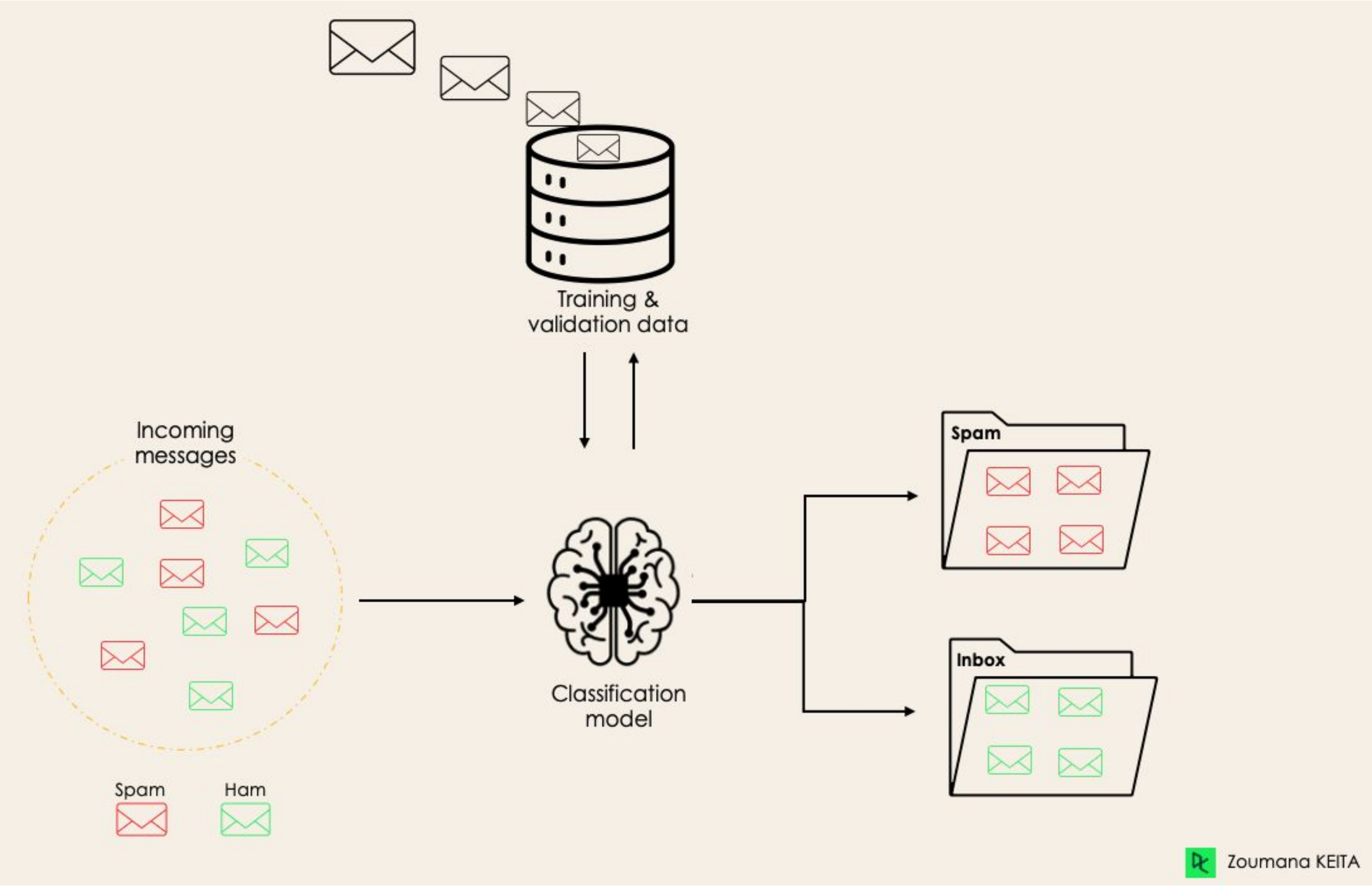
Chapter 4
Classification Techniques

What is Classification

- Classification is a supervised machine learning method where the model tries to predict the correct label of a given input data.

- In classification, the model is fully trained using the training data, and then it is evaluated on test data before being used to perform prediction on new unseen data.

For instance, an algorithm can learn to predict whether a given email is spam or ham (no spam), as illustrated below.



Two types of learners in classification

1) **Eager learners**

- These are machine learning algorithms that first build a model from the training dataset before making any prediction on future datasets.

- They spend more time during the training process because of their eagerness to have a better generalization during the training from learning the weights, but they require less time to make predictions.

- Example:
  - Logistic Regression.
  - Support Vector Machine.
  - Decision Trees.
  - Artificial Neural Networks.

Two types of learners in classification contd..

2) **Lazy learners or instance-based learners**

- Do not create any model immediately from the training data, this is where the lazy aspect comes from.

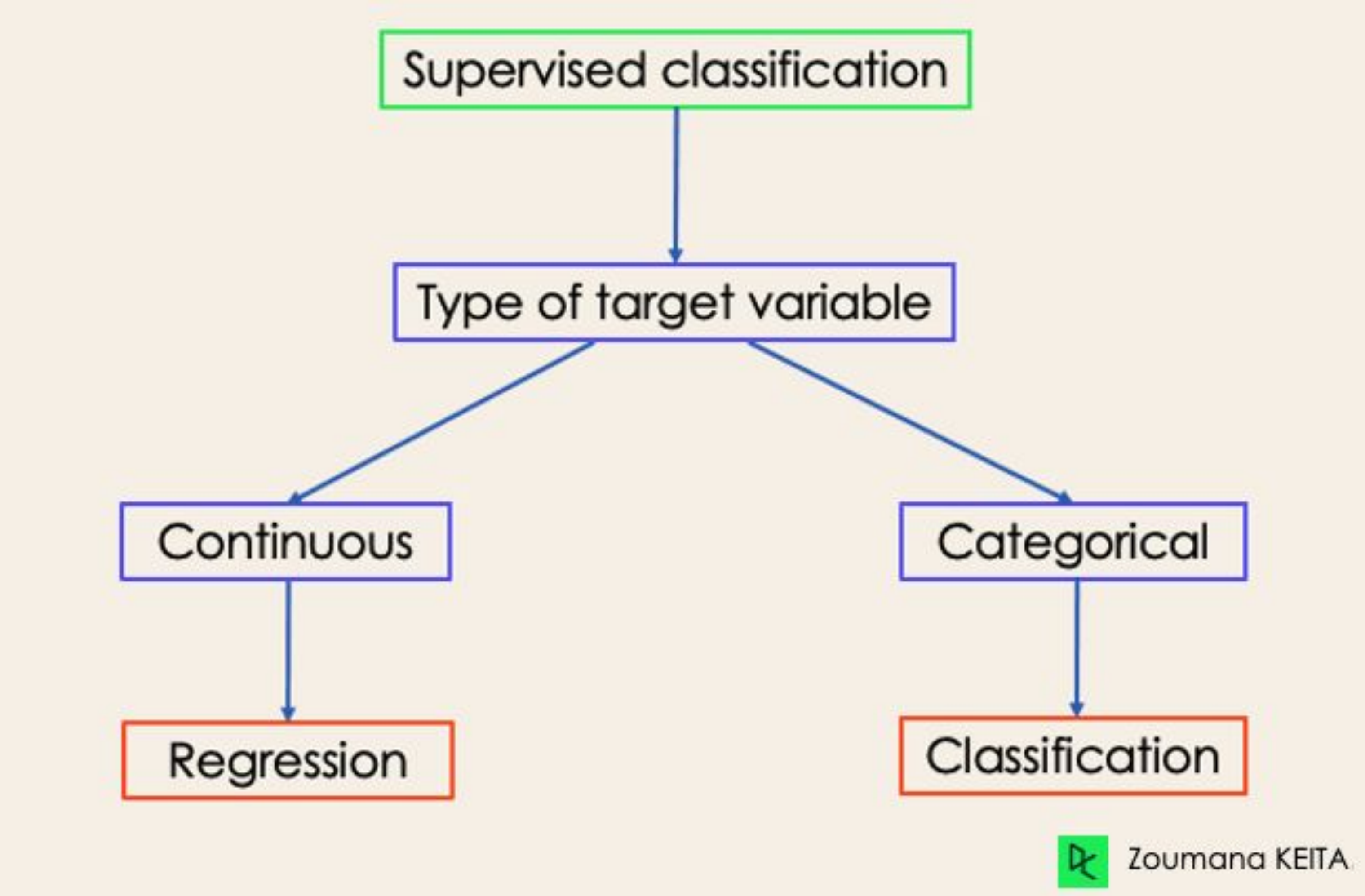
- They just memorize the training data, and each time there is a need to make a prediction, they search for the nearest neighbor from the whole training data, which makes them very slow during prediction.

- Example:
  - K-Nearest Neighbor.
  - Case-based reasoning.

Classification Vs. Regression

- The prediction task is a **classification** when the target variable is discrete.
  - An application is the identification of a given email as spam or ham

- The prediction task is a **regression** when the target variable is continuous.
  - An example can be the prediction of the salary of a person given their education degree, previous work experience, geographical location, and level of seniority.



Zoumana KEITA

Examples of Classification in Data Analytics

- Life Science: Predicting tumor cells as benign or malignant
- Security: Classifying credit card transactions as legitimate or fraudulent
- Prediction: Weather, voting, political dynamics, etc.
- Entertainment: Categorizing news stories as finance, entertainment, sports, etc.
- Social media: Identifying the current trend and future growth

Different Types of Classification Tasks

1) **Binary Classification**

- Binary is a type of problem in classification in machine learning that has only two possible outcomes.
- For example, yes or no, true or false, spam or not spam, etc.

