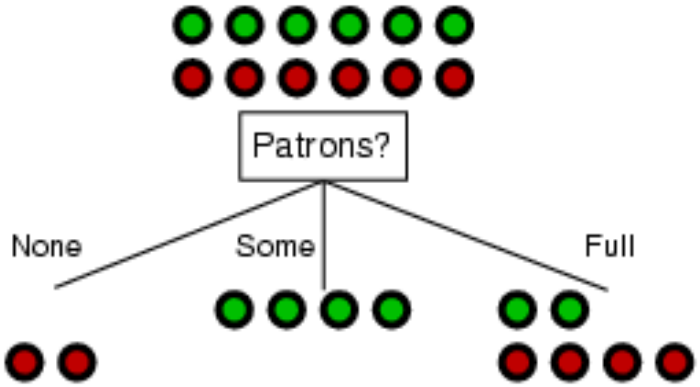
- Tree branch or Rule: walk from root to a class-labelled leaf.
- At each node, branching is based on the values of an attribute.

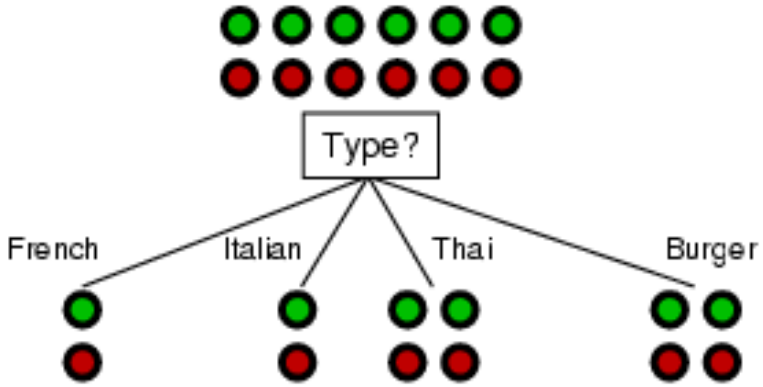
Example: Should we wait for a table at a restaurant?



Selecting an attribute

- Idea: a good feature splits the samples into subsets that are (ideally) "all positive" or "all negative"





To wait or not to wait is at 50%.

- *Patrons or type?*

Decision Tree Induction Process

- Decision tree generation consists of two phases
  - **Tree construction**
    - Divide and conquer: a recursive process
    - Successive partitions of the tree according to the best separator criterion
  - **Tree pruning**
    - Identify and remove branches that reflect noise or outliers
- Use of the decision tree: Classifying an unknown sample
  - Test the attribute values of the sample against the decision tree (e.g. nested if-then-else statements)

Algorithm for Tree Construction (1)

- Basic algorithm (a greedy algorithm)
  - Dataset represents attributes as **categorical** (if continuous-valued, they are discretized.)
  - Tree is constructed in a top-down recursive divide-and-conquer fashion.
  - All the training samples are considered at the root node.
  - Samples are partitioned recursively based on selected attributes
  - Test attributes are selected based on a heuristic or statistical measure (e.g., information gain)
    - choose an attribute that most effectively splits the data.
    - This means that the subsets produced by splitting the data according to the value of the attribute should be as homogeneous (or pure) as possible, with respect to the target attribute.

Algorithm for Tree Construction (2)

- Step 1: Split all the samples according to the attribute value that brings the best split
  - obtain *n* smaller sample sets for each branch
- Step 2: For each smaller sample set:
  - If the stopping conditions are not met:
    - apply Step 1 to the smaller sample set
  - Otherwise, the smaller sample set is a leaf (terminal)
- Conditions for stopping partitioning
  - All samples for a given node belong to the same class.
  - There are no remaining attributes for further partitioning.
  - There are no samples left.

An example: Building a decision tree

Age	CarType	HighRisk
23	Sedan	No
18	Sedan	No
23	Truck	Yes
36	Sedan	No
25	Sports	Yes
25	Truck	Yes
30	Sports	No
29	Truck	No

Training Data

Age	CarType	HighRisk
28	Truck	No
30	Sedan	No
32	Sedan	Yes
19	Sports	Yes

Test Data

The Problem Definition

- Example: “Is the insurer a high risk or not?”
  - a set of attributes and their possible values:

Age	Continuous values
Car type	Sedan, Sport, Truck

A particular *instance* in the training set is:

<Age, CarType>: Risk

<23, Sedan>: No

What is the target?
In this case, the target class is a binary attribute, so each instance represents a positive or a negative example.

