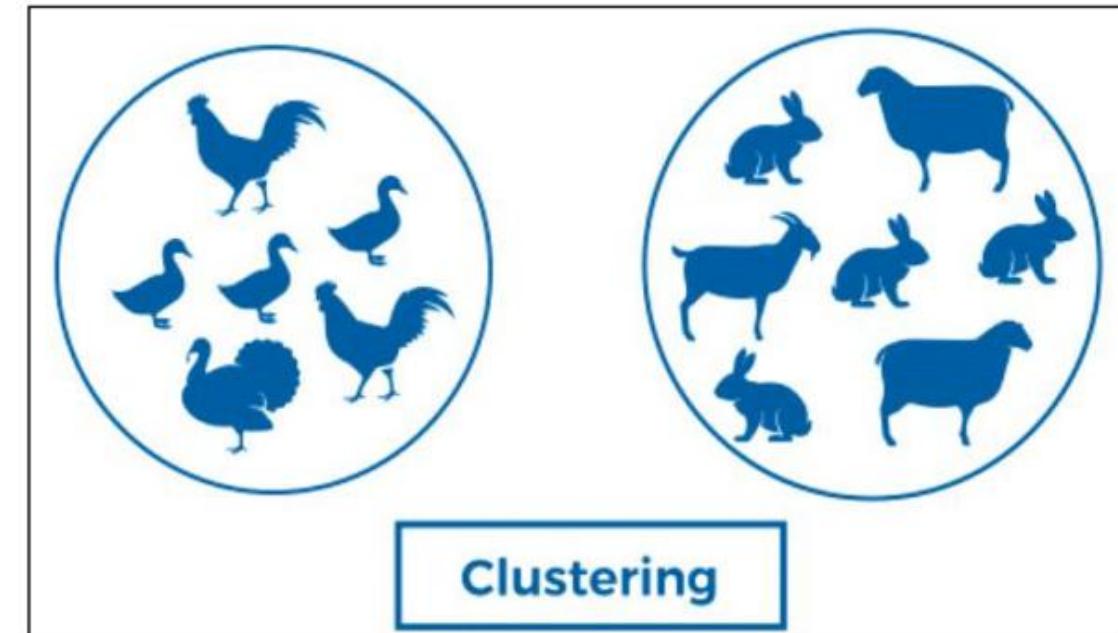
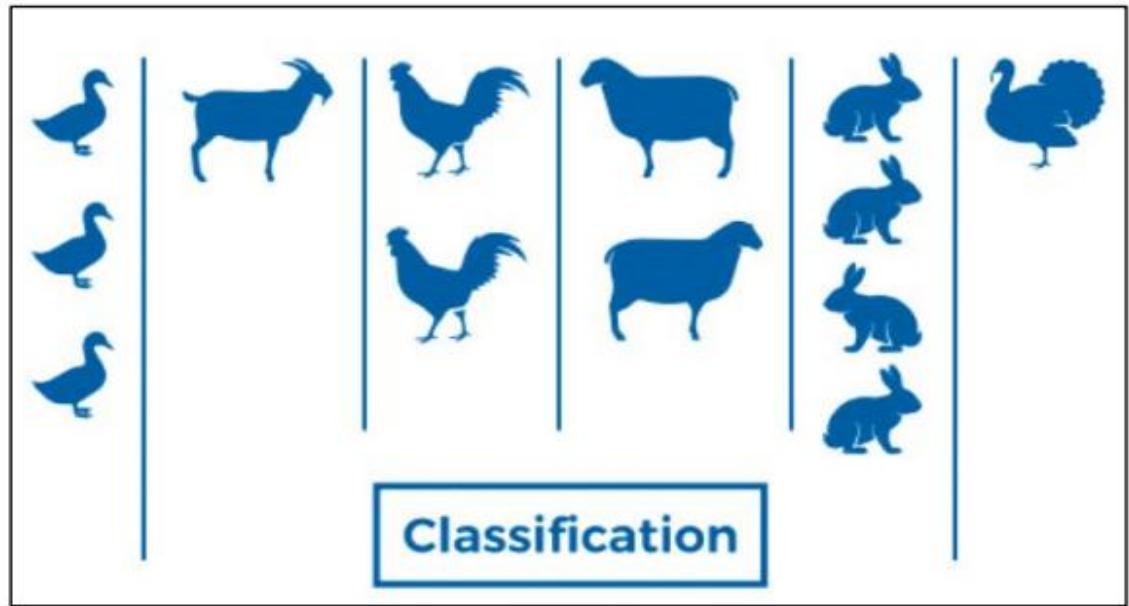
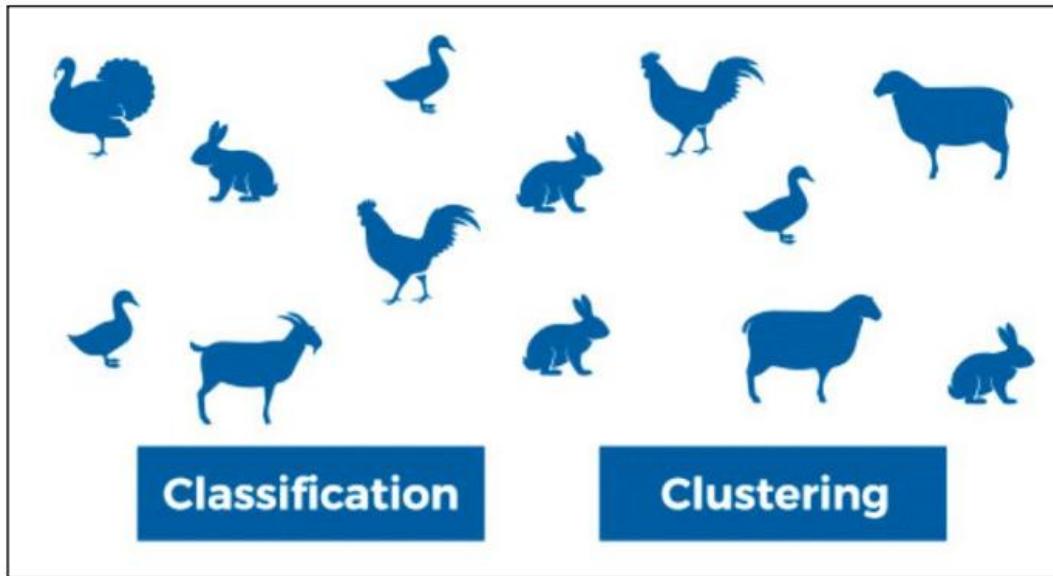


# Data Mining: Classification

# Classification and Clustering

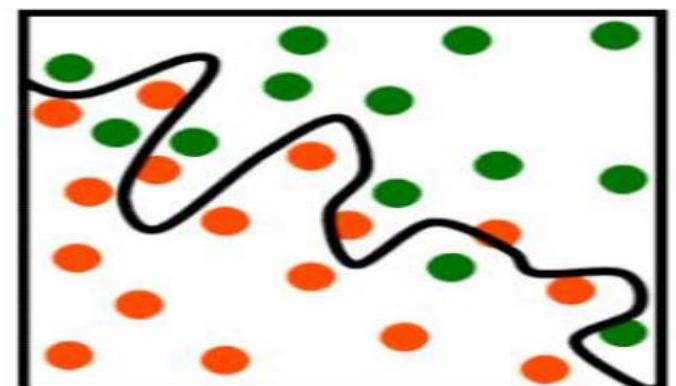
- Classification and clustering are two methods of pattern identification with some similarity and disimilarity.
- **Classification** uses predefined classes in which objects are assigned.
- **Clustering** identifies similarities between objects, which it groups according to those characteristics in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

# Classification and Clustering: Example



# Classification

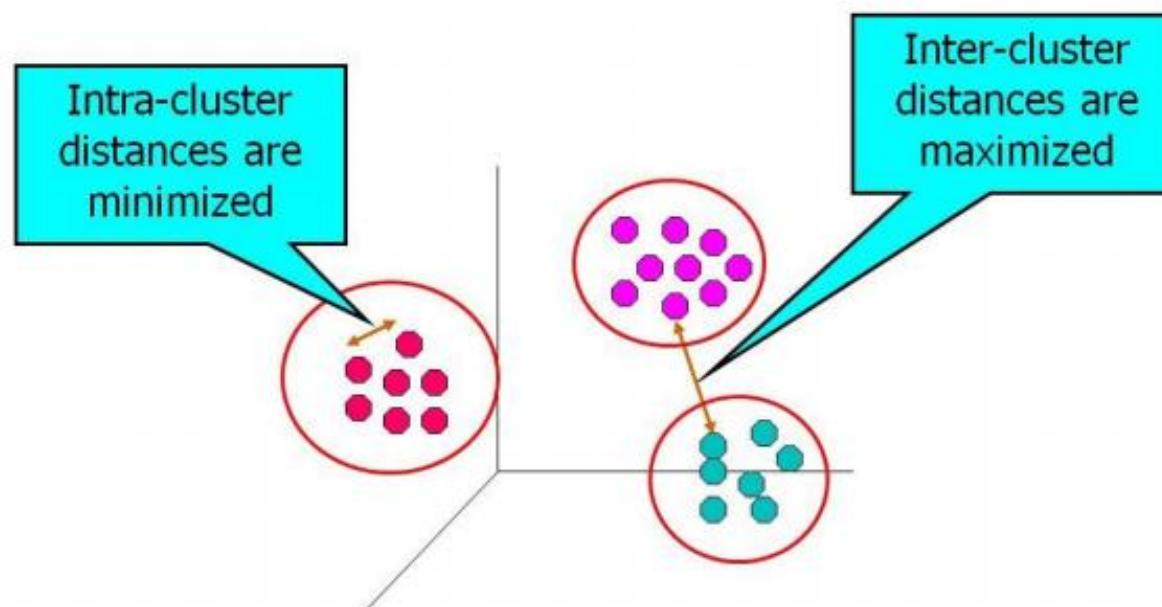
- Classification is the Data mining technique used to predict group membership for data instances.
  - There are two ways to assign a new value to a given class
    - Crispy Classification
    - Probabilistic Classification
- **Crispy Classification:** Given an input, the classifier exactly returns its 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
- Give only data  $> 90\%$
  - Assign the object to the class with the highest probability
  - Assign the object to the class only if its probability  $> 40\%$



For example, in a banking application, the customer who applies for a loan may be classified as a safe and risky according to his/her age and salary.

# Clustering

- Clustering 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 inter-cluster distance.

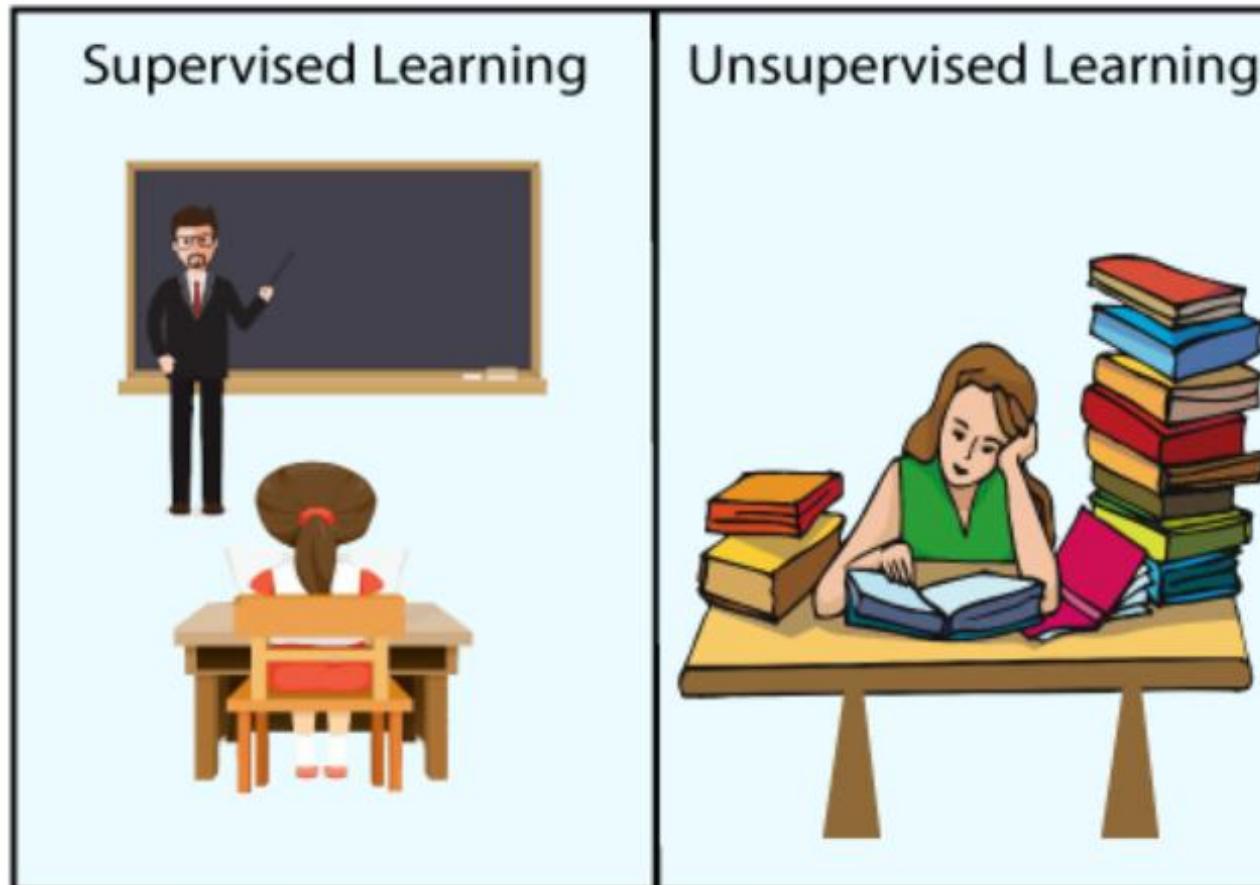


# Classification Vs Clustering

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 and Unsupervised learning are the two techniques of ML. Both are used in different scenarios and with different dataset.



# Supervised vs. Unsupervised Learning

## Supervised learning (classification) –

- In Supervised learning models are trained using labeled data.
- In supervised learning, models need to **find the mapping function** to map the input variable (X) with the output variable (Y).

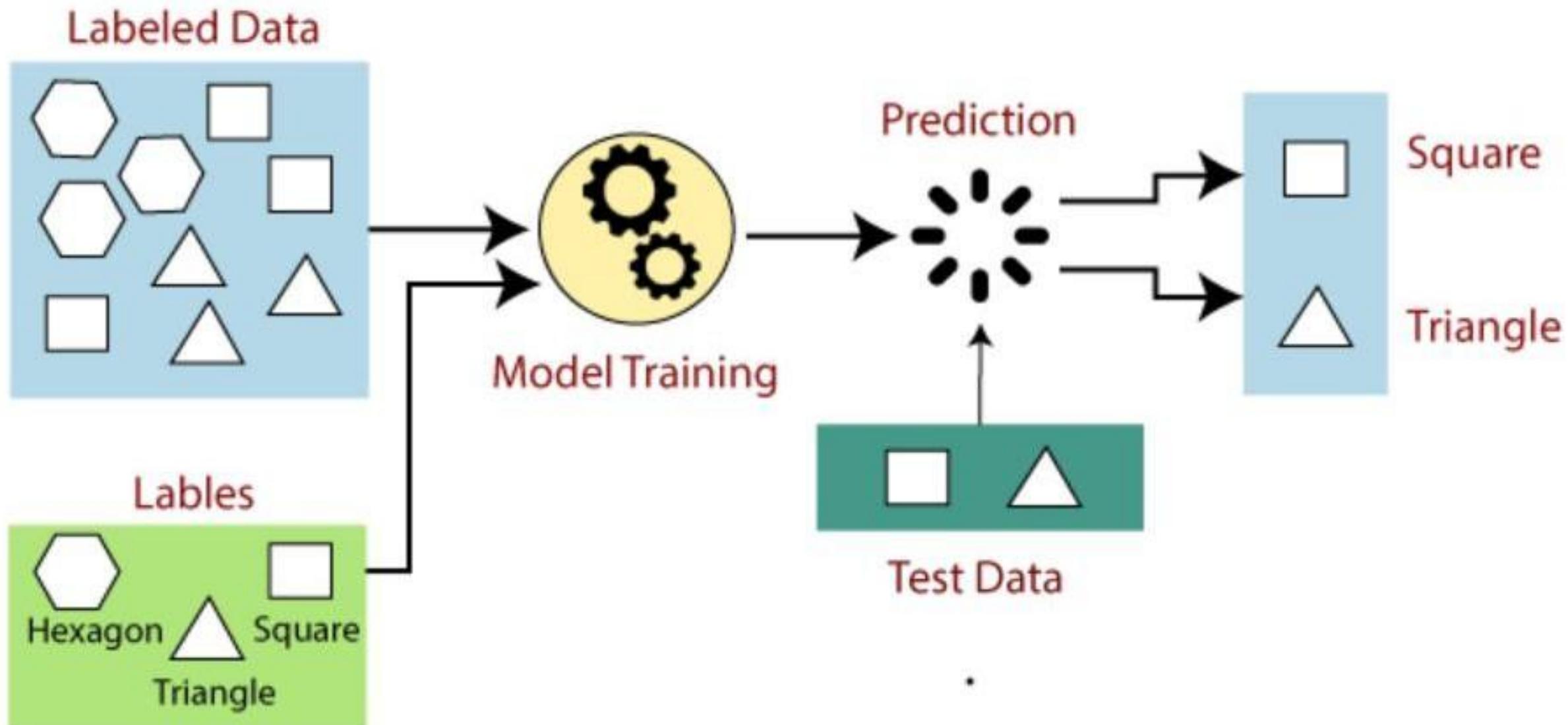
$$Y = f(X)$$

- Supervised learning **needs supervision to train the model**, which is similar to as a student learns things in the presence of a teacher.
- Supervised learning can be used for two types of problems: Classification and Regression.

