DATA ANALYTICS

Unit 4: Decision trees-CART, Ensemble Methods and Random Forests

Jyothi R. , R Bharathi

Assistant Professor

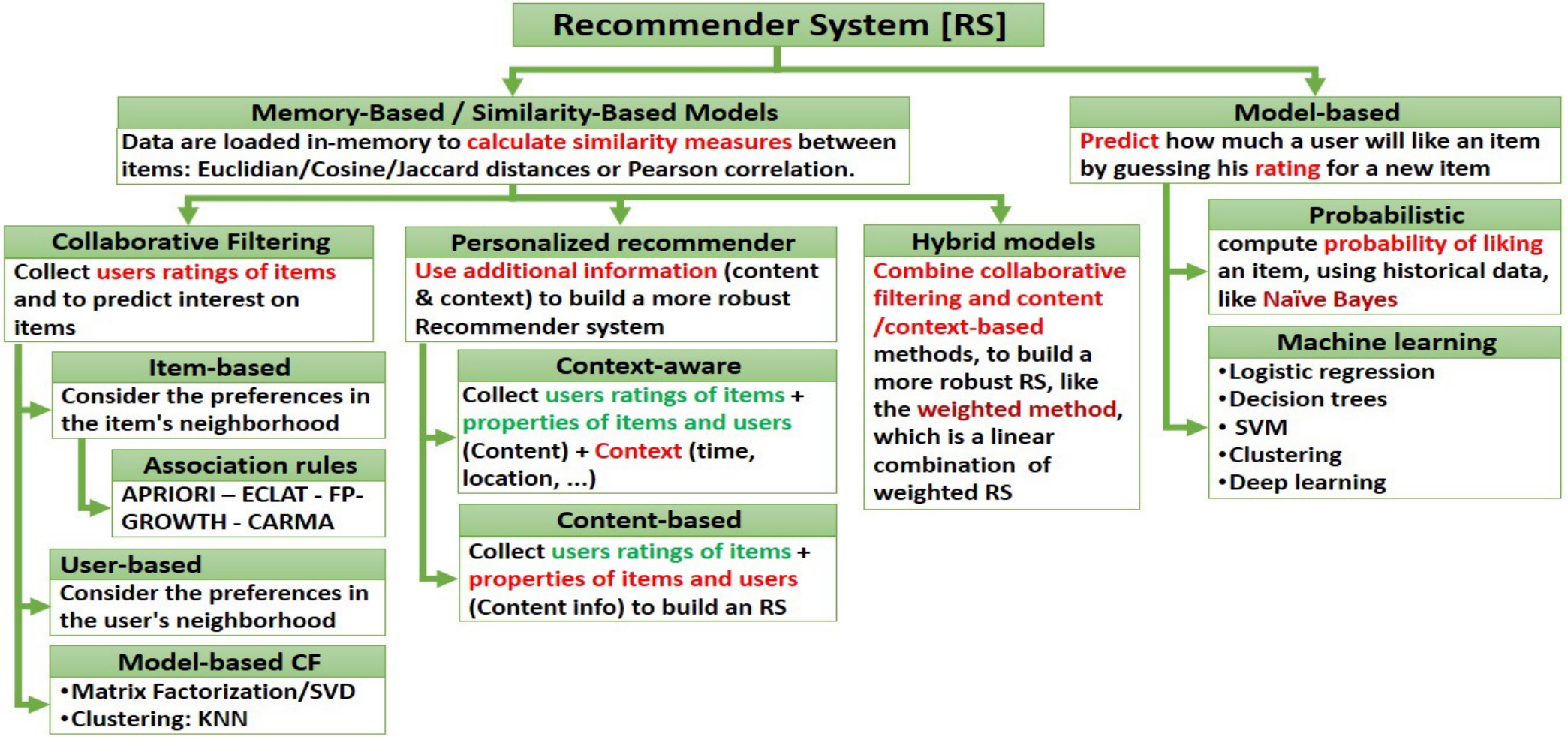
Department of Computer Science and Engineering

DATA ANALYTICS

Recommendations:

- Introduction to recommendation systems
- Collaborative filtering
- Knowledge based filtering using KNN
- **Decision trees** – CART,
- Ensemble methods and Random Forest
- Brief review of other classifiers: SVM, ANN and data driven approaches
- Brief review of unsupervised learning – clustering algorithms – DBSCAN
- Content based analysis – dealing with textual data
- Text classification and clustering
- Market basket analysis (Apriori algorithm)
- Generation and evaluation of association rules from frequent item sets
- Case Study

Types of Recommender Systems



Decision Tree Classification Task- General Approach

Base Classifiers

- Decision Tree based Methods
- Rule-based Methods
- Nearest-neighbor
- Neural Networks, Deep Neural Nets
- Naïve Bayes and Bayesian Belief Networks
- Support Vector Machines

Ensemble Classifiers

- Boosting, Bagging, Random Forests

DATA ANALYTICS

Decision Tree

- Decision Trees are collection of divide-conquer problem-solving strategies that use tree-like structure to predict the outcome of a variable.
- Decision trees are a collection of predictive analytics techniques that use tree-like graphs for predicting the value of a response variable or target variable based on the values of explanatory variables or predictors.
- It is one of the supervised learning algorithms used for predicting both the discrete and the continuous dependent variable.
- Decision trees are effective for solving classification problems in which the response variable or target variable takes discrete values.

