Machine Learning and Data Mining

Decision Trees: definitions, algorithms, applications, optimizations and implementation using R/Rattle

2016

Carlo Lipizzi
clipizzi@stevens.edu

SSE

Machine learning and our focus

- Like human learning from past experiences
- A computer does not have "experiences"
- A computer system learns from data, which represent some "past experiences" of an application domain
- Our focus: learn a target function that can be used to predict the values of a discrete class attribute, e.g.: approve or not-approved, and high-risk or low risk
- The task is commonly called: Supervised learning, classification, or inductive learning

The data and the goal

- Data: A set of data records (also called examples, instances or cases) described by
  - k attributes: A_1, A_2, … A_k.
  - a class: Each example is labelled with a pre-defined class
- Goal: To learn a classification model from the data that can be used to predict the classes of new (future, or test) cases/instances

An example: data (loan application)

Approved or not

ID	Age	Has_Job	Own_House	Credit_Rating	Class
1	young	false	false	fair	No
2	young	false	false	good	No
3	young	true	false	good	Yes
4	young	true	true	fair	Yes
5	young	false	false	fair	No
6	middle	false	false	fair	No
7	middle	false	false	good	No
8	middle	true	true	good	Yes
9	middle	false	true	excellent	Yes
10	middle	false	true	excellent	Yes
11	old	false	true	excellent	Yes
12	old	false	true	good	Yes
13	old	true	false	good	Yes
14	old	true	false	excellent	Yes
15	old	false	false	fair	No

An example: the learning task

- Learn a classification model from the data
- Use the model to classify future loan applications into
  - Yes (approved) and
  - No (not approved)
- What is the class for following case/instance?

Age	Has_Job	Own_house	Credit-Rating	Class
young	false	false	good	?

Supervised vs. Unsupervised Learning

- Supervised learning (classification)
  - Supervision: The training data (observations, measurements, etc.) are accompanied by labels indicating the class of the observations
  - New data is classified based on the training set
- Unsupervised learning (clustering)
  - The class labels of training data is unknown
  - Given a set of data, the task is to establish the existence of classes or clusters in the data

Classification by Decision Tree

- Decision tree
  - A flow-chart-like tree structure
  - Internal node denotes a test on an attribute
  - Branch represents an outcome of the test
  - Leaf nodes represent class labels or class distribution
- Decision tree generation consists of two phases
  - Tree construction
    - At start, all the training examples are at the root
    - Partition examples recursively based on selected attributes
  - Tree pruning
    - Identify and remove branches that reflect noise or outliers
- Use of decision tree: Classifying an unknown sample
  - Test the attribute values of the sample against the decision tree

Decision Trees

- One of the simplest and most successful forms of machine learning
- Takes as input a vector of attribute values. Returns a single output value decision

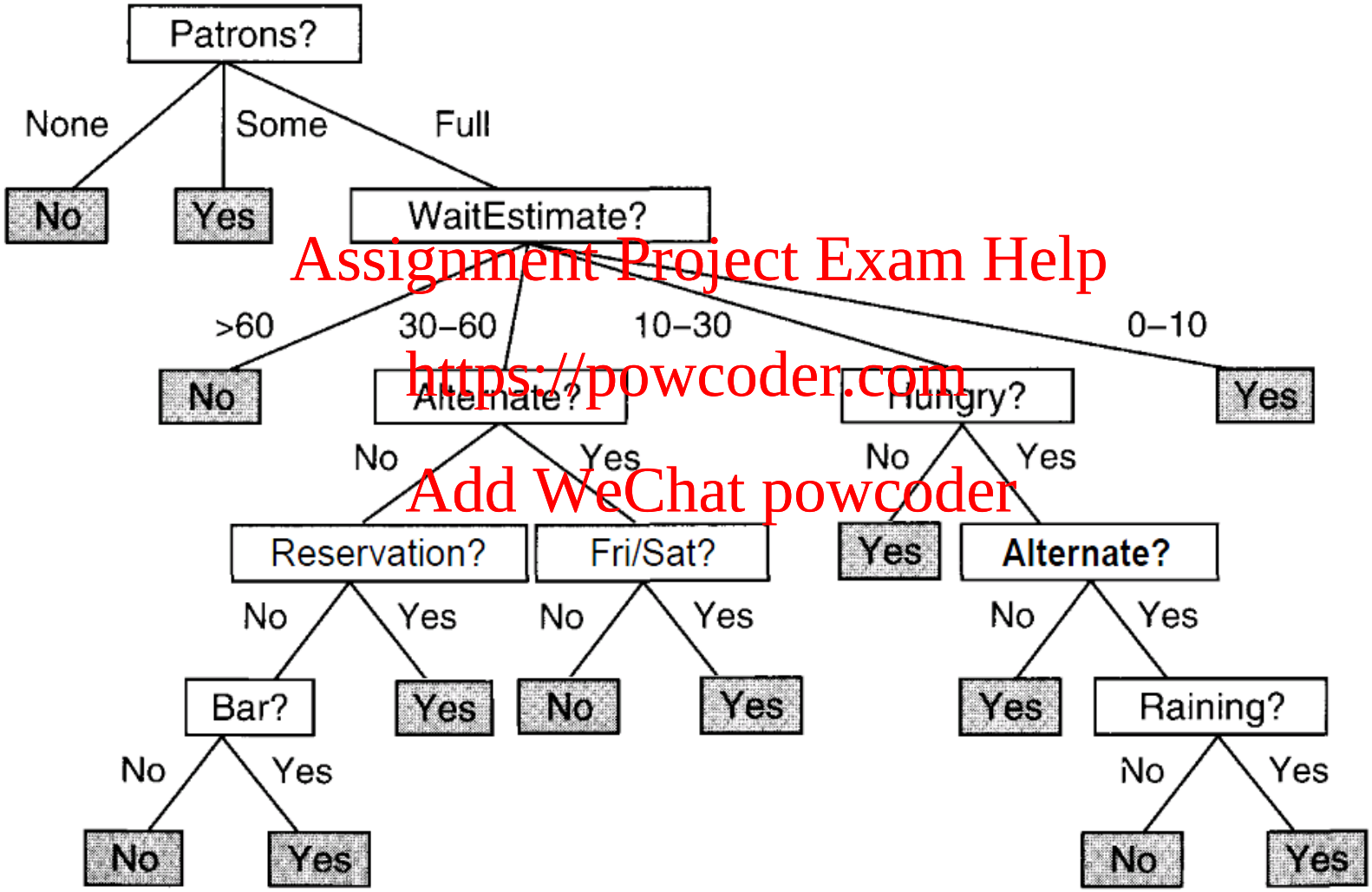
Decision - To wait for a table at a restaurant or not?