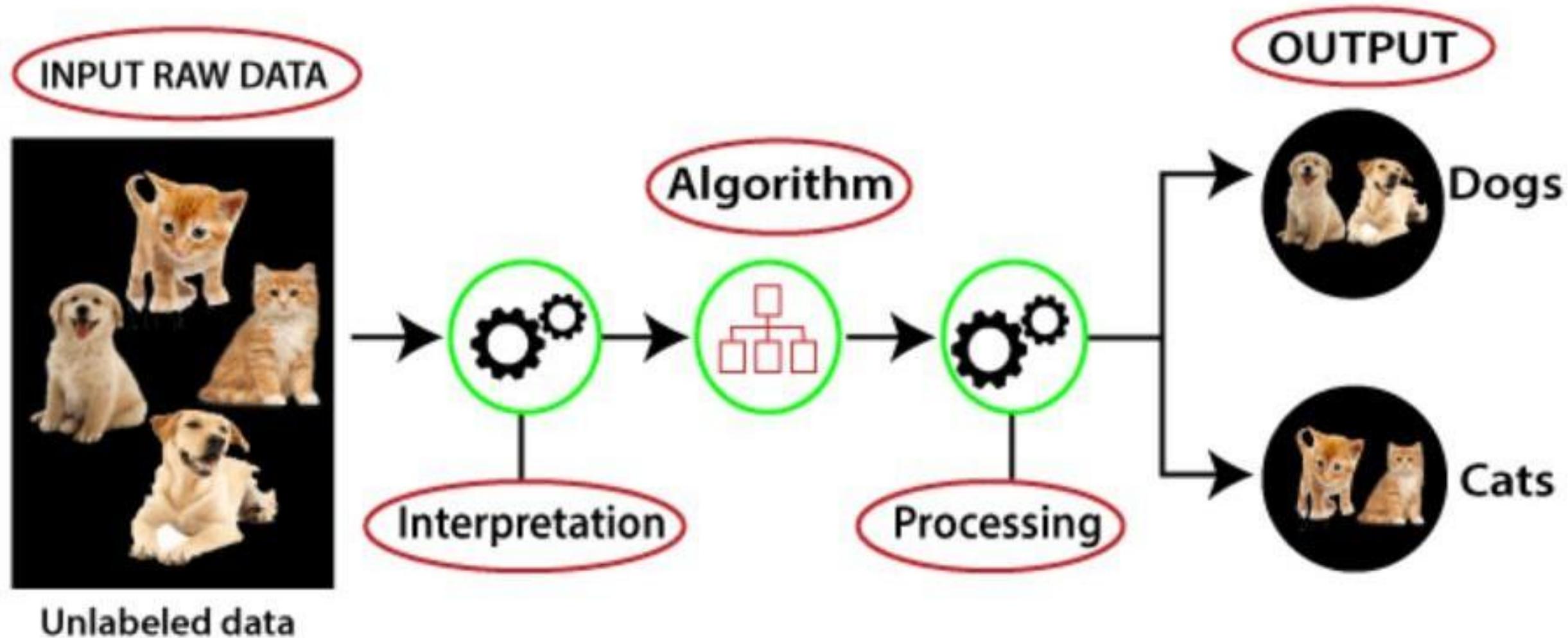
## Unsupervised learning (clustering) –

- 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
- Unsupervised learning can be used for two types of problems: Clustering and Association.

# How Supervised Learning Works?



# How Un-supervised Learning Works?



# Trainning Dataset Vs. Test Dataset

- The **training data** is the biggest subset ( $\geq 60\%$ ) 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 and Prediction

What is classification and What is prediction?

Issues regarding classification and prediction

# What is classification and What is prediction?

## Classification:

Classification is to identify the category or the class label of a new observation.

First a set of training data (input data with their class label) is given to the algorithm, the algorithm derives a model or classifier (that can be decision tree, mathematical formula, or a neural network). When a new unlabeled data is given to the classifier model, it should find the class to which it belongs.

## Prediction:

Prediction is to compute/predict the numeric output value of a new observation.

Same as classification, a set of training data (input data with their numeric output value) is given to the algorithm, the algorithm derives a model or predictor (that can be regression, K-nearest neighbor). When a new unlabeled data is given to the predictor model, it should find the continuous numeric output.

We can think of **prediction** as predicting the correct treatment for a particular disease for an individual person whereas the grouping of patients based on their medical records can be considered **classification**.

# Classification Learning: Definition

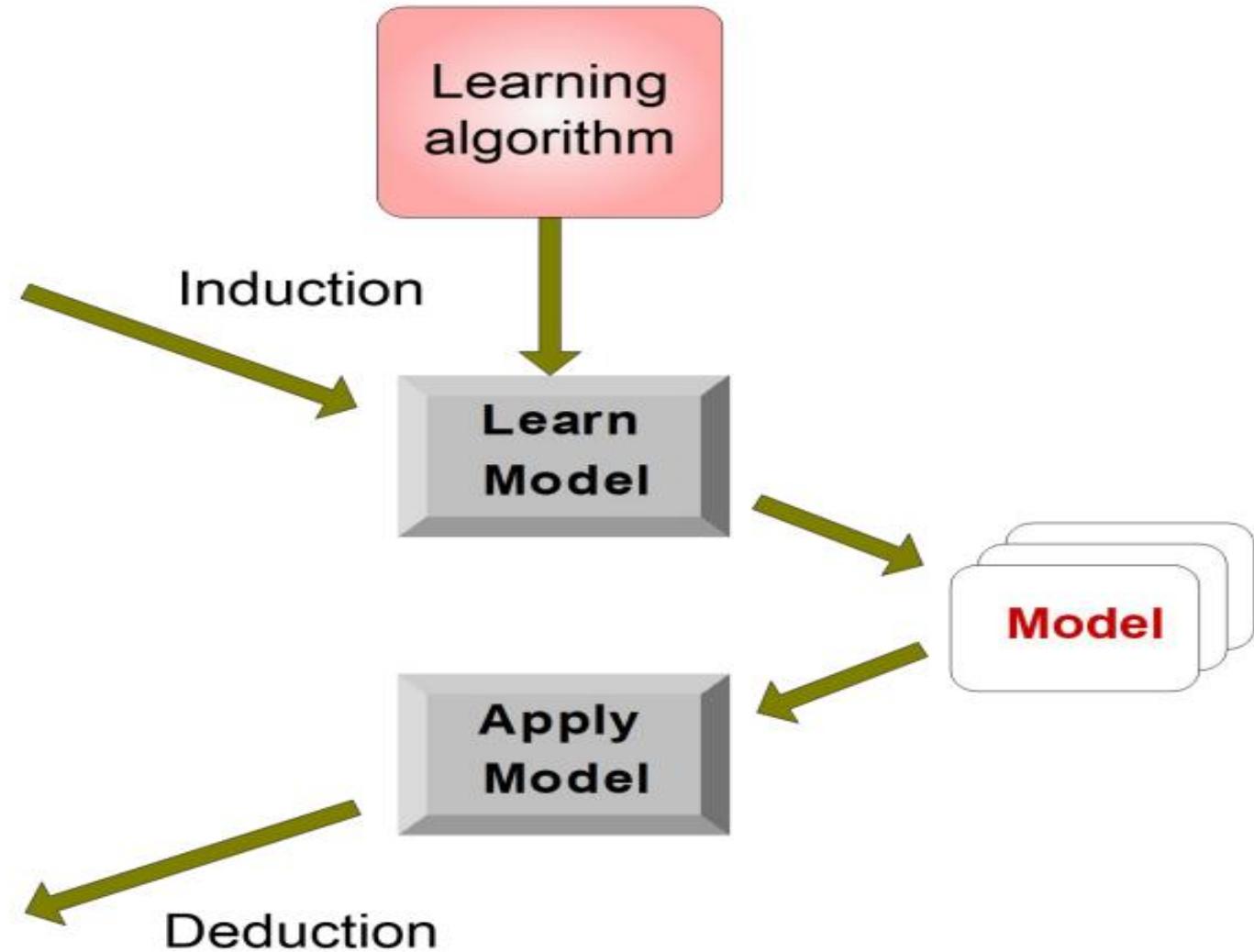
- Given a collection of records (**training set**)
  - Each record contains a set of attributes, one of the attributes is the **class**
- Find a model for the **class attribute as a function** of the values of the other attributes
- Use training set to construct the model and use test set to estimate the accuracy of the model
- **Goal:** previously unseen records should be assigned a class as accurately as possible

# Illustrating Classification Learning

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	?



# Classification - A Two-Step Process

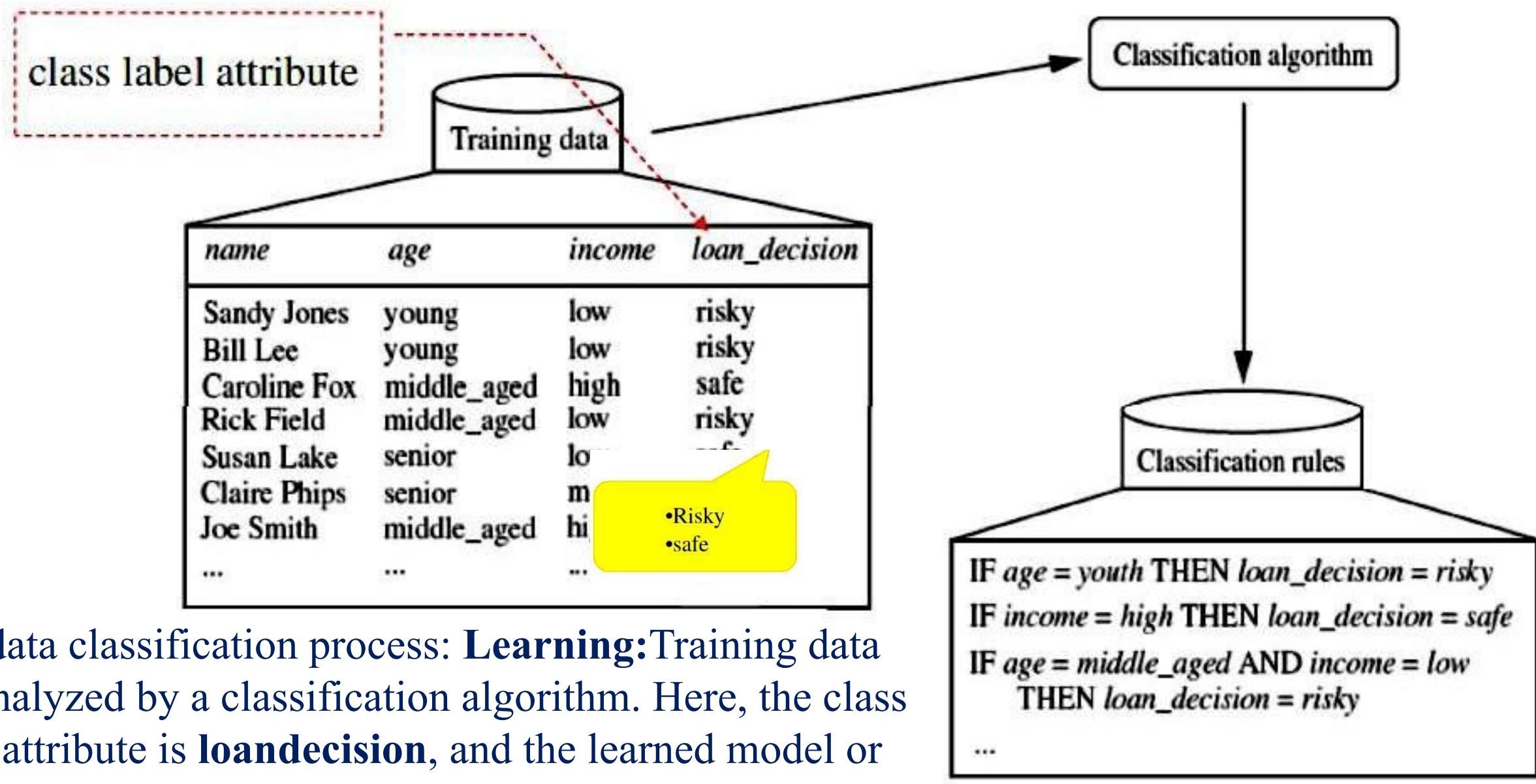
**Model construction:** describing a set of predetermined classes

- 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

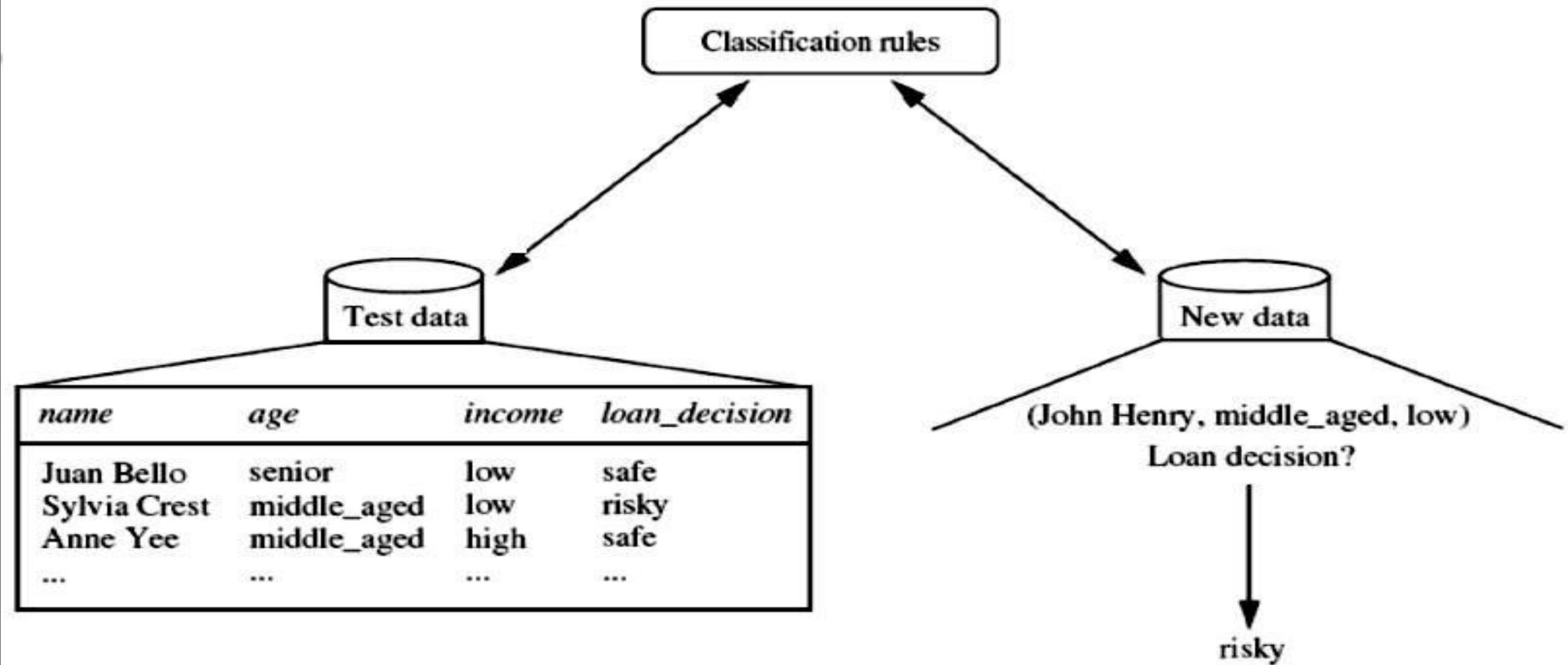
**Model usage:** for classifying future or unknown objects

- Using Classifier for Classification estimate accuracy of the 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

# Classification - A Two-Step Process: Model Construction



# Classification - A Two-Step Process: Model Usage



**Classification:** Test data are used to estimate the accuracy of the classification rules. If the accuracy is considered acceptable, the rules can be applied to the classification of new data tuples.

# Issues regarding classification and prediction:

## (1) Data Preparation

- **Data cleaning**
  - Preprocess 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

# Issues regarding classification and prediction: (2) Evaluating Classification Methods

- Predictive accuracy
- Speed and scalability
  - ✓ time to construct the model
  - ✓ time to use the model
- Robustness
  - ✓ handling noise and missing values
- Scalability
  - ✓ efficiency in disk-resident databases
- Interpretability
  - ✓ understanding and insight provided by the model
- Goodness of rules
  - ✓ decision tree size
  - ✓ compactness of classification rules

# Classification Technique

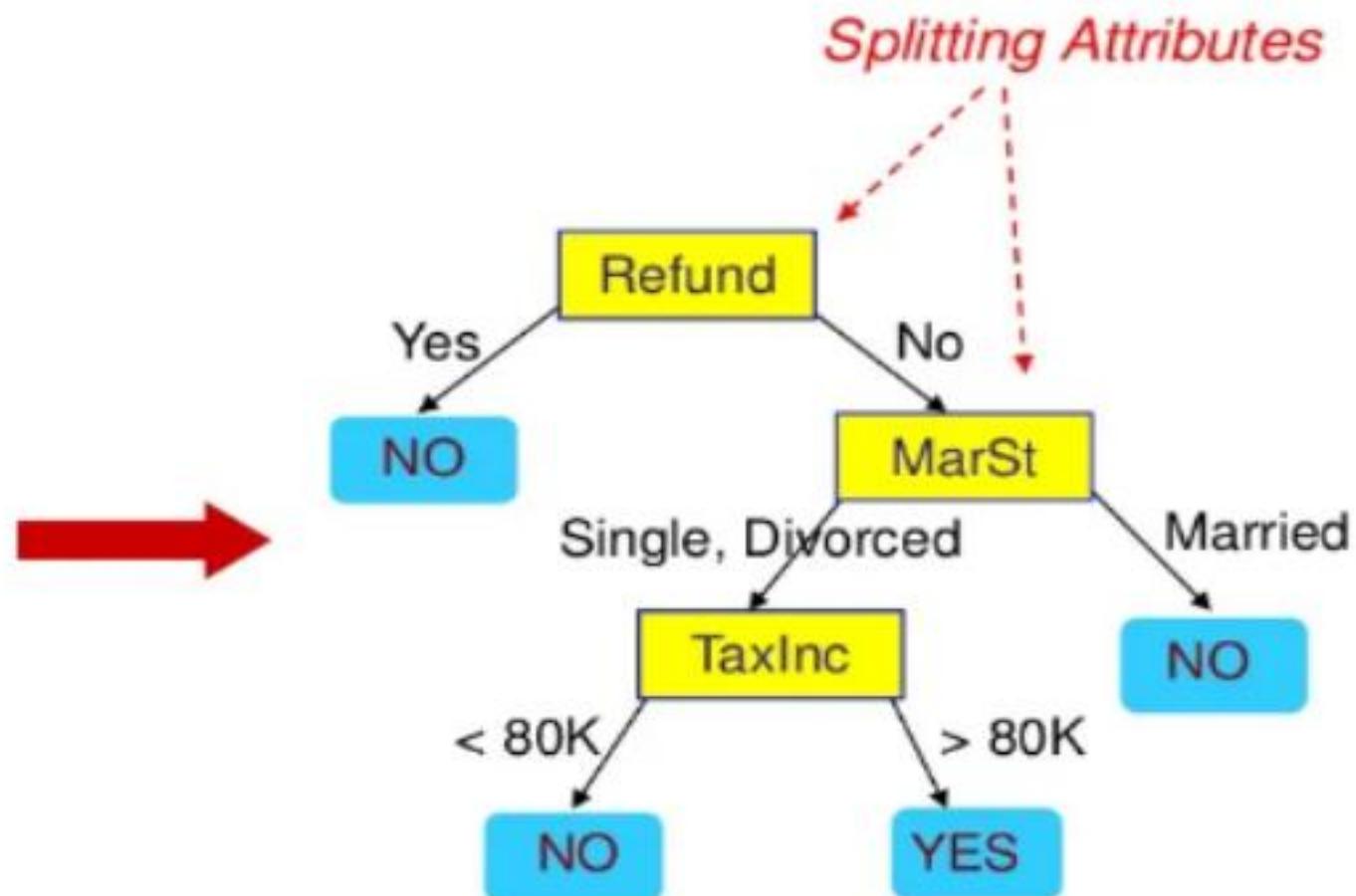
- Decision Trees
- Bayesian Classifiers
- Neural Networks
- K-Nearest Neighbour
- Support Vector Machines
- Linear Regression
- Logistic Regression
- Fuzzy Algorithm
- Genetic Algorithm