Selecting An Attribute

Age	HighRisk
23	No
18	No
23	Yes
36	No
25	Yes
25	Yes
30	No
29	No

Training Data

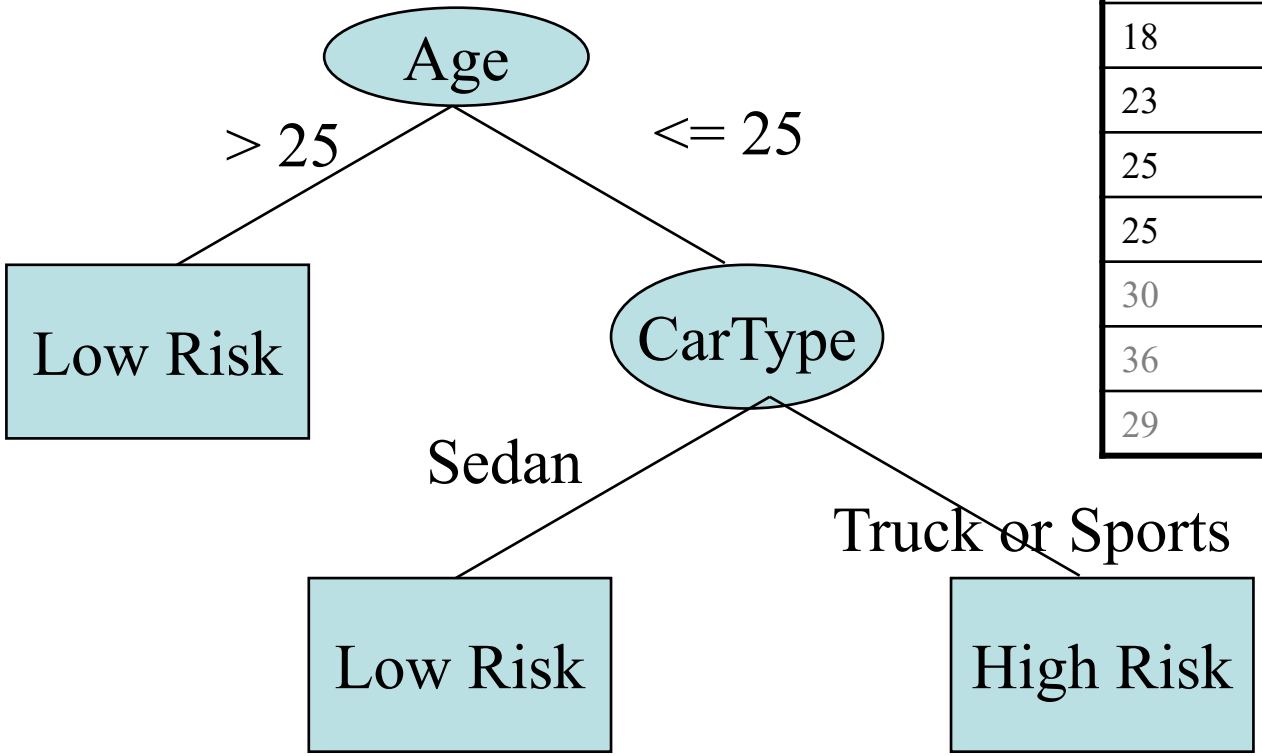
Age	HighRisk
23	No
18	No
23	Yes
25	Yes
25	Yes
30	No
36	No
29	No

Selecting An Attribute

CarType	HighRisk
Sedan	No
Sedan	No
Sedan	No
Truck	Yes
Truck	No
Truck	Yes
Sports	Yes
Sports	No

CarType	HighRisk
Sedan	No
Sedan	No
Truck	Yes
Sedan	No
Sports	Yes
Truck	Yes
Sports	No
Truck	No

Decision Tree: Construction



Age	CarType	HighRisk
23	Sedan	No
18	Sedan	No
23	Truck	Yes
25	Sports	Yes
25	Truck	Yes
30	Sports	No
36	Sedan	No
29	Truck	No

Testing the Decision Tree

- Classification Accuracy (%)
  - Number of correctly classified cases/
    Total Number of cases