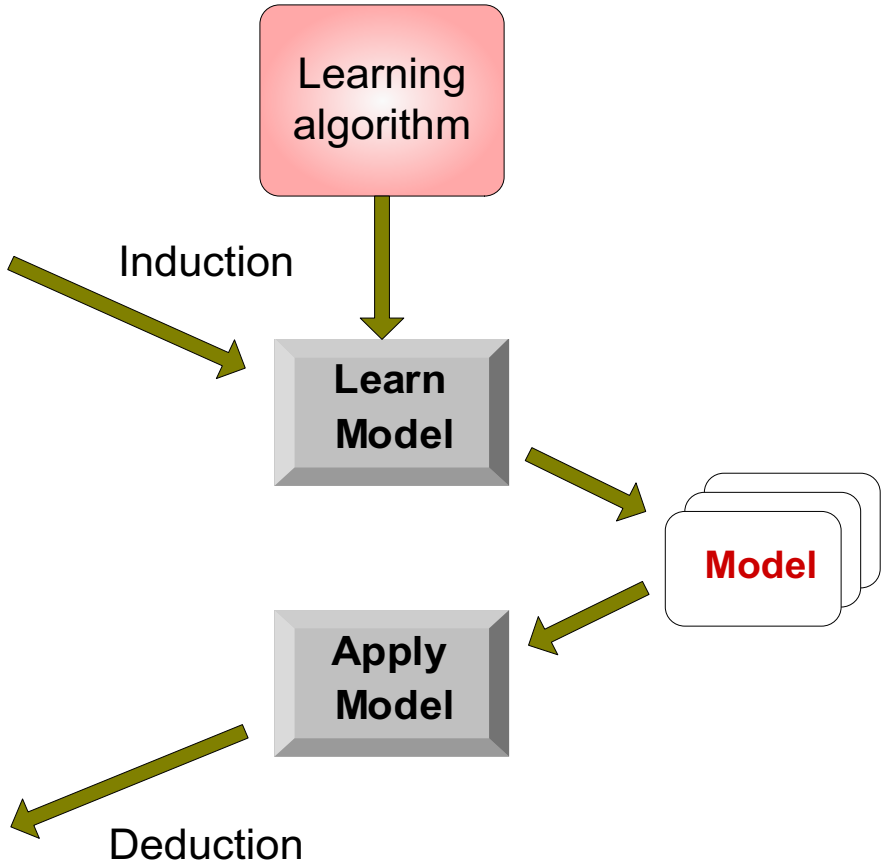
Decision Tree Classification Task- General Approach

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

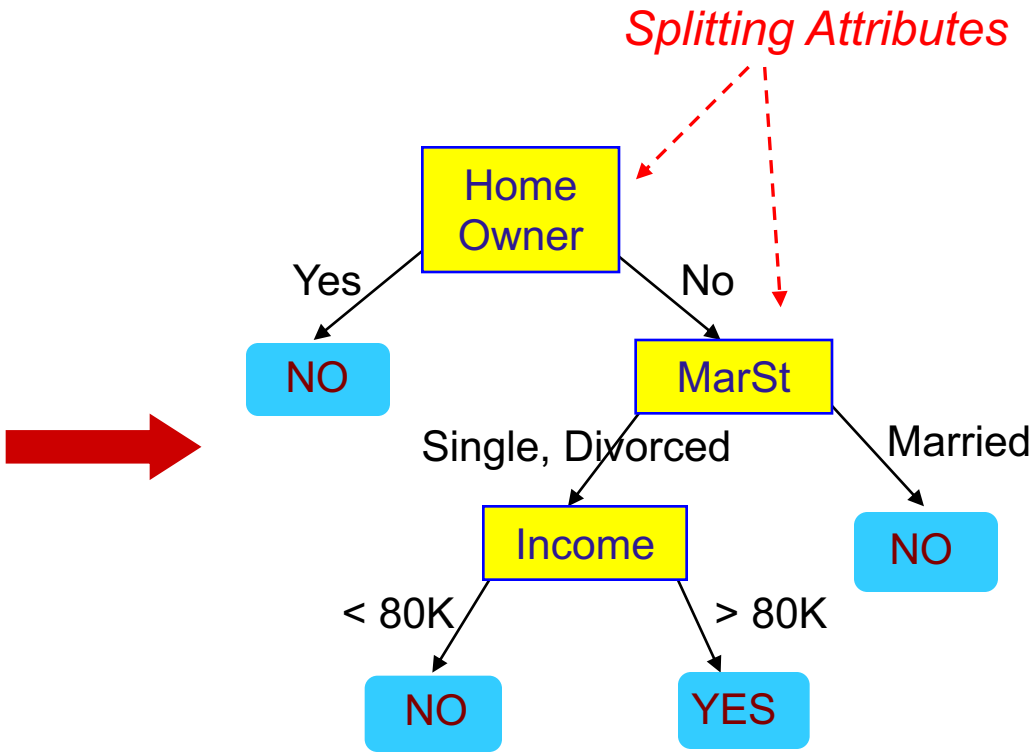


Decision Tree

Example1 of an Decision Tree

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
	categorical	categorical	continuous	class
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

