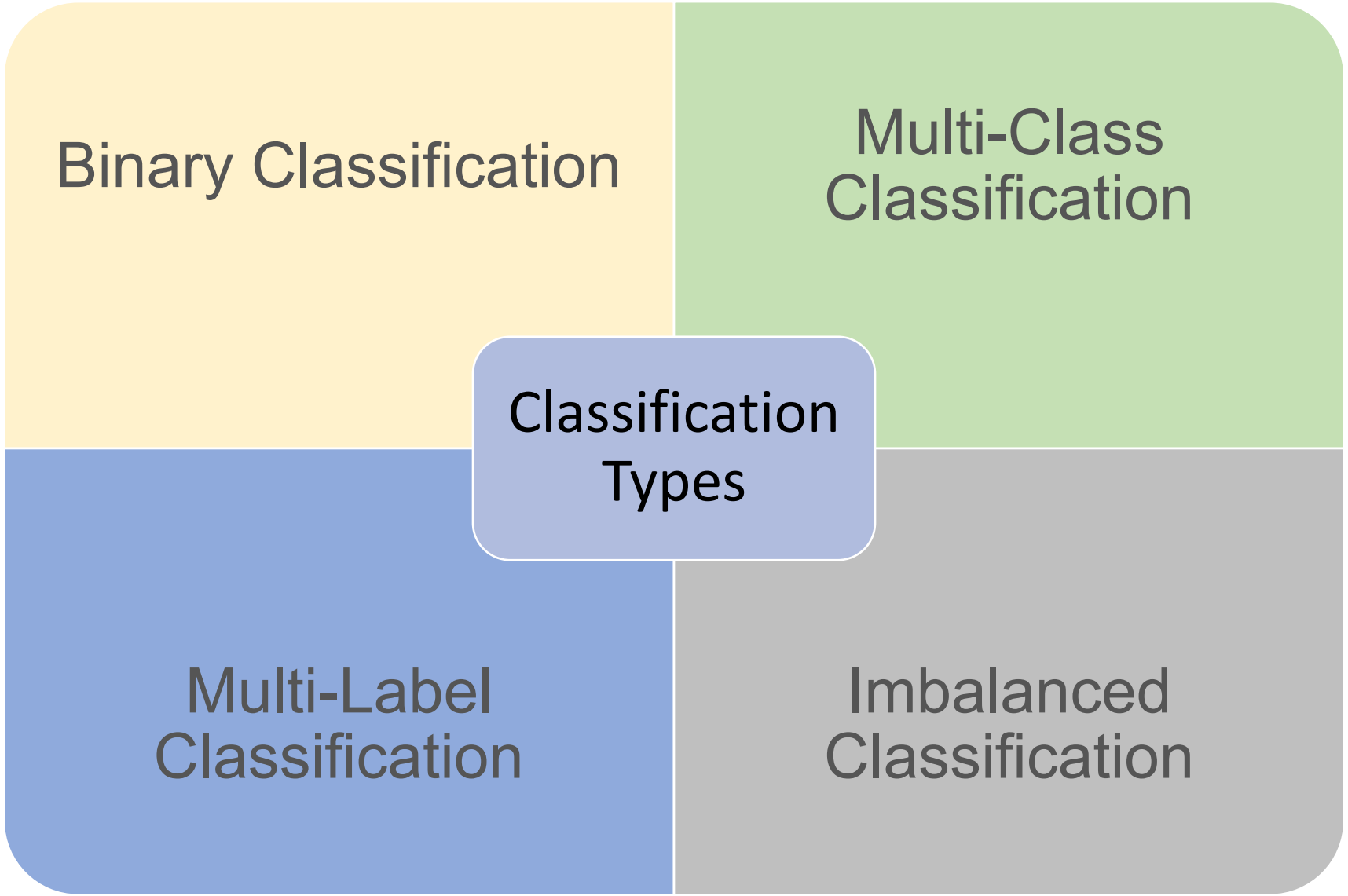
Classification—A Two-Step Process

- 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


Binary Classification
Multi-Class Classification
Classification Types
Multi-Label Classification
Imbalanced Classification

Classification Types

Binary Classification:

- Binary classification refers to those classification tasks that have two class labels.
- Examples include:
  - Email spam detection (spam or not).
  - Conversion prediction (buy or not).