Goal – To come up with a function, which gives a boolean output WillWait

Attributes:

- Alternate: Whether there is a suitable alternative restaurant nearby
- Bar: Whether the restaurant has a comfortable waiting lounge
- Fri/Sat: Is it a Friday Saturday
- Hungry: Whether we are hungry
- Patrons: How many people are in the restaurant (values are None, Some, Full)
- The restaurant's pricing range($, $$, $$$)
- Raining: Whether it is raining outside
- Type: The kind of restaurant(French, Italian, Thai, Burger)
- Reservation: Whether we made a reservation
- WaitEstimate: The wait estimated by the waiter (0-10, 10-30, 30-60 or 60>)

Decision Tree - Example



Decision Tree

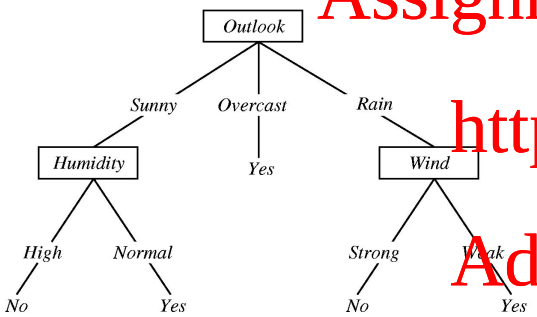
- It represents a human like thinking pattern. We take different attributes into consideration one by one and arrive at a conclusion for many problems
- A decision tree assigns a conclusion by performing a series of tests
- Each internal node in the tree corresponds to a test of the value of an attribute
- The branches from the nodes represent possible values of the attributes
- Each leaf node represents the final value to be returned by the function
- Decision trees are popular for pattern recognition because the models they produce are easier to understand

Decision Tree

- All paths …
  - start at the root node
  - end at a leaf
- Each path represents a decision rule
  - joining (AND) of all the tests along that path
  - separate paths that result in the same class are disjunctions (ORs)
- All paths - mutually exclusive
  - for any one case - only one path will be followed
  - false decisions on the left branch
  - true decisions on the right branch

From Tree to Rules

Converting A Tree to Rules



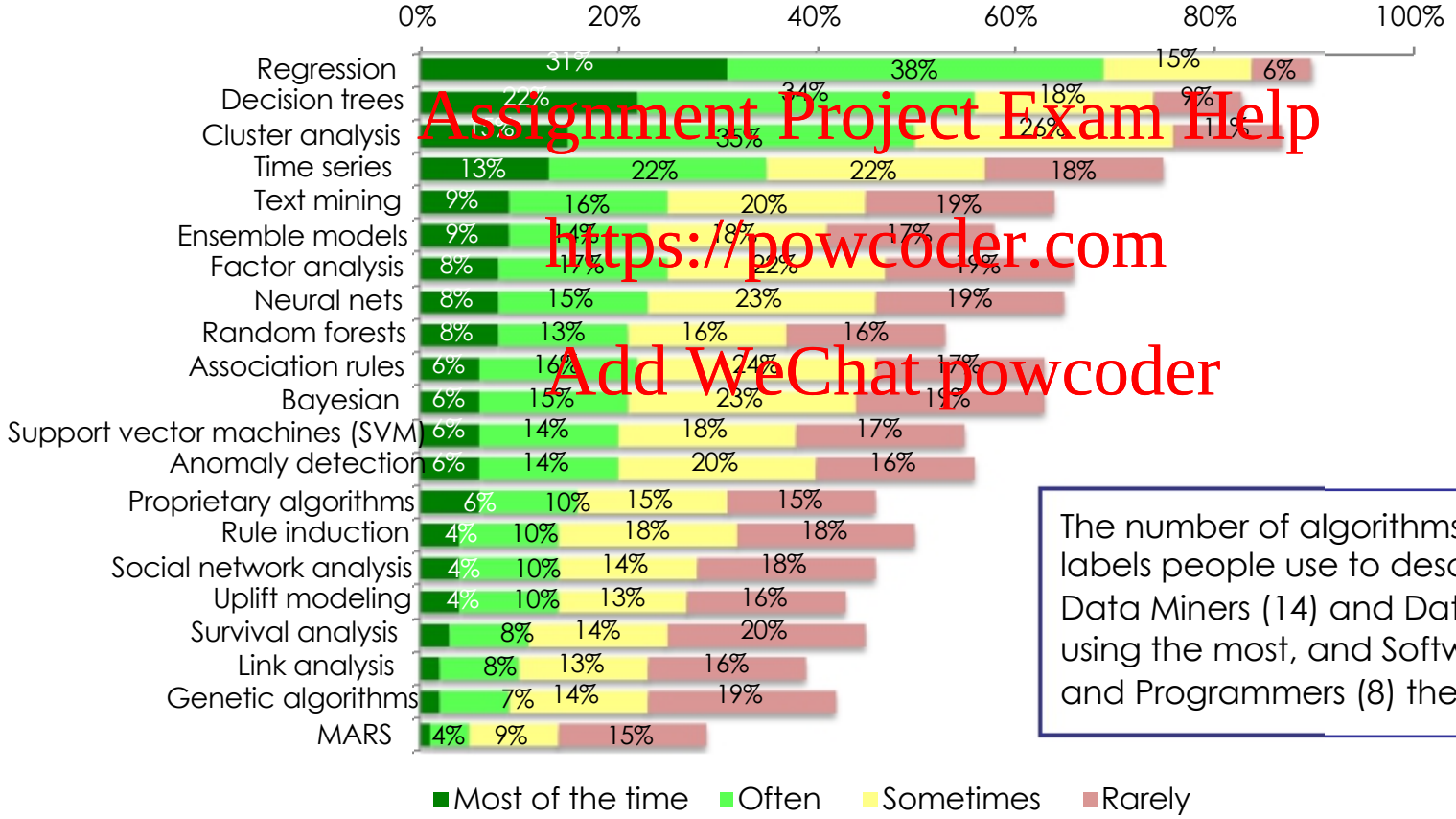
IF $(Outlook = Sunny)\ AND\ (Humidity = High)$
THEN $PlayTennis = No$