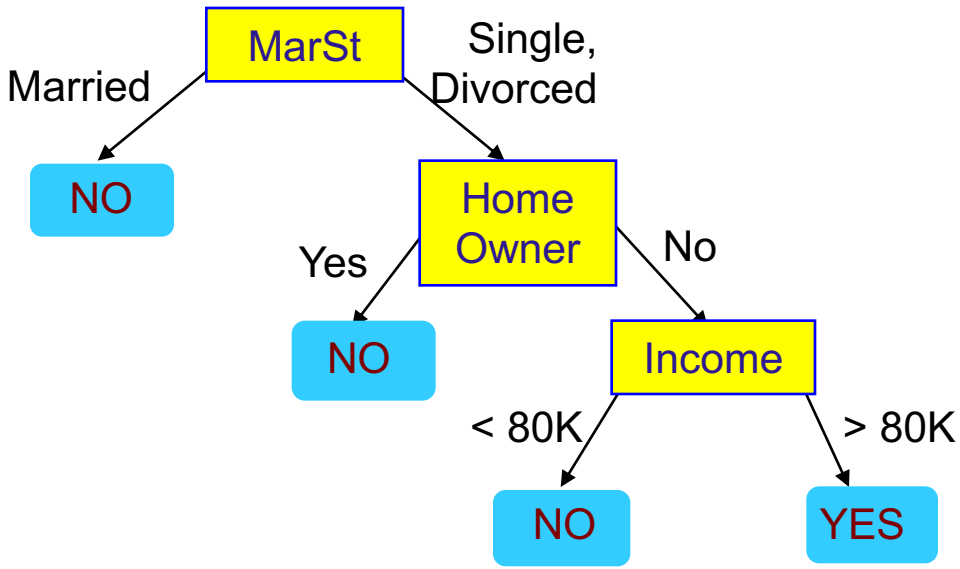
Training Data



Model: Decision Tree

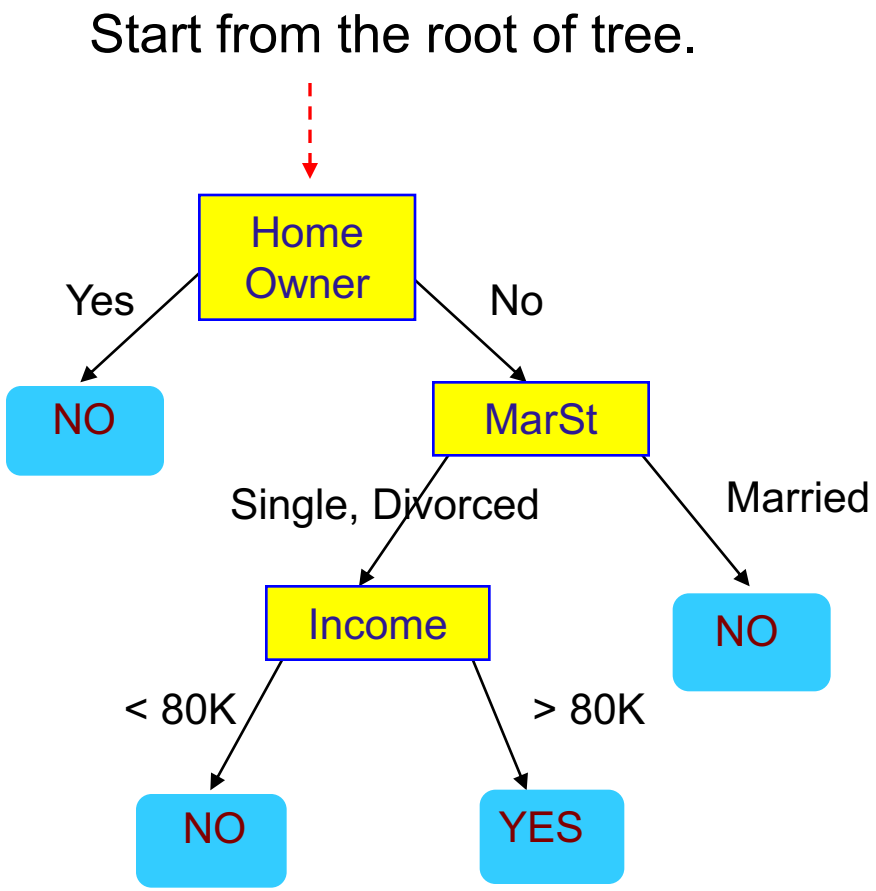
Decision Tree

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
	categorical	categorical	continuous	class
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



There could be more than one tree that fits the same data!

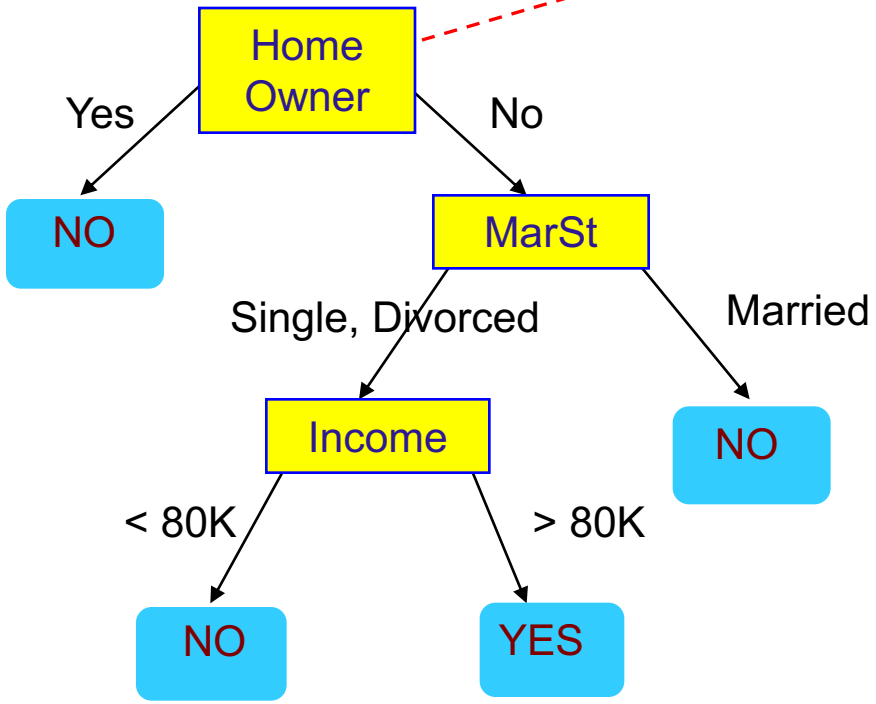
Apply Model to Test Data



Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

Apply Model to Test Data



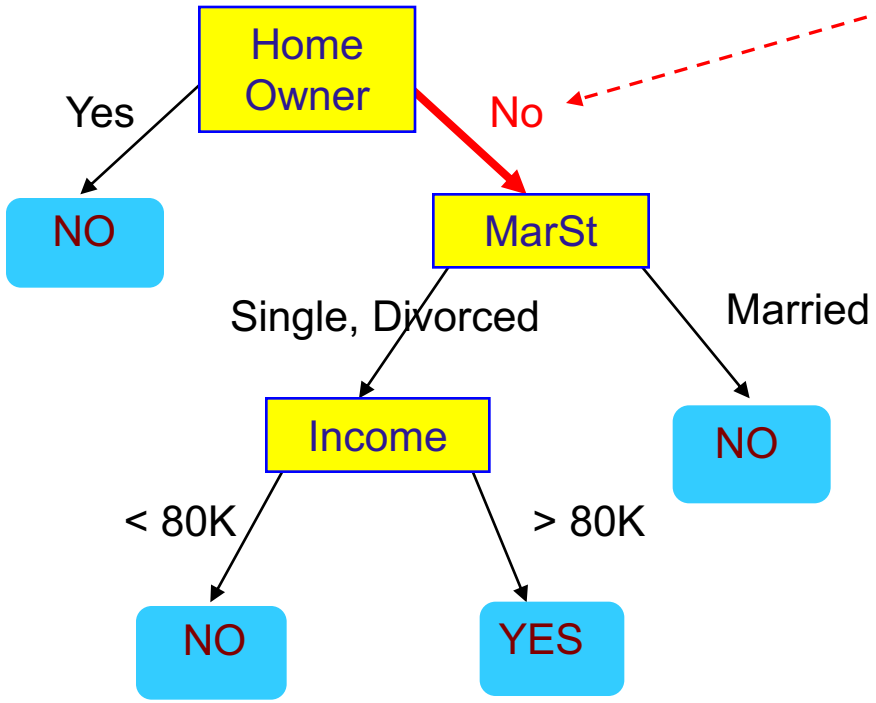
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