# 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

# Decision Tree for Cheat or not Cheat

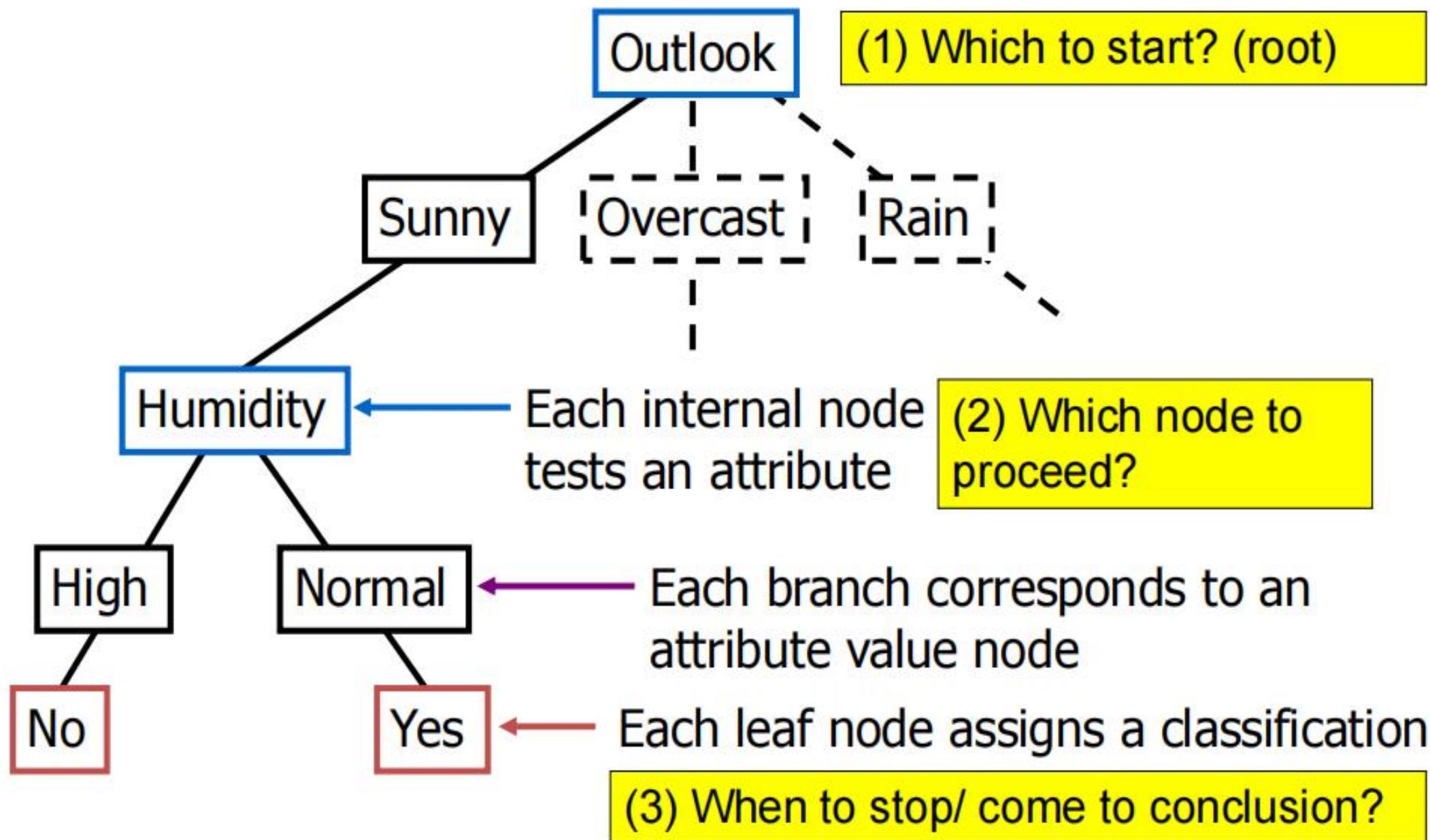
Tid	Refund	Marital Status	Taxable Income	Cheat	categorical	categorical	continuous	class
1	Yes	Single	125K	No				
2	No	Married	100K	No				
3	No	Single	70K	No				
4	Yes	Married	120K	No				
5	No	Divorced	95K	Yes				
6	No	Married	60K	No				
7	Yes	Divorced	220K	No				
8	No	Single	85K	Yes				
9	No	Married	75K	No				
10	No	Single	90K	Yes				



Training Data

Model: Decision Tree

# Decision Tree for PlayTennis

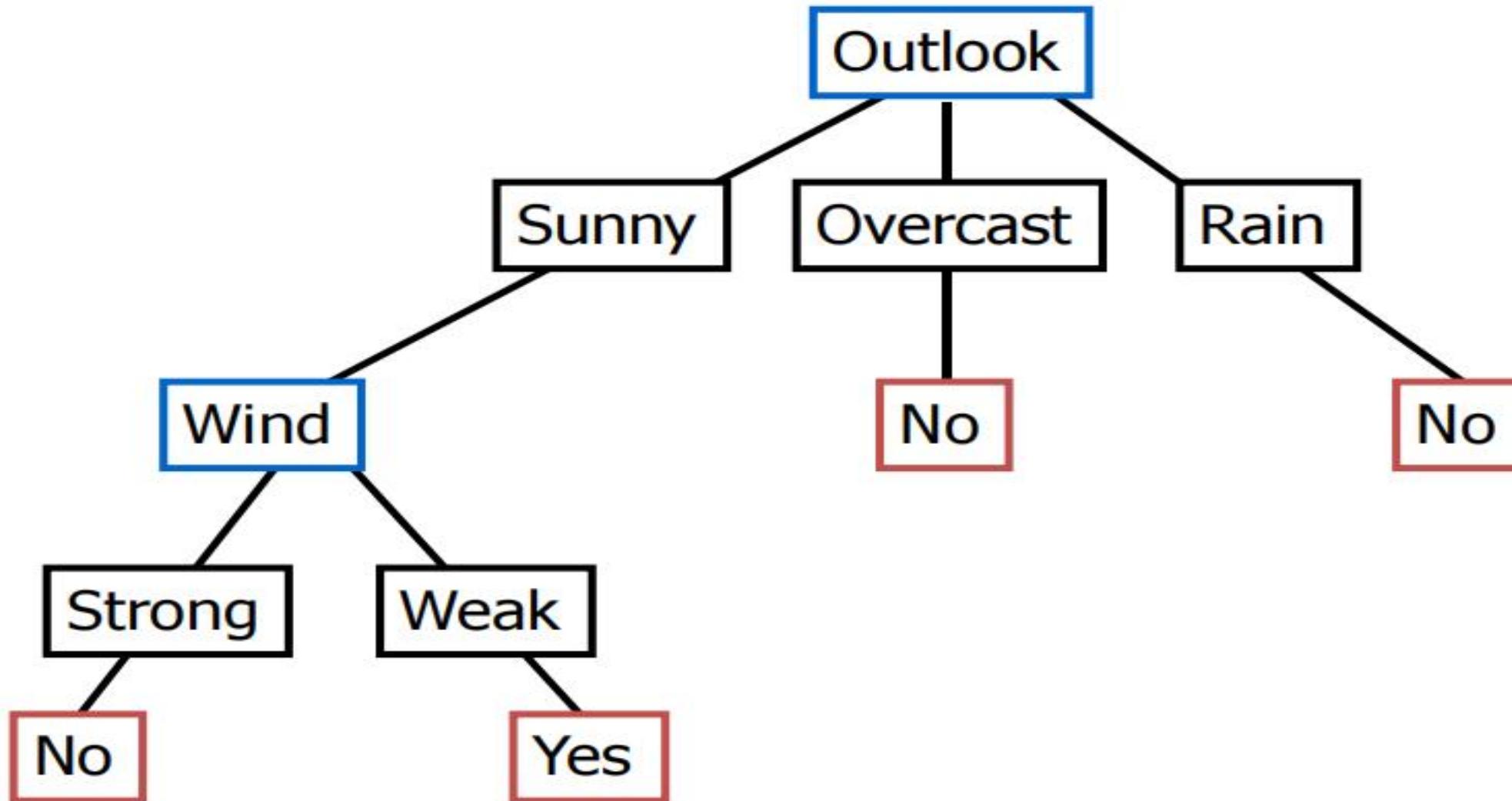


# Play-Tennis Data

DAY	OUTLOOK	TEMP	HUMIDITY	WIND	PLAY
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

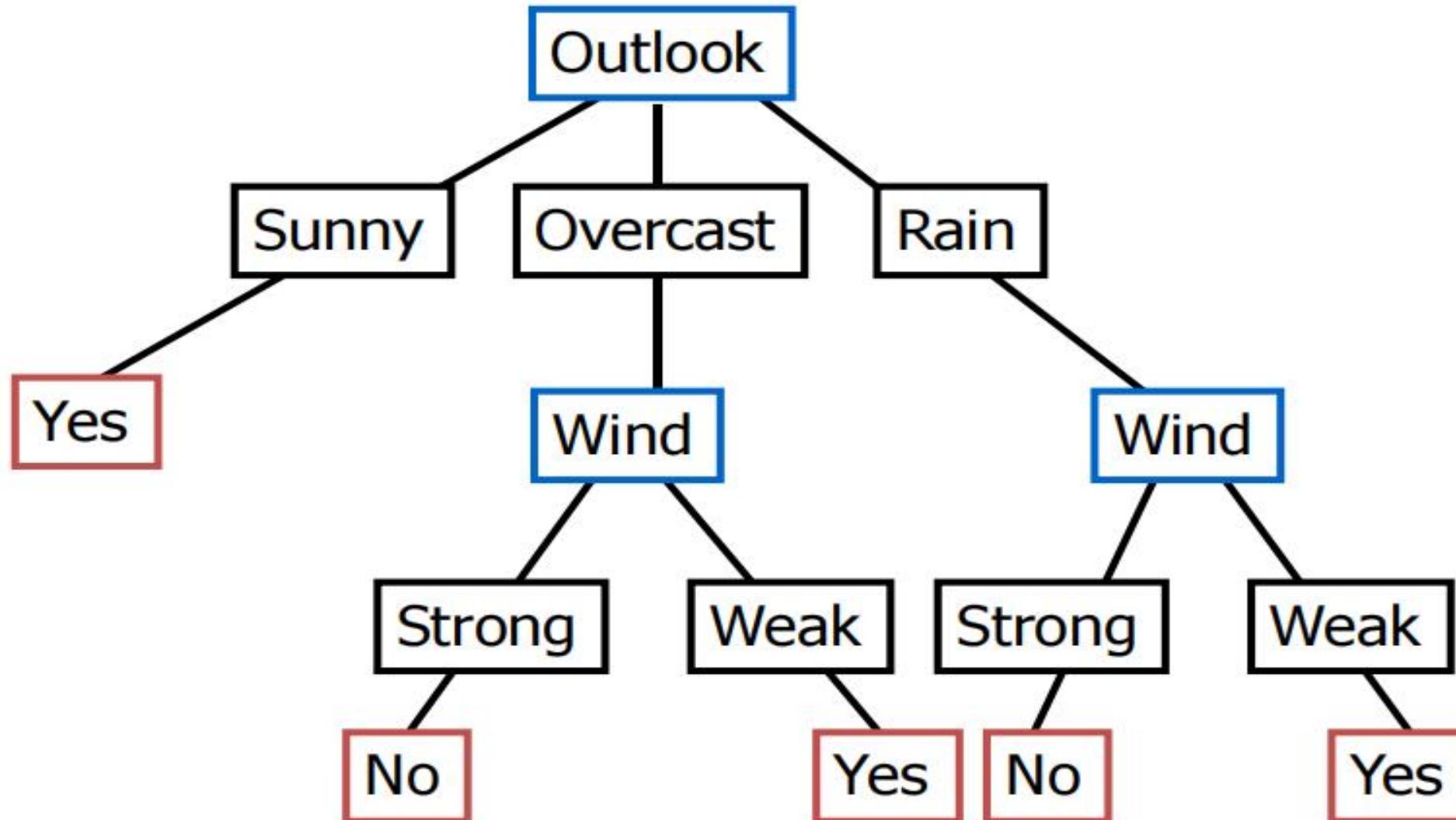
# Decision Tree for Conjunction

$\text{Outlook} = \text{Sunny} \wedge \text{Wind} = \text{Weak}$



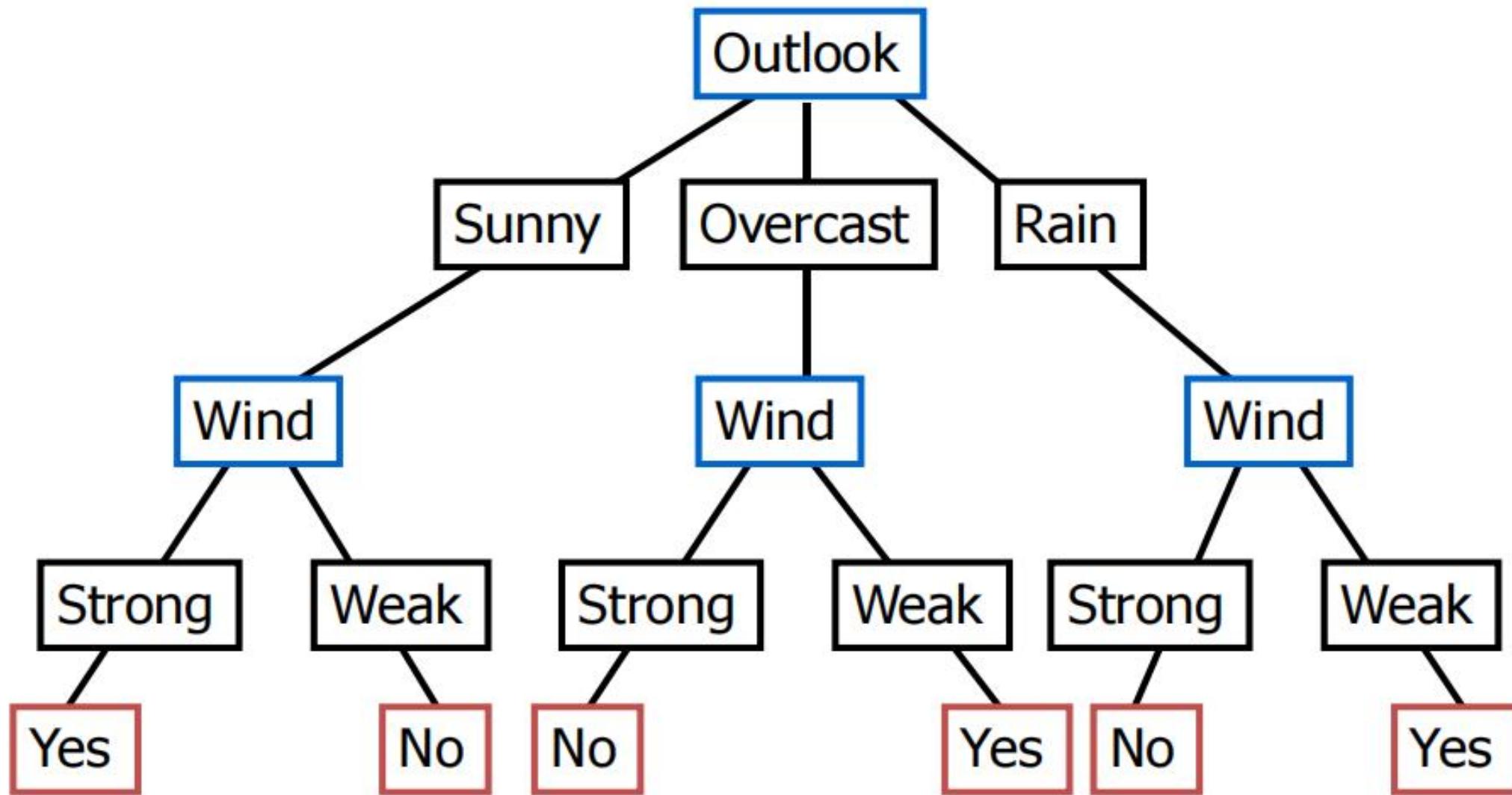
# Decision Tree for Disjunction

$\text{Outlook} = \text{Sunny} \vee \text{Wind} = \text{Weak}$

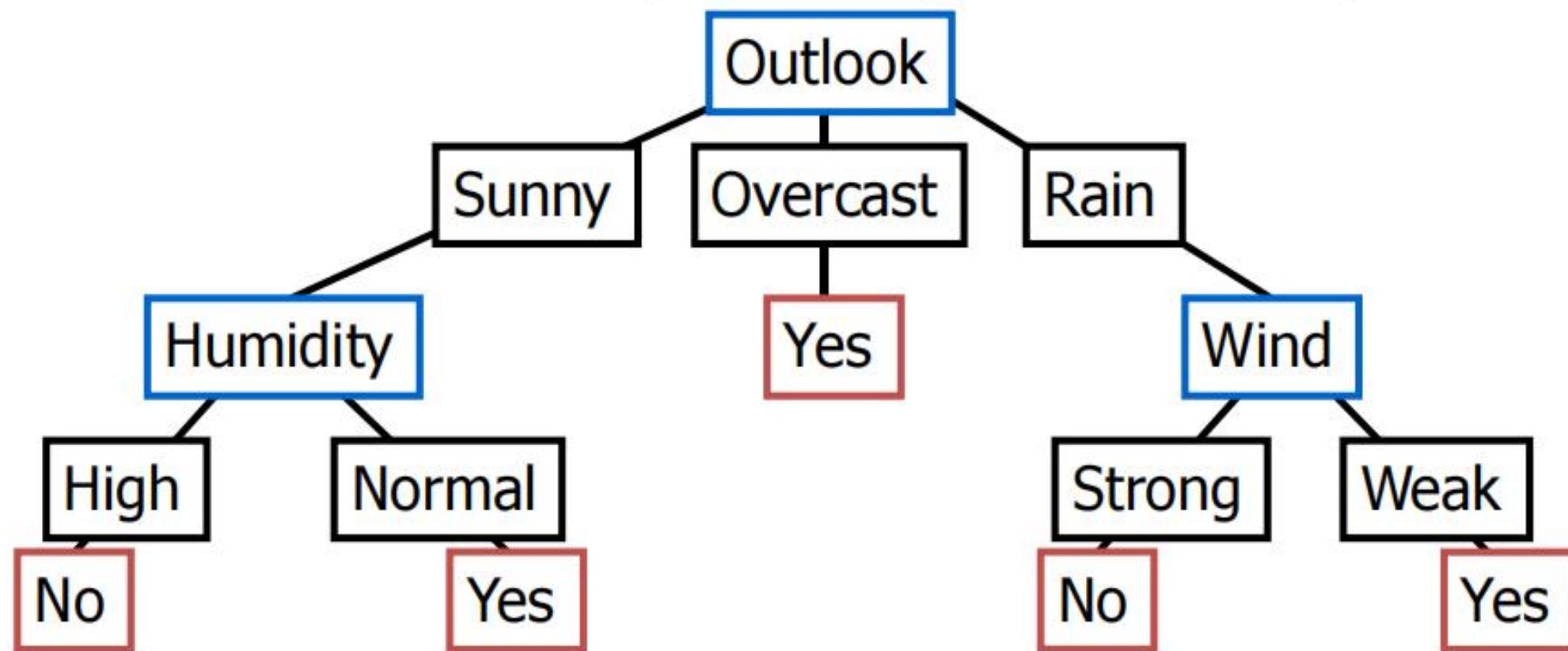


# Decision Tree for XOR

Outlook=Sunny XOR Wind=Weak



# Decision Tree for Disjunction of Conjunction



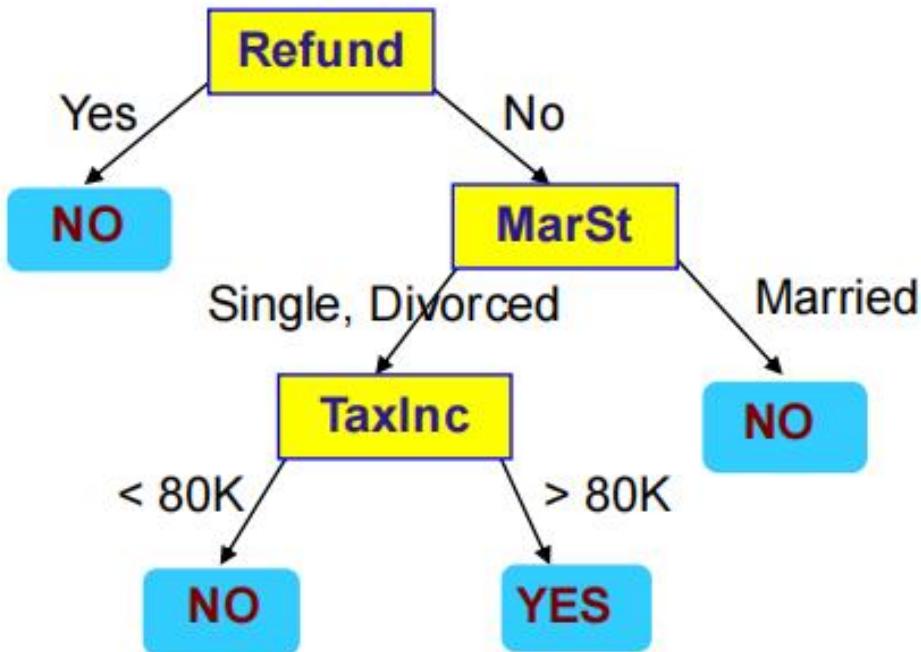
$(\text{Outlook}=\text{Sunny} \wedge \text{Humidity}=\text{Normal})$