2) Multi-Class Classification

- Multi-class is a type of classification problem with more than two outcomes.
- For example, classifying images, classifying species, and categorizing faces, among others.

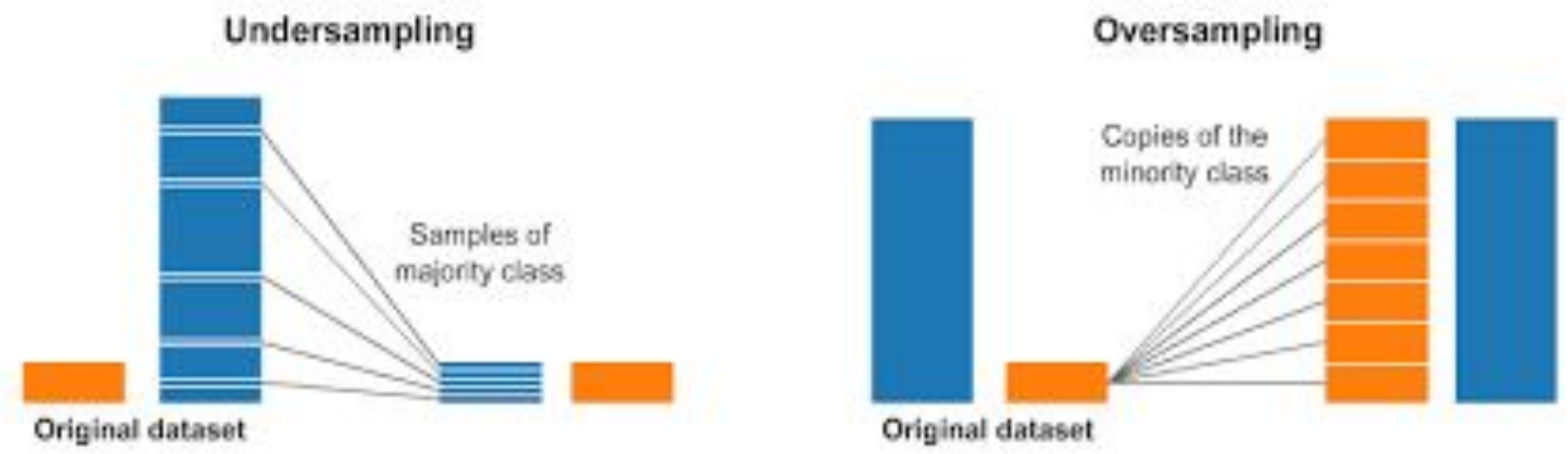
3) **Multi-Label Classification**

- Multi-label is a type of classification problem that may have more than one class label assigned to the data.
- For example, a book or a movie can be categorized into multiple genres, or an image can have multiple objects.

4) **Imbalanced Classification**

- For the imbalanced classification, the number of examples is unevenly distributed in each class, meaning that we can have more of one class than the others
- Some examples of imbalanced classification are disease screening, fraud detection.

Sampling Techniques for imbalanced classification



Techniques aim to balance the distribution of the data

- **Random undersampling:** random elimination of examples from the majority class.

- **SMOTE Oversampling:** random replication of examples from the minority class.

Classification example

Example

- Teacher classify students as A, B, C, D and F based on their marks. The following is one simple classification rule:

$\text{Mark} \geq 90$	:	A
$90 > \text{Mark} \geq 80$	:	B
$80 > \text{Mark} \geq 70$	:	C
$70 > \text{Mark} \geq 60$	:	D
$60 > \text{Mark}$	:	F

Note:

Here, we apply the above rule to a specific data
(in this case a table of marks as a dataset).

Classification

Classification consists of assigning a class label to a set of unclassified cases.

1. **Supervised Classification**

The set of possible classes is known in advance.

2. Unsupervised Classification

Set of possible classes is not known. After classification we can try to assign a name to that class. Unsupervised classification is called clustering.

Why Classification? A motivating application

- Credit approval
  - A bank wants to classify its customers based on whether they are expected to pay back their approved loans
  - The history of past customers is used to train the classifier
  - The classifier provides rules, which identify potentially reliable future customers
  - Classification rule:
    - If age = “31...40” and income = high then credit_rating = excellent
  - Future customers
    - Paul: age = 35, income = high ⇒ excellent credit rating
    - John: age = 20, income = medium ⇒ fair credit rating

Supervised Classification

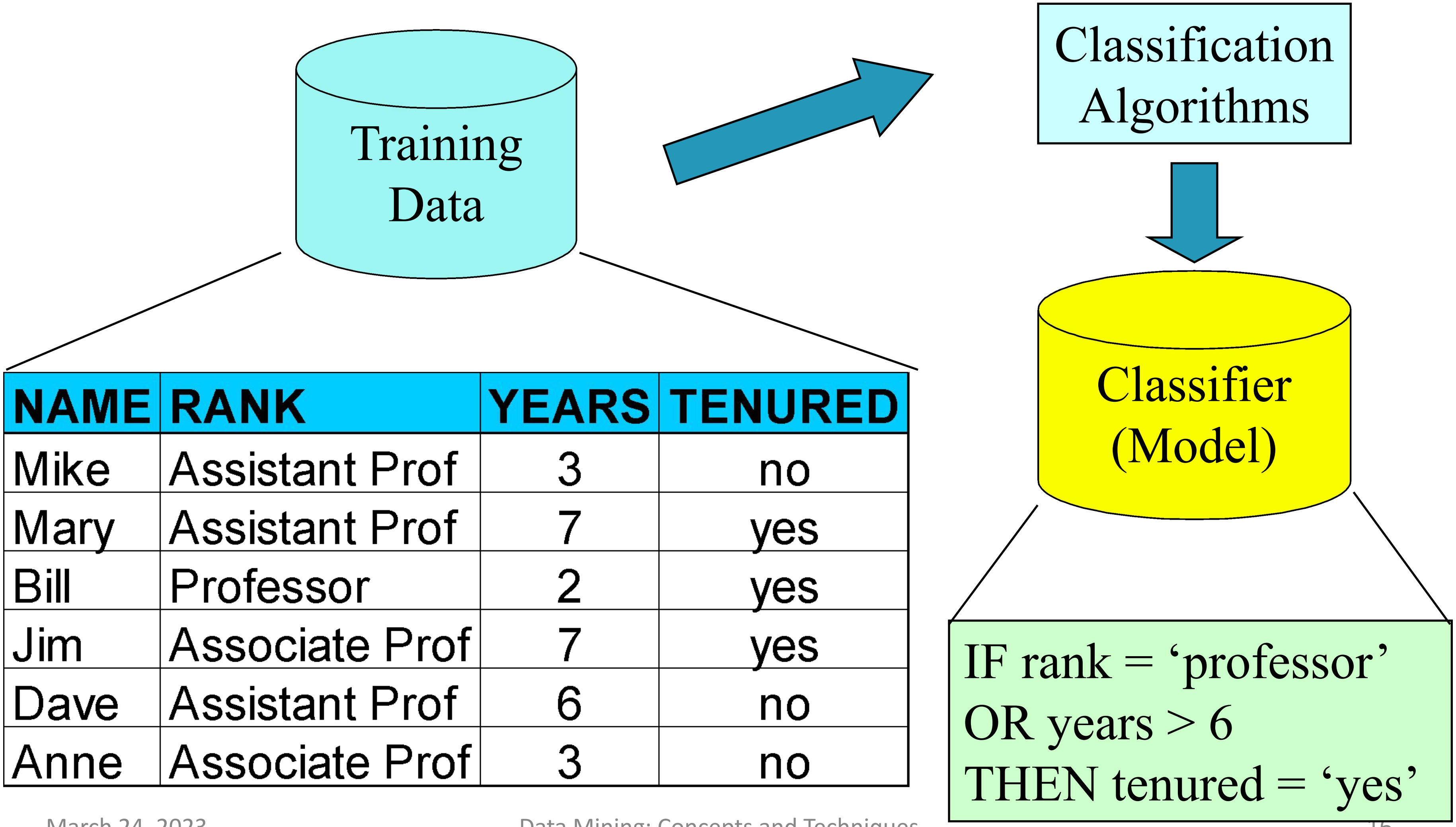
- The input data, also called the training set, consists of multiple records each having multiple attributes or features.
- Each record is tagged with a class label.
- The objective of classification is to analyze the input data and to develop an accurate description or model for each class using the features present in the data.
- This model is used to classify test data for which the class descriptions are not known.