Test Data

Apply Model to Test Data

Test Data

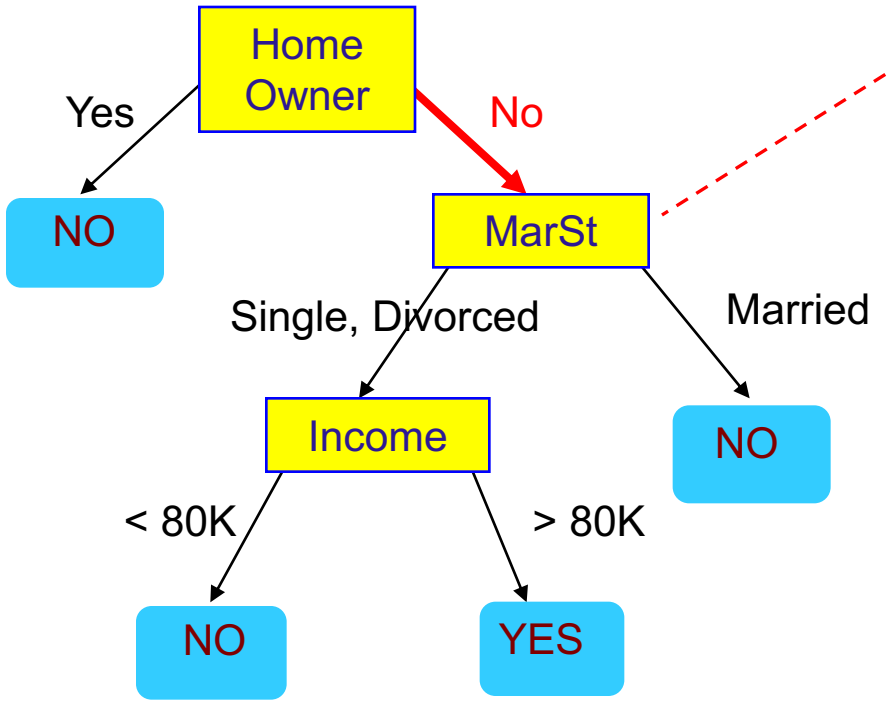
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



Apply Model to Test Data

Test Data

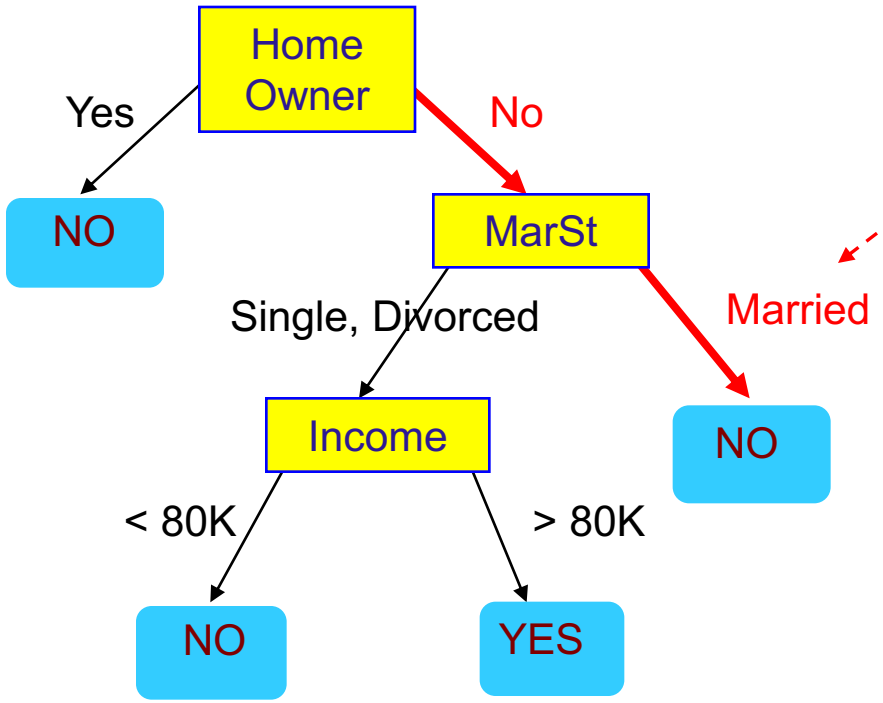
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



Apply Model to Test Data

Test Data

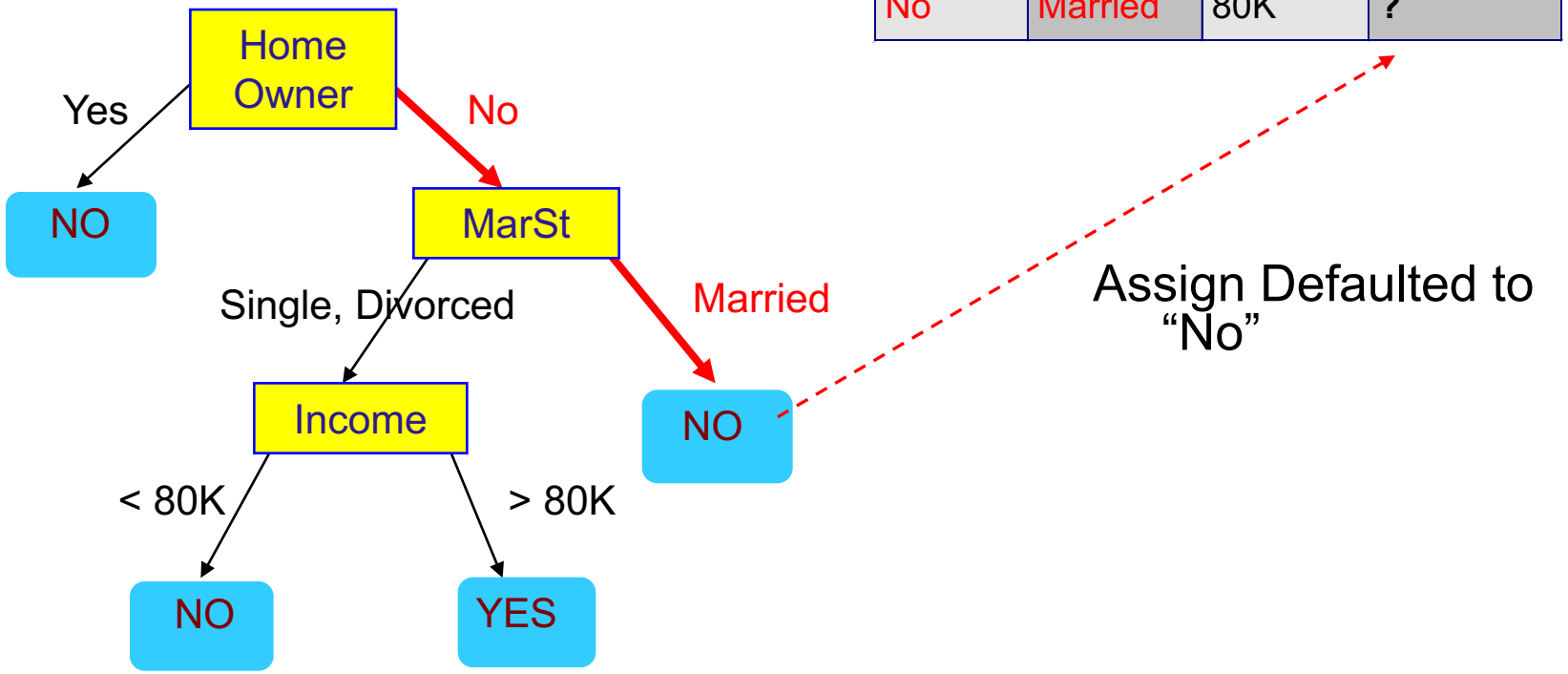
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