∨  $(\text{Outlook}=\text{Overcast})$

∨  $(\text{Outlook}=\text{Rain} \wedge \text{Wind}=\text{Weak})$

# Example of a Decision Tree

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

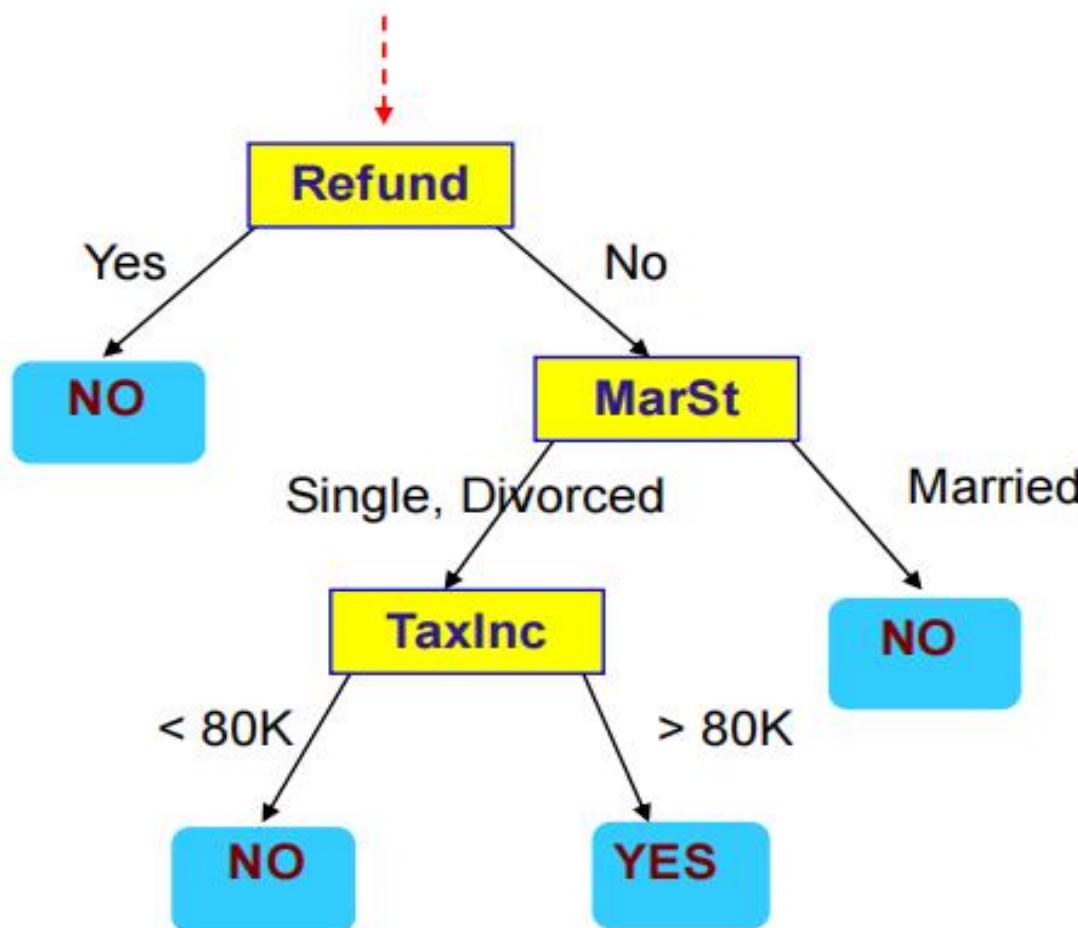


Training Data

Model: Decision Tree

# Apply Model to Test Data

Start at the root of tree



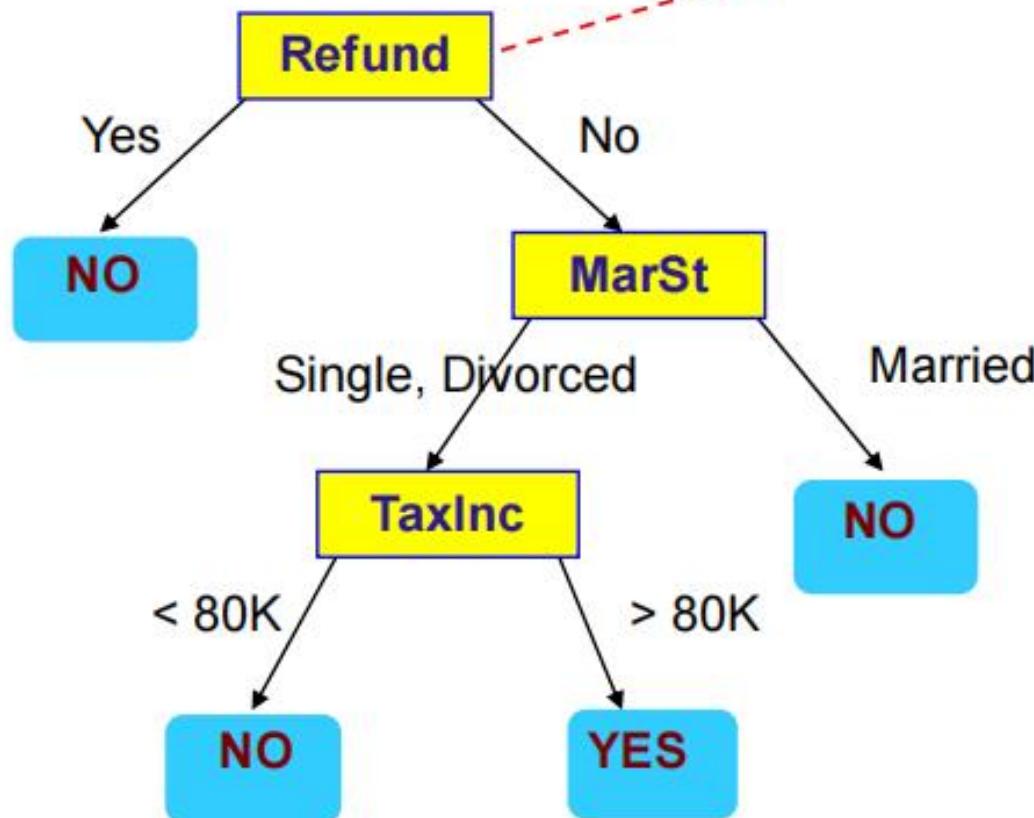
## Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

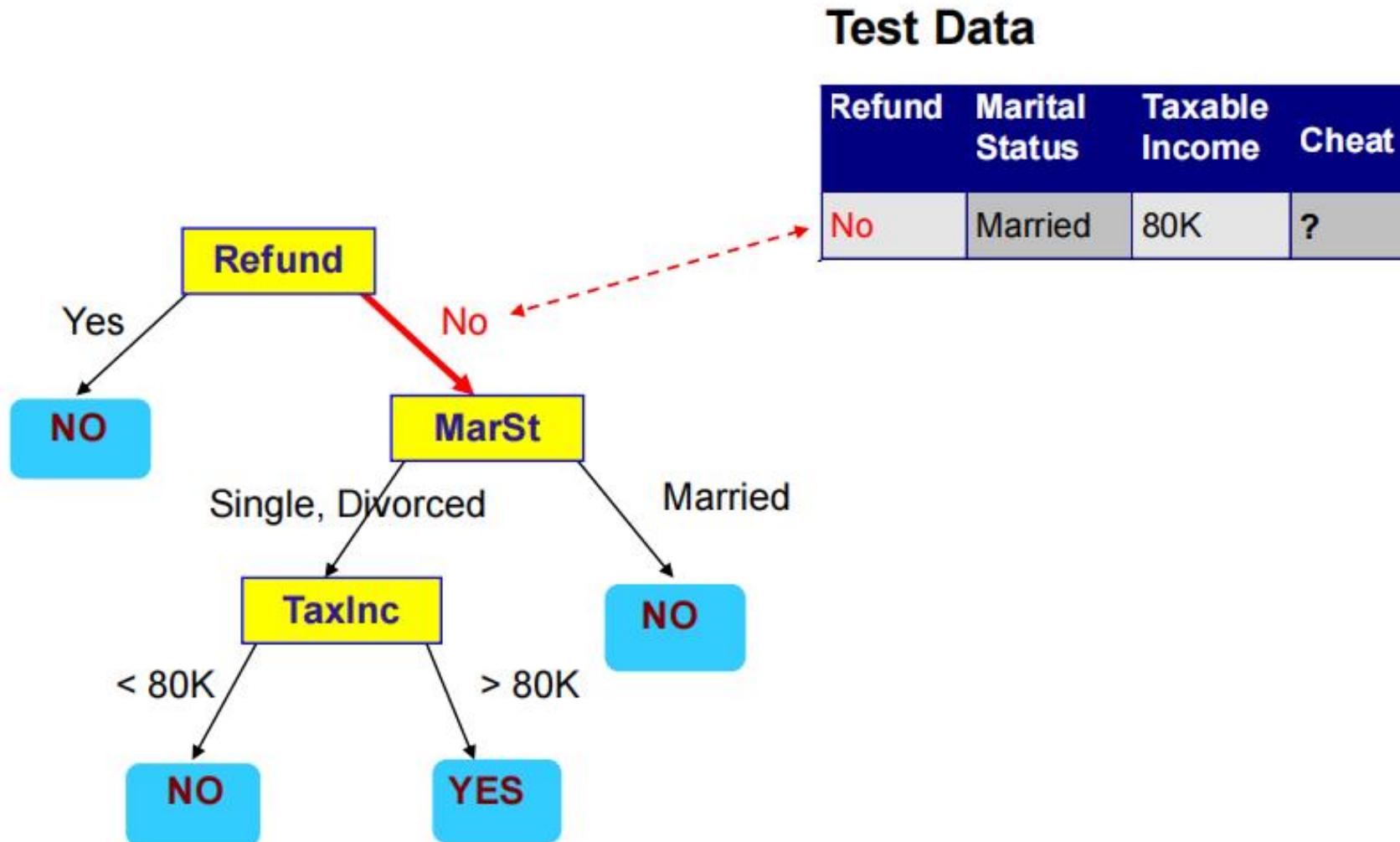
# Apply Model to Test Data

Test Data

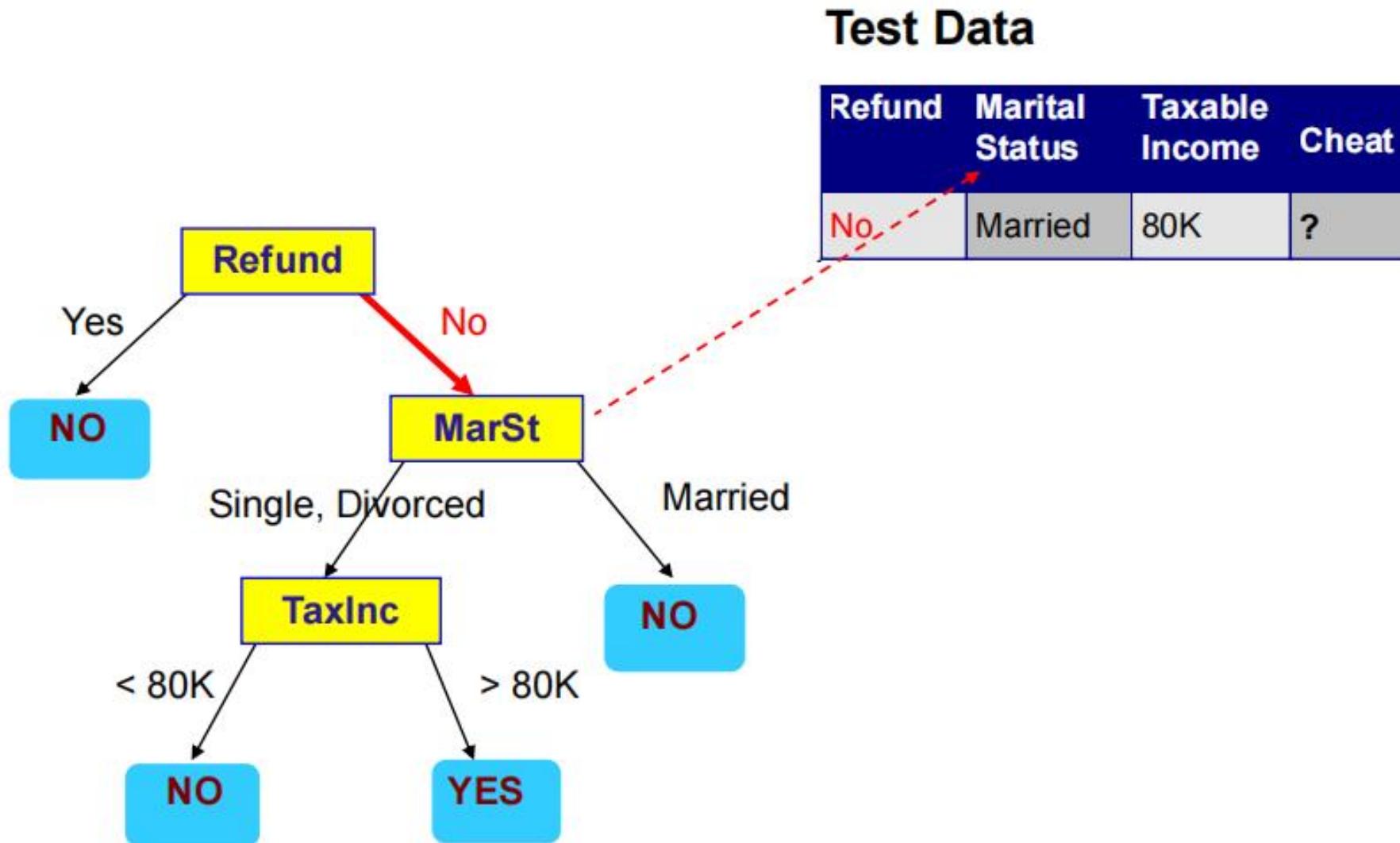
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



