Data Mining and Data Warehousing 8

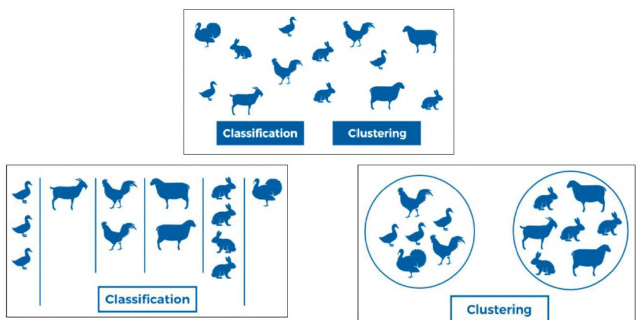
Classification and Prediction

Chittaranjan Pradhan
School of Computer Engineering,
KIIT University

Supervised vs. Unsupervised Learning

Supervised vs. Unsupervised Learning

- Classification and clustering are two methods of pattern identification with some similarity and disimilarity
- Classification uses predefined classes in which objects are assigned
- Clustering identifies similarities between objects, which it groups according to those characteristics in common and which differentiate them from other groups of objects. These groups are known as *clusters*



Supervised vs. Unsupervised Learning...

Supervised Learning (Classification)

- It is the data mining technique used to predict group membership for data instances
- There are two ways to assign a new value to a given class
  - Crispy Classification: Given an input, the classifier exactly returns its class label
  - Probabilistic Classification: Given an input, the classifier returns its probabilities to belong to each class. This is useful when some mistakes can be more costly than others

Unsupervised Learning (Clustering)

- It is a technique of organising a group of data into classes and clusters where the objects reside inside a cluster will have high similarity and the objects of two clusters would be dissimilar to each other
- In clustering, the similarity as well as distance between two objects is measured by the similarity function. Generally, Intra-cluster distance are less than intercluster distance

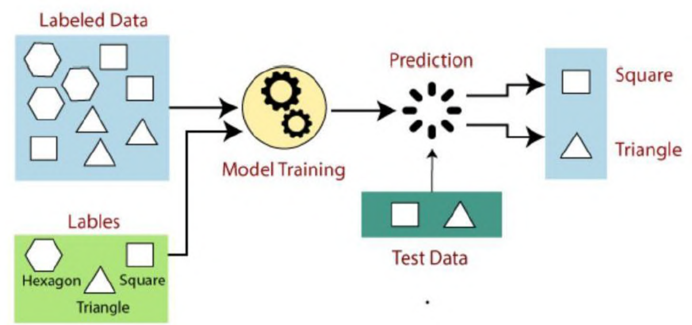
Supervised vs. Unsupervised Learning...

Parameter	Classification	Clustering
Type	Used for supervised learning	Used for unsupervised learning
Basic	process of classifying the input instances based on their corresponding class labels	grouping the instances based on their similarity without the help of class labels
Need	it has labels, so there is need of training and testing dataset for verifying the created model	there is no need of training and testing dataset
Complexity	more complex	less complex
Example Algorithms	Logistic regression, Naive Bayes classifier, Support vector machines, KNN, Decision Tree etc.	k-means clustering algorithm, Fuzzy c-means clustering algorithm, Gaussian (EM) clustering algorithm, etc.
Application	Detection of unsolicited email Recognition of the face Approval of a Bank Loan	Investigation of the social networks Segmentation of an image Recommendation Engines

Supervised vs. Unsupervised Learning...

Supervised Learning (Classification)

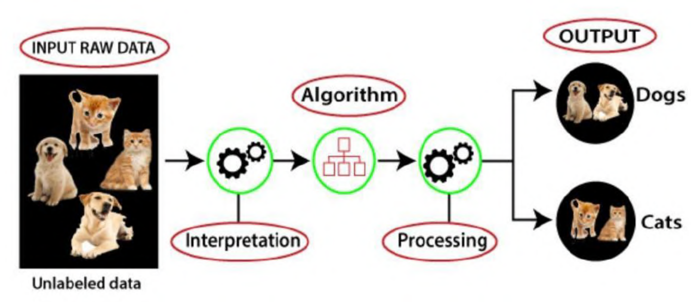
- Models are trained using labeled data
- Models need to find the mapping function to map the input variable (X) with the output variable (Y)
  Y= f(X)
- Supervised learning needs supervision to train the model, which is similar to as a student learns things in the presence of a teacher



Supervised vs. Unsupervised Learning...

Unsupervised Learning (Clustering)

- In Unsupervised learning patterns are inferred from the unlabeled input data
- The goal of unsupervised learning is to find the structure and patterns from the input data
- Unsupervised learning does not need any supervision. Instead, it finds patterns from the data by its own



Training Dataset vs. Test Dataset

Training Dataset vs. Test Dataset

- The training data is the biggest subset (>=60%) of the original dataset, which is used to train or fit the model. Firstly, the training data is fed to the algorithms, which lets them learn how to make predictions for the given task. The type of training data that we provide to the model is highly responsible for the model's accuracy and prediction ability. It means that the better the quality of the training data, the better will be the performance of the model
- The test dataset is another well-organized subset (20%-40%) of original data, which is independent of the training dataset. However, it has some similar types of features and class probability distribution for each type of scenario and uses it as a benchmark for model evaluation once the model training is completed

Classification vs. Prediction

Classification vs. Prediction

- Classification and prediction are two forms of data analysis that can be used to extract models describing important data classes or to predict future data trends
- Classification models predict categorical class labels; and prediction models predict continuous valued functions
- **Classification** is the process of identifying the category or class label of the new observation to which it belongs. For this, the classification extracts the classification model (classifier) from the given training set and then uses this model to decide the category on new data tuple to be classfied

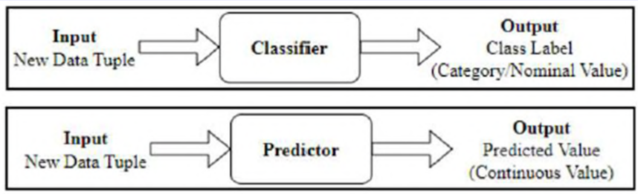
  Ex: Before starting of your 7th sem project, you want to predict whether it is accepted or rejected in final project defense

Classification vs. Prediction...