Popular algorithms :

- Logistic Regression
- k-Nearest Neighbors
- Decision Trees
- Support Vector Machine
- Naive Bayes

Classification Types

Multi-Class Classification

- Multi-class classification refers to those classification tasks that have more than two class labels.
- Examples include:
  - Face classification.
  - Plant species classification.
  - Optical character recognition.

Popular algorithms :

- k-Nearest Neighbors.
- Decision Trees.
- Naive Bayes.
- Random Forest.
- Gradient Boosting.

Classification Types

Multi-Label Classification

- Multi-label classification refers to those classification tasks that have two or more class labels, where one or more class labels may be predicted for each example.
- Examples include:
  - photo classification, where a given photo may have multiple objects in the scene and a model may predict the presence of multiple known objects in the photo, such as "*bicycle*," "*apple*," "*person*," etc

Popular algorithms :

- Multi-label Decision Trees
- Multi-label Random Forests
- Multi-label Gradient Boosting

Classification Types

Imbalanced Classification

- Imbalanced classification refers to classification tasks where the number of examples in each class is unequally distributed.

- Examples include:
  - Fraud detection.
  - Outlier detection.
  - Medical diagnostic tests.

Popular algorithms :

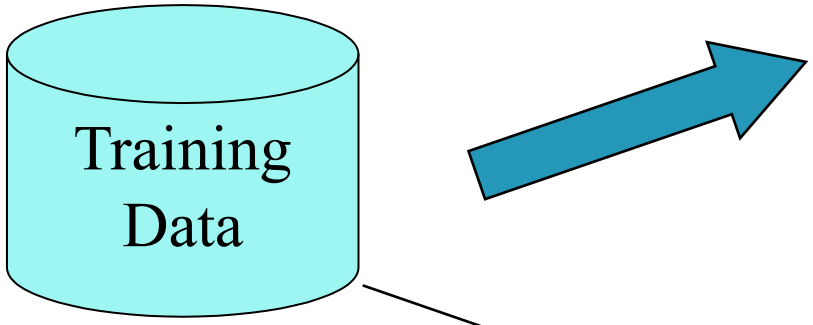
- Cost-sensitive Logistic Regression.
- Cost-sensitive Decision Trees.
- Cost-sensitive Support Vector Machines.

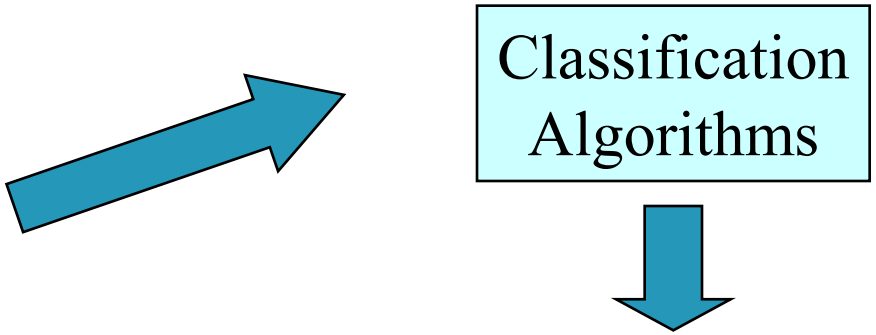
Classification—A Two-Step Process

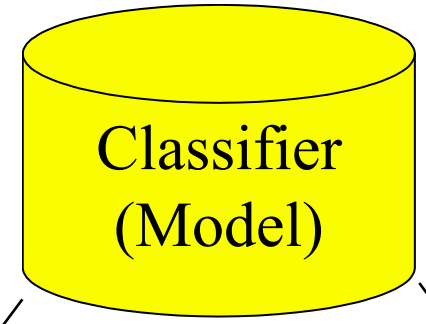
- Model construction:
  - describing a set of predetermined classes
  - The model is represented as classification rules, decision trees, or mathematical formulae