Classification—A Two-Step Process

- Model construction:
  - describing a set of predetermined classes
  - The model is represented as classification rules, decision trees, or mathematical formulae

- Model usage:
  - for classifying future or unknown objects
  - test sample is compared with the classified result from the model

Process (1): Model Construction



Training Data

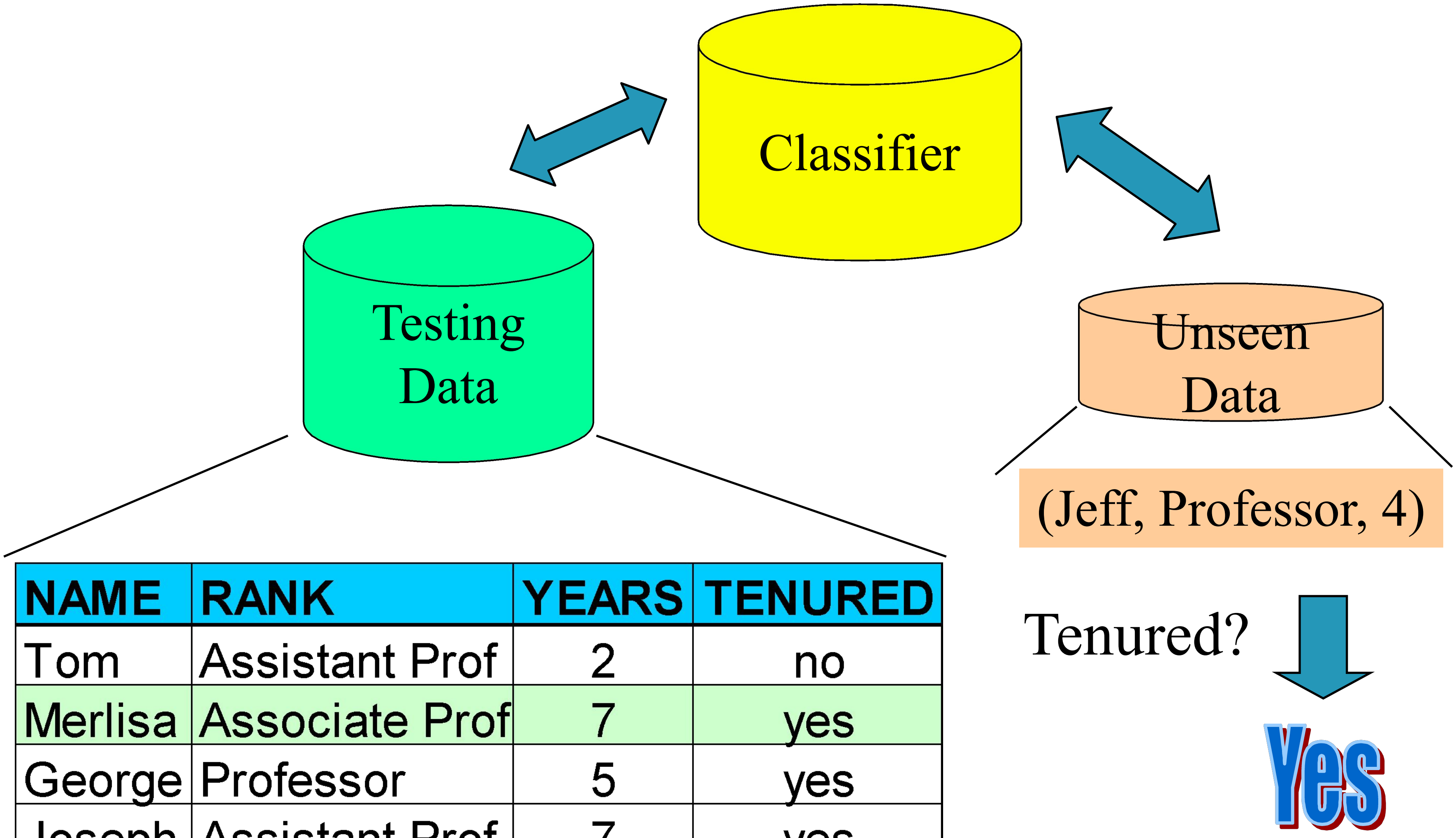
NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

Classification Algorithms

Classifier (Model)

IF rank = ‘professor’
OR years > 6
THEN tenured = ‘yes’

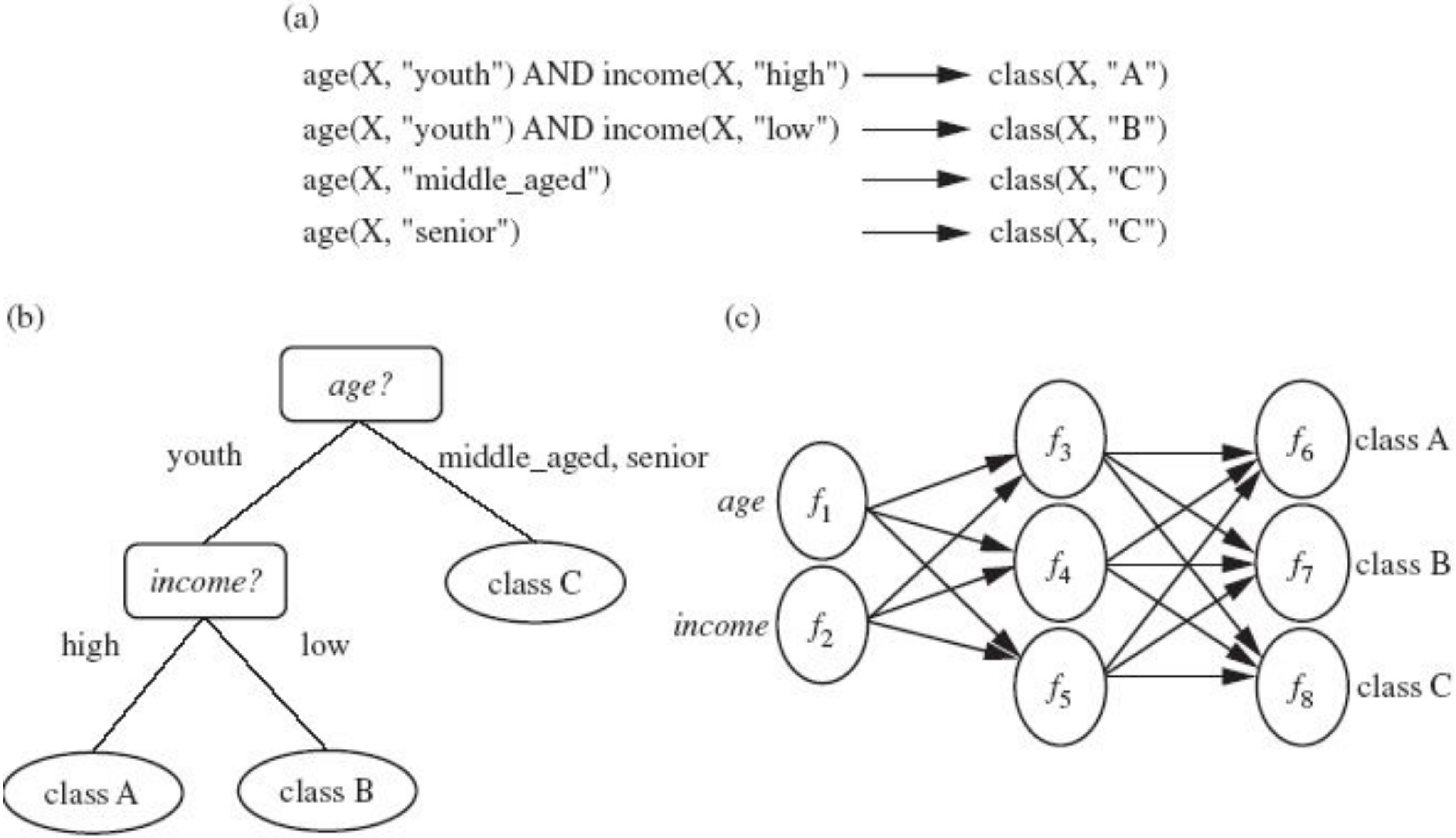
Process (2): Using the Model in Prediction



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

- Classification model can be represented in various forms such as
    - » IF-THEN Rules
    - » A decision tree
    - » Neural network

Classification Model



(a)

age(X, "youth") AND income(X, "high") → class(X, "A")

age(X, "youth") AND income(X, "low") → class(X, "B")

age(X, "middle_aged") → class(X, "C")

age(X, "senior") → class(X, "C")

Bayesian classification

- A **Bayes classifier** is a simple probabilistic classifier
- In simple terms, a naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature.
- For example, a fruit may be considered to be an apple if it is red, round, and about 4" in diameter. Even if these features depend on each other or upon the existence of the other features, a naive Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple
- Bayesian classifier are able to predict class membership probabilities such as the probability that a given tuple belongs to a particular class.

Naïve Bayesian classification

- Naïve Bayes Algorithm (for discrete input attributes)
  - Learning Phase: Given a training set S,

```
For each target value of $c_i$ $(c_i = c_1, \cdots, c_L)$
    $\hat{P}(C = c_i) \leftarrow$ estimate $P(C = c_i)$ with examples in $\mathbf{S}$;
    For every attribute value $x_{jk}$ of each attribute $X_j$ $(j = 1, \cdots, n;\ k = 1, \cdots, N_j)$
        $\hat{P}(X_j = x_{jk} | C = c_i) \leftarrow$ estimate $P(X_j = x_{jk} | C = c_i)$ with examples in $\mathbf{S}$;
```

Output: conditional probability tables; for $X_j, N_j \times L$ elements

  - Test Phase: Given an unknown instance $\mathbf{X}' = (a'_1, \cdots, a'_n)$,

Look up tables to assign the label c^* to $\mathbf{X}'$ if

$$[\hat{P}(a'_1 | c^*) \cdots \hat{P}(a'_n | c^*)]\hat{P}(c^*) > [\hat{P}(a'_1 | c) \cdots \hat{P}(a'_n | c)]\hat{P}(c), \quad c \neq c^*, c = c_1, \cdots, c_L$$

Prior and Posterior Probabilities

-
  - P(A) and P(B) are called prior probabilities
  - P(A|B), P(B|A) are called posterior probabilities

Prior versus Posterior Probabilities

- This table shows that the event Y has two outcomes namely A and B, which is dependent on another event X with various outcomes like x_1, x_2 and x_3.
- **Case1:** Suppose, we don't have any information of the event A. Then, from the given sample space, we can calculate $P(Y = A) = \frac{5}{10} = 0.5$
- **Case2:** Now, suppose, we want to calculate $P(X = x_2|Y = A) = \frac{2}{5} = 0.4$.