Classification vs. Prediction...

- **Prediction** is the process of identifying the missing or unavailable numerical data for a new observation. For this, prediction learns the continuous valued function from the given training set and then this function to predict the continuous value of new data tuple for which prediction is required

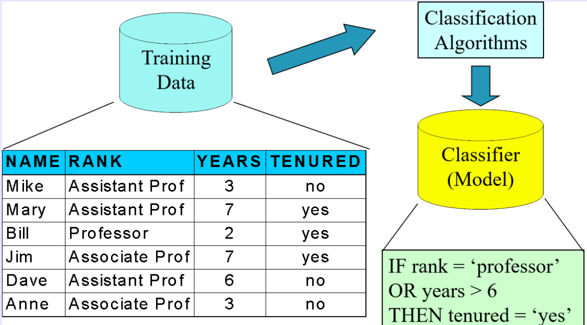
  Ex: Before starting 7th sem, if you want to predict how much score you want to obtain, here you use a prediction model by consulting previously obtained marks



Classification

Classification - A Two-Step Process

- **Model or Classifier Construction**: describing a set of predetermined Classes
  - Also called training phase or learning stage
  - Building the Classifier or Model
  - Each tuple/sample is assumed to belong to a predefined class, as defined by the class label attribute
  - The set of tuples used for model construction: training set
  - The model is represented as classification rules / decision trees / mathematical formulae

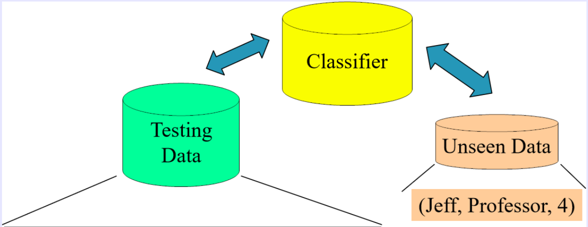


NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

Classification...

Classification - A Two-Step Process...

- **Model Usage**: for classifying future or unknown objects
  - Using classifier for classification estimate accuracy of model
  - The known label of test sample is compared with the classified result from the model
  - Accuracy rate is the percentage of test set that are correctly classified by the model
  - Test set is independent of training set, otherwise over-fitting will occur



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

Classification...

Issues regarding classification and prediction

- **Data Preparation**
  - Data cleaning: preprocesses data in order to reduce noise and handle missing values
  - Relevance analysis (feature selection): remove the irrelevant or redundant attributes
  - Data transformation: generalize and/or normalize data
- **Evaluating Classification Methods**
  - Predictive accuracy
  - Speed and scalability: time to construct the model; and time to use the model
  - Robustness: handling noise and missing values
  - Scalability: efficiency in disk-resident databases
  - Interpretability: understanding and insight provded by the model
  - Goodness of rules: decision tree size; and compactness of classification rules

Classification by Decision Tree Induction

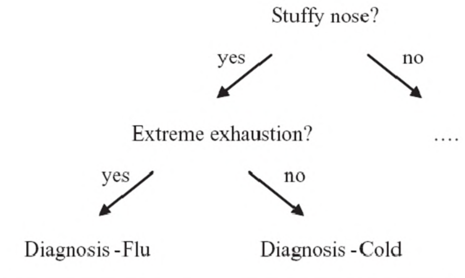
Classification by Decision Tree Induction

- 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
  - The **topmost** node in the tree is the root node
- 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

Classification by Decision Tree Induction...

Decision tree Example

Ex: You may visit a doctor and your doctor may ask you to describe your symptoms. You respond by saying you have a stuffy nose. In trying to diagnose your condition the doctor may ask you further questions such as whether you are suffering from extreme exhaustion. Answering yes would suggest you have the flu, where as answering no would suggest you have a cold. This line of questioning is common to many decision making processes and can be shown visually as a decision tree

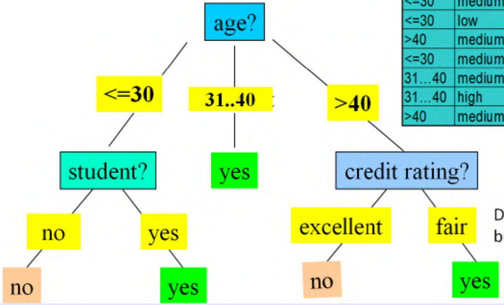


Decision Tree for the diagnosis of cold and Flu

Classification by Decision Tree Induction...

Decision tree Example

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no



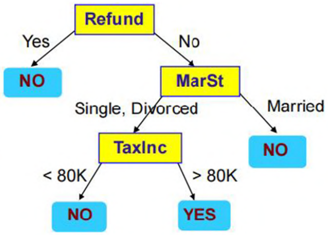
Decision tree for the concept buys computer

Classification by Decision Tree Induction...

Decision tree Example

Tid	Refund	Marital Status	Taxable Income	Cheat
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

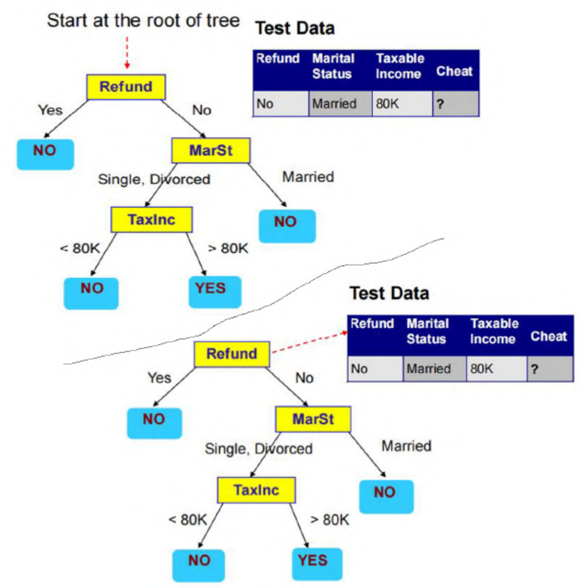
Training Data



Model: Decision Tree

Classification by Decision Tree Induction...