Apply Model to Test Data

Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



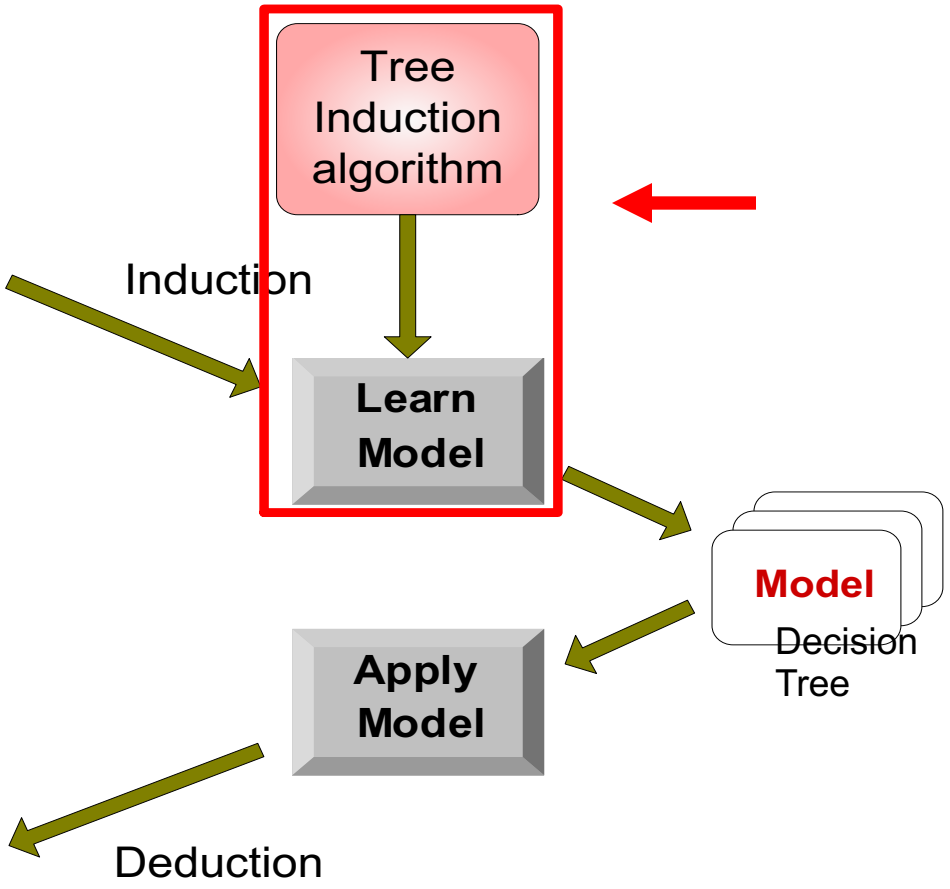
Decision Tree Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