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

Process (1): Model Construction



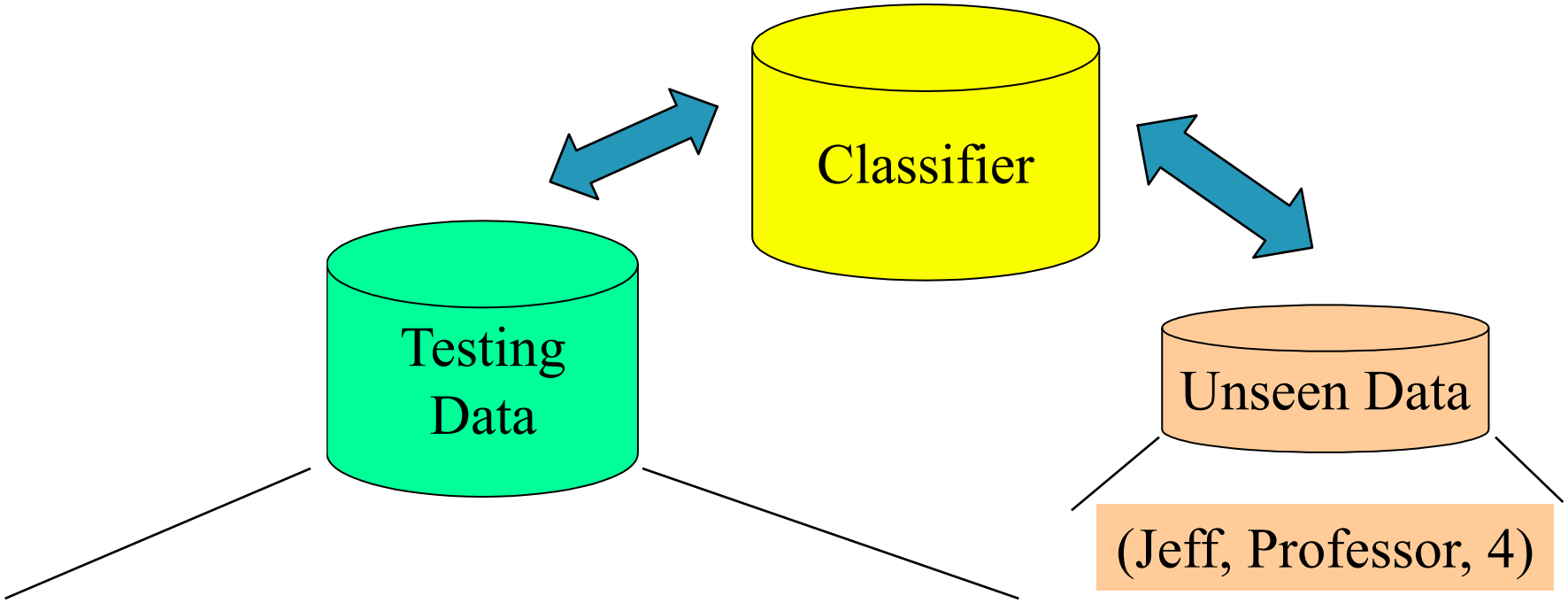




NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

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)