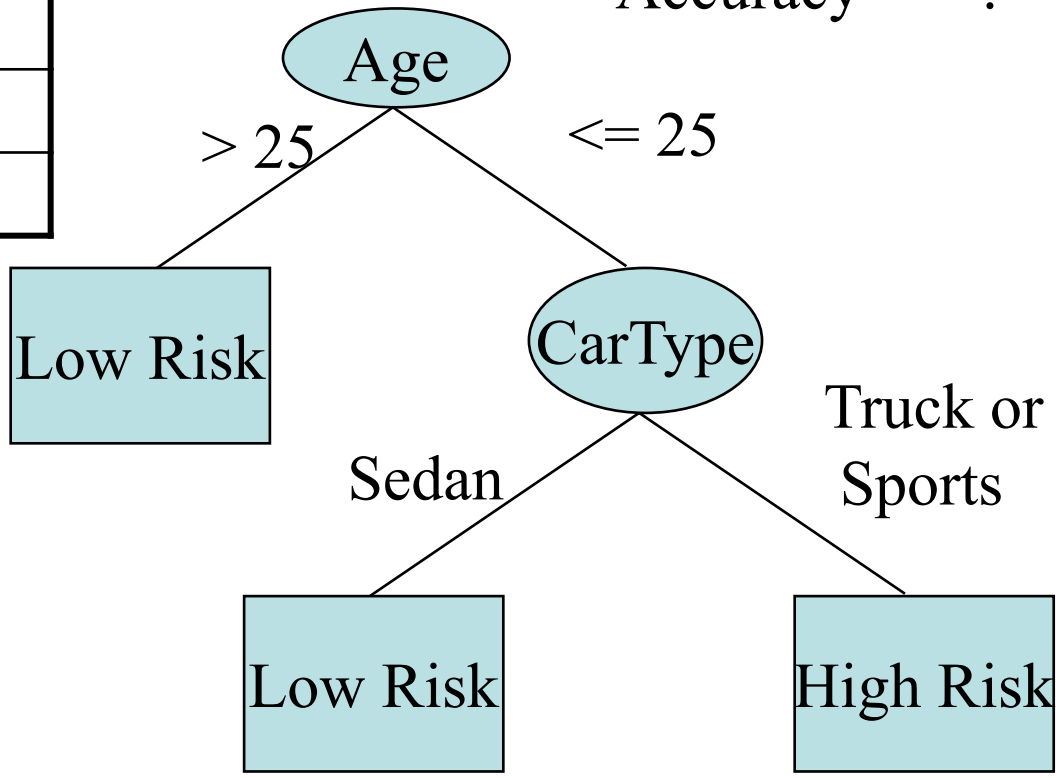
- Comprehensibility
  - Total Number of Rules Generated

Age	CarType	HighRisk
23	Sedan	No
18	Sedan	No
23	Truck	Yes
36	Sedan	No
25	Sports	Yes
25	Truck	Yes
30	Sports	No
29	Truck	No

Accuracy = ?

Age	CarType	HighRisk
28	Truck	No
30	Sedan	No
32	Sedan	Yes
19	Sports	Yes

Accuracy = ?



No of Rules = ?

Summary: Decision Tree Mining

- The tree starts as a single node representing the training samples.
- If the samples are all of the same class, then the node becomes a leaf and is labelled with that class.
- Otherwise, the heuristic of "selecting the attribute based on that it will best separate the samples into individual classes" is used.
  - This attribute becomes a node of the tree.
- A branch is created for each known value of the test attribute and the samples are partitioned accordingly.
- The same process is recursively repeated to form a decision tree for the samples at each partition.
- The recursive partitioning stops only when any one of the following conditions is met:
  - All samples for a given node belong to the same class.
  - There are no remaining attributes on which the samples may be further partitioned.
  - There are no samples for the branch.

Decision Tree: Classification

Tree cultivation

Bias and Variance

Benefits and Drawbacks

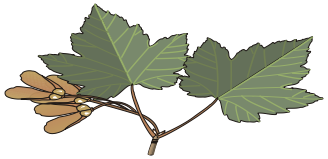
The Cultivation of Trees

- Splitting Criterion
  - Which split is best?
    - Heuristic: choose the attribute that produces the "purest" node
- Stopping Rule
  - When should the splitting stop?
    - Maximum depth of the tree
    - Minimum samples per leaf node

Splitting Criteria

- Three most widely used:
  - Information Gain or Entropy
  - Gini index
  - Pearson chi-squared test
- All three criteria measures the difference in class distributions for an attribute.

✓ All three methods usually give similar results.



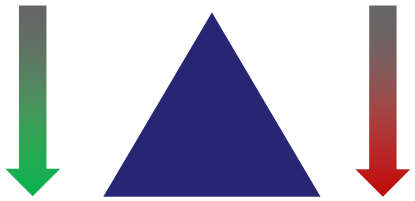
AID, THAID, CHAID:
chi-squared test, prepruning

CART:
Gini index, postpruning

ID3, C4.5, C5.0:
Entropy , postpruning

Tree Depth: Bias vs Variance

- As tree depth increases, bias decreases and variance increases.
  - May result in overfitting.