IF $(Outlook = Sunny)\ AND\ (Humidity = Normal)$
THEN $PlayTennis = Yes$

...

Algorithms

- Regression, decision trees, and cluster analysis continue to form a triad of core algorithms for most data miners. This has been consistent since the first Data Miner Survey in 2007
- The average respondent reports typically using 12 algorithms. People with more years of experience use more algorithms, and consultants use more algorithms (13) than people working in other settings (11).



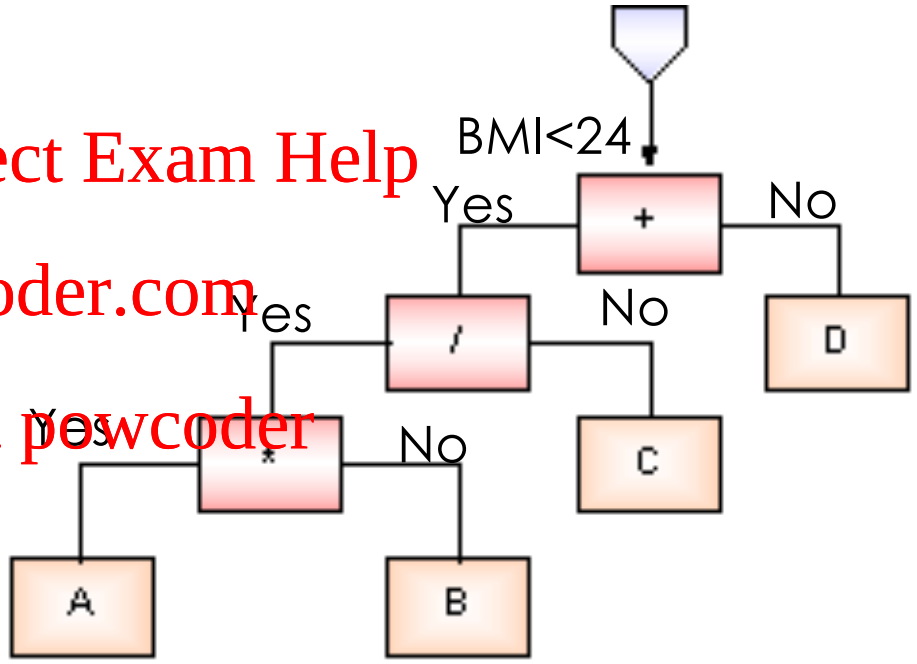
The number of algorithms used varies by the labels people use to describe themselves, with Data Miners (14) and Data Scientists (14) using the most, and Software Developers (9) and Programmers (8) the fewest.

Question: What algorithms / analytic methods do you TYPICALLY use? (Select all that apply)

Source: Rexer Analytics, 2013

Decision trees - Binary decision trees

- Classification of an input vector is done by scanning the tree beginning at the root node, and ending the leaf
- Each node of the tree computes an inequality (ex. BMI<24, yes or no) based on a single input variable
- Each leaf is assigned to a particular class



Decision Trees

- Decision trees can be:
  - **Classification trees,** when the predicted outcome is the class to which the data belongs
  - **Regression trees,** when the predicted outcome can be considered a real number (e.g.: the price of a house, or a patient's length of stay in a hospital)
- The term **Classification And Regression Tree** (**CART**) analysis is an umbrella term used to refer to both of the above procedures, first introduced by Breiman et al.
- Decision trees can be used either as supervised and unsupervised learning tools
  - when in supervised mode, they can be used to create models for future predictions
  - when in unsupervised, they are pure classifiers

Decision trees
- Classification and regression trees (CART)

- CLASSIFICATION AND REGRESSION TREES (CART) are **binary** decision trees, which split a **single variable** at each node
- The CART algorithm recursively goes though an exhaustive search of all variables and split values to find the optimal splitting rule for each node