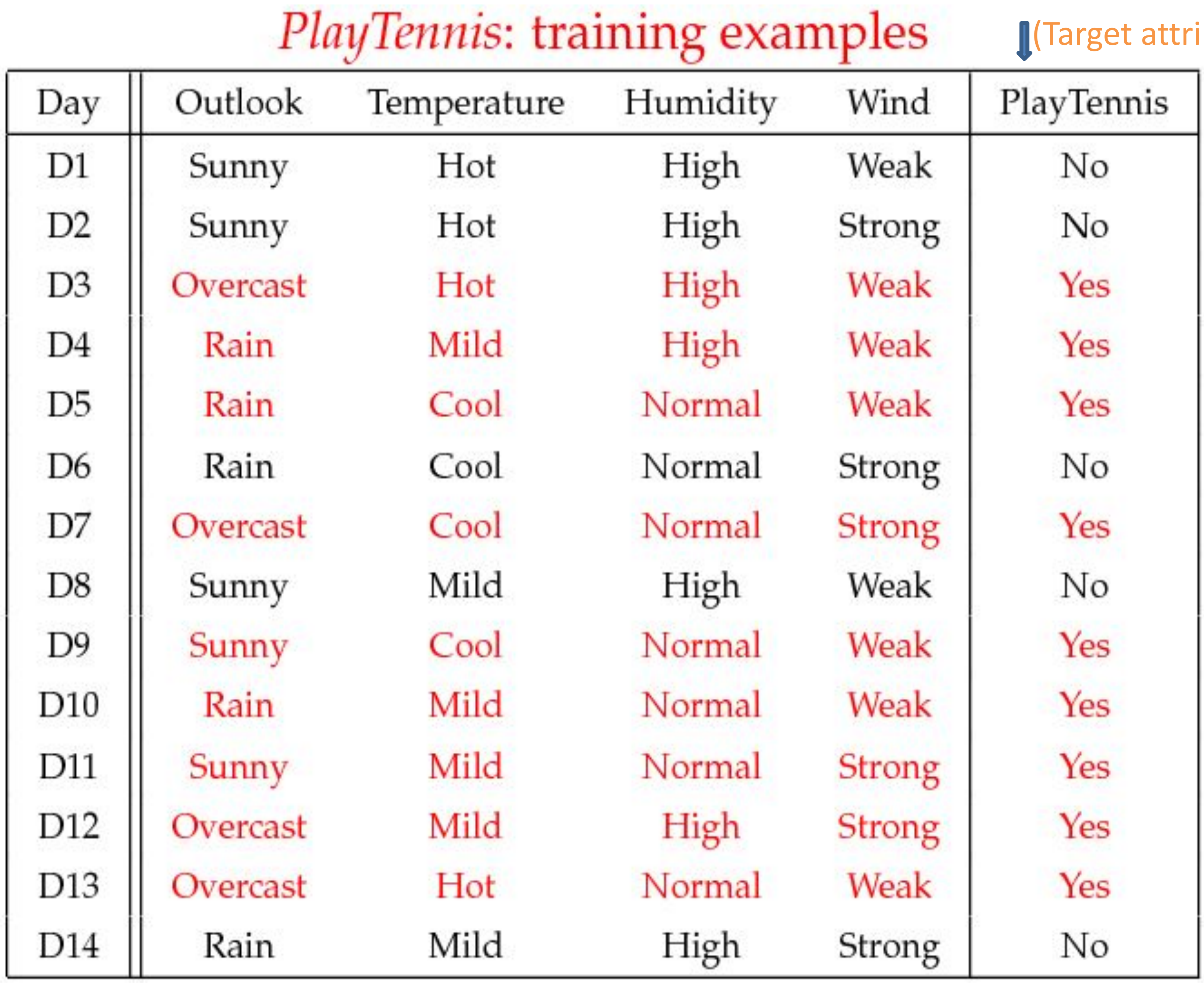
The later is the conditional or posterior probability, where as the former is the prior probability.

X	*Y*
	A
	A
	B
	A
	B
	A
	B
	B
	B
	A

Example 1: Naïve Bayes Classifier Example

Predict the class label for an unknown sample "X" using Naïve Bayesian classification.

'X'= (Outlook=*Sunny,* Temperature=*Cool,* Humidity=*High,* Wind=*Strong*)



PlayTennis: training examples

(Target attribute)

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

Learning phase:

P(Play=*Yes*) = 9/14 (prior probability) *P*(Play=*No*) = 5/14

Outlook	Play=*Yes*	Play=*No*
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

Temperature	Play=*Yes*	Play=*No*
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

Humidity	Play=*Yes*	Play=No
High	3/9	4/5
Normal	6/9	1/5

Wind	Play=*Yes*	Play=*No*
Strong	3/9	3/5
Weak	6/9	2/5

Posterior probability ↑

- Test Phase
  - Given a new instance,

    x’ = (Outlook=*Sunny,* Temperature=*Cool,* Humidity=*High,* Wind=*Strong*)

    - MAP rule

$$P(\mathbf{x}' | Yes) = P(Outlook=Sunny / Yes) *$$
$$P(Temperature=Cool / Yes)*$$
$$P(Humidity=High / Yes) *$$
$$P(Wind=Strong / Yes) *$$
$$P(Yes)$$

= 2/9 * 3/9 * 3/9 * 3/9 * 9/14

= = 0.0053

- Map rule:

$$P(x'|No) = P(Outlook=Sunny/No) * P(Temperature=Cool/No) * P(Humidity=High/No) * P(Wind=Strong/No) * P(No)$$

$= 3/5 * 1/5 * 4/5 * 3/5 * 5/14$

$= 0.0206$

Given the fact P(X|Yes) < P(X|No),

we label X to be “Play tennis = *No*”.

Example 2: Naïve Bayesian classification Example

- Predict a class label of an unknown sample using Naïve Bayesian classification on the following training dataset from all electronics customer database.
- The unknown sample is_

X’={age=“<=30, Income=“median”, Student=“yes”, credit rating=“fair”}

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

- P(**x**’|Yes) =0.028
- P(**x**’|No)=0.007
- Since 0.028>0.007, therefor the naïve Bayesian classifier predicts buyes computer=“yes” for sample X’

- Weka classifer demo naïve bay

https://www.youtube.com/watch?v=UzT4W1tOKD4

Decision Tree - Classification

- Decision tree builds classification models in the form of a tree structure.
- It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed.
- The final result is a tree with **decision nodes** and **leaf nodes**.
- A decision node has two or more branches
- Leaf node represents a classification or decision.
- The topmost decision node in a tree which corresponds to the best predictor called **root node**.
- Decision trees can handle both categorical and numerical data.

How to determine the Best Split

- Greedy approach:
  - Creation of nodes with homogeneous class distribution is preferred
- Need a measure of node impurity:

C0: 5
C1: 5

Non-homogeneous,

High degree of impurity

C0: 9
C1: 1

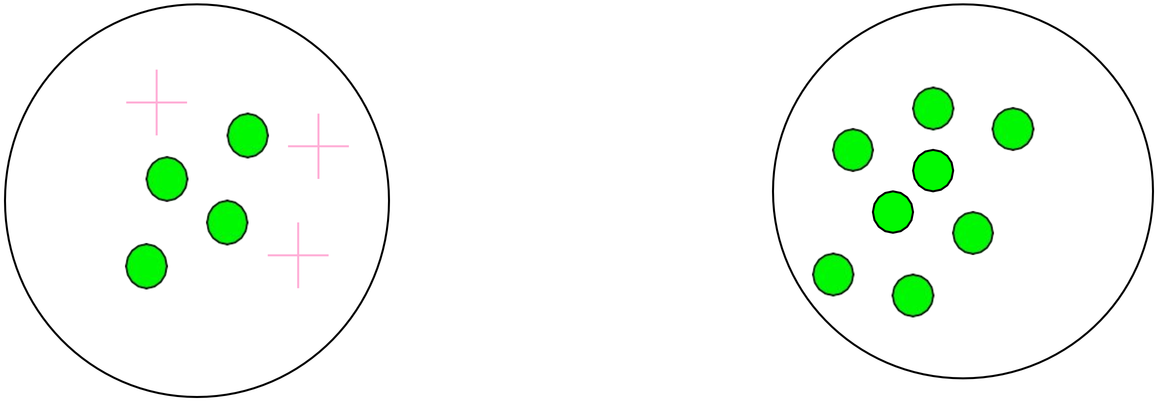
Homogeneous,

Low degree of impurity

Information Gain

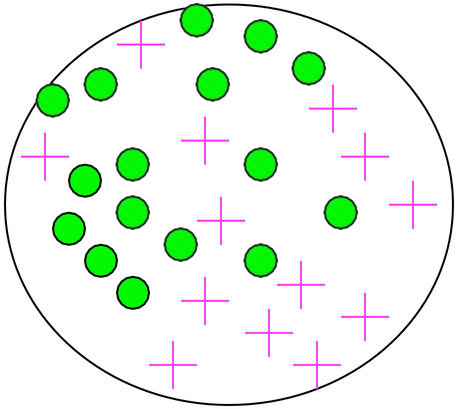
Entropy

– Measures the level of **impurity** in a group of examples

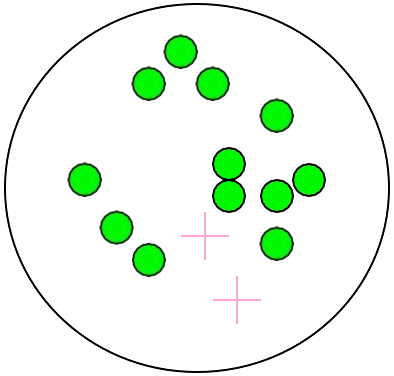


Impurity

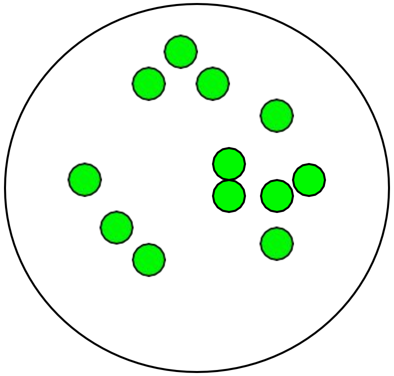
Very impure group



Less impure



Minimum impurity



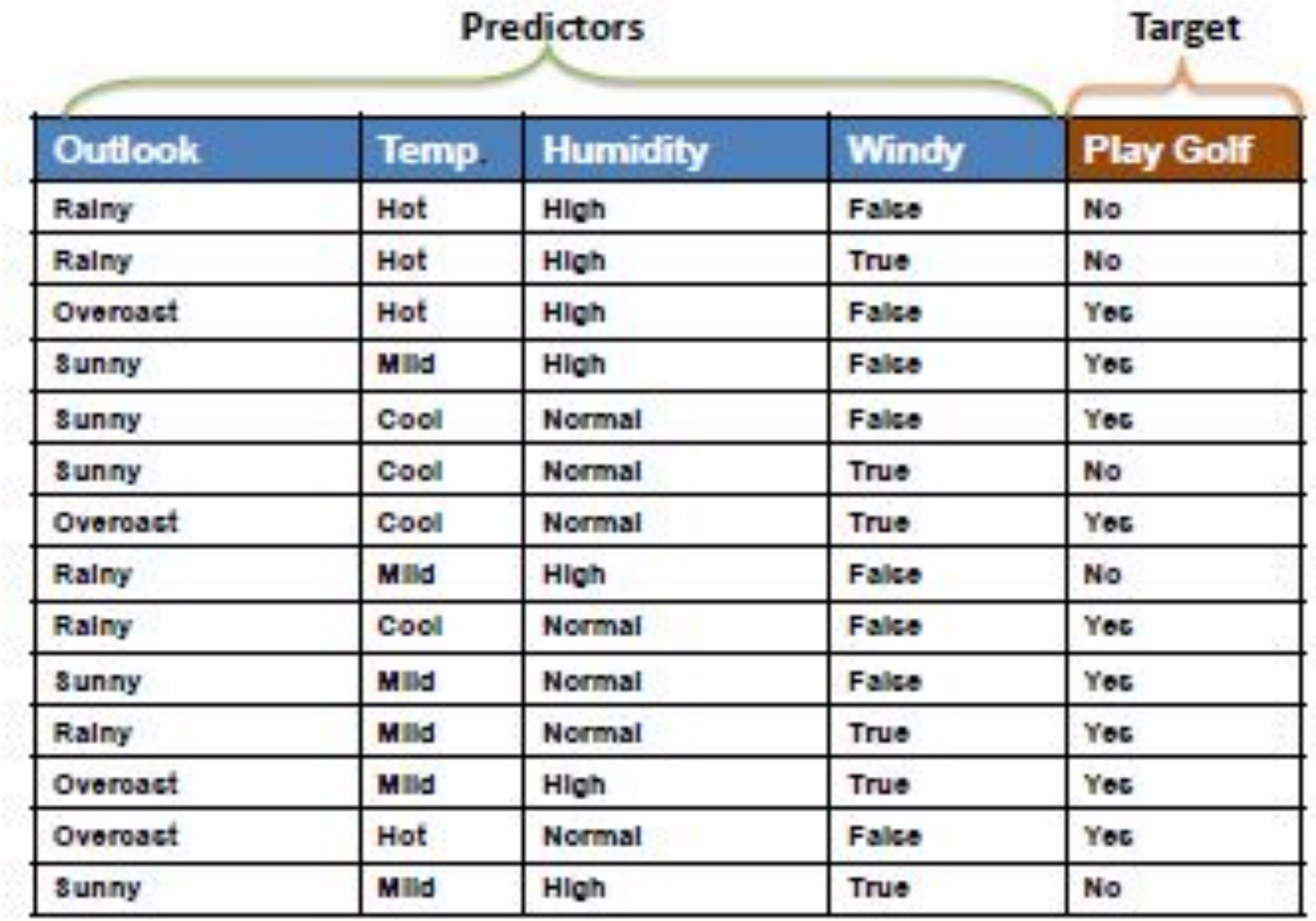
Based on slide by Pedro Domingos

Which is the best-split attribute?

- In a decision tree, the attribute with the highest information gain or the lowest impurity is usually chosen to be the attribute used at the root node and most useful for splitting the data

- Alternatively, other impurity metrics such as Gini impurity or misclassification error can also be used to determine the best attribute to use at the root node.

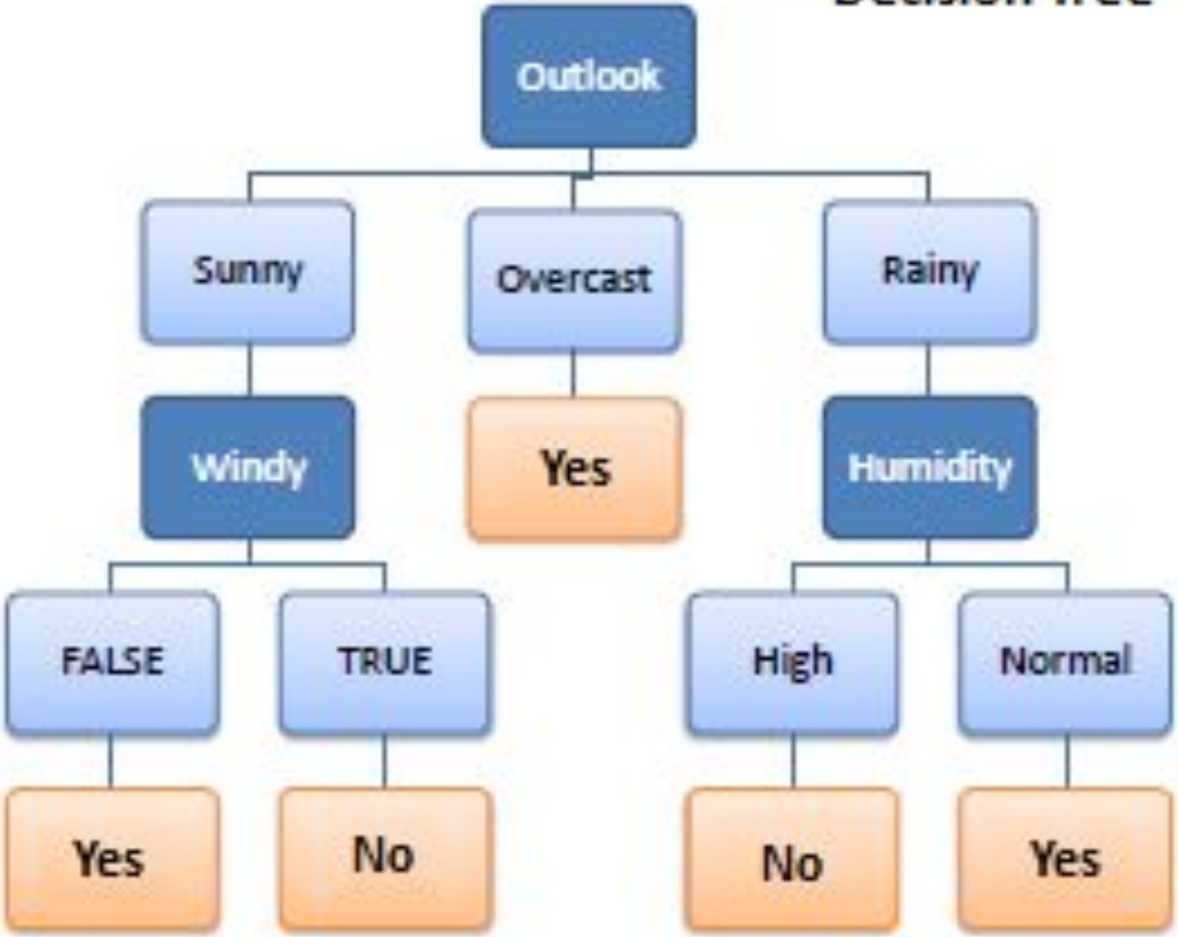
- The attribute with the lowest impurity or the highest reduction in impurity is usually chosen.


Predictors
Target

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

Example 1 : using given training data set, create classification model using decision tree

Decision Tree



$$E(S)=\sum_{i=1}^{c}-p_i\log_2 p_i$$

Play Golf	
Yes	No
9	5

Entropy(PlayGolf) = Entropy (5,9)

= Entropy (0.36, 0.64)

= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)

= 0.94

$$E(T,X)=\sum_{c\in X}P(c)E(c)$$

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14

E(PlayGolf, Outlook) = **P**(Sunny)*$\mathbf{E}$(3,2) + **P**(Overcast)*$\mathbf{E}$(4,0) + **P**(Rainy)*$\mathbf{E}$(2,3)