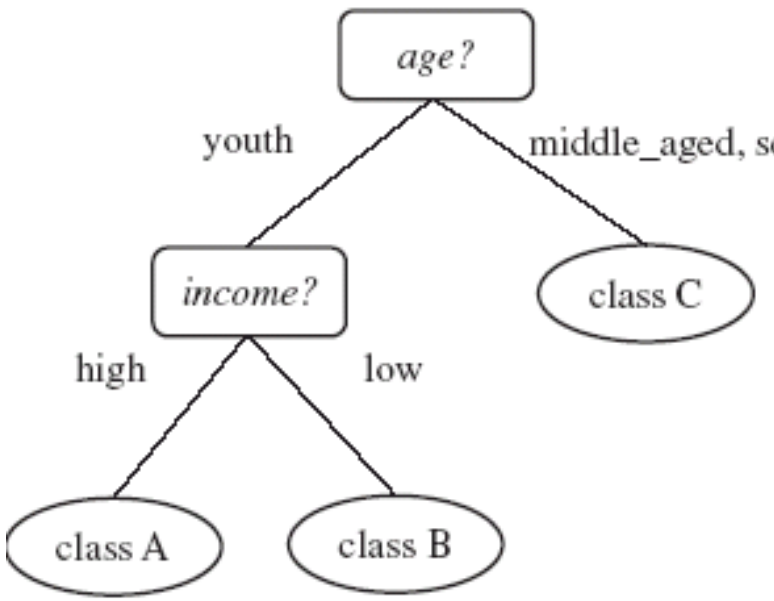
age(X, "youth") AND income(X, "high") ⟶ class(X, "A")

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

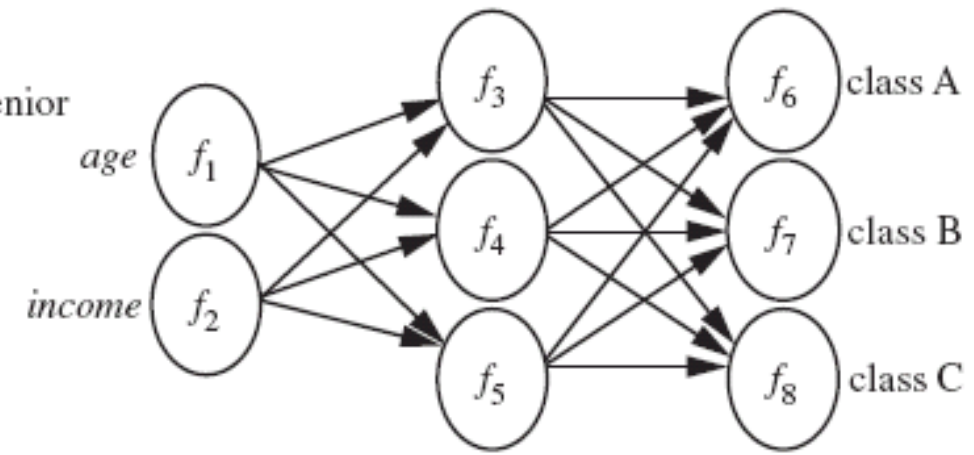
age(X, "middle_aged") ⟶ class(X, "C")

age(X, "senior") ⟶ class(X, "C")

(b)



(c)



Classification Techniques

- A number of classification techniques are known, which can be broadly classified into the following categories:

1. Statistical-Based Methods
   - Regression
   - Bayesian Classifier
2. Distance-Based Classification
   - K-Nearest Neighbours
3. Decision Tree-Based Classification
   - ID3, C 4.5, CART
5. Classification using Machine Learning (SVM)
6. Classification using Neural Network (ANN)