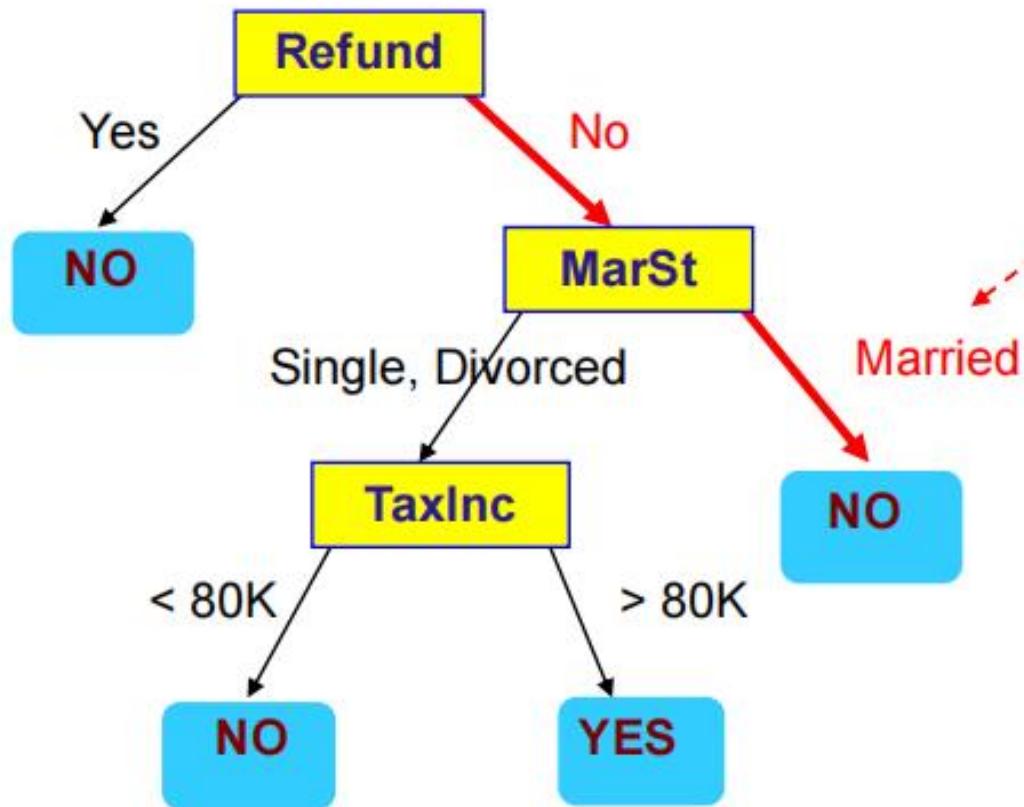
# Apply Model to Test Data



# Apply Model to Test Data



# Apply Model to Test Data



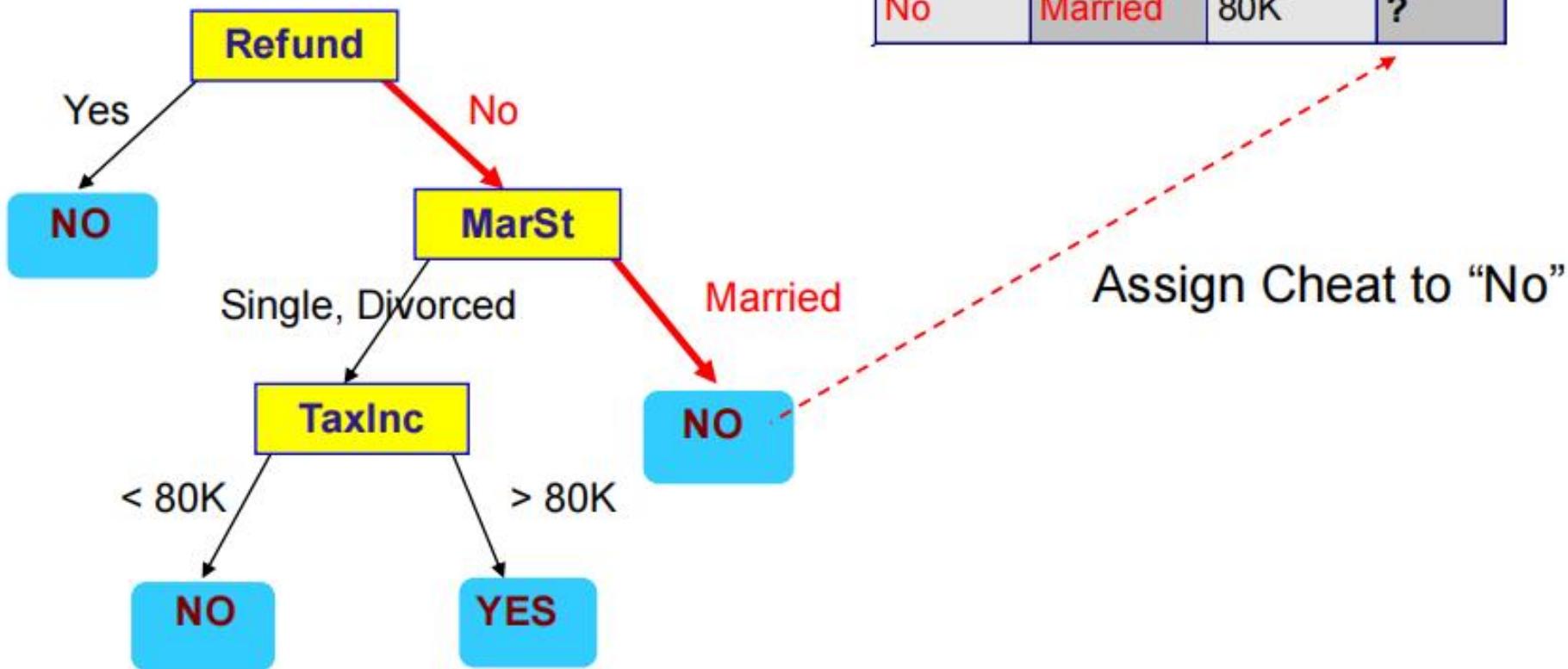
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

# Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



# Decision Tree Algorithm

## Principle

- Basic algorithm (adopted by ID3, C4.5 and CART): a **greedy algorithm**
- Tree is constructed in a top-down recursive divide-and-conquer manner
- Attributes are categorical (if continuous-valued, they are discretized in advance)
- Choose the *best* attribute(s) to split the remaining instances and make that attribute a decision node

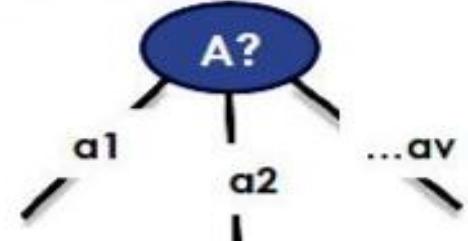
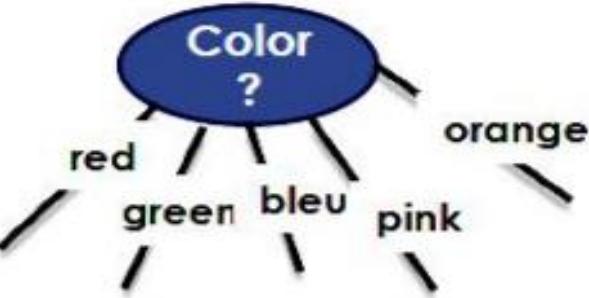
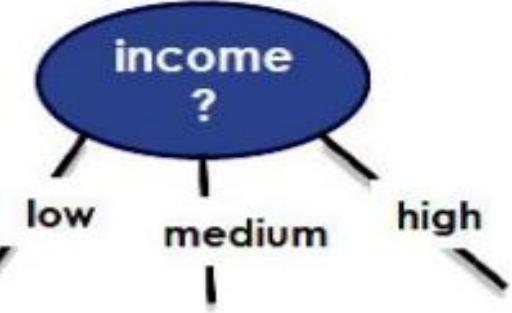
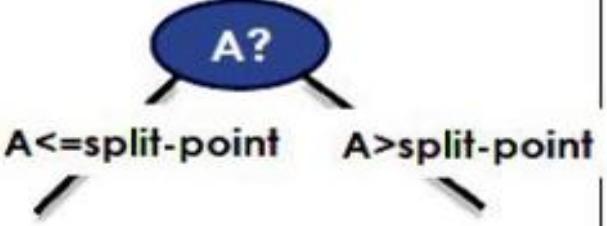
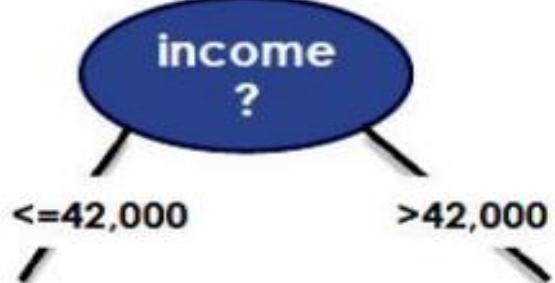
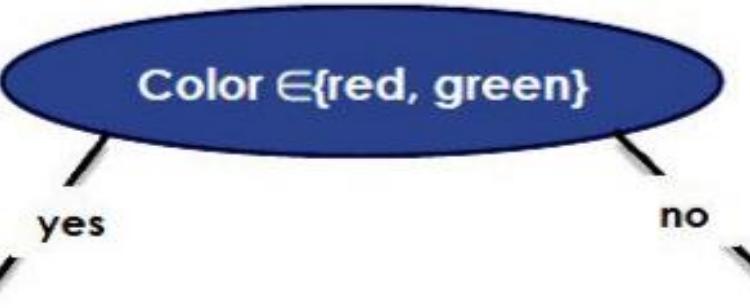
## Iterations

- At start, all the training tuples are at the root
- Tuples are partitioned recursively based on selected attributes
- Test attributes are selected on the basis of a heuristic or statistical measure (e.g., **information gain**)

## Stopping conditions

- All samples for a given node belong to the same class
- There are no remaining attributes for further partitioning – **majority voting** is employed for classifying the leaf
- There are no samples left

# Three Possible Partition Scheme

Partitioning scenarios	Examples
Discrete-valued	  
Continuous-valued	 
Discrete-valued+ binary tree	 

# Attribute Selection Measures

## How to choose An Attribute?

- An attribute selection measure is a heuristic for selecting the splitting criterion that “best” separates a given data partition, D, of class labeled training tuples into individual classes.

### **Ideally**

- Each resulting partition would be pure
- A pure partition is a partition containing tuples that all belong to the same class
- **Attribute selection measures (splitting rules)**
  - Determine how the tuples at a given node are to be split
  - Provide ranking for each attribute describing the tuples
  - The attribute with highest score is chosen
  - Determine a split point or a splitting subset
- **Methods**
  - Information gain (ID3 (Iterative Dichotomiser 3) /C4.5)
  - Gain ratio
  - Gini Index (IBM IntelligentMiner)

# 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.
- Three popular attribute selection measure:
  - Information Gain
  - Gain Ratio
  - Gini Index

# Information Gain Method

- **Entropy/Information** is the probabilistic measure of uncertainty/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 chosen

# Information Gain Method

To find the best feature that serves as a root node in terms of information gain:

1. Find the Expected Entropy/Information of Dataset: the average amount of information needed to identify the class label of a tuple in D.

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

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

2. Find the Entropy/Information of individual feature/attribute

$$\text{Entropy}_A(D) \text{ or } \text{Info}_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} \times \text{Info}(D_j)$$

- v: no. of variations of attribute A
- $p_i$ : Probability of Class i

3. Find the information gain of individual attribute(s).
- A is the particular Attribute

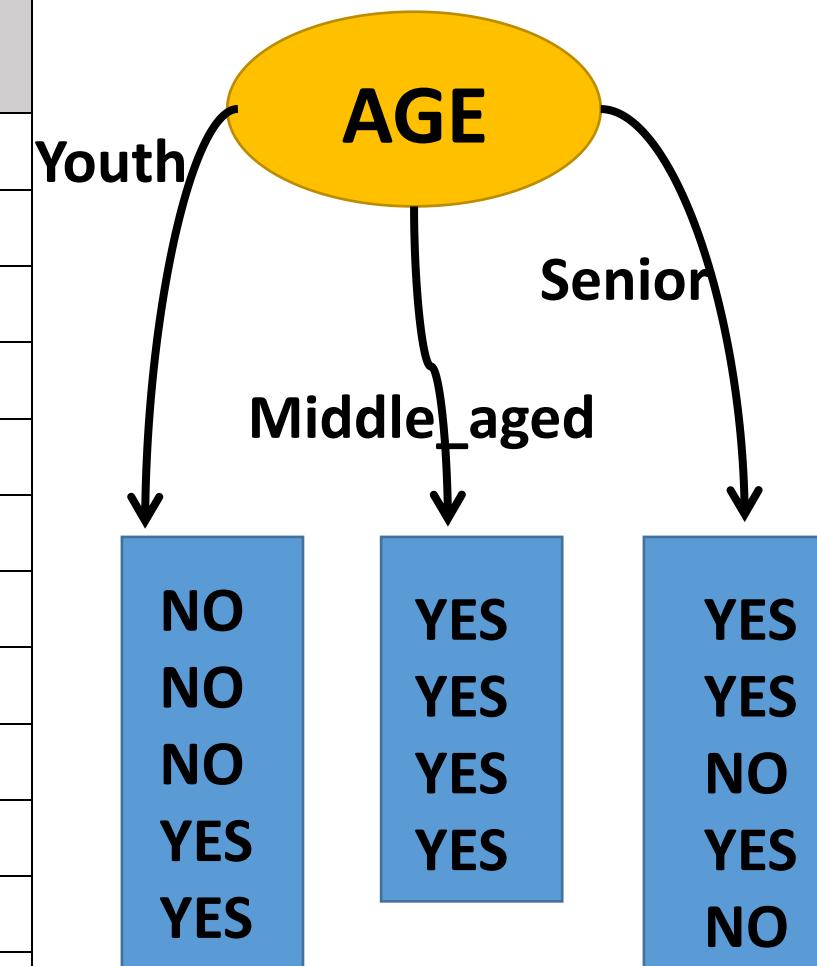
$$\text{Gain}(A) = \text{Info}(D) - \text{Info}_A(D)$$

3. Choose the best Information gain i.e Information gain with highest value.

Repeat this process to construct the complete decision tree.

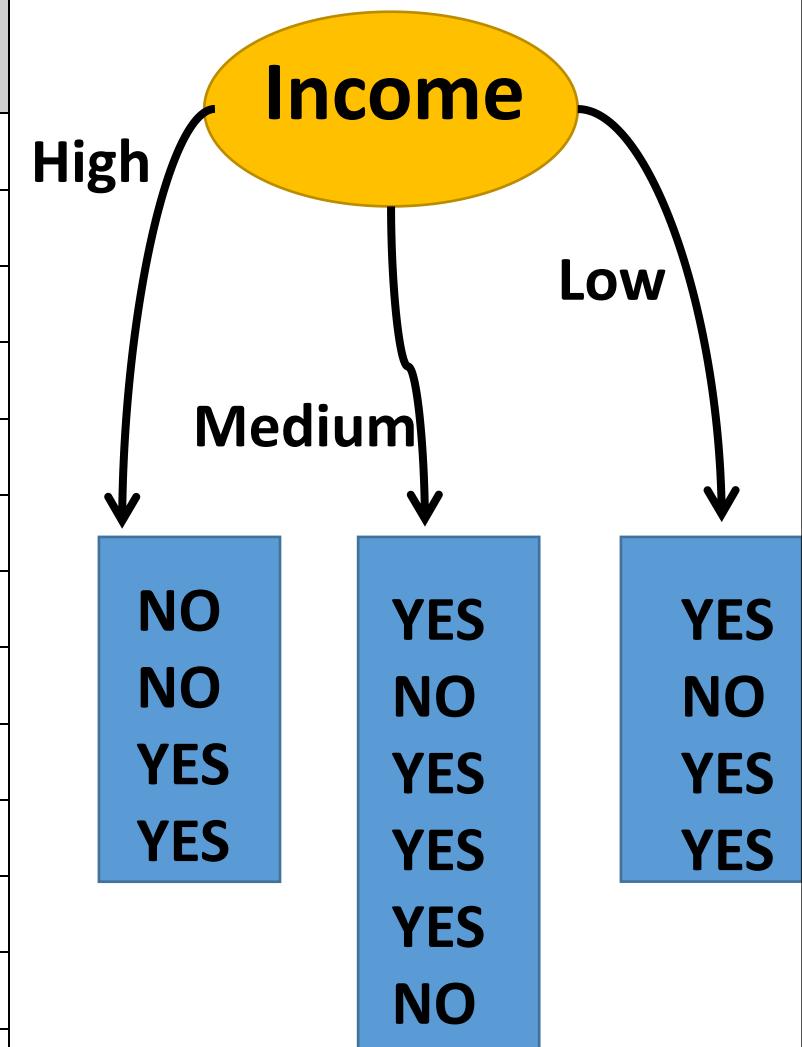
# Which Attribute to Select

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



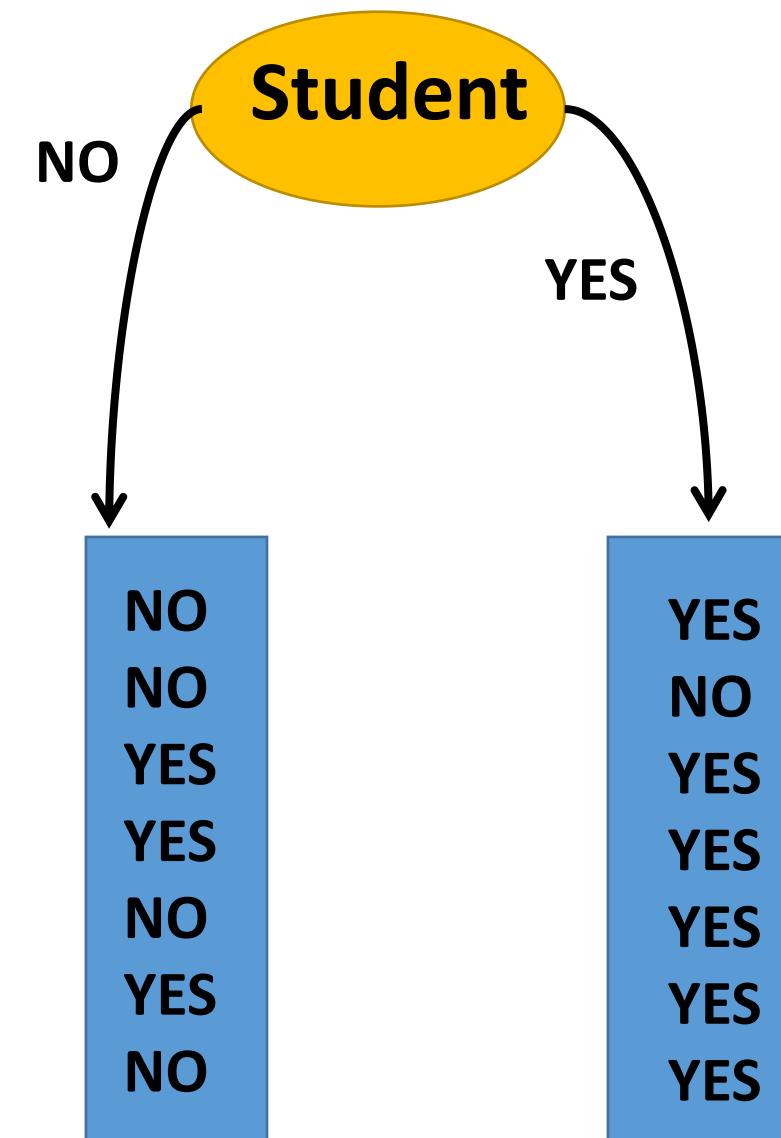
# Which Attribute to Select

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



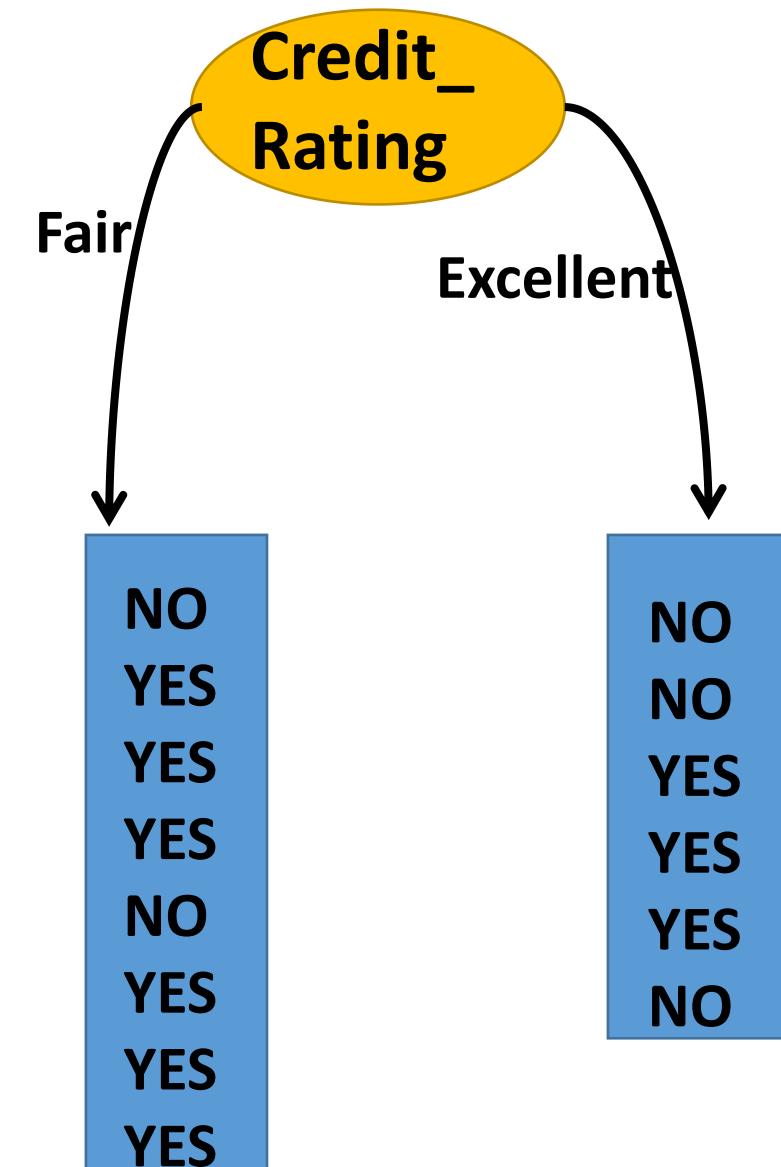
# Which Attribute to Select

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



# Which Attribute to Select

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 Approach

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

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

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

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

# Information Gain Approach

Step 2: Iteratively find the **Best Attribute** for each level of the Decision tree.