Bias decreases with tree depth

Variance increases with tree depth

Bias is the simplifying assumptions made by the **model** to make the target function easier to approximate.
Variance is the amount that the estimate of the target function will change given different training data.

- A tree perfectly fitting to training data: zero bias, high variance.

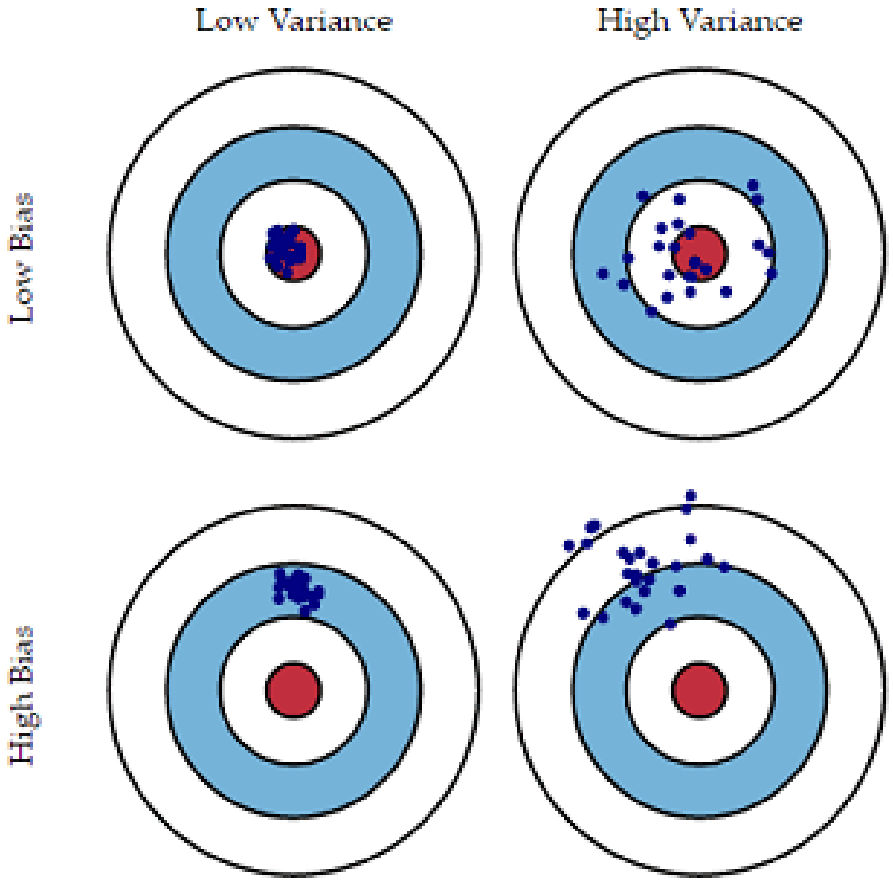
High Bias = A simpler model (the model may not fit the data very well - **underfitting**)
High Variance = A complex model (may fit the data too well, and learn the noise in addition to the inherent patterns in the data - **overfitting**)

High Bias Low Variance: Models are consistent but inaccurate on average.
High Bias High Variance: Models are inaccurate and inconsistent on average.
Low Bias High Variance: Models are accurate but inconsistent on averages.
Low Bias Low Variance: Models are accurate and consistent on averages.

- Look for the bias-variance trade-off

Bias Variance Trade-off

- Target: the red ball
- Any hit close to it is considered as low bias data points.
- Each subsequent hit, that is close to the previous hit, is considered as low variance cases.



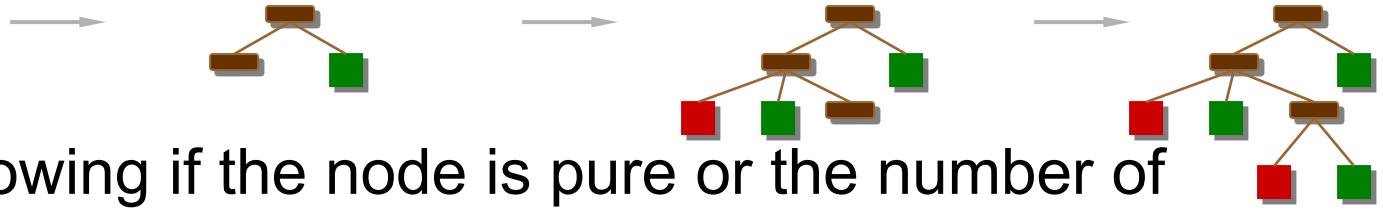
https://www.listendata.com/2017/02/bias-variance-tradeoff.html