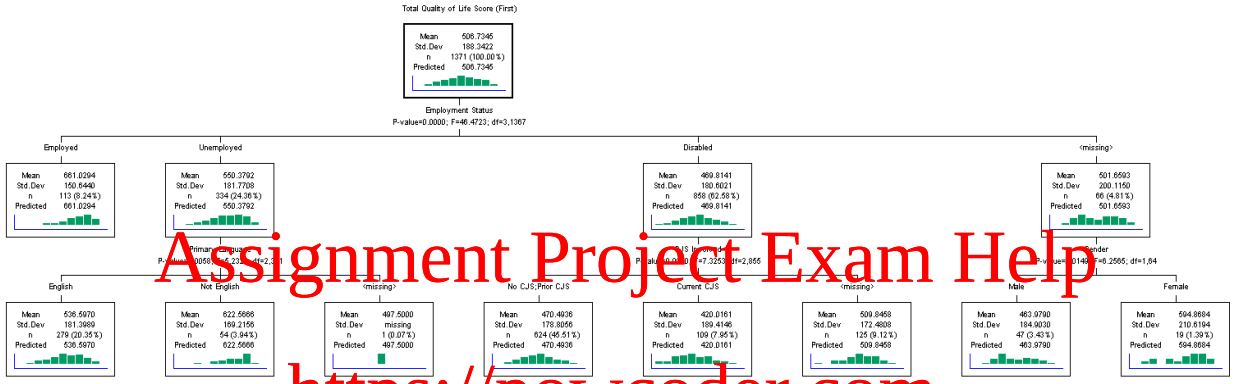
Trees families

- **ID3**, or Iterative Dichotomizer, was the first of three Decision Tree implementations developed by Ross Quinlan (Quinlan, J. R. 1986. Induction of Decision Trees. Mach. Learn. 1, 1 (Mar. 1986), 81-106.)
- **C4.5**, Quinlan's next iteration. The new features (versus ID3) are: (i) accepts both continuous and discrete features; (ii) handles incomplete data points; (iii) solves over-fitting problem by (very clever) bottom-up technique usually known as "pruning"; and (iv) different weights can be applied the features that comprise the training data
- **CART** (Classification And Regression Tree) is often used as a generic acronym for the term Decision Tree. The CART implementation is very similar to C4.5
- **CHAID** (chi-square automatic interaction detector) is a non-binary decision tree. The decision or split made at each node is still based on a single variable, but can result in multiple branches. The split search algorithm is designed for categorical variables

C4.5 and CART

- C4.5 trees differ from CART in several aspects, eg:
- Tests:
  - CART: always binary
  - C4.5: any number of branches
- Test selection criterion:
  - CART: diversity index (Gini)
  - C4.5: information-based criteria
- Pruning:
  - CART: cross-validated using cost-complexity model
  - C4.5: single pass based on binomial confidence limits

Chi-Squared Automatic Interaction Detector (CHAID)



- Continuous variables must be grouped into a finite number of bins to create categories
  - A reasonable number of "equal population bins" can be created for use with CHAID
  - ex. If there are 1000 samples, creating 10 equal population bins would result in 10 bins, each containing 100 samples
- A Chi^2 value is computed for each variable and used to determine the best variable to split on

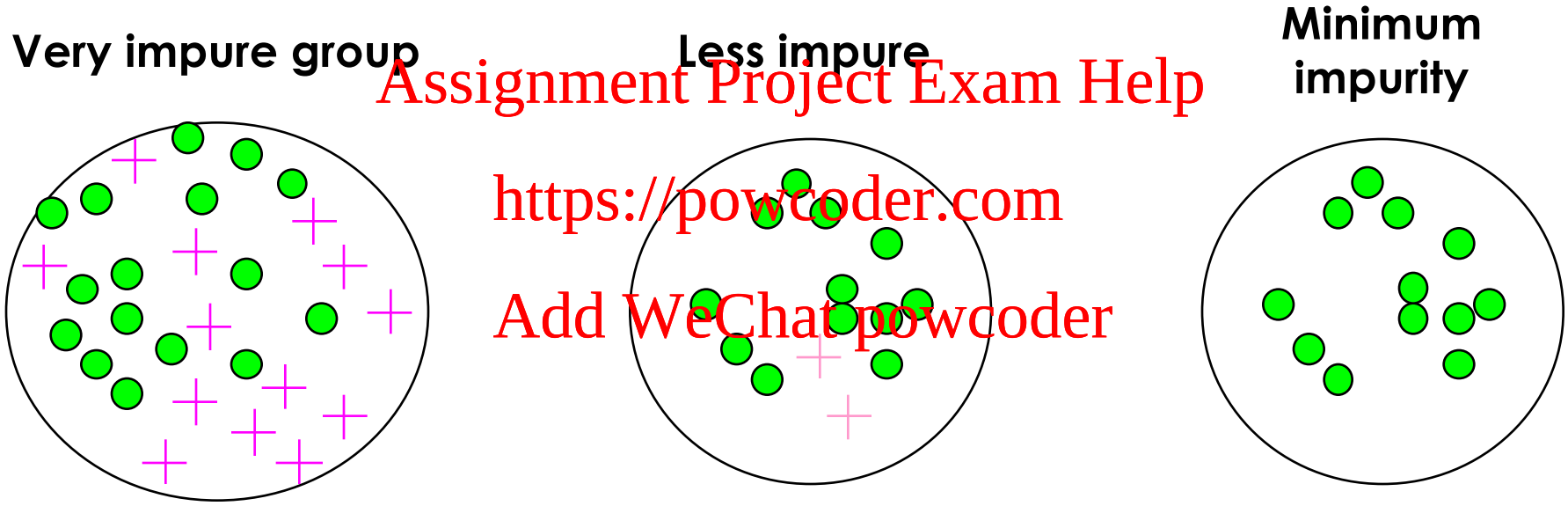
Plan for Construction of a Tree

- Selection of the Splits
- Decisions when to decide that a node is a terminal node (i.e. not to split it any further)
- Assigning a class to each terminal node

Impurity of a Node

- Need a measure of impurity of a node to help decide on how to split a node, or which node to split
- The measure should be at a maximum when a node is equally divided amongst all classes
- The impurity should be zero if the node is all one class

Impurity



Information Gain

- We want to determine which attribute in a given set of training feature vectors is most useful for discriminating between the classes to be learned.
- Information gain tells us how important a given attribute of the feature vectors is
- We will use it to decide the ordering of attributes in the nodes of a decision tree

Information Gain

- Impurity/Entropy (informal)
  - Measures the level of impurity in a group of examples

Measures of Impurity

- Misclassification Rate
- Information, or Entropy
- Gini Index

In practice the first is not used for the following reasons:

- Situations can occur where no split improves the misclassification rate

- The misclassification rate can be equal when one option is clearly better for the next step