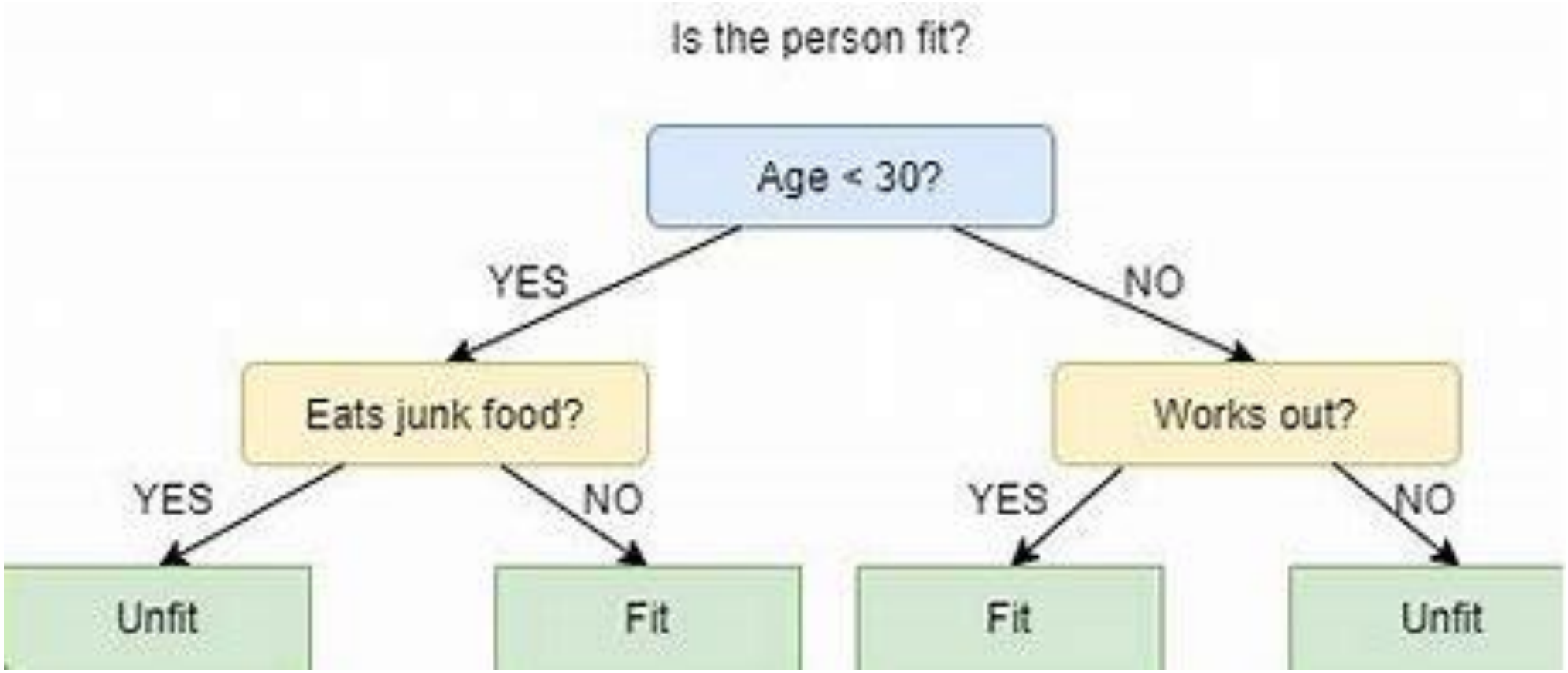
Decision Tree Algorithm

By Namdeo Badhe

Outline

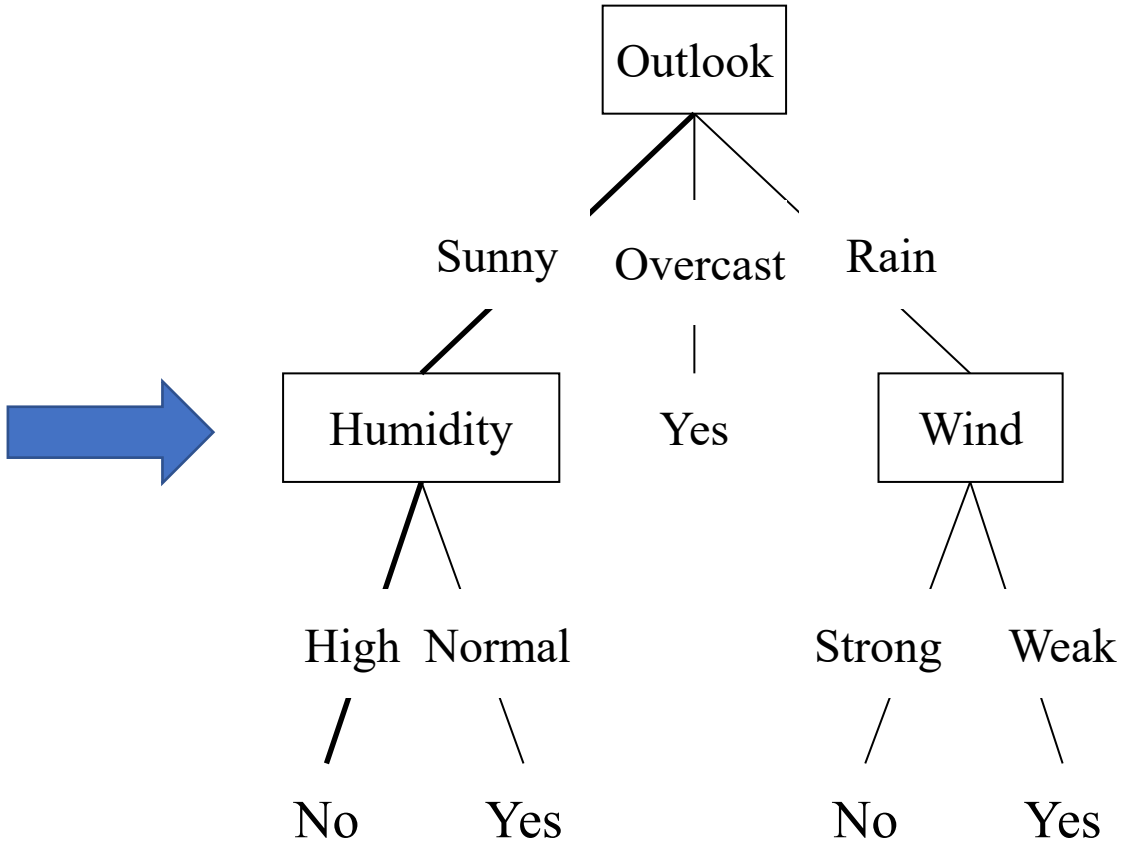
- Introduction to Decision Tree
- Examples of Decision Tree
- Decision Tree Algorithms
- ID3(Iterative Dichotomiser 3) Algorithm