Stoping and Pruning Rules

- Model Complexity: Determined by the Number of leaves
- **Maximal Tree**: A tree is built until all leaves are pure or the number of cases in a node falls below a specified limit.
  - Perfect fit on training data, but poor prediction on new data (overfitted, high variance)
- **Optimal Tree**: choose from a sequence of trees based on performance on validation/test data
  - Chooses a tree that has good generalisation capability with comparatively lower complexity

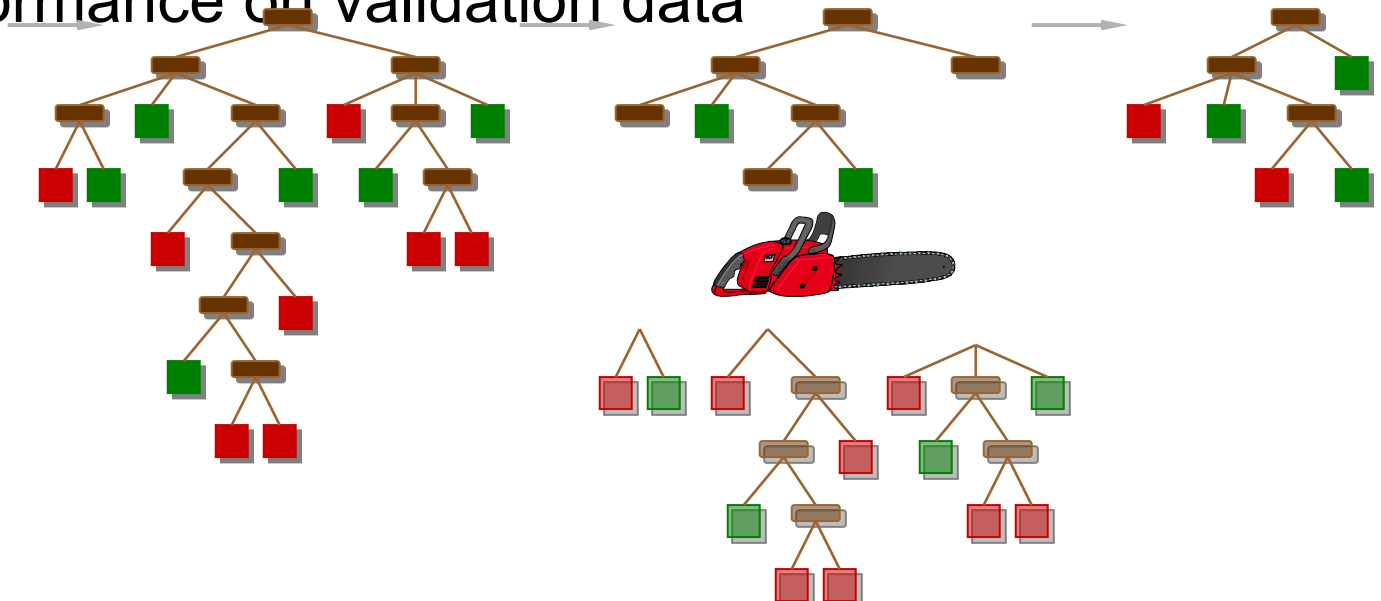
The Right-Sized Tree: Two Approaches

- Stunting or Prepruning: forward stopping rule



Stop growing if the node is pure or the number of cases in a node falls below a specified limit

- Post-pruning: choose from a sequence of trees based on performance on validation data

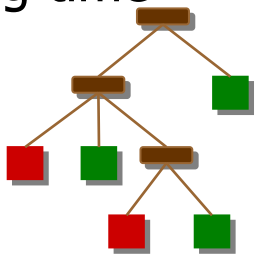


Pros and Cons of Decision Trees

· **Pros**

+ Reasonable training time
+ Fast application
+ Easy to interpret
+ Easy to implement
+ Can handle large number of features
  – Mixed Measurement Scales
    nominal, ordinal, interval
+ Robust
  – Can handle different data distributions and formats
  – Can handle Missing Values