Decision tree Example

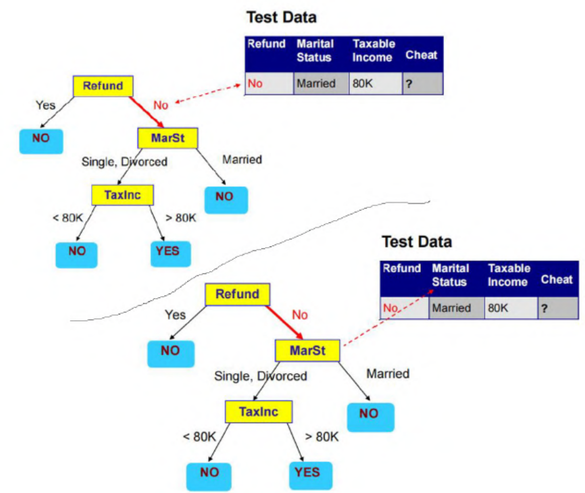


Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

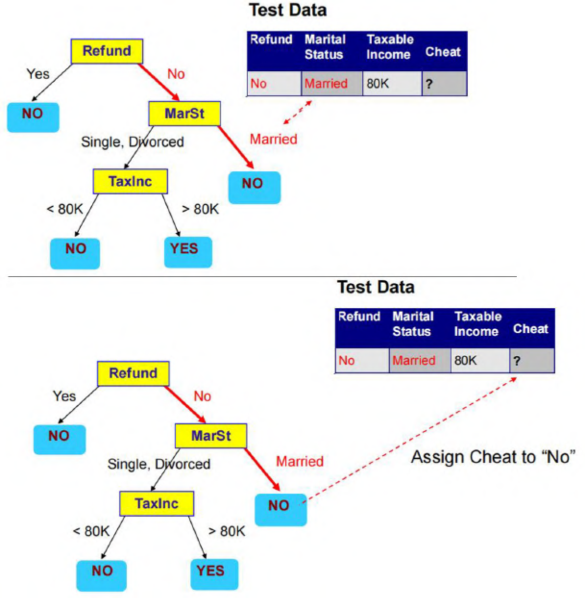
Classification by Decision Tree Induction...

Decision tree Example



Classification by Decision Tree Induction...

Decision tree Example



Classification by Decision Tree Induction...

Why Decision tree classifier are so popular?

- The construction of a decision tree doesn't require any domain knowledge or parameter setting
- They can handle high dimensional data
- The learning and classification steps of decision tree are simple and fast
- They have a high accuracy
- It is easily understandable by humans

How are Decision trees used for Classification?

- Given a tuple, X, for which the associated class label is unknown, the attribute values of the tuple are tested against the decision tree
- A path is traced from the root to a leaf node, which holds the class prediction for that tuple
- Decision trees can easily be converted to classification rules

Algorithm for Decision Tree

Basic Algorithm (A Greedy algorithm)

1. At start, all the training tuples are at the root
2. Tuples are partitioned recursively based on selected attributes
3. If all samples for a given node belong to the same class -> Label the class
4. If there are no remaining attributes for further partitioning -> Majority voting is employed for classifying the leaf
5. There are no samples left -> Label the class and terminate
6. Else -> Go to step 2

Attribute Selection Measure

Attribute Selection Measure

- Attribute selection measure finds the best attributes in sequence which can give a smallest decision tree. It provide a splitting rule that determine how the tuple at a give node can be best split
- On the basis of sample dataset/traning dataset, attribute selection measure provide a ranking for each attibute
- The attribute with best rank/score for the measure is choosen as the splitting attribute for the given tuples
- **Information Gain (ID3 (Iterative Dichotomiser 3)/C4.5)**
  - All attributes are assumed to be categorical
  - Can be modified for continuous-valued attributes
- **Gini Index (IBM IntelligentMiner)**
  - All attributes are assumed continuous-valued
  - Assume there exist several possible split values for each attribute
  - May need other tools, such as clustering, to get the possible split values
  - Can be modified for categorical attributes

Attribute Selection Measure...

Attribute Selection Measure...

- Is a heuristic measure for selecting the splitting criterion that best separates a given data partition D, of class-labeled training tuples into individual classes
- If we split the D into smaller partitions according to the outcomes of the splitting criterion, then ideally, each partition will be PURE
- Therefore, best splitting criterion is needed
- It is also known as splitting rules, as they determine how the tuples at a given node are to be split
- It provides a ranking for each attribute, describing the given tuples
- The attributes having best score is chosen as splitting attribute for the given tuples

Information Gain (ID3/C4.5) Method

Information Gain (ID3/C4.5) Method

- ID3 uses information gain as its attribute selection measure
- The attribute with the highest information gain is chosen as the splitting attribute for node N

- Entropy/Information is the probabilistic measure of uncertanity/variance/randomness present in the dataset/attribute
- Information Gain is the decrease/increase in Entropy value when the node is split. It is the measure of a reduction of uncertainty/variance
- Based on the computed values of Entropy and Information Gain, the best attribute at the respective level is choosen

Information Gain (ID3/C4.5) Method...

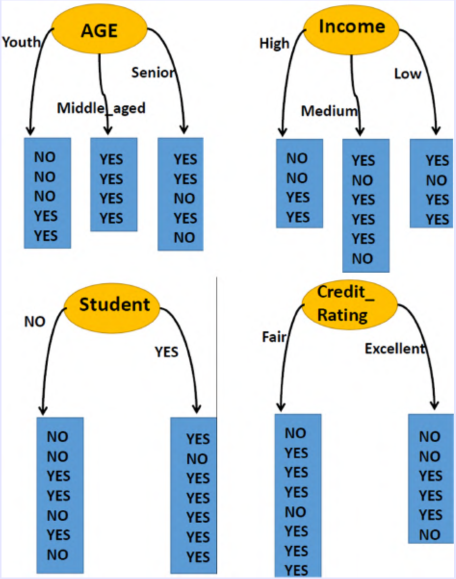
- To find the best feature that serves as a root node in terms of information gain:
  - Find the Expected Entropy/Information of Dataset: the average amount of information ned to identify the class label of a tuple in D
    $Entropy(D) or Info(D) = -\sum_{i=1}^{m} p_i log_2(p_i)$
    where, m-> number of Classes or labels and p_i -> probability of class i
  - Find the Etropy/Information of individual feature/attribute:
    $Entropy_A(D) or Info_A(D) = \sum_{j=1}^{v} \frac{D_j}{D} x Info(D_j)$
    where, v-> number of variations of attribute A
  - Find the information gain of individual attribute(s):
    $Gain(A) = Info(D) - Info_A(D)$
  - Choose the best Information gain i.e. Information gain with highest value
- Repeat this process to construct the complete decision tree

Information Gain (ID3/C4.5) Method...