Entropy

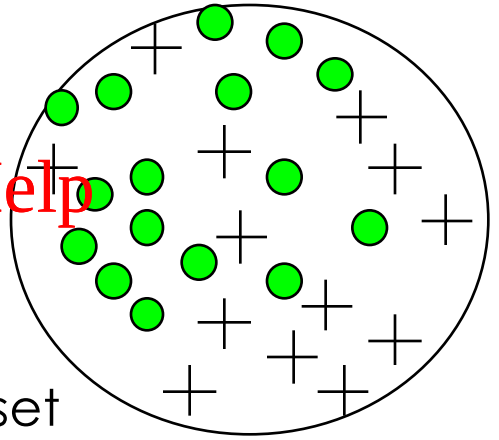
- Is a measure of information or uncertainty of a random variable
- Entropy comes from information theory. More the uncertainty, the higher the entropy, the more the information content
- For example if we toss a coin which always falls on head, we gain no information by tossing it, so zero entropy. However if we toss a fair coin, we are unsure of the outcome. So we get some information out of tossing it

Entropy

- Entropy = $\sum_i p_i \log_2 p_i$

  p_i is the probability of class i

  Compute it as the proportion of class i in the set



- For a fair coin H(Fair) = - (0.5$\log_2$ 0.5 + 0.5 $\log_2$ 0.5) = 1
- For a Biased coin H(Biased) = - 1 $\log_2$ (1) = 0

2-Class Cases:

- What is the entropy of a group in which all examples belong to the same class?

  entropy = - 1 $\log_2 1 = 0$

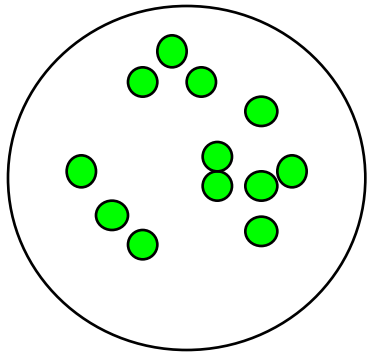
  not a good training set for learning

- What is the entropy of a group with 50% in either class?

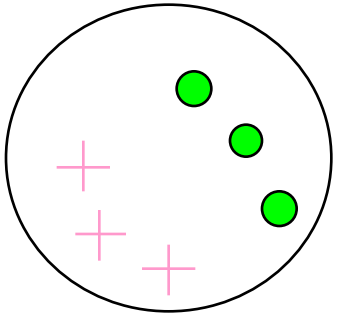
  entropy = -0.5 $\log_2 0.5$ – 0.5 $\log_2 0.5$ =1

  good training set for learning

Minimum impurity



Maximum impurity

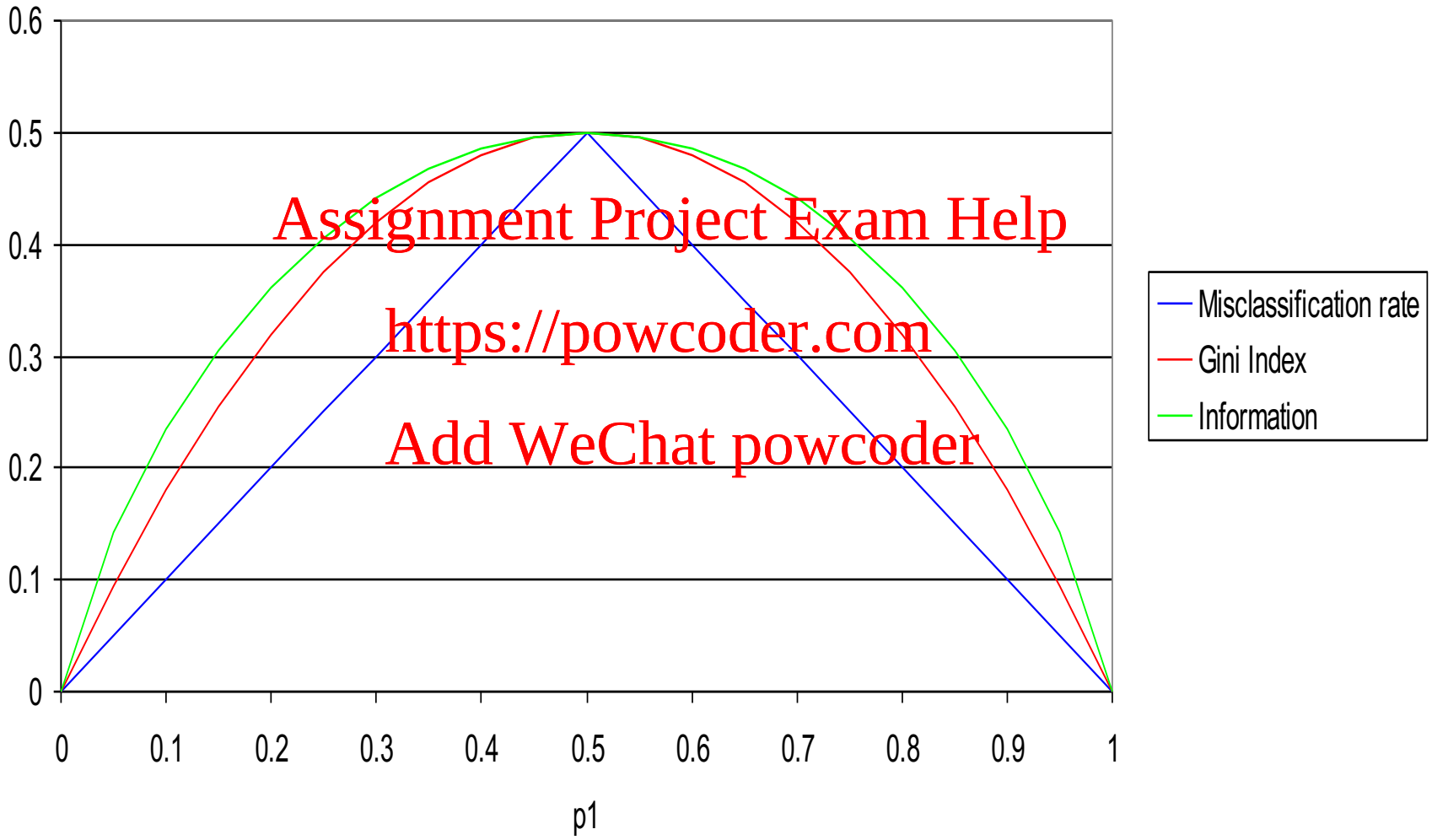


Gini Index

- This is the most widely used measure of impurity (at least by CART)
- Gini* index is:

$$i(p) = \sum_{i} \sum_{j \neq i} p_i p_j = 1 - \sum_{j} p_j^2$$

*: Corrado Gini (1884 – 1965) developed the Gini coefficient, a measure of the income inequality in a society



Tree Impurity

- We define the impurity of a tree to be the sum over all terminal nodes of the impurity of a node multiplied by the proportion of cases that reach that node of the tree
- Example: Impurity of a tree with one single node, with both A and B having 400 cases, using the Gini Index:
  - Proportions of the two cases = 0.5
  - Therefore Gini Index = $1 - (0.5)^2 - (0.5)^2 = 0.5$

Decision Tree - Exercise

Occupation	Gender	Age	Salary
Service	Female	45	$48,000
	Male	25	$25,000
	Male	33	$35,000
Management	Male	25	$45,000
	Female	35	$65,000
	Male	26	$45,000
	Female	45	$70,000
Sales	Female	40	$50,000
	Male	30	$40,000
Staff	Female	50	$40,000
	Male	25	$25,000

Using the dataset above (available at the Dataset page in the course website as "Salaries.csv"), construct a classification and regression tree to classify Salary, based on the other variables.
Use Excel first

Decision Tree – Exercise – Hints 1

Possible Splits

Candidate Split	Left Child Node, tL	Right Child Node, tR
1	Occupation=Service	Occupation=Management|Sales|Staff
2	Occupation=Management	Occupation=Service|Sales|Staff
3	Occupation=Sales	Occupation=Service|Management|Staff
4	Occupation=Staff	Occupation=Service|Sales|Management
5	Gender=Female	Gender=Male
6	Age<=25	Age>25
7	Age<=35	Age>35
8	Age<=45	Age>45

Possible Criteria: Salary <=$45,000 vs >$45,000

Decision Tree – Exercise – Hints 2

Possible Execution

Number of cases:	11	COUNT(C2:C12)
Salary <=$45K	7	COUNTIF(D2:D12,"<=45000")
Salary >$45K	4	COUNTIF(D2:D12,">45000")
1st H(T):	0.9457	-(E24/E23)*LOG((E24/E23),2)-(E25/E23)*LOG((E25/E23),2)
Number of Staff:	2	COUNTIF(B2:B12,"Staff")
Salary <=$45K - Staff:	2	
Salary >$45K - Staff:	0	
Salary <=$45K - Staff H(T):	0.0000	-(E20/E19)*LOG((E20/E19),2)-(E21/E19)*LOG((E21/E19),2)
Number of not Staff:	9	E14-E19
Salary <=$45K - Not Staff:	5	
Salary >$45K - Not Staff:	4	
Salary <=$45K - Not Staff H(T):	0.8108804	(E24/E$14)*(-(E25/E24)*LOG((E25/E24),2)-(E26/E24)*LOG((E26/E24),2))
Total Impurity After Split:	0.8108804	E22+E27
Gain:	0.1347799	E17-E29

Selection of Splits

- We select the split that most decreases the Gini Index/better entropy reduction. This is done over all possible places for a split and all possible variables to split
- We keep splitting until the terminal nodes have very few cases or are all pure
- This is may lead to large trees that will then need to be pruned

Overfitting

- Overfitting is a scenario where the decision tree algorithm will generate a large tree when there is actually no pattern to be found. This is a common problem with all types of learners

- Reasons for overfitting

  - Some attributes have little meaning

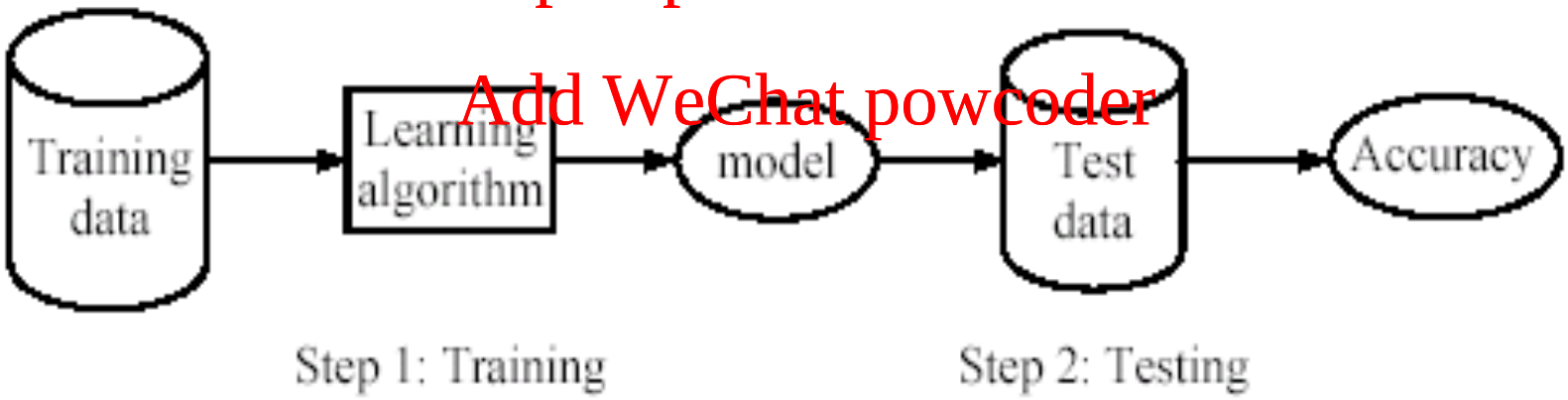
  - Number of attributes too high
  - Small training data set