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

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

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

$$= \frac{5}{14} \left( -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} \right) + \frac{4}{14} \left( -\frac{4}{4} \log_2 \frac{4}{4} \right) + \frac{5}{14} \left( -\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5} \right) = 0.694$$

Class Variation in Youth      Class Variation in Middle\_aged      Class Variation in Senior

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

# Information Gain Approach

**Step 2:** Iteratively find the **Best Attribute** for each level of the Decision tree.

Iteration 1: Find Best Attribute for For 0<sup>th</sup> level Cont...

Similarly find the gain of attribute “income”, “student”, “credit\_rating”

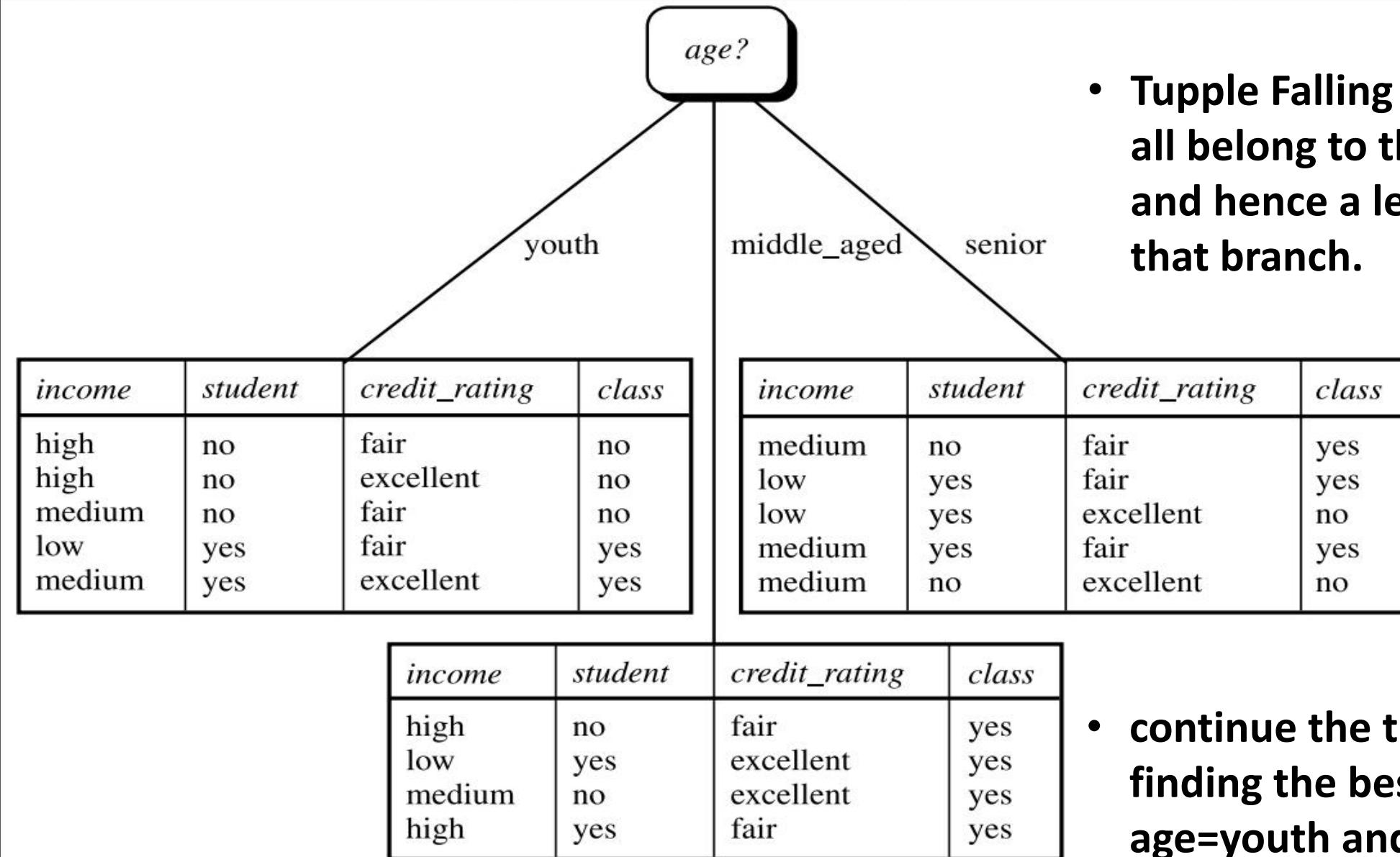
b. **Gain of income = 0.029**

c. **Gain of student = 0.151**

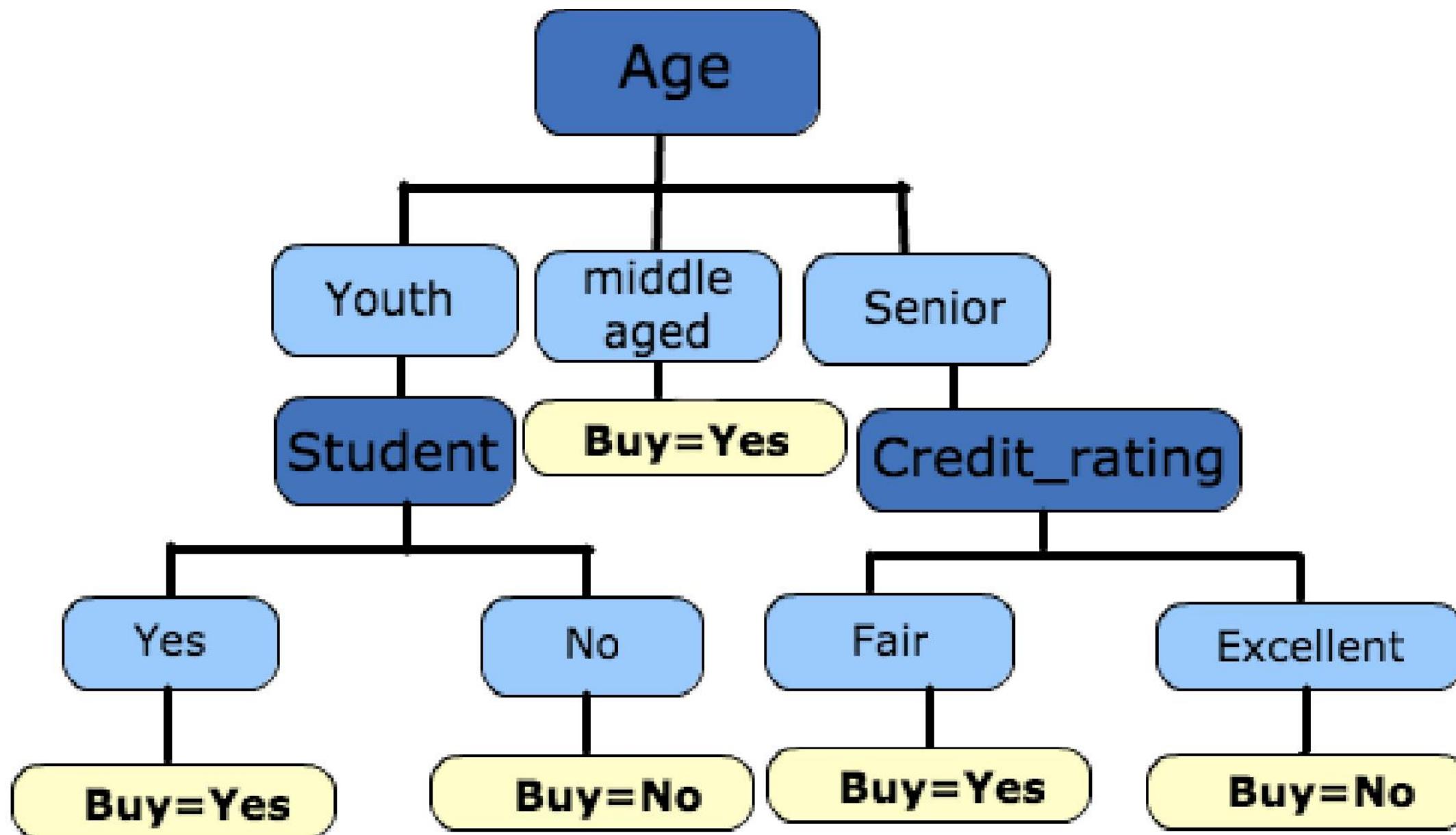
d. **Gain of credit\_rating = 0.048**

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

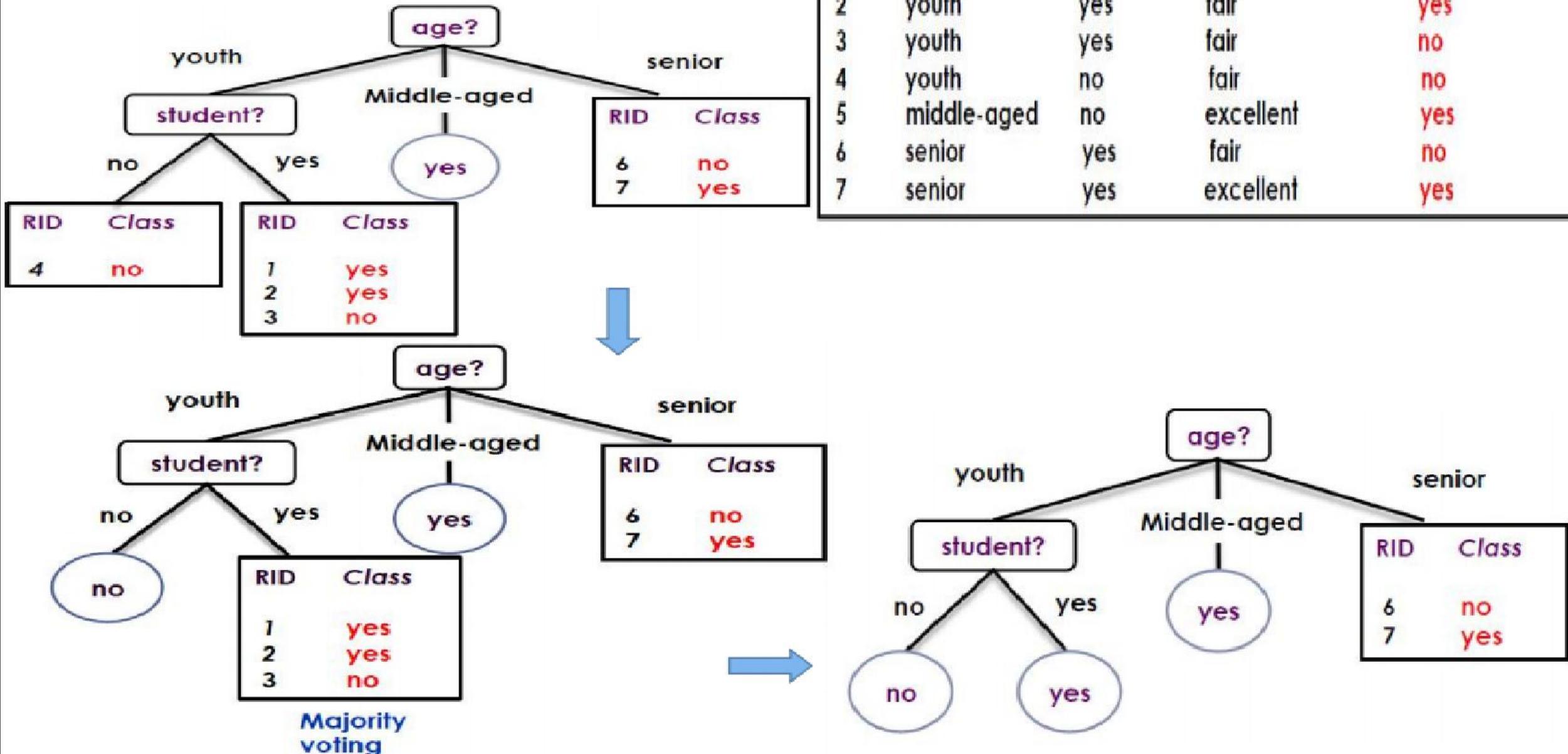
# Level 0 Decision Tree using 1<sup>st</sup> Splitting Attribute Age



# Final Decision Tree



# Example of Majority Voting



# Extracting Classification Rule from Trees

- Represent the knowledge in the form of **IF-THEN** rules
- One rule is created for each path from the root to leaf.
- Each attribute-value pair along a path **forms a conjunction**
- The leaf node holds the class prediction

## Example:

- If (Age == Youth) AND (Student == No) then buys\_computer="No"
- If (Age == Youth) AND (Student == No) then buys\_computer="Yes"
- If (Age == middle\_aged) then buys\_computer="Yes"
- If( Age == senior) AND credit\_rating == "Excellent" then buys\_computer="No"
- If(Age == senior) AND credit\_rating == "fair" then buys\_computer="Yes"

# Drawback of Information gain as attribute selection measure

- The information gain measure is biased towards tests with many outcomes.  
i.e. it prefers to select attribute having a large number of values.
- For instance, for the this dataset information gain prefers to split on attribute “Table income” that would result in a large number of pure partitions but each one containing one tuple.

$$\text{Info}_{\text{income}}(D)=0 \text{ [variation = 0]}$$

$$\Rightarrow \text{Gain}_{\text{income}}(D) = \text{Info}(D) - \text{Info}_{\text{income}}(D) = \text{Info}(D)$$

- Hence, Information gain w.r.t. income is maximal.

**Solution:**

C4.5, a successor of ID3

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

# Gain Ratio

- C4.5, a successor of ID3, uses an extension to information gain known as gain ratio, which attempts to overcome this bias.
- It attempts a kind of normalization to information gain using a “*split information*” value defined analogously with  $Info(D)$  as follows.

$$SplitInfo_A(D) = - \sum_{j=1}^v \frac{|D_j|}{D} \times \log_2 \left( \frac{|D_j|}{|D|} \right)$$

- If there are “ $v$ ” partitions in attribute A then  $SplitInfo$  finds the number of tuples having that outcome w.r.t. total number of tuple.
- Now, to compute the best splitting attribute the criteria “GainRatio” will be used instead of “Gain” which intern uses “ $SplitInfo$ ”.

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(D)}$$

- Conclusion: Gain Ratio applies a kind of normalization to information gain using “*split information*” value. the attribute with max gain ratio is selected as splitting attribute

# Example of Gain Ratio

In the process of selecting best partition attribute instead of finding gain of each attribute, now find the gain ratio of each attribute and then find the max gain ration

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

Let computed Gain of attribute income = 0.029

Attribute income is having tree partitions:

Low, Medium, High

Low -> 4 tuples

Medium -> 6 tuples

High -> 4 tuples

$SplitRatio_{income}(D)$

$$= \frac{4}{14} \log_2 \frac{4}{14} + \frac{6}{14} \log_2 \frac{6}{14} + \frac{4}{14} \log_2 \frac{4}{14} = 1.557$$

$$GainRatio = \frac{Gain_A(D)}{SplitRatio_A(D)} = \frac{0.029}{1.557} = 0.019$$

# Gini Index

- Gini Index is used in CART algorithm. It measures the impurity of database (D) and tries to minimize the impurity
- It assumes all attributes to be categorical/continuous valued and tries to find the best binary splitting position that gives minimum Gini Index and then find the best splitting attribute that can best reduce the impurity of the dataset D.
  - Step 1: Finds the impurity of dataset D  $[Gini(D)]$
  - Step 2: Find best binary splitting position of each attribute that gives the minimum Gini Index of that attribute.  $[Min\ of\ Gini_A(D)]$
  - Step 3: Find the reduction in impurity due to each attribute [due to best splitting pos] and choose the best splitting attribute which can maximize the reduction of impurity of dataset D.  $[Max\ of\ \Delta Gini(A)]$

# Gini Index

Step1: Finds the impurity of dataset D  $Gini(D) = 1 - \sum_{i=1}^m P_i^2$

m: number of class

$P_i$ : Probability of  $i^{th}$  class

Step 2: Find the best splitting position of each attribute that minimizes  $Gini_A(D)$

$$gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$$

- For instance income={Low, Medium, High} Number of splitting position=  $(2^v - 2)$
- Find the minimum  $gini_A(D)$  [best binary splitting position] from these split positions for income.
- Similarly find best splitting position for all other attributes using  $gini_A()$  index

Step3: Compute the reduction of impurity  $gini(A)$  of each attribute as follows and consider the attribute that maximizes the reduction impurity.

$$\Delta gini(A) = gini(D) - gini_A(D)$$

# Advantages and Shortcomings of Decision Tree Classifications

- A decision tree construction process is concerned with identifying the splitting attributes and splitting criterion at every level of the tree.
- Major strengths are:
  - Decision tree able to generate understandable rules.
  - They are able to handle both numerical and categorical attributes.
  - They provide clear indication of which fields are most important for prediction or classification.
- Weaknesses are:
  - The process of growing a decision tree is computationally expensive. At each node, each candidate splitting field is examined before its best split can be found.
  - Some decision tree can only deal with binary-valued target classes.

# Other Attribute Selection Measures

- **CHAID:** a popular decision tree algorithm, measure based on  $\chi^2$  test for independence
- **C-SEP:** performs better than info. gain and gini index in certain cases
- **G-statistics:** has a close approximation to  $\chi^2$  distribution
- **MDL (Minimal Description Length) principle** (i.e., the simplest solution is preferred):
  - The best tree as the one that requires the fewest # of bits to both (1) encode the tree, and (2) encode the exceptions to the tree
- **Multivariate splits** (partition based on multiple variable combinations)
  - CART: finds multivariate splits based on a linear comb. of attrs.
- **Best attribute selection measure**
  - Most give good results, none is significantly superior than others

# **Classifier -II**

# **Naive Bayesian Classifier**

# Classification by Naive Bayesian Classifier

- **Bayesian Classifier** is a statistical classifier based on Bayes Theorem which can predict the probability that a given tuple belongs to a particular class.
- **Naive Bayesian Classifier** is a simple Bayesian classifier that exhibits high accuracy and speed compared to Decision tree & Neural Network classifier.
- **Naive Assumption: [Class-Conditional Independence]:** The effect of an attribute value on a particular class is independent of other attribute value.