•Regression Tree



· **Cons**

- Cannot handle complicated relationship between features

-Roughness

- simple decision boundaries

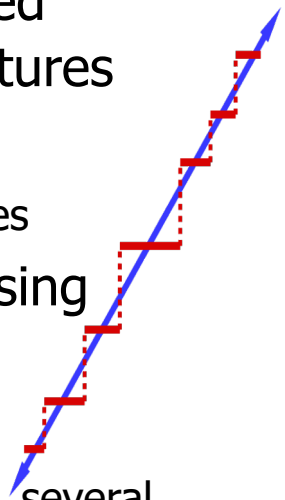
- problems with lots of missing data

-Instability

- gives a local solution, i.e., several decision trees can be built from the same dataset
- Need to find the right split attribute

-Visualizing trees can sometimes be tedious

– particularly as data dimensionality increases



Decision Tree: Classification

Tree Ensembles – A quick tour

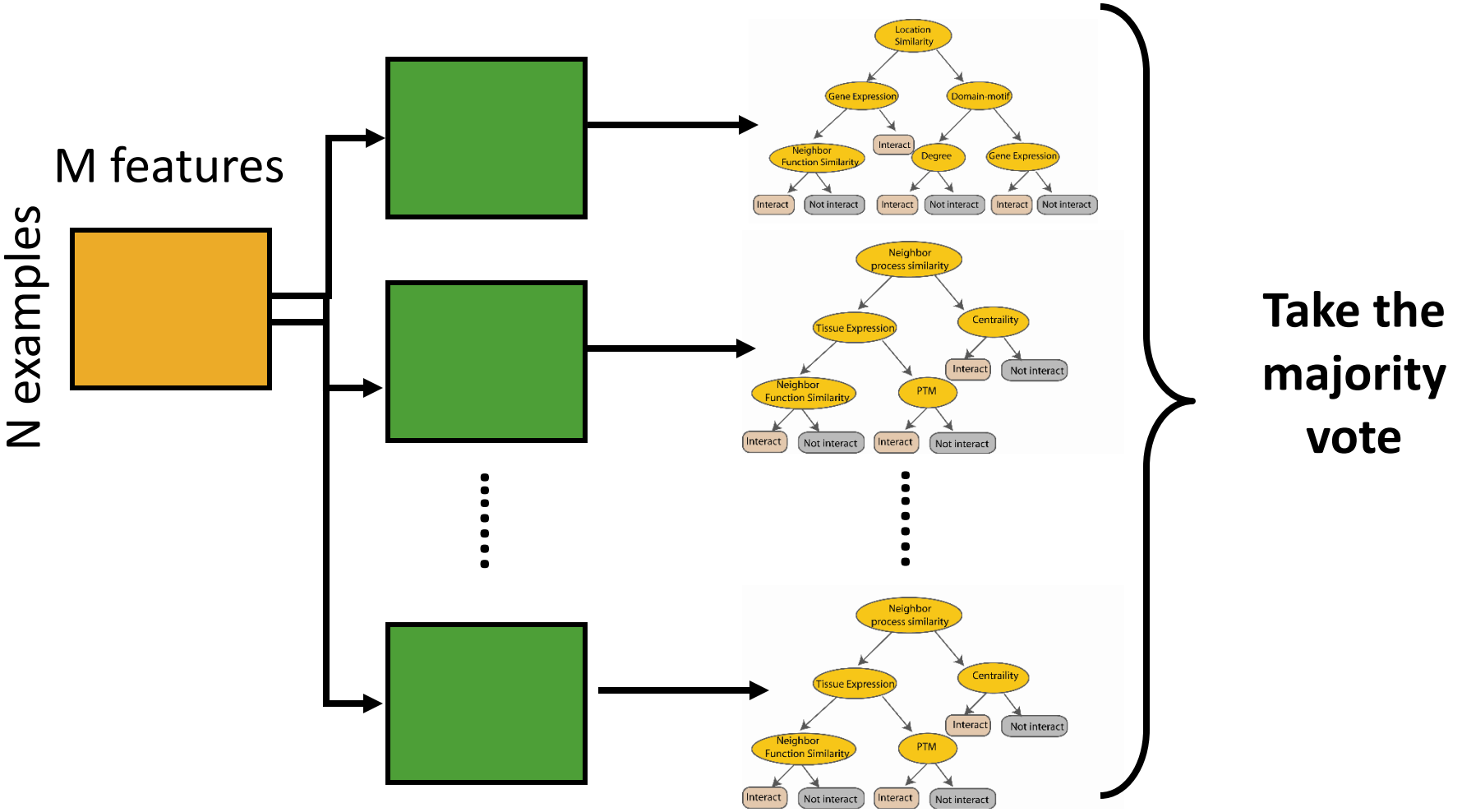
Ensemble Methods

- Take a collection of simple or *weak* learners
- Combine their results to make a single, better learner