Decision Tree



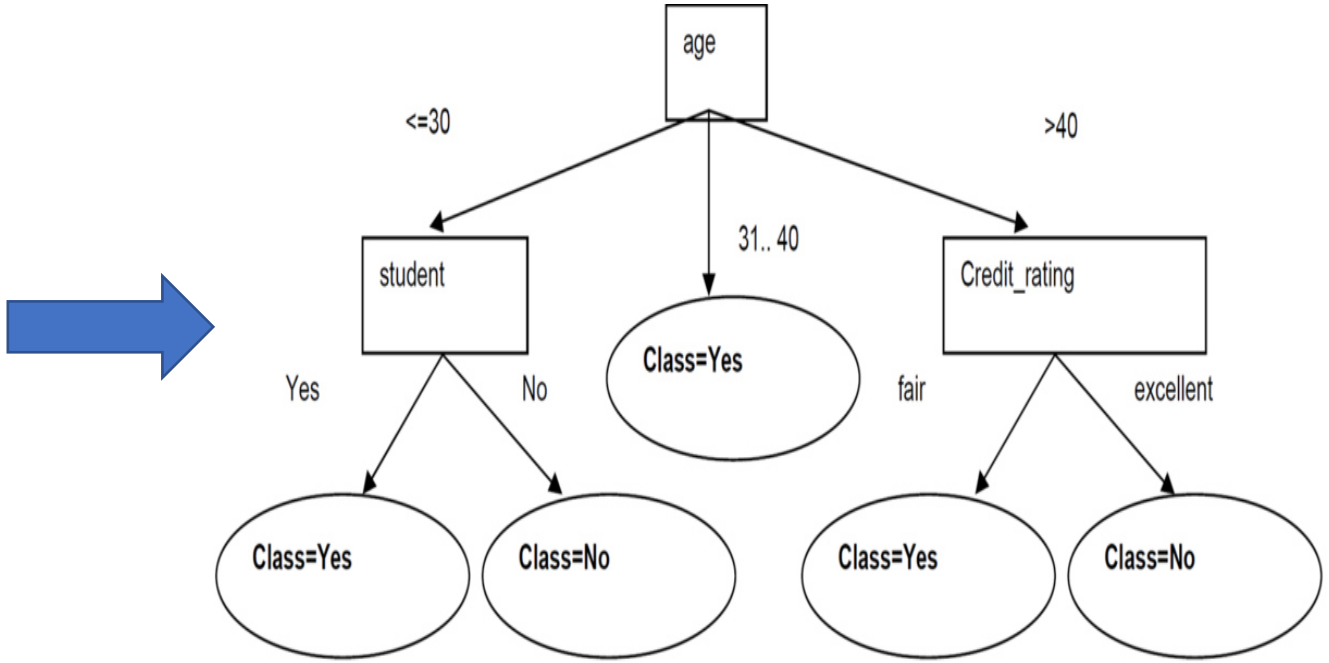
Examples of Decision Tree

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
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



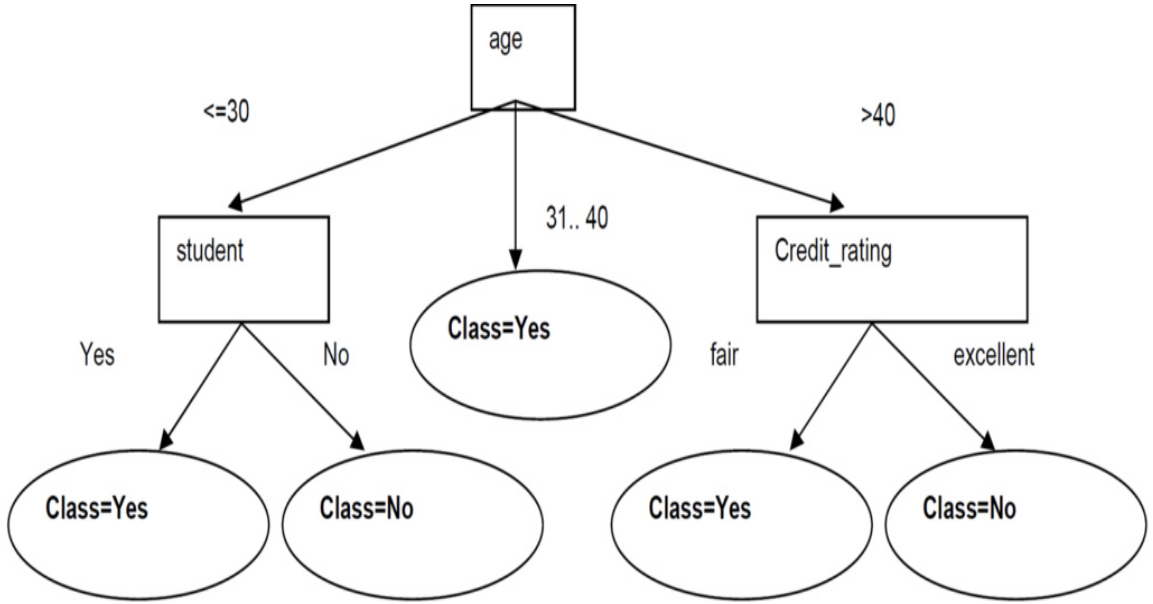
Examples of Decision Tree

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



How are decision trees used for classification?

- Given a tuple, X, for which the associated class label is unknown,
- For Example
- **If X={age<=30,student=yes class=?}**
- Decision trees can easily be converted to classification rules



How does the Decision Tree algorithm Work?