= (5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971

= 0.693

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
Gain = 0.029			

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1
Gain = 0.152			

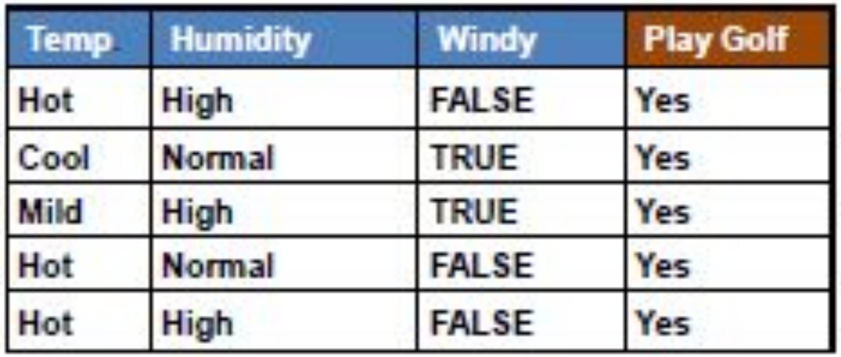
		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Gain = 0.048			

$$Gain(T,X) = Entropy(T) - Entropy(T,X)$$

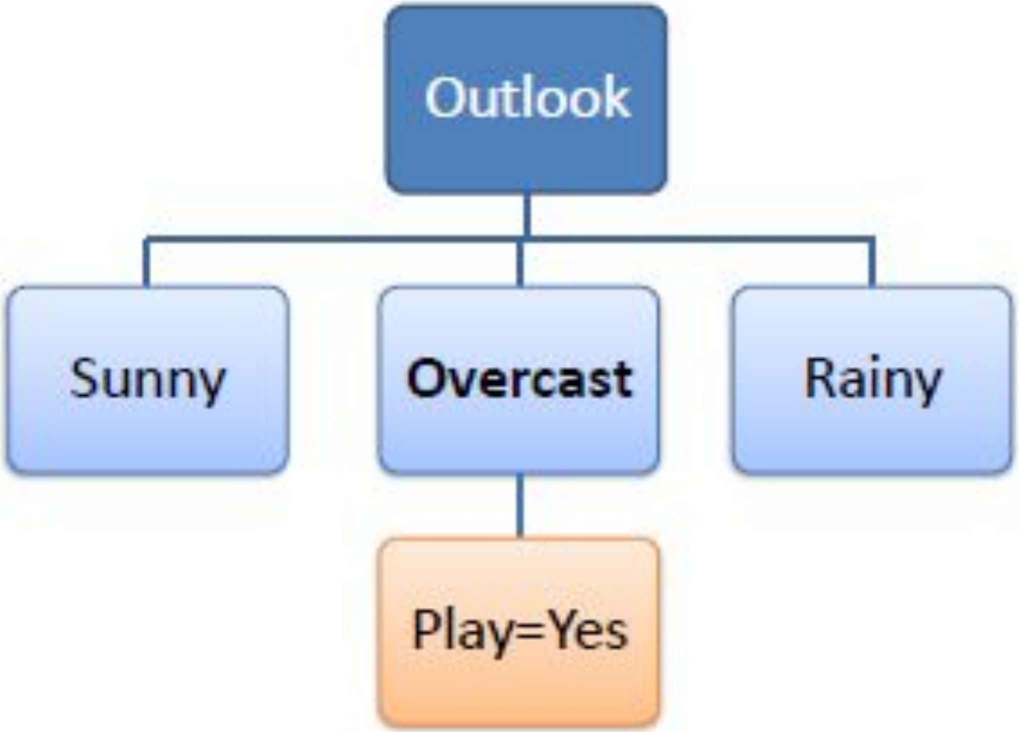
G(PlayGolf, Outlook) = **E**(PlayGolf) – **E**(PlayGolf, Outlook)

= 0.940 – 0.693 = 0.247

Outlook= Overcast



Temp	Humidity	Windy	Play Golf
Hot	High	FALSE	Yes
Cool	Normal	TRUE	Yes
Mild	High	TRUE	Yes
Hot	Normal	FALSE	Yes
Hot	High	FALSE	Yes



Outlook

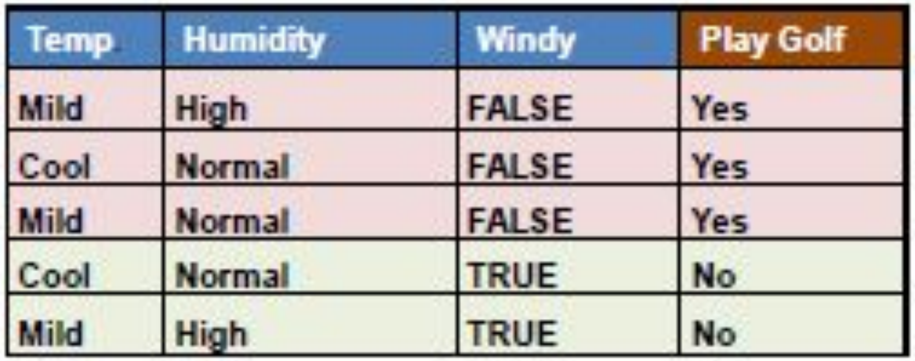
Sunny

Overcast

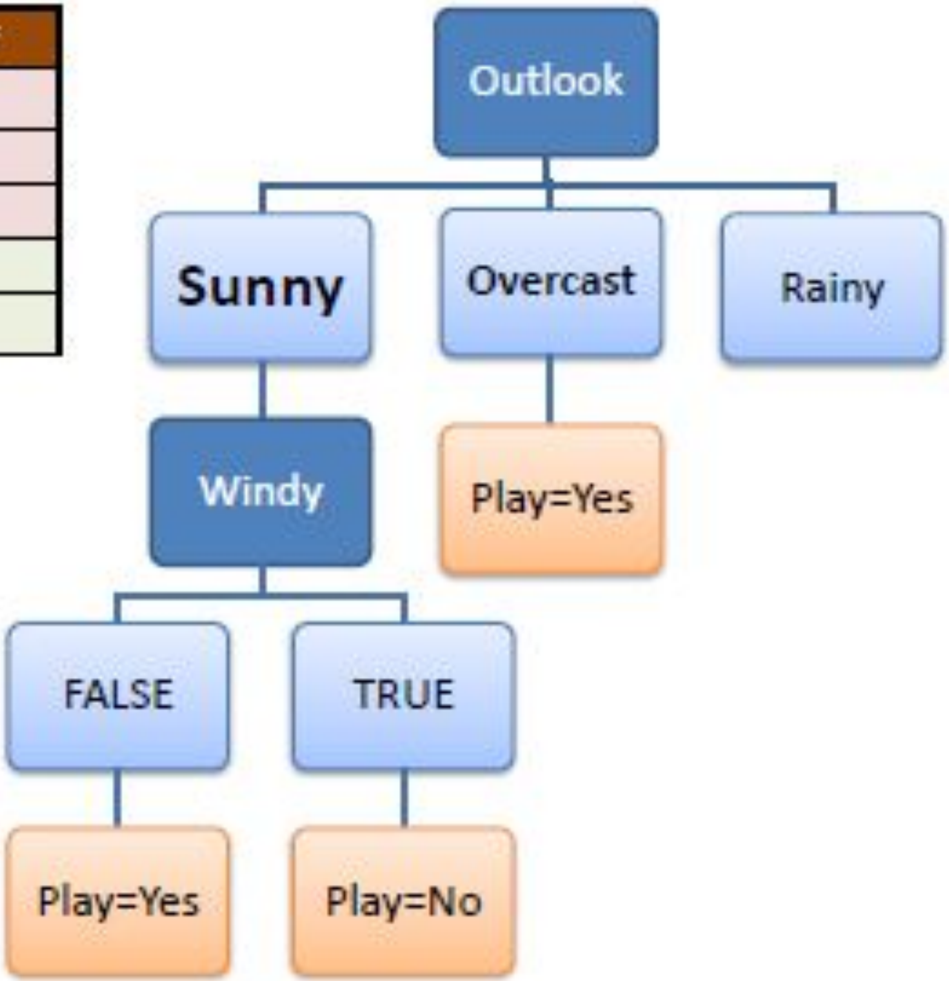
Rainy

Play=Yes

Outlook= Sunny



Temp	Humidity	Windy	Play Golf
Mild	High	FALSE	Yes
Cool	Normal	FALSE	Yes
Mild	Normal	FALSE	Yes
Cool	Normal	TRUE	No
Mild	High	TRUE	No



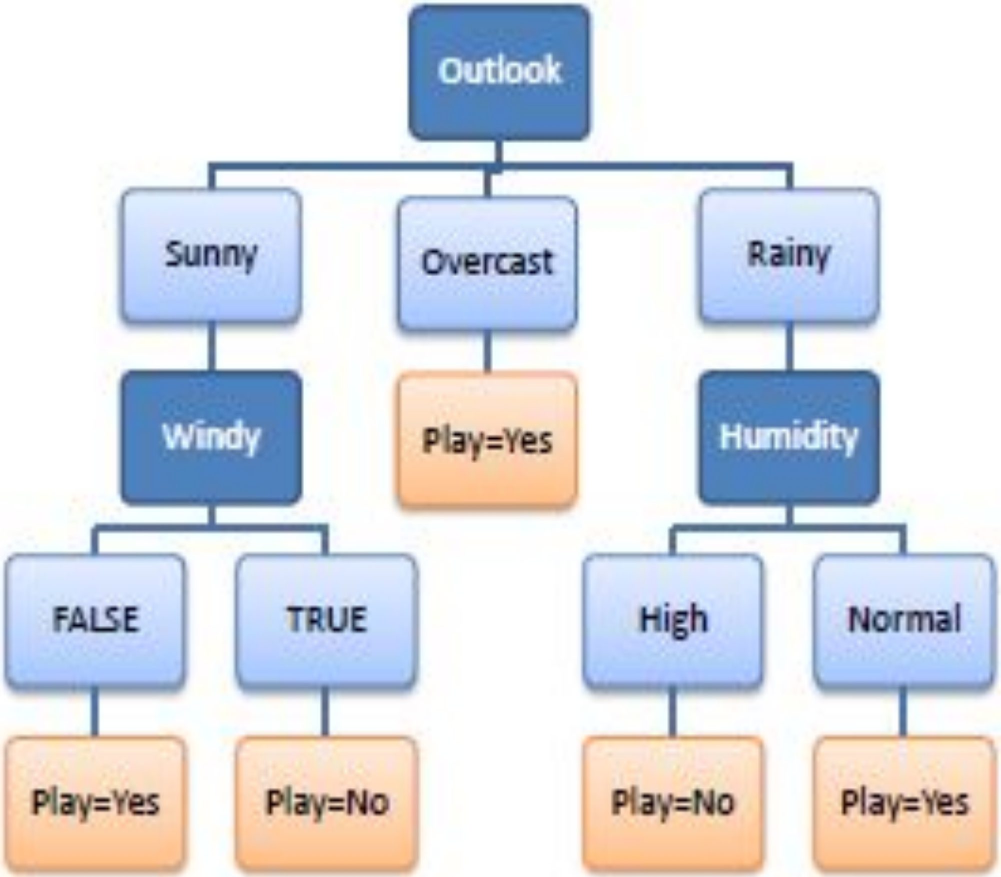
R_1: IF (Outlook=Sunny) AND (Windy=FALSE) THEN Play=Yes