Information Gain (ID3/C4.5) Method...

RID	Age	Income	Student	Credit_Rating	Class: Buys Computer
1	Youth	High	No	Fair	No
2	Youth	High	No	Excellent	No
3	middle_aged	High	No	Fair	Yes
4	senior	Medium	No	Fair	Yes
5	senior	Low	Yes	Fair	Yes
6	senior	Low	Yes	Excellent	No
7	middle_aged	Low	Yes	Excellent	Yes
8	Youth	Medium	No	Fair	No
9	Youth	Low	Yes	Fair	Yes
10	senior	Medium	Yes	Fair	Yes
11	Youth	Medium	Yes	Excellent	Yes
12	middle_aged	Medium	No	Excellent	Yes
13	middle_aged	High	Yes	Fair	Yes
14	senior	Medium	No	Excellent	No

Information Gain (ID3/C4.5) Method...



Information Gain (ID3/C4.5) Method...

Information Gain (ID3/C4.5) Method...

Step 1: Find the Expected Entropy/Information of Dataset:

$$Entropy(D) \, or \, Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

- m: number of Classes or labels=2
- p_i: Probability of Class i

$$Info(D) = -\left(\underbrace{\frac{9}{14}\log_2\frac{9}{14}}_{\text{Class YES of D}} + \underbrace{\frac{5}{14}\log_2\frac{5}{14}}_{\text{Class NO of D}}\right) = 0.94$$

Step 2: Iteratively find the **Best Atrribute** for **each level** of the Decision ree.

Iteration 1: Find Best Attribute for 0th level by computing **Gain of each Attribute**

a. Age attribute: $Entropy_{Age}(D) \, or \, Info_{Age}(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$

- m: number of Classes or labels=2
- p_i: Probability of Class i

$$= \underbrace{\frac{5}{14}\left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right)}_{\text{Class Variation in Youth}} + \underbrace{\frac{4}{14}\left(-\frac{4}{4}\log_2\frac{4}{4}\right)}_{\text{Class Variation in Middle\_aged}} + \underbrace{\frac{5}{14}\left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right)}_{\text{Class Variation in Senior}} = \mathbf{0.694}$$

Gain(Age)= Entropy (D) - Entropy$_{Age}$(D) = 0.940 - 0.694 = 0.246

Information Gain (ID3/C4.5) Method...

Information Gain (ID3/C4.5) Method...

Step 2: Iteratively find the **Best Atrribute** for **each level** of the Decision ree cont..

Iteration 1: Find Best Attribute for For 0^{th} level Cont...

Similarly find the gain of attribute "income", "student", "credit_rating"

b. Gain of income = 0.029

c. Gain of student = 0.151

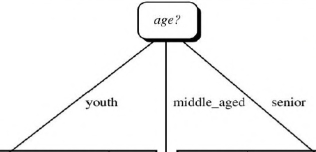
d. Gain of credit_rating = 0.048

Age has highest information gain from rest of the attribute, and hence is seleceted as splitting attribute.

Information Gain (ID3/C4.5) Method...

Information Gain (ID3/C4.5) Method...

Level 0 Decision Tree using 1st Splitting Attribute Age



age = youth:

income	*student*	*credit_rating*	*class*
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

age = senior:

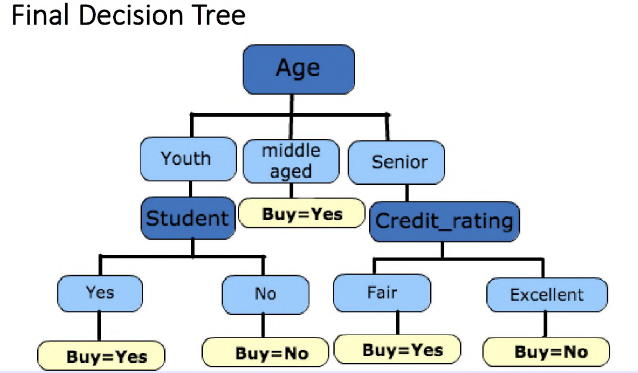
income	*student*	*credit_rating*	*class*
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

age = middle_aged:

income	*student*	*credit_rating*	*class*
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

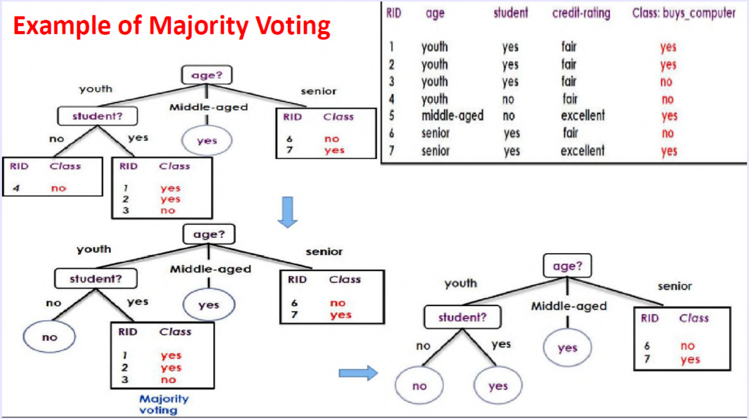
- **Tupple Falling on age=middle_aged all belong to the same class "YES" and hence a leaf can be created for that branch.**
- **continue the tree construction with finding the best gain for data-partion age=youth and age=senior**

Information Gain (ID3/C4.5) Method...



Information Gain (ID3/C4.5) Method...

Majority Voting



RID	age	student	credit-rating	Class: buys_computer
1	youth	yes	fair	yes
2	youth	yes	fair	yes
3	youth	yes	fair	no
4	youth	no	fair	no
5	middle-aged	no	excellent	yes
6	senior	yes	fair	no
7	senior	yes	excellent	yes

Information Gain (ID3/C4.5) Method...