Combating Overfitting – Pruning

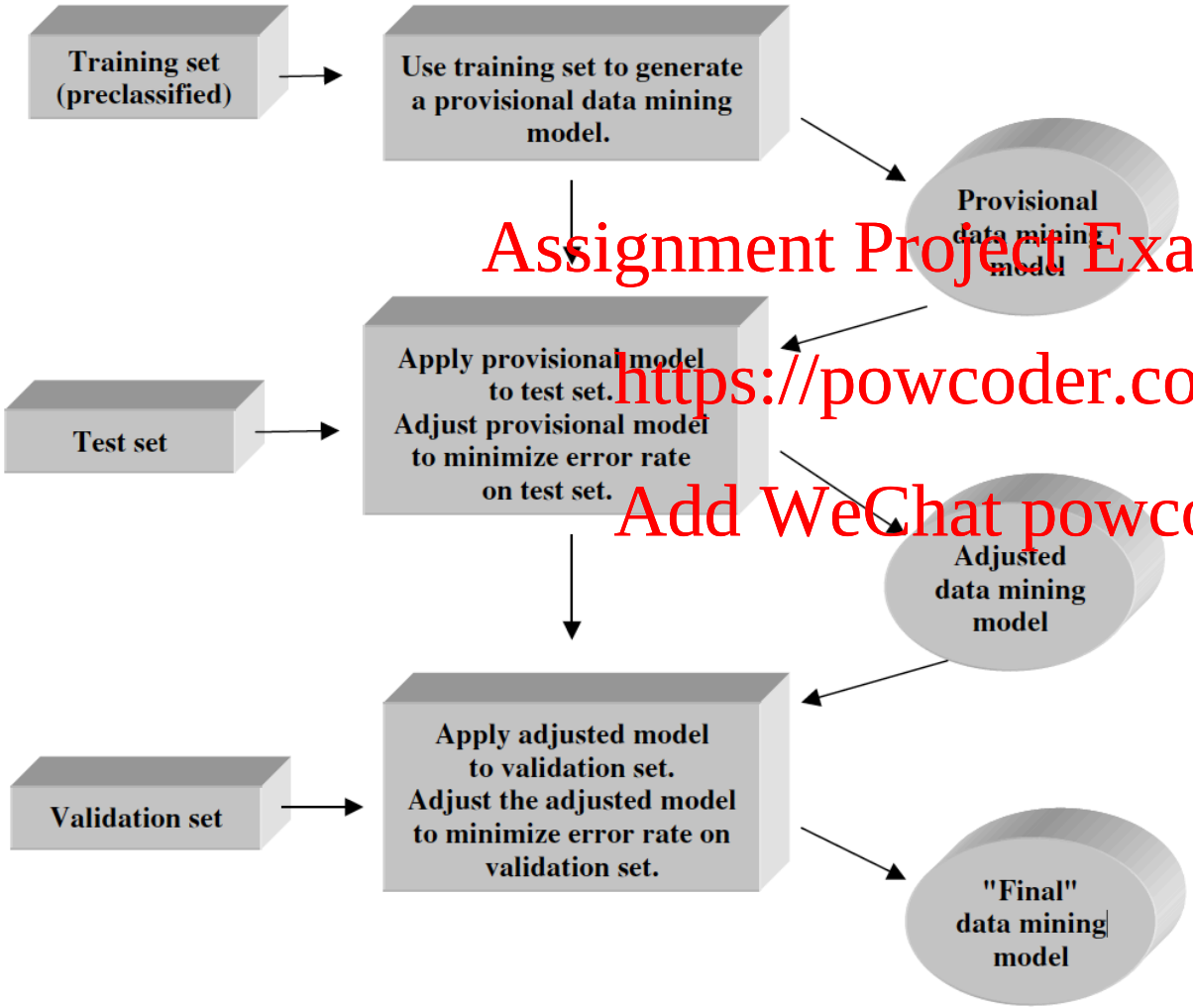
- It works by eliminating nodes that are not clearly relevant
- One of the possible and most used way to determine the relevance of a branch is to measure the gain in entropy reduction it provides compared to the previous/other branches
- Pruning is cutting the branches providing relative minor contribution

Supervised learning process: two steps

- Learning (training): Learn a model using the training data
- Testing: Test the model using unseen test data to assess the model accuracy



Supervised learning process: Three steps



- A third step can be added, splitting the original dataset in 3 parts: **Training**, **Testing** and **Validate**
- This is the default for Rattle

Learning in Data Mining

- Given
  - a data set D
  - a task T
  - a performance measure M
  - a computer system is said to learn from D to perform the task T if after learning the system's performance on T improves as measured by M
- In other words, the learned model helps the system to perform T better as compared to no learning

Fundamental assumption of learning

Assumption: The distribution of training examples is identical to the distribution of test examples (including future unseen examples)

- In practice, this assumption is often violated to certain degree
- Strong violations will clearly result in poor classification accuracy
- To achieve good accuracy on the test data, training examples must be sufficiently representative of the test data

Model Evaluation

- Evaluation metrics: How can we measure accuracy?
- Use validation test set of class-labeled tuples instead of training set when assessing accuracy
- Methods for estimating a classifier's accuracy:
  - Holdout method, random subsampling
  - Cross-validation
  - Bootstrap
- Comparing classifiers:
  - Confidence intervals
  - Cost-benefit analysis and ROC Curves