R_2: IF (Outlook=Sunny) AND (Windy=TRUE) THEN Play=No

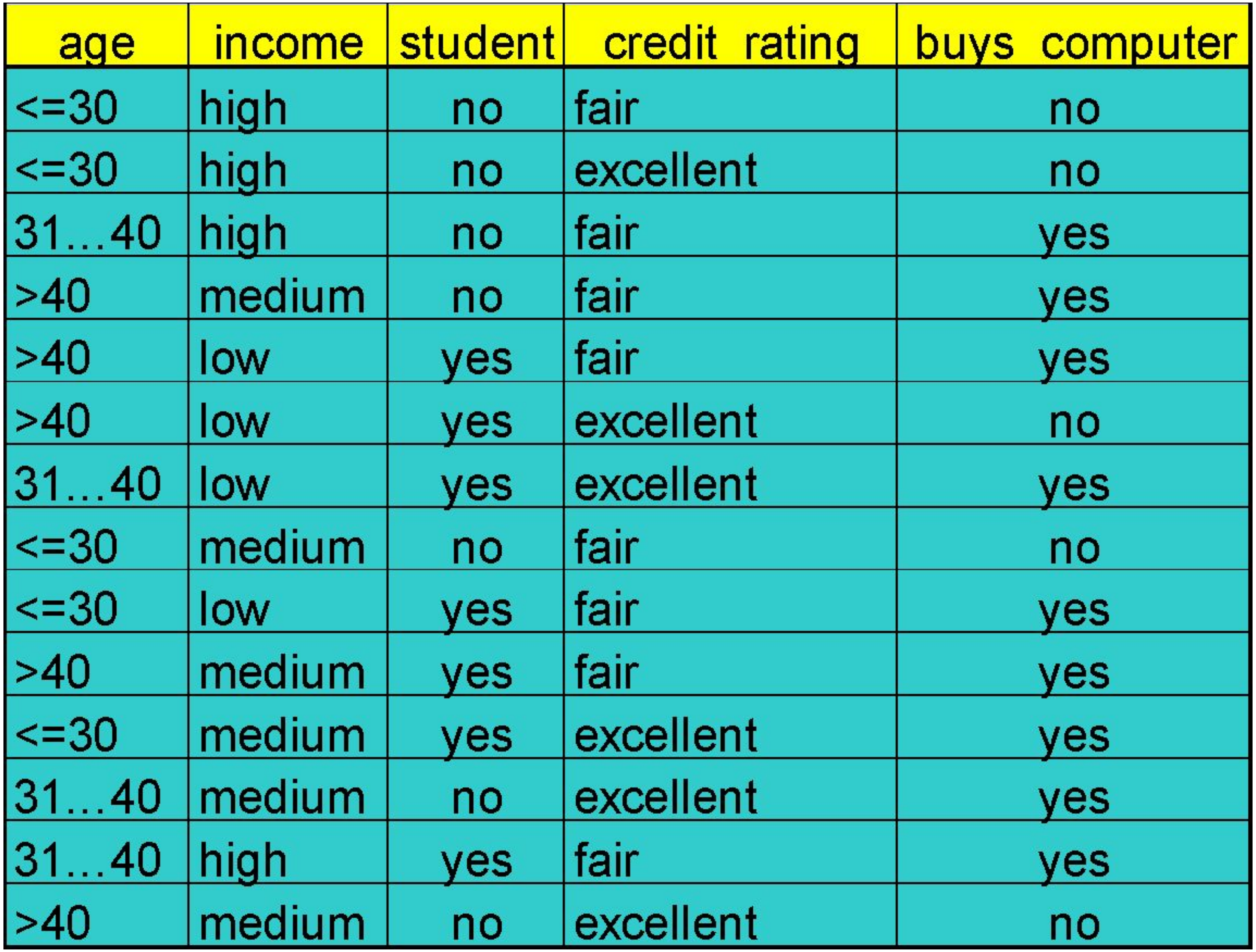
R_3: IF (Outlook=Overcast) THEN Play=Yes

R_4: IF (Outlook=Rainy) AND (Humidity=High) THEN Play=No

R_5: IF (Outlook=Rain) AND (Humidity=Normal) THEN Play=Yes

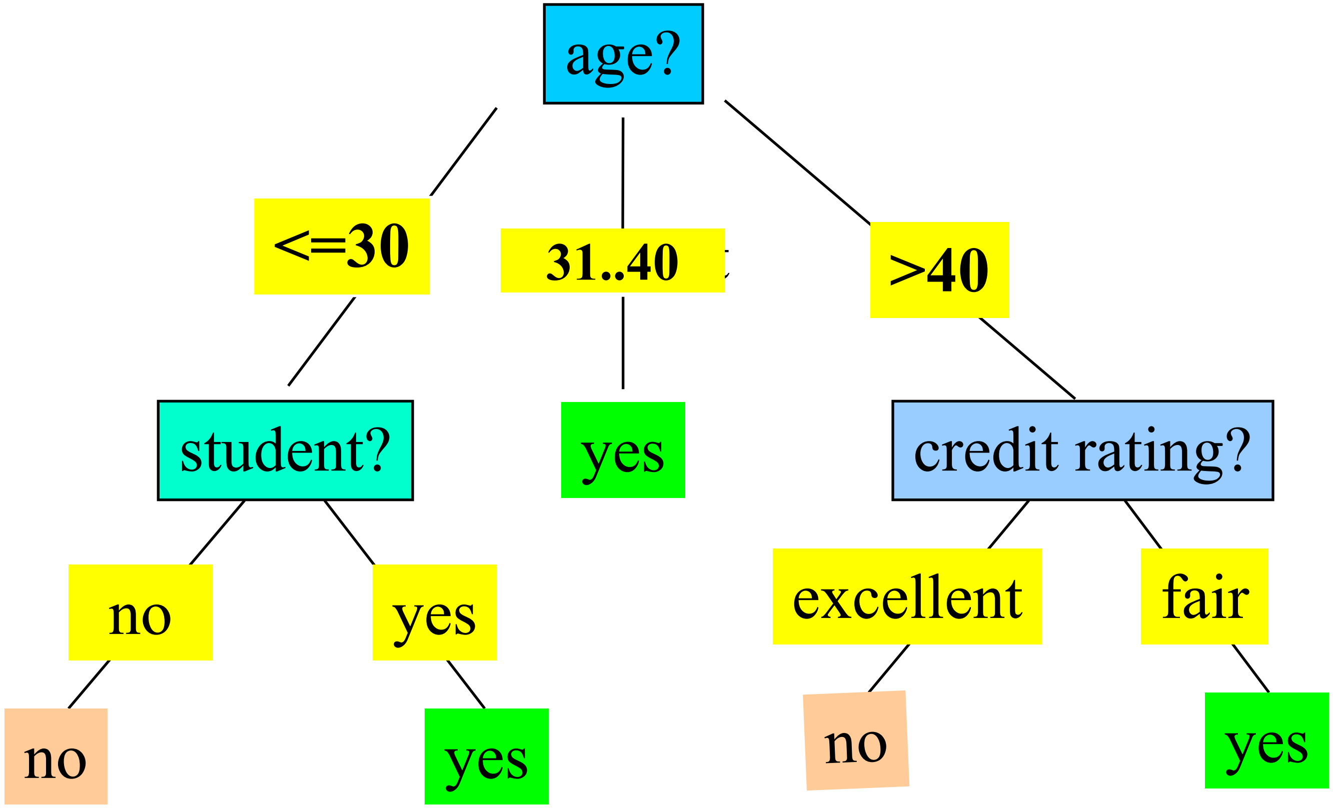


Example 2: using given training data set, create classification model using decision tree



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

Output: A Decision Tree for “*buys_computer*”



Example 3: Create classification model using decision tree for the following training data set

Sr. No.	Income	Age	Own House
1	Very high	Young	yes
2	High	Medium	yes
3	Low	Young	Rented
4	High	Medium	Yes
5	Very high	Medium	yes
6	Medium	Young	yes
7	High	Old	yes
8	Medium	Medium	Rented
9	low	medium	Rented
10	Low	Old	Rented
11	High	Young	yes
12	Medium	old	Rented

Total no. of records=12,

yes=7, rented=5

Entropy(own house) = E([7/12],[5/12])

=E(0.58, 0.41)

= - (0.58 $\log_2$ 0.58) - (0.42$\log_2$ 0.42)

= 0.98

$$E(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$$

- Step1: for E(Income) we have $E(T,X)=\sum_{c\in X} P(c)E(c)$

Income	Yes	rented	total
Very high	2	0	2
High	4	0	4
Low	0	3	3
medium	1	2	3
			12

E(Own house, Income) = p(vh)*E(vh) + p(h)*E(h) + p(l)*E(l) + p(m)*E(m)

E(O,I) = [2/12*E(2/2,0/2)]+ [4/12*E(4/4,0/4)]+ [3/12*E(0/3,3/3)] + [3/12*E(1/3,2/3)]