[Simplify the computation] => Naive

# Introduction to Bayesian Classification

- Statistical method for classification
- Supervised Learning method.
- Assumes an underlying probabilistic model, the Bayes theorem
- Can solve problems involving both categorical and continuous valued attributes.
- Named after Thomas Bayes, who proposed the Bayes Theorem

# Probability Basics: Prior Probability

A prior probability is the probability that an observation will fall into a group before one collects the data.

Example:

1. John conducts a single coin toss. What is the a priori probability of landing a head?  
A priori probability of landing a head =  $1 / 2 = 50\%$ .
  
2. A six-sided fair die is rolled. What is the a priori probability of rolling a 2, 4, or 6, in a die roll?

The number of desired outcomes is 3 (rolling a 2, 4, or 6), and there are 6 outcomes in total.

A priori probability of rolling a 2, 4, or 6 =  $3 / 6 = 50\%$ .

# Probability Basics: Posterior Probability

A **posterior probability / Conditional probability** is the probability of an event occurring given that another event has already occurred.

For **example**, we might be interested in finding the probability of some event “A” occurring after we account for some event “B” that has just occurred.

The formula to calculate a posterior probability of A occurring given that B occurred:

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

Bayes' Theorem

Where, A, B = Events

$P(B|A)$ :- The probability of B occurring given that A has happened

$P(A)$  &  $P(B)$ :- The prior probability of event A & event B

# Probability Basics: Example Calculating Posterior Probability

A forest is composed of **20% Oak trees** and **80% Maple trees**. Suppose it is known that **90% of the Oak trees are healthy** while just **50% of the Maple trees are healthy**.

Q:- Suppose that from a distance one can tell that a particular tree is healthy, what is the probability that the tree is an Oak tree?

$$P(A|B) = \frac{P(A) \times P(B|A)}{P(B)} \Rightarrow P(Oak|Healthy) = \frac{P(Oak) \times P(Healthy|Oak)}{P(Healthy)}$$

P(Oak): Probability that a tree is an oak is 0.2 (20%)

P(Healthy): Probability that a given tree is healthy=  $(0.2 \times 0.9) + (0.8 \times 0.5) = 0.58$

P(Healthy|Oak): The probability that a tree is healthy given that it's an Oak tree is 0.9 [90%]

$$P(Oak|Healthy) = \frac{0.2 \times 0.9}{0.58} = 0.3103$$

# Probability Basics: Posterior Probability contd...

For an intuitive understanding of this probability, suppose the following grid represents this forest with 100 trees. Exactly 20 of the trees are Oak trees (20%) and 18 (90%) of them are healthy. The other 80 (80%) trees are Maple and 40 (50%) of them are healthy.

(O = Oak, M = Maple, Green = Healthy, Red = Unhealthy)

O	O	M	M	M	M	M	M	M	M
O	O	M	M	M	M	M	M	M	M
O	O	M	M	M	M	M	M	M	M
O	O	M	M	M	M	M	M	M	M
O	O	M	M	M	M	M	M	M	M
O	O	M	M	M	M	M	M	M	M
O	O	M	M	M	M	M	M	M	M
O	O	M	M	M	M	M	M	M	M
O	O	M	M	M	M	M	M	M	M
O	O	M	M	M	M	M	M	M	M

Out of all exactly 58 of them are healthy and 18 of these healthy ones are Oak trees. Thus, if a healthy tree has selected then the probability that it's an Oak tree is  $18/58 = 0.3103$ .

# Bayes' Theorem [useful in finding posterior probability]

**Given:**

- **X:** Data tuple described by measurement made on a set on n attributes
- **H:** Some hypothesis that the data tuple “X” belongs to class C

**Problem:** Determine  $P(H|X)$ :- Given the evidence/observed data tuple “X”, find the probability that the hypothesis H holds. i.e. Probability that tuple “X” belongs to class C

According to Bayes Theorem

$$P(H|X) = \frac{P(X|H) \times P(H)}{P(X)}$$

**P(H|X):-** Posterior Probability of class “H” where attribute values  $\langle x_1, \dots, x_n \rangle$  are given.

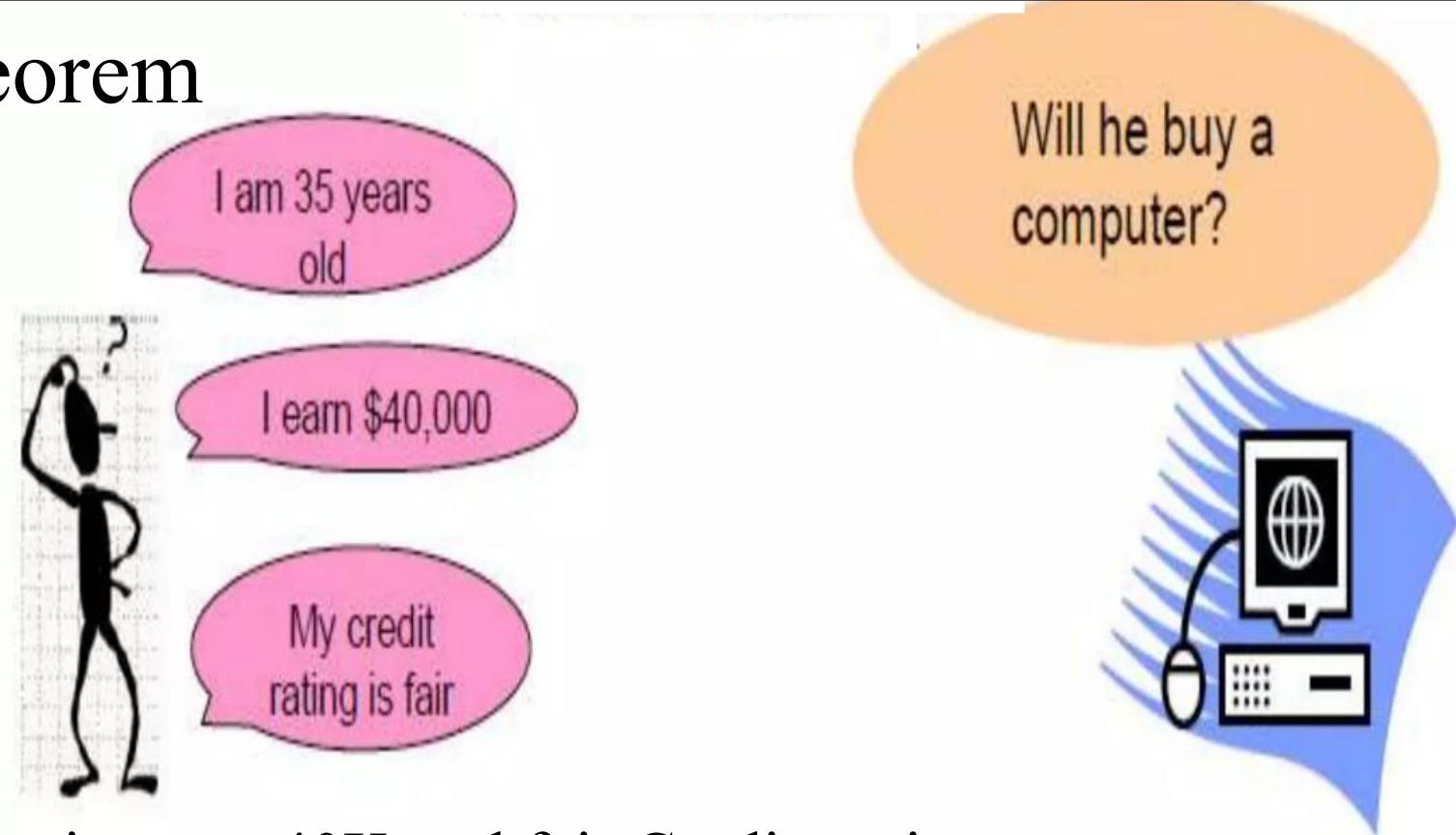
**P(H):** Prior Probability of class “H” regardless of attributes values  $\langle x_1, \dots, x_n \rangle$

**P(X|H):** Posterior Probability of attribute values  $\langle x_1, \dots, x_n \rangle$  where class “H” is given

**P(X):** Prior Probability of attribute values  $\langle x_1, \dots, x_n \rangle$  regardless of class “H”

# An Example of Bayes' Theorem

**Example:** Let customers are only described by attributes age, income and credit\_rating



- X: 35 years old customer with an income 40K and fair Credit\_rating
  - H: Hypothesis that the customer will buy a computer

# Determine:-

# An Example of Bayes' Theorem contd...

According to Bayes Theorem

$$P(H|X) = \frac{P(X|H) \times P(H)}{P(X)}$$

Let hypothesis H is buying computer = “YES”

**P(H|X):-** Posterior Probability that the customer will buy computer = "YES" given that age = 35, income = 40K and credit\_rating = fair.

**P(H):** Prior Probability of class Yes regardless of attributes values <age, income, credit\_rating>

**P(X|H):** Posterior Probability of attribute values <age=35, income=40K credit\_rating = fair> where class buying computer = "YES" is given.

**P(X):** Prior Probability of attribute values <age=35, income=40K credit\_rating = fair> regardless of class buying computer

# Naive Bayes Classifier

- Given: D: - Set of tuples.  
 $X: -\langle x_1, \dots, x_n \rangle$  where  $x_i$  is the value of attribute  $A_i$   
m: Number of classes -  $C_1, \dots, C_m$

- Bayesian classifier predicts  $X$  belongs to Class  $C_i$  iff

$$P(C_i|X) > P(C_j|X) \quad \text{for } 1 \leq j \leq m \text{ & } j \neq i$$

or

Maximize Posterior Hypothesis:  $P(C_i|X) = \frac{P(X|C_i) \times P(C_i)}{P(X)}$

# Naive Bayes Classifier

In the maximization of hypothesis

$$P(C_i|X) = \frac{P(X|C_i) \times P(C_i)}{P(X)},$$

Find three values 1.  $P(X)$  2.  $P(C_i)$  3.  $P(X|C_i)$

**Naive Assumption: [Class-Conditional Independence]: The effect of an attribute value on a particular class is independent of other attribute value.**

1. Due to conditional independence of the attribute  $P(X) = P(x_1, \dots, x_n) = 1$
2.  $P(C_i)$ : Probability that class  $i$  is occurring in the dataset

$$P(C_i) = \frac{|C_{i,D}|}{|D|} \text{ where } |C_{i,D}| \text{ is the number of tuple containing class } i$$

3.  $P(X|C_i) = P(x_1, \dots, x_n | C_i)$ .

Due to conditional independence,  $P(X|C_i) = \prod_{k=1}^n P(x_k|C_i) = P(x_1|C_i) \times \dots \times P(x_n|C_i)$

**As  $P(X)$  is 1, Hence only maximize  $P(X|C_i) \times P(C_i)$**

# Problem

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)

# Problem

**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**

$$\begin{aligned} P(X|\text{Yes}) &= \prod_{k=1}^n P(x_k|\text{Yes}) \\ &= P(x_1: \text{age}=\text{youth}|\text{Yes}) \times \\ &\quad P(x_2: \text{income}=\text{medium}|\text{Yes}) \times \\ &\quad P(x_3: \text{student}=\text{yes}|\text{Yes}) \times \\ &\quad P(x_4: \text{credit\_rating}=\text{fair}|\text{Yes}) \\ &= \frac{2}{9} \times \frac{4}{9} \times \frac{6}{9} \times \frac{6}{9} = 0.22 \times 0.44 \times 0.66 \times 0.66 = 0.044 \end{aligned}$$

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

$$\begin{aligned} P(X|\text{Yes}) &= \prod_{k=1}^n P(x_k|\text{No}) \\ &= P(x_1: \text{age}=\text{youth}|\text{No}) \times \\ &\quad P(x_2: \text{income}=\text{medium}|\text{No}) \times \\ &\quad P(x_3: \text{student}=\text{yes}|\text{No}) \times \\ &\quad P(x_4: \text{credit\_rating}=\text{fair}|\text{No}) \\ &= \frac{3}{5} \times \frac{2}{5} \times \frac{1}{5} \times \frac{2}{5} = 0.6 \times 0.4 \times 0.2 \times 0.4 = 0.019 \end{aligned}$$

# Problem

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

$$\begin{aligned} & \text{Max} \{ P(X|\text{buys\_Comp}=\text{Yes}) \times p(\text{buys\_computer} = \text{Yes}), \\ & \quad P(X|\text{buys\_Comp}=\text{No}) \times p(\text{buys\_computer} = \text{No}) \\ & = \{0.44 \times 0.643, 0.019 \times 0.357\} \\ & = \{0.028, 0.007\} = 0.028 \text{ } \{ \text{Class "Yes" Maximizes the conditional probability of X} \} \end{aligned}$$

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

# Problem

OUTLOOK	TEMP	HUMIDITY	WINDY	PLAY GOLF?
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

For the given training dataset let the new customer X value as follows. Find the probability that Golf will be played or not?

Find the query X belongs to which class?

X = (Outlook=Rainy

Temp = Hot

Humidity = High

Windy = False)

# Partial Solution

$$P(p) = 9/14$$

$$P(n) = 5/14$$

outlook	
$P(\text{sunny} p) = 2/9$	$P(\text{sunny} n) = 3/5$
$P(\text{overcast} p) = 4/9$	$P(\text{overcast} n) = 0$
$P(\text{rain} p) = 3/9$	$P(\text{rain} n) = 2/5$
temperature	
$P(\text{hot} p) = 2/9$	$P(\text{hot} n) = 2/5$
$P(\text{mild} p) = 4/9$	$P(\text{mild} n) = 2/5$
$P(\text{cool} p) = 3/9$	$P(\text{cool} n) = 1/5$
humidity	
$P(\text{high} p) = 3/9$	$P(\text{high} n) = 4/5$
$P(\text{normal} p) = 6/9$	$P(\text{normal} n) = 2/5$
windy	
$P(\text{true} p) = 3/9$	$P(\text{true} n) = 3/5$
$P(\text{false} p) = 6/9$	$P(\text{false} n) = 2/5$

# **Advantage**

- Simple, fast and easy to implement. Needs less training data. Highly scalable.
- Handles continuous and discrete data.
- Easily updatable if training dataset increases

# **Uses**

- Text Classification
- Spam Filtering
- Hybrid Recommender
- Online Application

# **Disavantages**

- Assumption: Class conditional independence, therefore loss of accuracy
- Practically, dependencies exist among variables.

Example: Hospitals, patients profile, family history, disease symptoms etc

# Major Issues

- **Violations of Independence Assumption:** It assumes that the effect of an attribute values on a given class is independent of the values of the other attribute.
  - Solution: Bayesian belief network classifier allows the representation of dependencies among subsets of attributes
- **Zero Conditional Probability problem:** If a given class and feature value never occur together in the training set then the frequency based probability estimate will be zero. This is problematic since it will wipe out all information in the other probabilities when they are multiplied.
  - Therefore often it is desirable to incorporate a small sample correction in all possible option. Or consider the occurrence/probability as 1 instead of zero which can be negligible in the consideration of maximum.

# K-Nearest Neighbor

(Lazy Learners or Learning from your Neighbor)

# Eager Learners Vs. Lazy Learners

## Eager Learner

- Eager Learner approach of classification constructs a generalized classification model just after receiving the training tuples (much before receiving new/test tuple), and be ready and eager to classify the new unseen tuple.
- More work when training tuple is received and less work when making a classification.
- Less Expensive but not support incremental learning
- e.g. Decision Tree, Bayesian Classifier, Backpropagation, Association Rule Mining

## 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
- e.g. K-Nearest Neighbor, case-based learning

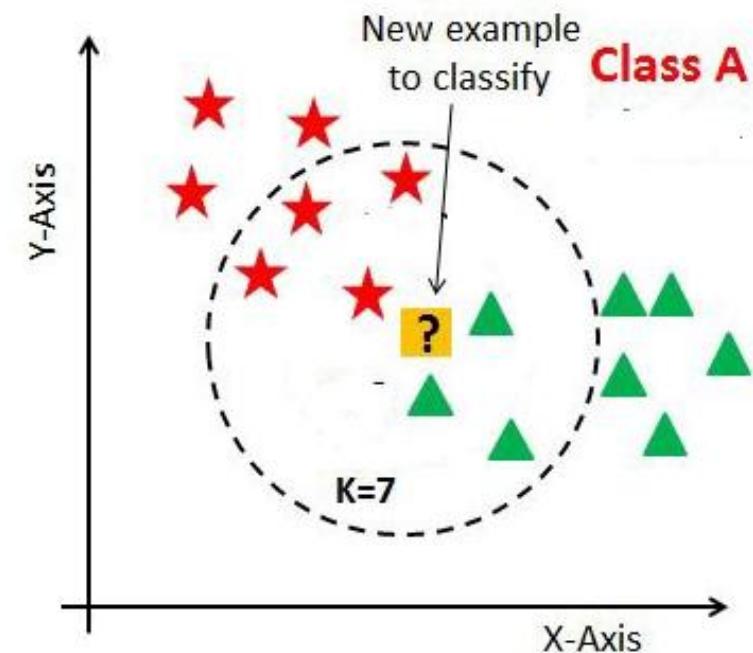
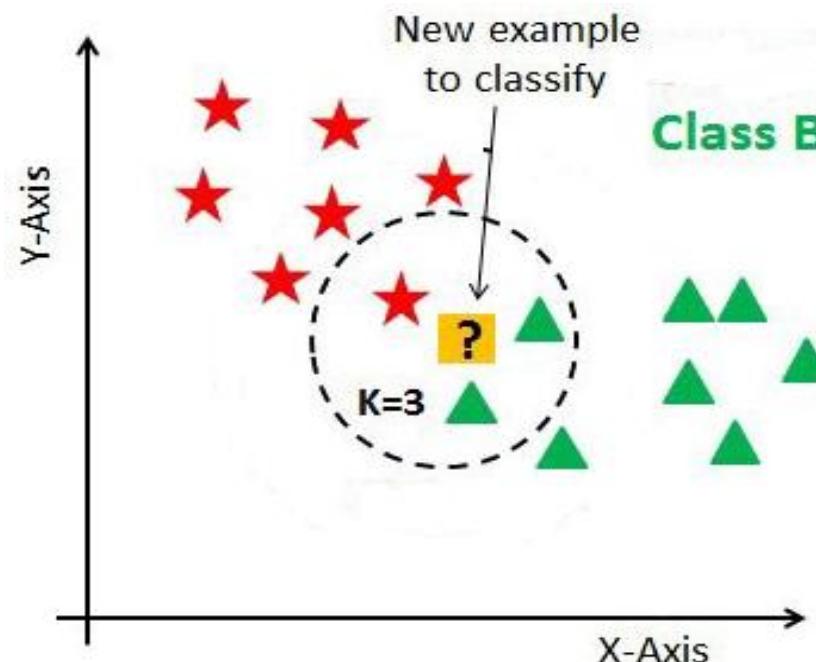
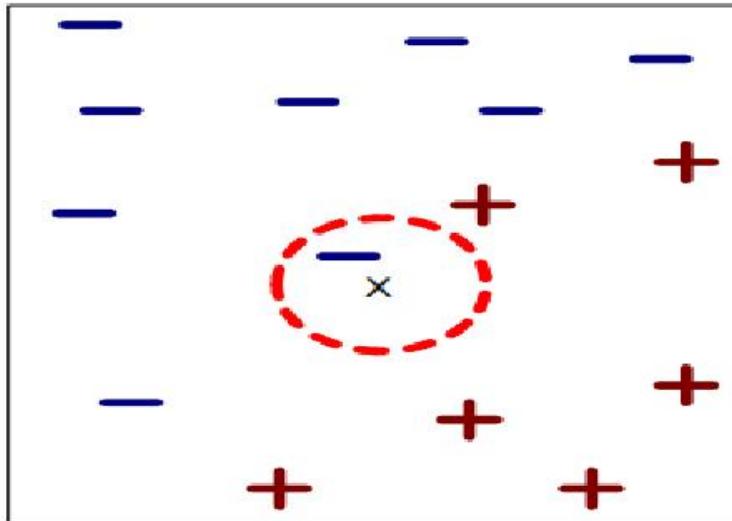
# K-Nearest Neighbor Classifier

**Nearest based classifiers** are based on learning by analogy i.e. by comparing a given test tuples with training tuples that are similar to it.