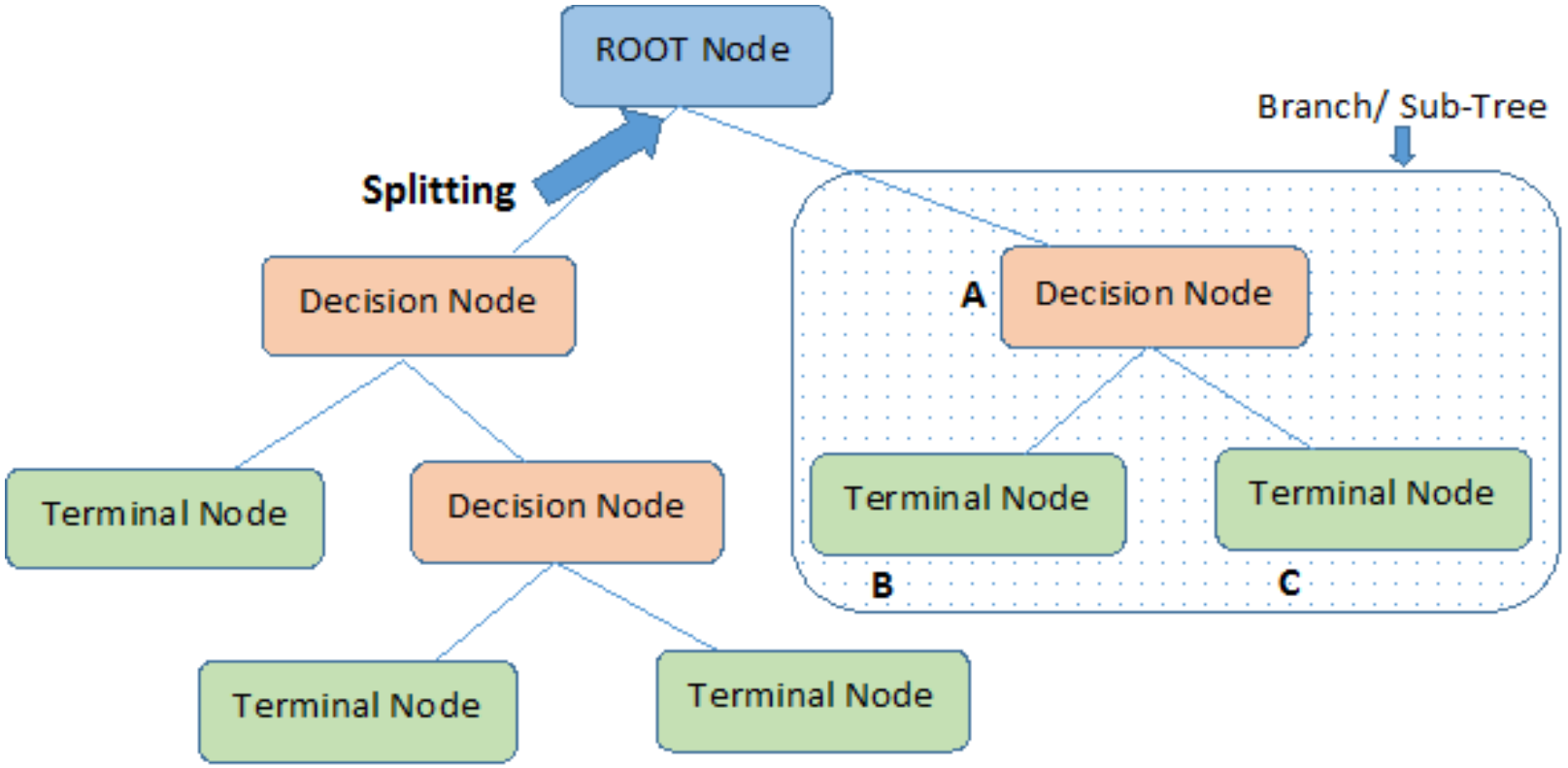
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



DATA ANALYTICS

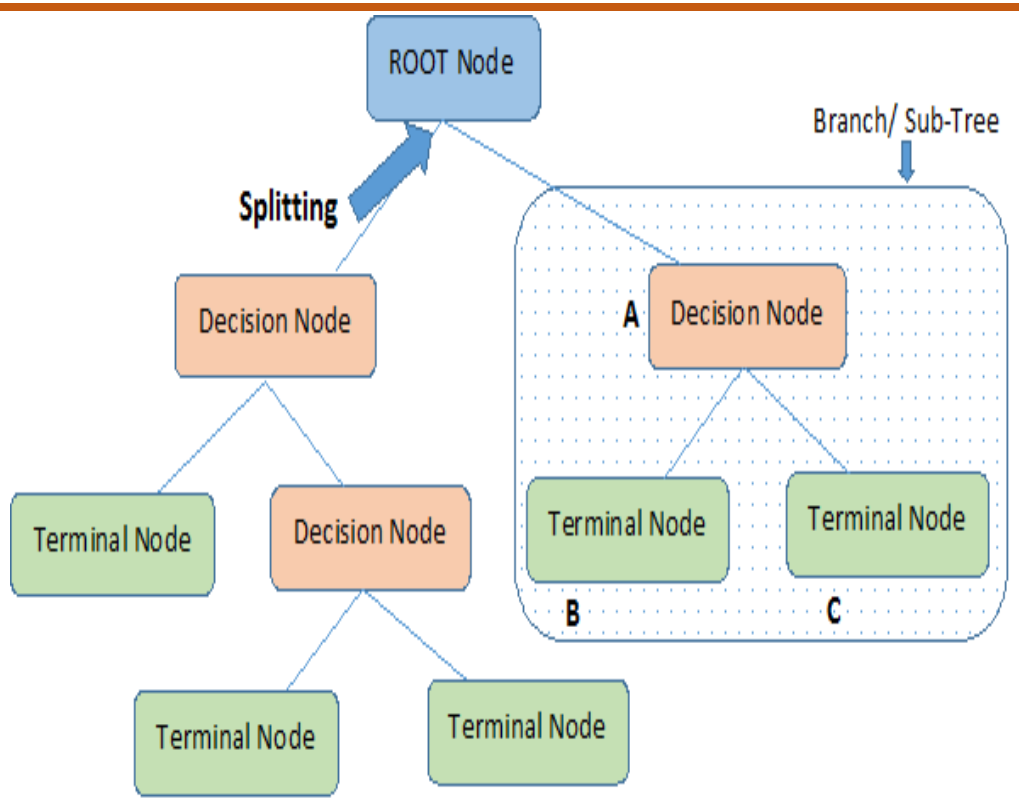
Important Terminology related to Decision Trees



Note:- A is parent node of B and C.

DATA ANALYTICS

Important Terminology related to Decision Trees



Note:- A is parent node of B and C.

1.Root Node: It represents entire population or sample and this further gets divided into two or more homogeneous sets.

2.Splitting: It is a process of dividing a node into two or more sub-nodes.

3.Decision Node: When a sub-node splits into further sub-nodes, then it is called decision node.

4.Leaf/ Terminal Node: Nodes with no children (no further split) is called Leaf or Terminal node.

5.Pruning: When we reduce the size of decision trees by removing nodes (opposite of Splitting), the process is called pruning.

6.Branch / Sub-Tree: A sub section of decision tree is called branch or sub-tree.

7.Parent and Child Node: A node, which is divided into sub-nodes is called parent node of sub-nodes where as sub-nodes are the child of parent node.

Steps are used for generating decision trees

- The following steps are used for generating decision trees:

1. Start with the root node in which all the data is present.
2. Decide on a splitting criterion and stopping criteria: The root node is then split into two or more subsets leading to tree branches (called edges) using the splitting criterion. Nodes thus created are known as internal nodes. Each internal node has exactly one incoming edge.
3. Further divide each internal node until no further splitting is possible or the stopping criterion is met. The terminal nodes (aka leaf nodes) will not have any outgoing edges.
4. Terminal nodes are used for generating business rules.
5. Tree pruning (a process for restricting the size of the tree) is used to avoid large trees and overfitting the data. Tree pruning is achieved through different stopping criteria.