Classifier Evaluation Metrics: Confusion Matrix

Actual class\Predicted class	yes	no
yes	**True Positives (TP)**	**False Negatives (FN)**
no	**False Positives (FP)**	**True Negatives (TN)**

Example of Confusion Matrix:

Actual class\Predicted class	buy_computer = yes	buy_computer = no	Total
buy_computer = yes	**6954**	**46**	7000
buy_computer = no	**412**	**2588**	3000
Total	7366	2634	10000

- TP and TN are the correctly predicted tuples
- May have extra rows/columns to provide totals

Classifier Evaluation Metrics: Accuracy, Error Rate, Sensitivity and Specificity

A\P	Y	N	
Y	**TP**	**FN**	**P**
N	**FP**	**TN**	**N**
	P'	**N'**	**All**

- Classifier Accuracy, or recognition rate: percentage of test set tuples that are correctly classified
  - **Accuracy** = (TP + TN)/All
- Error rate: misclassification rate
- 1 – accuracy, or
  - **Error rate** = (FP + FN)/All

- Class Imbalance Problem:
  One class may be rare, e.g. fraud, or HIV-positive
  Significant majority of the negative class and minority of the positive class

Sensitivity: True Positive recognition rate
- Sensitivity = TP/P

Specificity: True Negative recognition rate
- Specificity = TN/N

Classifier Evaluation Metrics: Precision and Recall, and F-measures

- **Precision**: exactness – what % of tuples that the classifier labeled as positive are actually positive

$$precision = \frac{TP}{TP+FP}$$

- **Recall**: completeness – what % of positive tuples did the classifier label as positive?

$$recall = \frac{TP}{TP+FN}$$

- Perfect score is 1.0
- **F measure** (F1 or F-score): harmonic mean of precision and recall,

$$F = \frac{2 \times precision \times recall}{precision + recall}$$

Classifier Evaluation Metrics: Example

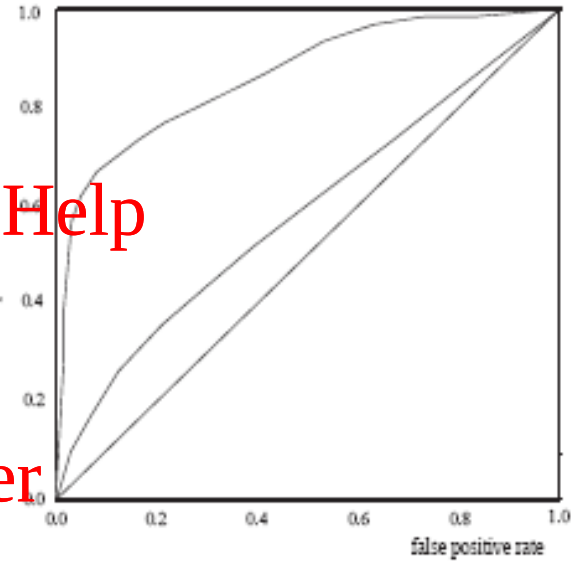
Actual Class\Predicted class	cancer = yes	cancer = no	Total	Recognition(%)
cancer = yes	**90**	**210**	300	30.00 (*sensitivity*)
cancer = no	**140**	**9560**	9700	98.56 (*specificity*)
Total	230	9770	10000	96.40 (*accuracy*)

- *Precision = ??* *Recall = ??*

$$precision = \frac{TP}{TP+FP} \qquad recall = \frac{TP}{TP+FN}$$

Model Selection: ROC Curves

- ROC (Receiver Operating Characteristics) curves: for visual comparison of classification models
- Originated from signal detection theory
- Shows the trade-off between the true positive rate and the false positive rate
- The area under the ROC curve is a measure of the accuracy of the model
- Diagonal line: for every TP, equally likely to encounter FP
- The closer to the diagonal line (i.e., the closer the area is to 0.5), the less accurate is the model



- Vertical axis represents the true positive rate
- Horizontal axis rep. the false positive rate

Information from R/Rattle

- Rpart (the R algorithm used by Rattle to generate the trees) provides a text view of the tree, containing per each split:

  node), split, n, loss, yval, (yprob)

  Where node) is the node number, split is name/condition of the node, n is the number of entities at the node, loss is the number of entities that are incorrectly classified at that node, yval the default classification for the node, yprob contains the distribution of classes in that node. The following is an example: 1) root 256 41 No (0.83984375 0.16015625)

- Using "**CP**", that is the threshold complexity parameter. It is an argument that is used to control the maximum size of the tree selecting a value of the cost complexity parameter. It is always monotonic with the number of splits. The smaller the value of CP, the more complex will be the tree (the greater the number of splits).

	CP	nsplit	rel error	xerror	xstd
1	0.082621	0	1.00000	1.00000	0.049197
2	0.076923	4	0.66952	0.76638	0.043951
3	0.047009	5	0.59259	0.67236	0.041495
4	0.024217	8	0.41880	0.48718	0.036884
5	0.015670	10	0.37037	0.45299	0.034679
6	0.012821	13	0.32194	0.45869	0.034880
7	0.011396	16	0.28205	0.46154	0.034980
8	0.010000	18	0.25926	0.45584	0.034780

  - For a regression tree, the relative error (**rel error**) is the average deviance of the current tree divided by the average deviance of the null tree
  - The cross-validation error (**xerror**) is based on a 10-fold cross-validation and is measured relative to the deviance of the null model. The cross-validation error is greater than the relative error

Decision Tree - Exercise

Using the dataset "weather.csv, construct a classification and regression tree to classify "RainTomorrow", based on the other variables

- Consider **Date** as "Ident" variable
- Ignore variable **RISK_MM**
- Consider first a partition 70/30 (no validation set)
- Use first
  - Complexity = 0, Min. Split = 2, Min. Bucket = 1

  and then use the model metrics to prune it by tuning the above parameters
- Explain the results