- Let the training tuples are described by n attributes. All instances of training tuples are correspond to points in the n-dimensional space and accordingly each tuple represents a point in an n-dimensional space
- A **k-nearest-neighbor classifier** searches the pattern space for the k training tuples that are closest to the unknown tuple.

Prediction for test data is done on the basis of its neighbour

# Properties of KNN



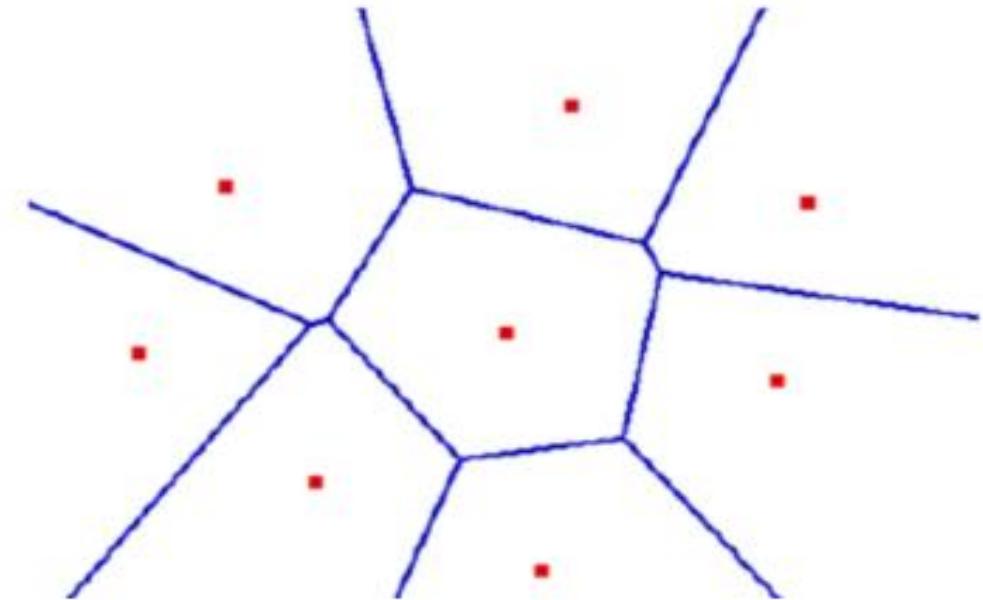
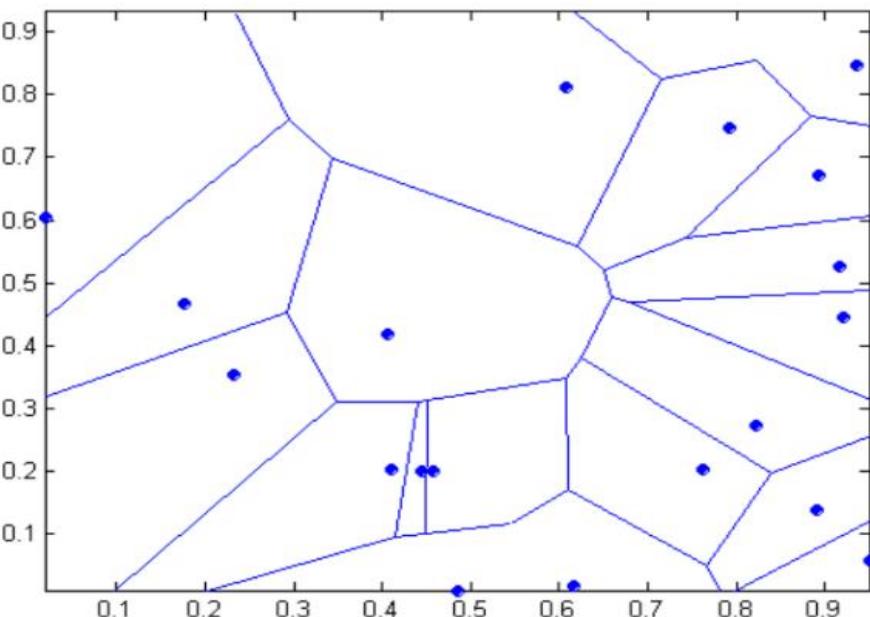
- How many neighbors should be consider? i.e. what is k?
- How do we measure distance?
- Should all points be weighted equally, or some points have more influence than others?

## ➤ How many neighbors should be consider? (what is k)

- At **K=1**, the KNN model tends to closely **follow the training data** and thus shows a **high training score and low test score**, indicating overfitting.
- The problem can be solved by tuning the value of `k_neighbors` parameter. As we increase the number of neighbors k, the model starts to generalize well, but increasing the value too much would again drop the performance.
- Therefore, it's important to find an optimal value of K, such that the model is able to classify well on the test data set with minimum error.
- Setting K value to the square root of number of training tuple or half of the square root of number of training tuple can lead to better result.
- For an even number of classes (e.g. 2 or 4) it is a good idea to choose a K value with an odd number **to avoid a tie**. And the inverse, use an even number for K when you have an odd number of classes

# Decision Boundary: Voronoi Diagram

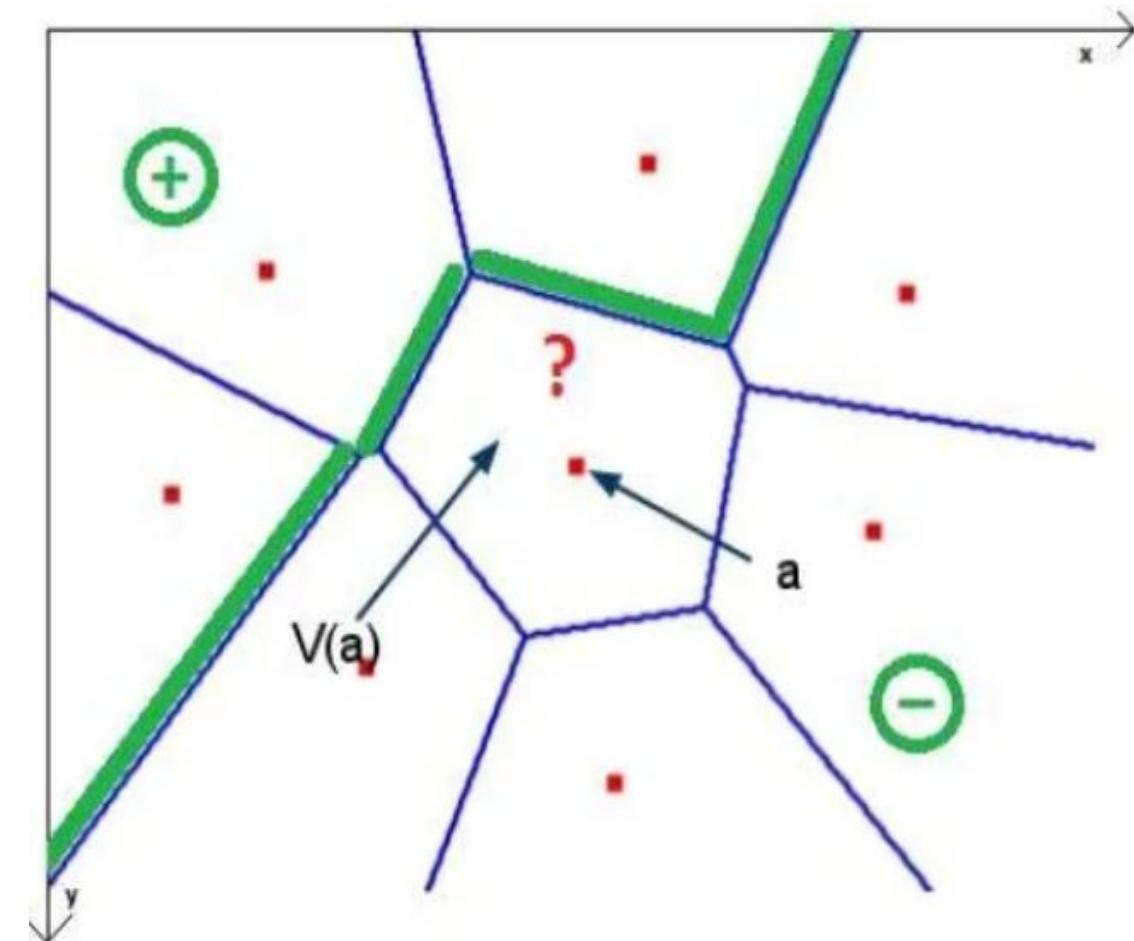
- Given a set of data, Voronoi diagram describes the areas that are nearest to any given point.
- Each line segment is at equidistance between two points of opposite class.



A Voronoi diagram divides the space into such cells that each cell contains one sample and every location within the cell is closer to that sample than to any other sample. .

# Decision Boundary: Voronoi Diagram

- Let the requirement is to classify data “?” using the voronoi diagram.
- For this it is required to construct the voronoi diagram decision boundary.
- Let the data “?” is located in the voronoi area of “a” (diagram) which is in the negative side of the decision boundary or belongs to negative class. As this area contains all the points closest to “a” compared to other instance, the data “?” belongs to class to which a belongs, i.e. negative class.



# How to Measure Distance

$$\text{Euclidean distance} : d(x, y) = \sqrt{\sum (x_i - y_i)^2}$$

$$\text{Squared Euclidean distance} : d(x, y) = \sum (x_i - y_i)^2$$

$$\text{Manhattan distance} : d(x, y) = \sum |(x_i - y_i)|$$

From these Euclidean distance is more preferable as it gives equal importance to each feature

# Example: 1

- Let given is the data from the questionnaires survey with two attributes ( $X_1$ : Acid durability and  $X_2$ : Strength) to classify whether a special paper tissue is good or not. Classify the new paper tissue with features:  $X_1=3$  and  $X_2=7$ .

<b>X1: Acid Durability</b>	<b>X2: Strength</b>	<b>Y: Class</b>
7	7	Bad
7	4	Bad
3	4	Good
1	4	Good

# Example-1: Solution

## Step 1: Initialize and Define K

K=1	Overall Rule 1	X1: Number of Acid Durability	X2: Strength	Y: Class	Distance from (3, 7)
Let K = 3					

Step 2: Compute distance measure	7	7	Bad	$\sqrt{(7-3)^2 + (7-7)^2} = 4$	sing Euclidian b) $^2$ ))
	7	4	Bad	$\sqrt{(7-3)^2 + (4-7)^2} = 5$	
	3	4	Good	$\sqrt{(3-3)^2 + (4-7)^2} = 3$	
	1	4	Good	$\sqrt{(1-3)^2 + (4-7)^2} = 3.6$	

# Example-1: Solution

**Step 3: Determine K nearest neighbor those are at minimum distance and rank them.**

X1: Acid Durability	X2: Strength	Y: Class	Distance from (3, 7)	Rank
7	7	Bad	4	3
7	4	Bad	5	4
3	4	Good	3	1
1	4	Good	3.6	2

**Step 4: Apply Simple Majority**

There are 2 “Good” and 1 “Bad”. Thus the new paper tissue will be classified as Good due to simple majority

## Example 2

Given is the customer detail as customer's age, income, credit card no. and their corresponding class. Find the class label for the new tuple/sample:

Customer: RRR, Age: 37, Income: 50K, Cr Card: 2, Class- ?

Customer	Age	Income	No. of Credit Cards	Class
XXX	35	35K	3	No
YYY	22	50K	2	Yes
ZZZ	63	200K	1	No
PPP	59	170K	1	No
QQQ	25	40K	4	Yes
<b>RRR</b>	<b>37</b>	<b>50K</b>	<b>2</b>	<b>????</b>

# Example-2: Solution

Find the Euclidian Distance of new tuple from each training data

Customer	Age	Income	No. of Credit Cards	Class	Distance
XXX	35	35K	3	No	$\sqrt{(35-37)^2 + (35-50)^2 + (3-2)^2} = 15.16$
+YYY	22	50K	2	Yes	$\sqrt{(22-37)^2 + (50-50)^2 + (2-2)^2} = 15$
ZZZ	63	200K	1	No	$\sqrt{(63-37)^2 + (200-50)^2 + (1-2)^2} = 152.2$
PPP	59	170K	1	No	$\sqrt{(59-37)^2 + (170-50)^2 + (1-2)^2} = 122$
QQQ	25	40K	4	Yes	$\sqrt{(25-37)^2 + (40-50)^2 + (4-2)^2} = 15.74$
<b>RRR</b>	<b>37</b>	<b>50K</b>	<b>2</b>	<b>????</b>	

Majority of {Yes, Yes, No} is Yes. Hence Class of new tuple is Yes

# Should all Attributes be weighted equally, or Some Attributes have more influence than others?

Some attributes (like income, loan) outweighs some smaller ranged attributes (like binary or categorical or numeric like age) and thus, the result obtained can be a biased one.

Normalization is a solution to it: Max-Min Normalization generally used

Age	Loan	Default	Distance
25	\$40,000	N	102000
35	\$60,000	N	82000
45	\$80,000	N	62000
20	\$20,000	N	122000
35	\$120,000	N	22000
52	\$18,000	N	124000
23	\$95,000	Y	47000
40	\$62,000	Y	80000
60	\$100,000	Y	42000
48	\$220,000	Y	78000
33	\$150,000	Y	8000
48	\$142,000	?	

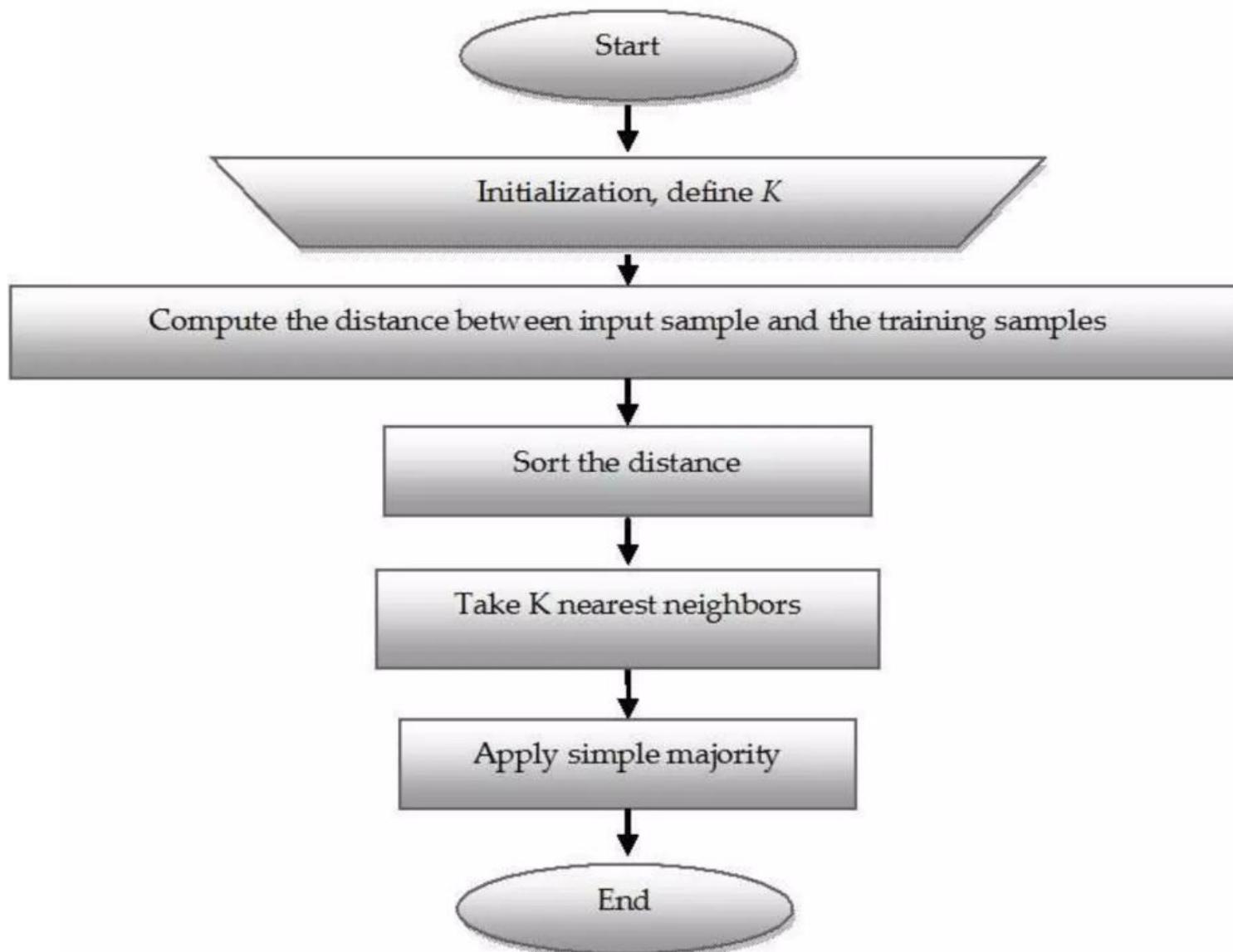
# Solution : Normalization

Age	Loan	Default	Distance
0.125	0.11	N	0.7652
0.375	0.21	N	0.5200
0.625	0.31	N	0.3160
0	0.01	N	0.9245
0.375	0.50	N	0.3428
0.8	0.00	N	0.6220
0.075	0.38	Y	0.6669
0.5	0.22	Y	0.4437
1	0.41	Y	0.3650
0.7	1.00	Y	0.3861
0.325	0.65	Y	0.3771
0.7	0.61	?	

Standardized Variable

$$X_s = \frac{X - \text{Min}}{\text{Max} - \text{Min}}$$

# KNN Classifier Algorithm



## **Advantage:**

- No training period: There is no discriminative function or model build at the traing stage
- 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

## **Disadvantage:**

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

# Application

- Used in classification and regression
- Used in pattern recognition
- Used in gene expression, protein-protein prediction, get 3D structure of the protein
- Used to measure document similarity