Decision Trees use the following criteria to develop the tree

- Decision trees use the following criteria to develop the tree:

1. Splitting Criteria: Splitting criteria are used to split a node i.e. set of data into subsets.

2. Merging Criteria: When the predictor variable is categorical with n categories, it is possible that all n categories may be statistically significant. Thus, few categories may be merged to create a compound or aggregate category.

3. Stopping Criteria: Stopping criteria is used for pruning the tree (stopping the tree from further branching) to reduce the complexity associated with business rules generated from the tree. Usually levels (depth) from root node (where each level corresponds to adding a predictor variable), minimum number of observation in a node for splitting are used as stopping criteria.

Decision Tree Induction

There are many Decision tree techniques and they differ in the strategy that they use for splitting the nodes.

Algorithms:

- Hunt's Algorithm (one of the earliest)
- CART
- ID3, C4.5
- SLIQ,SPRINT

Note : ID3

The core algorithm for building decision trees is called **ID3.** Developed by J. R. Quinlan, this algorithm employs a top-down, greedy search through the space of possible branches with no backtracking.

ID3 uses *Entropy* and *Information Gain* to construct a decision tree.

Classification and Regression Tree (CART)

- CART is used for a -Classification Tree when the dependent variable is discrete and - a Regression Tree when the dependent variable is continuous.

- Classification tree uses various impurity measures such as the **Gini Impurity Index** and **Entropy** to split the nodes.

- Regression Tree splits the node that minimizes the **Sum of Squared Errors (SSE).** CART is a binary tree wherein every node is split into only two branches.

Classification and Regression Tree (CART) - Gini Index

- Gini index says, if we select two items from a population at random then they must be of same class and probability for this is 1 if population is pure.
- It works with categorical target variable “Success” or “Failure”.
- It performs only Binary splits
- Higher the value of Gini higher the homogeneity.
- CART (Classification and Regression Tree) uses Gini method to create binary splits.

Steps to generate Classification and Regression Tree (CART)

1. Start with the complete training data in the root node.
2. Decide on the measure of impurity (usually Gini impurity index or Entropy). Choose a predictor variable that minimizes the impurity when the parent node is split into children nodes. This happens when the original data is divided into two subsets using a predictor variable such that it results in the maximum reduction in the impurity in the case of discrete dependent variable or the maximum reduction in SSE in the case of a continuous dependent variable.
3. Repeat step 2 for each subset of the data for each internal node using the independent variables until:
   (a) All the dependent variables are exhausted.
   (b) The stopping criteria are met. Few stopping criteria used are number of levels of tree from the root node, minimum number of observations in parent/child node (e.g. 10% of the training data), and minimum reduction in impurity index.
4. Generate business rules for the leaf (terminal) nodes of the tree.

Classification and Regression Tree (CART)

Steps to Calculate Gini for a split

Calculate Gini for sub-nodes, using formula sum of square of probability for success and failure

Calculate Gini for split using weighted Gini score of each node of that split

Example: — Referring to example where we want to segregate the students based on target variable (playing cricket or not). In the snapshot below, we split the population using two input variables Gender and Class. Now, I want to identify which split is producing more homogeneous sub-nodes using Gini index.

DATA ANALYTICS

Classification and Regression Tree (CART)

Objective

Decision rules will be found by GINI index value.

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Classification and Regression Tree (CART)

Objective

Decision rules will be found by GINI index value.

Gini index

Gini index is a metric for classification tasks in CART. It stores sum of squared probabilities of each class. We can formulate it as illustrated below.

$$Gini = 1 - \Sigma (Pi)^2 \text{ for } i=1 \text{ to number of classes}$$

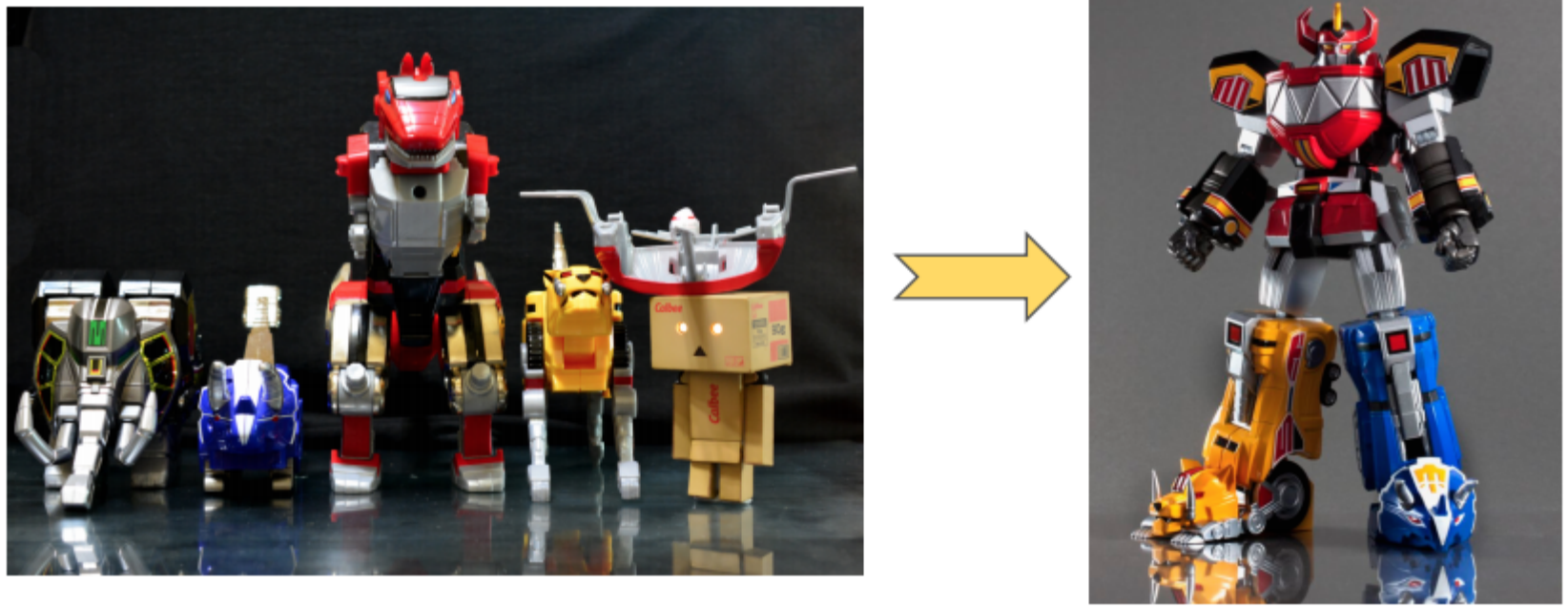
Classification and Regression Tree (CART)

- The following steps are used to generate a classification and a regression tree:

1. In Classification and regression tree(CART) impurity measures such as
2. Gini impurity index or entropy are used as splitting criteria when the dependent variable is categorical and
3. Sum of squared errors (SSE) is used when the dependent variable is continuous.