Example

Q. Consider the following 14 days weather dataset with attributes outlook, temperature, humidity and wind. The outcome variable will be playing ball or not. Construct decision tree from the given dataset using ID3 algorithm and use the constructed decision tree to classify the new data X={Outlook=Sunny, Temperature=Cool, Humidity=Normal, Wind=Strong}.

Day	Outlook	Temperature	Humidity	Wind	Play ball
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

Information Gain (ID3/C4.5) Method...

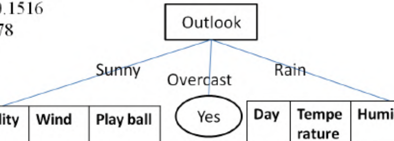
Example...

Total records = 14
Yes-> 9; No->5

So, Entropy(D)=0.94

Gain(D, Outlook) = 0.2464
Gain(D, Temperature) = 0.0289
Gain(D, Humidity) = 0.1516
Gain(D, Wind) = 0.0478

Thus Root is Outlook

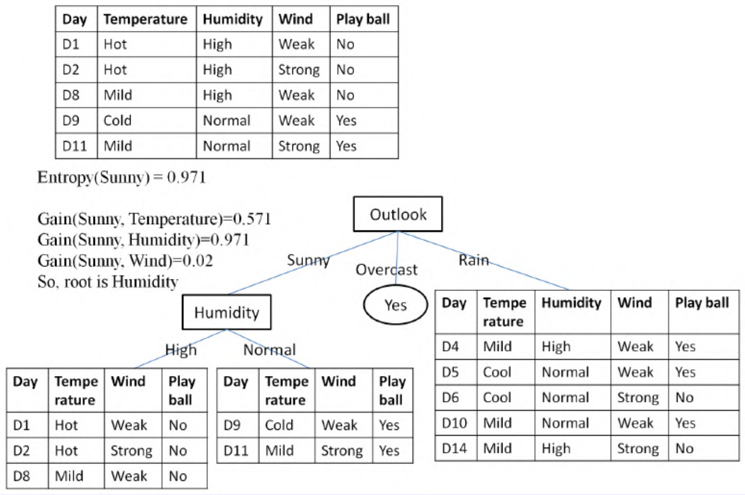


Day	Temperature	Humidity	Wind	Play ball
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cold	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

Day	Temperature	Humidity	Wind	Play ball
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
D10	Mild	Normal	Weak	Yes
D14	Mild	High	Strong	No

Information Gain (ID3/C4.5) Method...

Example...



Day	Temperature	Humidity	Wind	Play ball
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cold	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

Entropy(Sunny) = 0.971

Gain(Sunny, Temperature)=0.571
Gain(Sunny, Humidity)=0.971
Gain(Sunny, Wind)=0.02
So, root is Humidity

Day	Temperature	Wind	Play ball
D1	Hot	Weak	No
D2	Hot	Strong	No
D8	Mild	Weak	No

Day	Temperature	Wind	Play ball
D9	Cold	Weak	Yes
D11	Mild	Strong	Yes

Day	Temperature	Humidity	Wind	Play ball
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
D10	Mild	Normal	Weak	Yes
D14	Mild	High	Strong	No

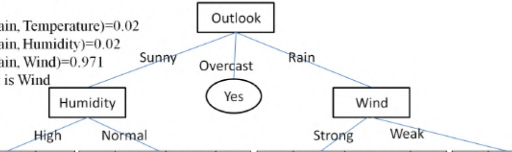
Information Gain (ID3/C4.5) Method...

Example...

Day	Temperature	Humidity	Wind	Play ball
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
D10	Mild	Normal	Weak	Yes
D14	Mild	High	Strong	No

Entropy(Rain)=0.971

Gain(Rain, Temperature)=0.02
Gain(Rain, Humidity)=0.02
Gain(Rain, Wind)=0.971
So, root is Wind



Da y	Tempe rature	Wind	Play ball
D1	Hot	Weak	No
D2	Hot	Strong	No
D8	Mild	Weak	No

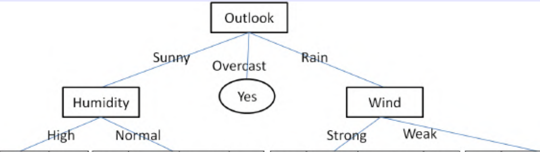
Day	Tempe rature	Wind	Play ball
D9	Cold	Weak	Yes
D11	Mild	Strong	Yes

Day	Tempe rature	Humidity	Play ball
D6	Cool	Normal	No
D14	Mild	High	No

Day	Tempe rature	Humidity	Play ball
D4	Mild	High	Yes
D5	Cool	Normal	Yes
D10	Mild	Normal	Yes

Information Gain (ID3/C4.5) Method...

Example...

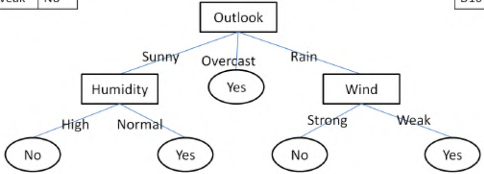


Day	Temperature	Wind	Play ball
D1	Hot	Weak	No
D2	Hot	Strong	No
D8	Mild	Weak	No

Day	Temperature	Wind	Play ball
D9	Cold	Weak	Yes
D11	Mild	Strong	Yes

Day	Temperature	Humidity	Play ball
D6	Cool	Normal	No
D14	Mild	High	No

Day	Temperature	Humidity	Play ball
D4	Mild	High	Yes
D5	Cool	Normal	Yes
D10	Mild	Normal	Yes



X={Outlook=Sunny, Temperature=Cool, Humidity=Normal, Wind=Strong}

Predicted class: Play ball=Yes

Information Gain (ID3/C4.5) Method...