$E(O,I) = (3/12) * E([1/3],[2/3])$

$= 0.25 * [- (0.33 \log_2 0.33) - (0.67 \log_2 0.67)]$

$= 0.25 * 0.92$

$= 0.23$

- Gain(O,I) = E(O) – E(O,I)
  = 0.98 – 0.23
  = 0.75

- **Information Gain**

  The information gain is based on the decrease in entropy after a dataset is split on an attribute. Constructing a decision tree is all about finding attribute that returns the highest information gain (i.e., the most homogeneous branches).

$$Gain(T,X) = Entropy(T) - Entropy(T,X)$$

Step 2: For E(Age) we have,

Age	Yes	Rented	total
Young	3	1	4
Medium	3	2	5
Old	1	2	3
			12

$$E(\text{Own house, Age}) = p(y)*E(y) + p(m)*E(m) + p(o)*E(o)$$

$$= [(4/12)*E(3/4, 1/4)] + [(5/12)*E(3/5,2/5)] + [(3/12)*E(1/3,2/3)]$$

$$= [(4/12)*E(0.75,0.25)] + [(5/12)*E(0.6,0.4)] + [(3/12)*E(0.33,0.67)]$$

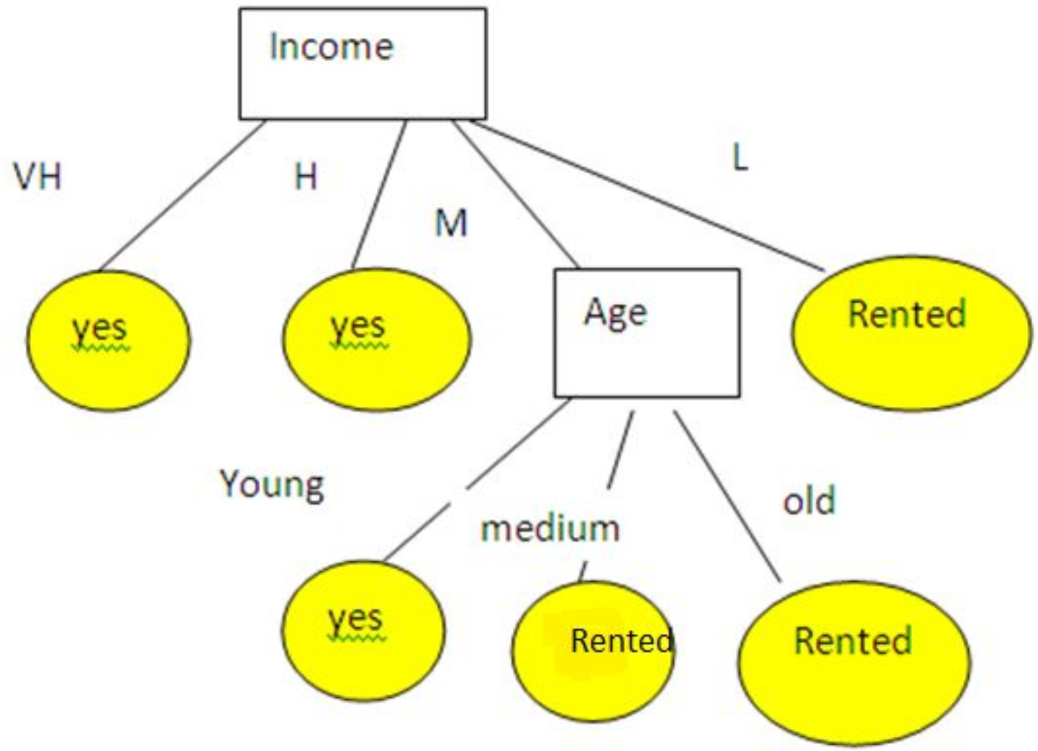
$$= 0.90$$

- $G(O,A) = E(O) – E(O,A)$

  $= 0.98 – 0.90$

  $= 0.08$

  Income attribute has highest gain, so used as a decision attribute in the root node



R1: If(Income=VH) then Own house=yes

R2: If (Income=Medium) and (Age=old) then own house=rented

For income medium we have three values □

Income	Age	Own house
Medium	Young	yes
Medium	Medium	Rented
Medium	old	Rented

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
- 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

Algorithm for Decision Tree Induction

- Basic algorithm (a greedy algorithm)
  - Tree is constructed in a top-down recursive divide-and-conquer manner
  - At start, all the training examples are at the root
  - Attributes are categorical (if continuous-valued, they are discretized in advance)
  - Samples are partitioned recursively based on selected attributes
  - Test attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain)
- Conditions for stopping partitioning
  - 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

Algorithm for Decision Tree Induction (pseudocode)

```
Algorithm GenDecTree(Sample S, Attlist A)
1. create a node N
2. If all samples are of the same class C then label N with C; terminate;
3. If A is empty then label N with the most common class C in S (majority voting); terminate;
4. Select a∈A, with the highest information gain; Label N with a;
5. For each value v of a:
    a. Grow a branch from N with condition a=v;
    b. Let $S_v$ be the subset of samples in S with a=v;
    c. If $S_v$ is empty then attach a leaf labeled with the most common class in S;
    d. Else attach the node generated by GenDecTree($S_v$, A-a)
```

Attribute Selection Measure

- Information gain (ID3/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

Information Gain (ID3/C4.5)

- Select the attribute with the highest information gain
- Assume there are two classes, *P* and *N*
  - Let the set of examples *S* contain *p* elements of class *P* and *n* elements of class *N*
  - The amount of information, needed to decide if an arbitrary example in *S* belongs to *P* or *N* is defined as

$$I(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

Prediction

- In prediction we can predict the continues values of response variable with the help of predictor variable
- Prediction can be done with the help of statistical technique of **regression**
- It assumes the data to fit in some kind of function and involve study of those function
- Most widely used approach for numeric prediction is regression

Regression

Regression can be of following kind

- Linear regression
- Multiple linear regression
- Non-linear regression

Linear regression(single predictor variable)

- Data is modeled using straight line
- This regression line is represented by following expression

$$y = \alpha + \beta x$$

Where x □ predictor variable

Y□ response variable

Multiple linear regression(multiple predictor variable)

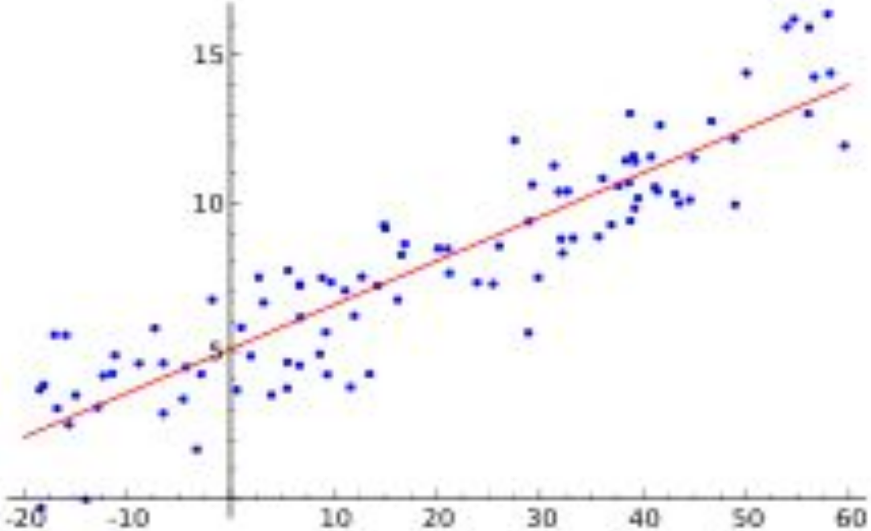
$$y = \alpha + \beta 1x1 + \beta 2x2 + \beta 3x3$$

- In linear and multiple linear regression always predictor and response variable got the linear relationship

Non-linear regression :

- If the given response variable and predictor variable have got polynomial relationship then it is called non-linear regression

$$y = \alpha + \beta 1x + \beta 2x^2 + \beta 3x^3$$


15
10
5
-20
-10
10
20
30
40
50
60

How α and β is calculated?

- using **least square method**

$$\beta = \frac{\sum_{i=1}^{n}(Xi - \overline{X}) * (Yi - \overline{Y})}{\sum_{i=1}^{n}(Xi - \overline{X})^2}$$

$$\alpha = \overline{y} - \beta\overline{x}$$

$\overline{X}$□ mean value of x

$\overline{Y}$□ mean value of y

Example

The below table shows the marks obtain by student in midterm and final year exam

Midterm(x)	Final year(y)
45	60
70	70
60	54
84	82
75	68
84	76

Find:

1)Equation of predication(linear Regression formula)

2) What will be the final year marks if the midterm marks is 40?

- Solution:
- compute

$$\beta = \frac{\sum_{i=1}^{n}(Xi - \overline{X}) * (Yi - \overline{Y})}{\sum_{i=1}^{n}(Xi - \overline{X})^2}$$

$$\alpha = \overline{y} - \beta\overline{x}$$

x	y	$(x-\bar{x})$	$(y-\bar{y})$	$(x-\bar{x})(y-\bar{y})$	$(x-\bar{x})^2$
45	60				
70	70				
60	54				
84	82				
75	68				
84	76				
				Σ = 648.62	Σ= 1141.36

$$\beta = \frac{\sum_{i=1}^{n}(Xi - \overline{X}) * (Yi - \overline{Y})}{\sum_{i=1}^{n}(Xi - \overline{X})^2}$$

β =0.558

$$\alpha = \overline{y} - \beta\overline{x}$$

α =28.84

- Now, $y = \alpha + \beta x$

- So, prediction equation□ Y=28.84 + 0.558X

- For X=40, we get Y=52

This is the final year marks for students getting 40 marks in midterm.