Ensemble Methods

- General Idea Combine multiple classifiers usually with different strengths to build a bigger, better classifier



DATA ANALYTICS

Ensemble Methods

- The ensemble method is a machine-learning-algorithm that generates several classifiers using different sampling strategies such as bootstrap aggregating.

- A majority-voting-approach may be used for classifying a new observation using the multiple classifiers that are part of the ensemble method.

- In the ensemble method, several techniques such as logistic regression, CHAID, CART etc. are used.
- For a new observation, its class is identified using all the classifiers that are part of the ensemble method.
- Different classifiers are likely to classify a new observation into different categories.

- The final class of a new observation is decided based on a majority vote in which each classifier is given equal weightage.

Ensemble Methods

- Adaboosting: Boosting algorithms assign weights to each classifier based on their accuracy.
- A majority-voting-approach may be used for classifying a new observation using the multiple classifiers that are part of the ensemble method.
- In the ensemble method, several techniques such as logistic regression, CHAID, CART etc. are used.
- For a new observation, its class is identified using all the classifiers that are part of the ensemble method.
- Different classifiers are likely to classify a new observation into different categories.
- The final class of a new observation is decided based on a majority vote in which each classifier is given equal weightage.

Ensemble methods

Typical application: classification

Ensemble of classifiers is a set of classifiers whose individual decisions combined in some way to classify new examples.

Simplest approach:

1. Generate multiple classifiers
2. Each votes on test instance
3. Take majority as classification

Classifiers different due to different sampling of training data, or randomized parameters within the classification algorithm

Aim: take simple mediocre algorithm and transform it into a super classifier without requiring any fancy new algorithm.

Ensemble methods: Summary

Differ in training strategy, and combination method

1. Parallel training with different training sets: bagging

2. Sequential training, iteratively re-weighting training examples so current classifier focuses on hard examples: boosting

3. Parallel training with objective encouraging division of labor: mixture of experts

Notes:

- Also known as meta-learning
- Typically applied to weak models, such as decision stumps (single-node decision trees), or linear classifiers

Variance-bias tradeoff?

Minimize two sets of errors:

1. Variance: Error from sensitivity to small fluctuations in the training set
2. Bias: Erroneous assumptions in the model.
3. Variance-bias decomposition: is a way of analyzing the generalization error as a sum of 3 terms: variance, bias and irreducible error (resulting from the problem itself)

Why do ensemble methods work?

Based on one of two basic observations:

1. Variance reduction: if the training sets are completely independent, it will always helps to average an ensemble because this will reduce variance without affecting bias (e.g., bagging) -- reduce sensitivity to individual data points.

2. Bias reduction: for simple models, average of models has much greater capacity than single model (e.g., hyperplane classifiers, Gaussian densities). Averaging models can reduce bias substantially by increasing capacity, and control variance by fitting one component at a time (e.g., boosting).

Ensemble methods: Jusfficafon

Ensemble methods more accurate than any individual members:

- Accurate (better than guessing)
- Diverse (different errors on new examples)
- Diverse (different errors on new examples)

Independent errors: prob k of N classifiers (independent error rate ε) wrong:

Independent errors: prob k of N classifiers (independent error rate ε) wrong:

Ensemble methods: Netflix

- Clear demonstration of the power of ensemble methods
- Original progress prize winner (BellKor) was ensemble of 107 models!

- "Our experience is that most efforts should be concentrated in deriving substantially different approaches, rather than refining a simple technique."

- "We strongly believe that the success of an ensemble approach depends on the ability of its various predictors to expose different complementing aspects of the data. Experience shows that this is very different than optimizing the accuracy of each individual predictor."

Bootstrap estimation

- Repeatedly draw n samples from D
- For each set of samples, estimate a statistic
- The bootstrap estimate is the mean of the individual estimates
- Used to estimate a statistic parameter and its variance
- Bagging: bootstrap aggregation

Bagging

- Simple idea: generate M bootstrap samples from your original training set. Train on each one to get y , and average them
- For regression: average predictions
- For classification: average class probabilities (or take the majority vote if only hard outputs available)
- Bagging approximates the Bayesian posterior mean. The more bootstraps the better, so use as many as you have time for Each bootstrap sample is drawn with replacement, so each one contains some duplicates of certain training points and leaves out other training points completely

Cross-validated commi□ees

- Bagging works well for unstable algorithms: can change dramatically with small changes in training data
- But can hurt a stable algorithm: a Bayes optimal algorithm may leave out some training examples in every bootstrap
- Alternative method based on different training examples: cross-validated committees:
- Here k disjoint subsets of data are left out of training sets
- Again uses majority for combination

Boosting

- Also works by manipulating training set, but classifiers trained sequentially

- Each classifier trained given knowledge of the performance of previously trained classifiers: focus on hard examples

- Final classifier: weighted sum of component classifiers

References

Text Book:

"Business Analytics, The Science of Data-Driven Making", U. Dinesh Kumar, Wiley 2017

"Recommender Systems, The text book, Charu C. Aggarwal, Springer 2016 Section 1.and Section 2.

Image Courtesy

http://webcache.googleusercontent.com/search?q=cache:vW5NL8dqVQkJ:www.cs.kent.edu/~jin/DM07/ClassificationDecisionTree.ppt+&cd=5&hl=en&ct=clnk&gl=in

https://cmci.colorado.edu/classes/INFO-4604/fa17/files/slides-16_ensemble.pdf

https://sefiks.com/2018/08/27/a-step-by-step-cart-decision-tree-example/

https://medium.com/@rishabhjain_22692/decision-trees-it-begins-here-93ff54ef134

THANK YOU

Jyothi R.
Assistant Professor,
Department of Computer Science
jyothir@pes.edu