Characteristics of ID3 Algorithm

- It uses greedy approach, that's why doesn't guarantee for optimal solution
- It can overfit to the training data
- It usually produces small trees, but not always
- It is harder to use on continuous data

Advantages

- cheap to construct
- extremely fast in classifying unknown records
- easy to interpret for small sized trees

Disadvantages

- greedy approach is often unable to find best tree
- each decision boundary involves only a single attribute

Naive Bayesian Classification

Naive Bayesian (or Naive Bayes) Classification

- Bayesian classifiers are statistical classifiers based on Bayes theorem (named after Thomas Bayes)
- They can predict class membership probabilities
- Naive Bayes classifier is a simple Bayesian classifier that exhibits Naive assumption. It has high accuracy and speed compared to decision tree
- It can solve problems involving both categorical and continuous valued attributes

Bayes Theorem

Prior Probability

- A prior probability is the probability that an observation will fall into a group before one collect the data
- Ex: John conducts a single coin toss. What is the a priori probability of landing a head?
  A priori probability of landing a head = 1 / 2 = 50%

Posterior (or conditional) Probability

- A posterior probability is the probability of an event occurring given that another event has already occurred
- Posterior probability of A occurring given that B occurred:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} = \boxed{\frac{P(A) \times P(B \mid A)}{P(B)}}$$

Bayes' Theorem

where, A, B -> events
P(B|A) -> probability of B occurring given A has happened
P(A) & P(B) -> the prior probability of event A & event B

Bayes Theorem...

Bayes Theorem...

- Given training data set D and data sample to be classified X, where class label is unknown. Let H be a hypothesis that X belongs to class C. Classification is to determine P(H|X), the probability that hypothesis holds given the observed data sample X
- $P(H|X) = \frac{P(X|H)*P(H)}{P(X)}$
  This predicts $X \in classC_i$, iff $P(C_i|X)$ is the highest among all the $P(C_k|X)$ for all classes k

Naive Bayes Classifier

Naive Bayes Classifier

- Given, D-> set of tuples; $X = < x_1, ...x_n$ where x_n is the value of attribute A_i; m-> number of classes
- Bayesian classifier predicts X belongs to class C_i iff $P(C_i|X) > P(C_j|X)$, for $1 \leq j \leq m \& j \neq i$
  or Maximize Posterior Hypothesis: $P(C_i|X) = \frac{P(X|C_i) * P(C_i)}{P(X)}$
- *Naive Assumption: (Class - Condition Independence): The effect of an attribute value on a particular class is independent of other attribute values*
  - Due to conditional independence of attributes, P(X)=1
  - $P(C_i)$: Probability that class i is occurring in the dataset $P(C_i) = \frac{|C_{i,D}|}{|D|}$ where $|C_{i,D}|$ is the number of tuple containing class i
  - $P(X|C_i) = P(x_1, ...x_n|C_i) = \sum_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i * ... * P(x_n|C_i))$
- As P(X) is 1. Hence, only maximize $P(X|C_i) * P(C_i)$

Naive Bayes Classifier...

Example

RID	Age	Income	Student	Credit_Rating	Class: Buys Computer
1	Youth	High	No	Fair	No
2	Youth	High	No	Excellent	No
3	middle_aged	High	No	Fair	Yes
4	senior	Medium	No	Fair	Yes
5	senior	Low	Yes	Fair	Yes
6	senior	Low	Yes	Excellent	No
7	middle_aged	Low	Yes	Excellent	Yes
8	Youth	Medium	No	Fair	No
9	Youth	Low	Yes	Fair	Yes
10	senior	Medium	Yes	Fair	Yes
11	Youth	Medium	Yes	Excellent	Yes
12	middle_aged	Medium	No	Excellent	Yes
13	middle_aged	High	Yes	Fair	Yes
14	senior	Medium	No	Excellent	No

For the given training dataset let the new customer X value as follows.
Find the probability that customer will buy computer or not
or
Find the query X belongs to which class?
X = (Age=Youth income = medium student = Yes credit_rating = Fair)

Naive Bayes Classifier...

Example...

Step-1: Find Class Prior Probability $P(C_i)$ for i=1,2

$$P(C_1) = P(\text{buys\_computer} = \text{Yes}) = \frac{9}{14} = 0.643$$

$$P(C_2) = P(\text{buys\_computer} = \text{No}) = \frac{5}{14} = 0.357$$

Step-2: Find Conditional Probability of the query X or tuple X given is the class label

Conditional Probability of X w.r.t. Class Yes

$$P(X|Yes) = \prod_{k=1}^{n} P(x_k|Yes)$$

$= P(x1: age=youth|Yes) \times$

$P(x2: income=medium|Yes) \times$

$P(x3: student=yes|Yes) \times$

$P(x4: credit\_rating=fair|Yes)$

$$= \frac{2}{9} \times \frac{4}{9} \times \frac{6}{9} \times \frac{6}{9} = 0.22 \times 0.44 \times 0.66 \times 0.66 = 0.044$$

Conditional Probability of X w.r.t. Class No

$$P(X|No) = \prod_{k=1}^{n} P(x_k|No)$$

$= P(x1: age=youth|No) \times$

$P(x2: income=medium|No) \times$

$P(x3: student=yes|No) \times$

$P(x4: credit\_rating=fair|No)$

$$= \frac{3}{5} \times \frac{2}{5} \times \frac{1}{5} \times \frac{2}{5} = 0.6 \times 0.4 \times 0.2 \times 0.4 = 0.019$$

Step-3: Find the class that maximizes $P(X|C_i) \times P(C_i)$

Max{ P(X|buys_Comp=Yes) × p(buys_computer = Yes),

P(X|buys_Comp=No) × p(buys_computer = No)

= {0.044× 0.643, 0.019 × 0.357}

= {0.028, 0.007} = 0.028 {Class "Yes" Maximizes the conditional probability of X}

Hence, the Naive Bayesian classifier predicts buys_comp is yes for given tuple X

Naive Bayes Classifier...

Example

Confident	Studied	Sick	Result
Yes	No	No	Fail
Yes	No	Yes	Pass
No	Yes	Yes	Fail
No	Yes	No	Pass
Yes	Yes	Yes	Pass

Q. Find out whether the object with attribute Confident=Yes, Studied=Yes and Sick=No will Fail or Pass using Bayesian classification.