Types:

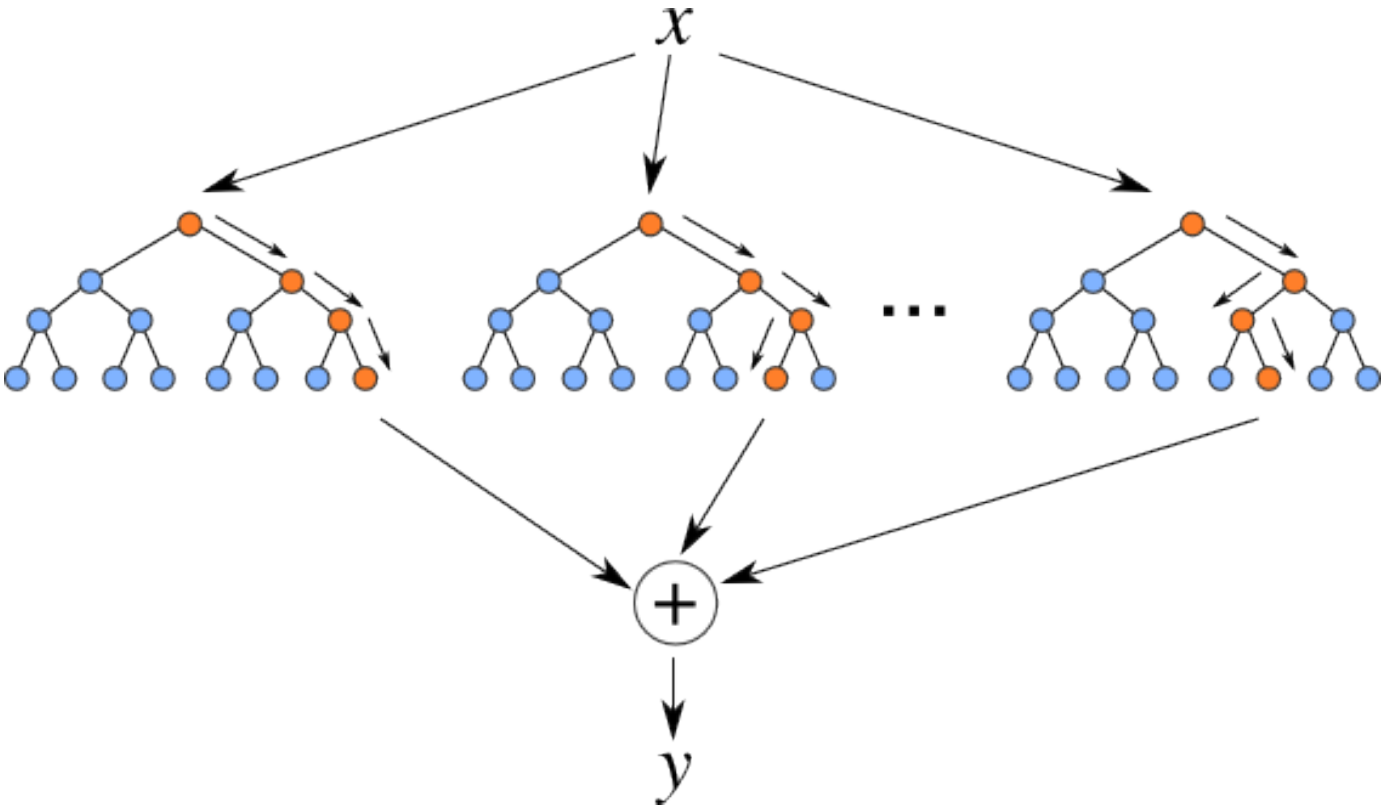
- **Bagging:** train learners in parallel on different samples of the data, then combine them by voting (discrete output) or by averaging (continuous output).
  - Example: Random Forest
- **Stacking:** combine model outputs using a second-stage learner like linear regression.
- **Boosting:** train learners on the filtered output of other learners.
  - Example: XGBoost