X={Confident=Yes, Studied=Yes, Sick=No}
P(Result=Pass)=3/5=0.6
P(Result=Fail)=2/5=0.4

P(Confident=Yes | Result=Pass) = 2/3
P(Studied=Yes | Result=Pass)=2/3
P(Sick=No | Result=Pass)=1/3
So, P(X | Result=Pass)= 2/3 * 2/3 * 1/3 = 0.148

P(Confident=Yes | Result=Fail) = 1/2
P(Studied=Yes | Result=Fail)=1/2
P(Sick=No | Result=Fail)=1/2
So, P(X | Result=Fail)= 1/2 * 1/2 * 1/2 = 0.125

$$P(Result = Pass| X) = P(X| Result = Pass) * P(Resutlt = Pass) = 0.148 * 0.6 = 0.089$$

$$P(Result = Fail| X) = P(X| Result = Fail) * P(Resutlt = Fail) = 0.125 * 0.4 = 0.05$$

As 0.089 > 0.05, the instance with $(Confident = Yes, Studied = Yes, Sick = No)$ has result as ***'Pass'***.

Naive Bayes Classifier...

Laplacian Correction

- Naive Bayes prediction requires each conditional probability be non-zero. Otherwise, the predicted probability will be zero. This is called as problem of zero probability

- Ex: Suppose a dataset with 1000 tuples, income=low (0), income=medium (990) and income=high (10)
  - Prob(income=low)=0/1000
    Prob(income=medium)=990/1000
    Prob(income=high)=10/1000
- **Laplacian correction (or Laplacian estimator or Laplace smoothing)** is a smoothing technique that handles the problem of zero probability in Naive Bayes by adding 1 to each count
  - Prob(income=low)=1/1003
    Prob(income=medium)=991/1003
    Prob(income=high)=11/1003
- The corrected probability estimates are close to their uncorrected counterparts

Naive Bayes Classifier...

Advantages

- SImple, fast and easy to implement
- Highly scalable
- Handles continuous and discrete data
- Easily updatable if training dataset increases

Disadvantages

- Assumptions: class conditional independence, therefore loss of accuracy
- Practically, dependencies exist among variables

Uses

- Text classification
- Spam filtering
- Hybrid recommender

Rule Based Classification

Rule Based Classification

- A rule-based classifier uses a set of IF-THEN rules for classification. An IF-THEN rule is an expression of the form **IF condition THEN conclusion**
- Ex: *R1: IF age=youth AND student='yes' THEN buys_ computer='yes'*
  OR
  R1: (age=youth) $\wedge$ *(student=yes) => (buys_ computer=yes)*
- The 'IF' part of a rule is known as the antecedent or precondition. The 'THEN' part is the consequent. The rule's consequent contains a class prediction

Rule Based Classification...

Rule Coverage

- The % of records that satisfy the antecedent condition of a particular rule are generated by rule based classifier
- The rules may not be exhaustive, i.e. there are some records that are not covered by any of the rules
- The decision boundaries created by the rules either be linear or much more complex as many rules are triggered for same record
- A rule's coverage is the percentage of tuples that satisfy rule's antecedent $coverage(R) = \frac{n_{covers}}{|D|}$
  where, n_{covers} -> Number of tuples covered by R
  |D| -> number of the tuples present in training dataset D

Rule Accuracy

- A rule's accuracy is the percentage of tuples that are correctly classified by the rule, i.e. satisfy both the antecedent and consequent of a rule $accuracy(R) = \frac{n_{correct}}{n_{covers}}$
  where, $n_{correct}$ -> Number of tuples correctly classified by R

Rule Based Classification...

Rule Coverage & Rule Accuracy...

- Find coverage and accuracy of the rule *R1: (age=youth)* $\wedge$ *(student=yes) => (buys_ computer=yes)*
- |D| = 14
- Number of tuples satisfying the condition or rule (n_{covers})=2. So, coverage(R1)=2/14=14.28 %
- It can correctly classify both tuples ($n_{correct}$)=2. So, accuracy(R1)=2/2=100%

RID	Age	Income	Student	Credit_Rating	Class: Buys Computer
1	Youth	High	No	Fair	No
2	Youth	High	No	Excellent	No
3	middle_aged	High	No	Fair	Yes
4	senior	Medium	No	Fair	Yes
5	senior	Low	Yes	Fair	Yes
6	senior	Low	Yes	Excellent	No
7	middle_aged	Low	Yes	Excellent	Yes
8	Youth	Medium	No	Fair	No
9	Youth	Low	Yes	Fair	Yes
10	senior	Medium	Yes	Fair	Yes
11	Youth	Medium	Yes	Excellent	Yes
12	middle_aged	Medium	No	Excellent	Yes
13	middle_aged	High	Yes	Fair	Yes
14	senior	Medium	No	Excellent	No

Building Classification Rules

Indirect Method

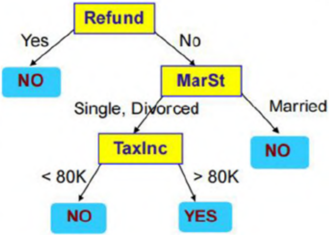
- It builds the model by extracting rules from other classification models, such as decision tree
- We can extract rules from given decision tree by using following steps:
  - One rule is created for each path from the root to a leaf
  - Each attribute value pair along a path forms a conjuction which is used as rule antecedent
  - The leaf holds the class prediction which is used as rule consequent

•(Refund=Yes)=> NO

•(Refund=No)∧(Marital Status=Married)=> NO

•(Refund=No)∧(Marital Status={Single, Divorced)∧(TaxIncome<80K)=>NO

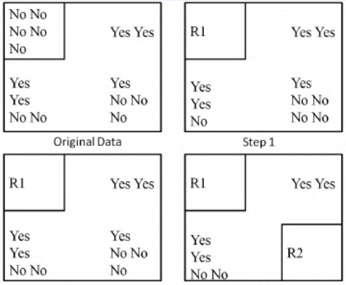
•(Refund=No)∧(Marital Status={Single, Divorced)∧(TaxIncome>80K)=>YES



Building Classification Rules...

Direct Method

- A model build by extracting rules directly from given training dataset. Ex: Sequential Covering Method
- Here, rules are learned once and that tuple from which rule is learned, is removed and learning another rule is done from next tuple and so on until all the remaining tuples are eliminated
- **Input**: D, a dataset class labeled tuples; Attribute values, set of all attributes and their possible values
- **Output**: A set of IF-THEN rules



Building Classification Rules...

Advantages

- As highly expressive as decision trees
- Easy to interpret
- Easy to generate
- Can classify new instances rapidly
- Performance comparable to decision trees

Disadvantages

- Harder to handle missing values in test set
- If rule set is large, then it is complex to apply rule for classification
- For large training set, large number of rules generated these require large amount of memory
- During rule generation, extra computation needed to simplify and prune the rules

KNN Classifier

Lazy Learner

- Lazy learner approach stores the received training tuples and wait to the last minute of receiving the new/test tuple. Only when it receives the test tuple, it constructs the generalized classification model based on similarity among test tuple and training tuples
- Less work (only store) when training tuple is received and more work when making a classification
- More expensive, but support incremental learning

Distance Measurement of Numeric Attributes

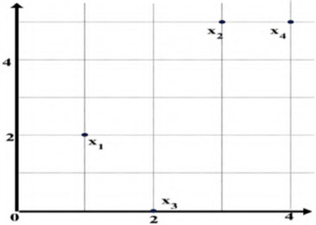
Distance Measurement of Numeric Attributes

Commonly used distance measures for computing dissimilarity are:

- **Euclidean Distance**
  $d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
  where,
  n->number of dimensions (attributes)
  x_i and y_i are the i^{th} attributes of data objects x and y
- **Squared Euclidean Distance**
  $d(x, y) = \sum_{i=1}^{n}(x_i - y_i)^2$
- **Manhattan Distance**
  $d(x, y) = \sum_{i=1}^{n}|x_i - y_i|$
- **Minkowski Distance**: is a generalization of the Euclidean and Manhattan distances. It is defined as
  $(\sum_{i=1}^{n}|x_i - y_i|^p)^{\frac{1}{p}}$
  When p=1 -> it is Manhattan
  When p=2 -> it is Euclidean

Distance Measurement of Numeric Attributes...

Distance Measurement of Numeric Attributes...



Point	Attribute1	Attribute2
x1	1	2
x2	3	5
x3	2	0
x4	4	5

Dissimilarity Matrix (Euclidean Distance)

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

	x1	x2	x3	x4
x1	0			
x2	3.61	0		
x3	2.24	5.1	0	
x4	4.24	1	5.39	0

Dissimilarity Matrix (Manhattan Distance)

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

	x1	x2	x3	x4
x1	0			
x2	5	0		
x3	3	6	0	
x4	6	1	7	0

Distance Measurement of Numeric Attributes...

Distance Measurement of Numeric Attributes...

Point	Attribute1	Attribute2
x1	1	2
x2	3	5
x3	2	0
x4	4	5

Minkowski Distance $$\left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}}$$

Where p-> real number >=1

Minkowski Distance with p value 1

Manhattan Distance $$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

	x1	x2	x3	x4
x1	0			
x2	5	0		
x3	3	6	0	
x4	6	1	7	0

Minkowski Distance with p value 2

Euclidean Distance $$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

	x1	x2	x3	x4
x1	0			
x2	3.61	0		
x3	2.24	5.1	0	
x4	4.24	1	5.39	0

KNN (K-Nearest Neighbor)Algorithm

KNN (K-Nearest Neighbor)Algorithm

- It does not require any assumptions about the underlying data distribution
- It can handle both numerical and categorical data, making it a flexible choice for various types of datasets in classification and regression tasks
- It is a non-parametric method that makes predictions based on the similarity of data points in a given dataset. K-NN is less sensitive to outliers compared to other algorithms
- The K-NN algorithm works by finding the K nearest neighbors to a given data point based on a distance metric, such as Euclidean distance
- It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset

KNN (K-Nearest Neighbor)Algorithm...

Algorithm

1. **Select K**: Choose the number of neighbors to consider
2. **Calculate distance**: Calculate the distance between the new data point and all the other data points
3. **Find the K nearest neighbors**: Rank the data points by distance and select the K closest neighbors
4. **Assign the new data point to a class**: Count the number of data points in each category and assign the new data point to the majority class

- The value of k is very crucial in the KNN algorithm to define the number of neighbors in the algorithm. The value of k should be chosen based on the input data
- If the input data has more outliers or noise, a higher value of k would be better. A low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model
- It is recommended to choose an odd value for k to avoid ties in classification. Ideally K=5

KNN (K-Nearest Neighbor)Algorithm...

Example

Perform KNN classification Algorithm on following dataset and Predict the class for [math=6, Computer=8], where K=3

Math	Computer	Result
4	3	F
6	7	P
7	8	P
5	5	F
8	8	P

Step-1 : Calculate the Euclidian distance between new data point **[math=6, Computer=8]** and all the other data points.

D1->5.38
D2->1
D3->1
D4->3.16
D5->2

Here, our K is 3. So, we choose the 3 nearest values, i.e. D2, D3 and D5

Math	Computer	Result
6	7	P
7	8	P
8	8	P

For our new data Math=6 and Computer=8, the result will be 'P' because all the 3 nearest data result are 'P'

If two are 'P' and one is 'F', the result will be 'P' because maximum is 'P'

KNN (K-Nearest Neighbor)Algorithm...

Example

Given data from the questionnaires survey with two attributes (X1: Acid durability and X2: Strength) to classify whether a special paper tissue is good or bad. Classify the new paper tissue with features X1=3 and X2=7

X1: Acid durability	X2: Strength	Class
7	7	Bad
7	4	Bad
3	4	Good
1	4	Good

- Number of tuples is even, so consider K as odd. Let K=3
- Compute distance between input sample and training samples using Euclidean distance
- D1->4; D2->5, D3->3, D4->3.6
- Choose 3 nearest values and rank them

X1: Acid durability	X2: Strength	Class	Rank
7	7	Bad	3
3	4	Good	1
1	4	Good	2

There are 2 'Good' and 1 'Bad'. So, the new paper tissue will be classified as 'Good' due to simple majority

KNN (K-Nearest Neighbor)Algorithm...

Advantages

- No training period
- As there is no training period, any new data can be added seamlessly without impacting accuracy of the result
- Easy to implement
- Best used where probability is unknown
- Useful for non-linear data as no assumption is required

Disadvantages

- Computationally expensive and more space complexity
- Output depends on choosen K value whic can reduce accuracy for some K value
- Doesn't work well with large data set and high dimension
- Sensitive to noisy, missing and outlier data
- Need normlization

Ref: J. Han, M. Kamber and J. Pei, "Data Mining: Concepts and Techniques", Morgan Kaufmann, 3rd edition