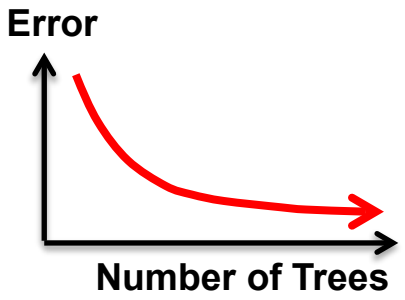
Random Forest Classifier



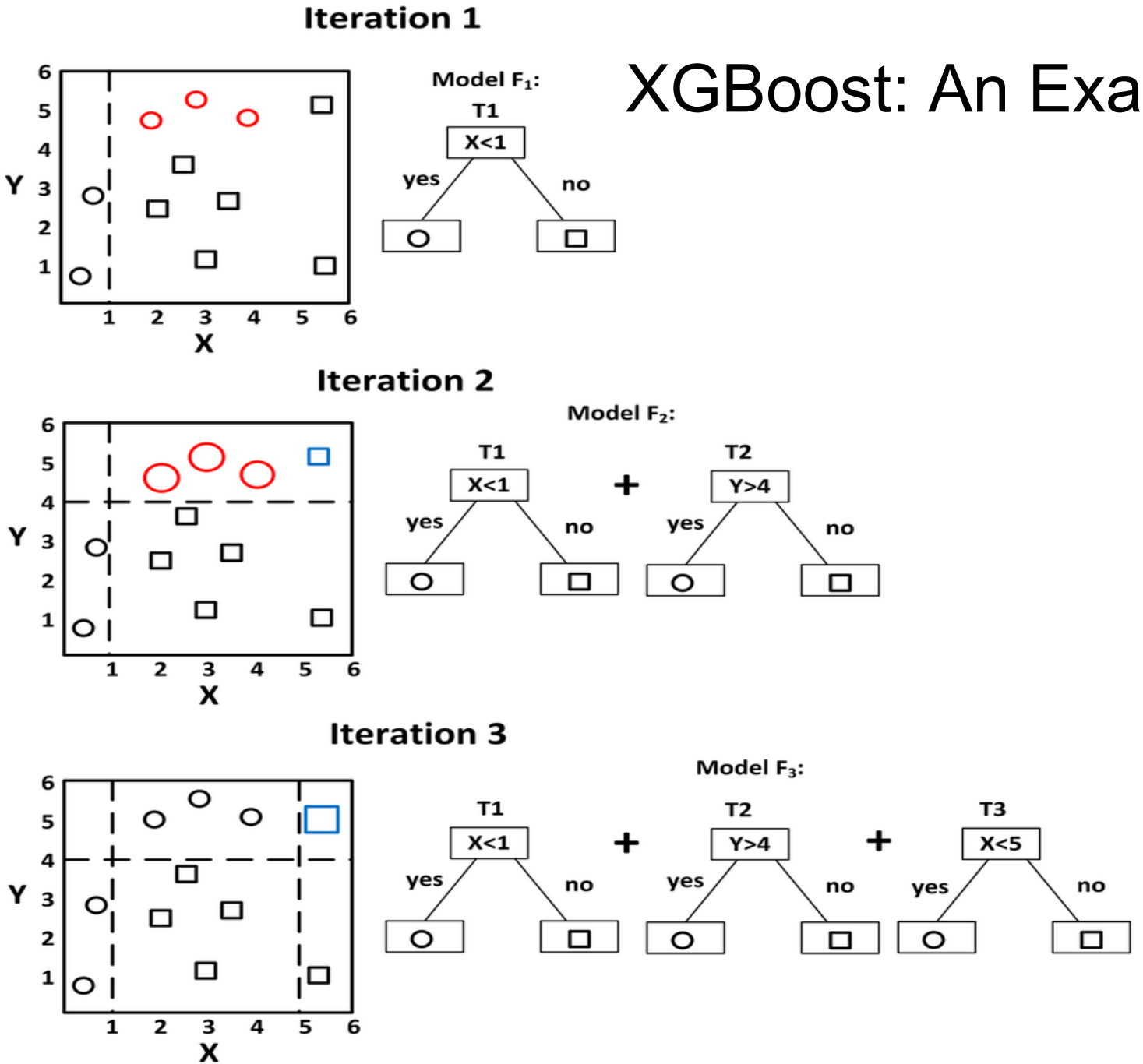
XGBoost

- Additive tree model: add new trees that complement the already-built ones
- Response is the optimal linear combination of all decision trees
- Popular in Kaggle competitions for efficiency and accuracy





XGBoost: An Example



References

- Data Mining techniques and concepts by Han J et al, 2011.
- Discovering Data Mining, by Cabena, et al., 1997.
- Predictive Data Mining, by Weiss and Indurkhya, 1999.