- **Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.
- **Step-2:** Find the best Splitting attribute in the dataset using **Attribute Selection Measure (ASM).**
- **Step-3:** Divide the S into subsets that contains possible values for the best attributes.
- **Step-4:** Generate the decision tree node, which contains the best attribute.
- **Step-5:** Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

Attribute Selection Measure (ASM)

- Information Gain
- Gain Ratio
- Gini Index

Popular Decision Tree algorithms

- ID3(Iterative Dichotomiser 3)
- C4.5
- CART(Classification and Regression Trees)

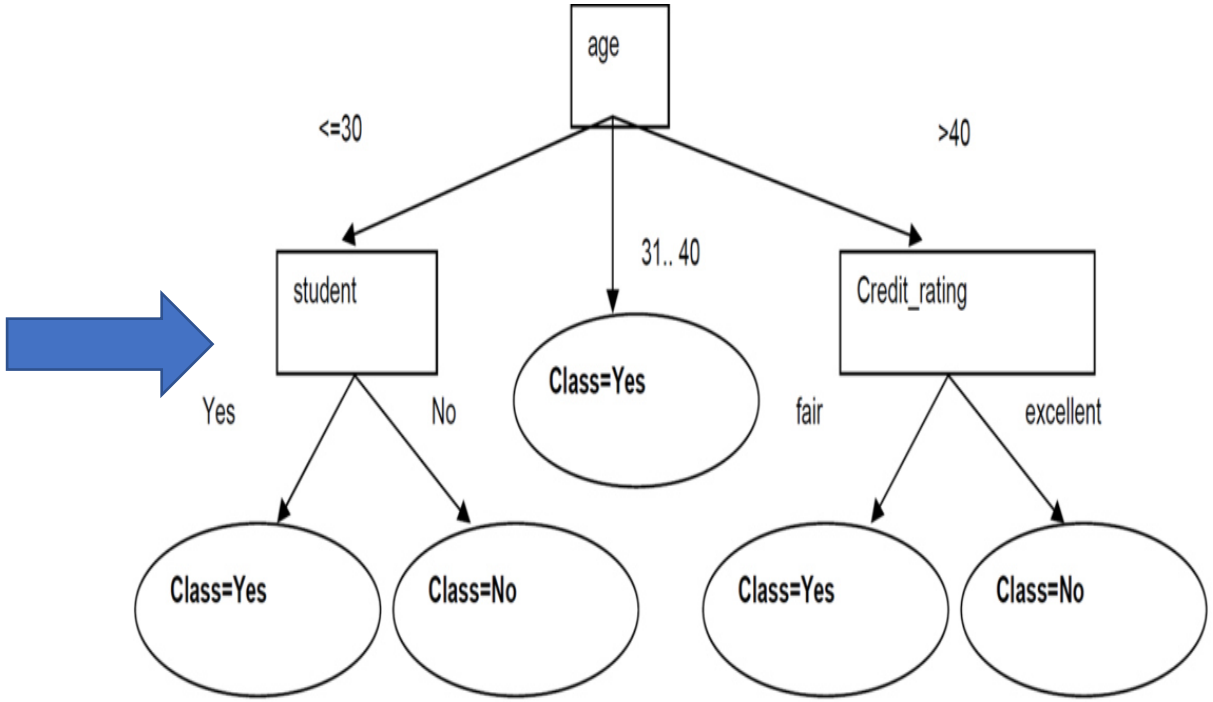
ID3 (Iterative Dichotomiser 3)

- ID3 is one of the earliest decision tree algorithms and was developed in the late 1970s.
- It uses the information gain metric to determine the best feature to split on at each node of the tree.
- ID3 is a greedy algorithm, No Backtracking .

ID3 Algorithm

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

- **Problem Definition:**
- Build a **decision tree** using ID3 algorithm for the given training data in the table (Buy Computer data), and predict the class of the following new example: **age<=30, income=medium, student=yes, credit-rating=fair**



ID3 Algorithm

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

- **Problem Definition:**
- Build a **decision tree** using ID3 algorithm for the given training data in the table (Buy Computer data), and predict the class of the following new example: **age<=30, income=medium, student=yes, credit-rating=fair**

Metrics used in ID3 Algorithm are

Information Gain = 1 – Entropy

$Entropy(S) = -\sum p(I) \cdot \log_2 p(I)$

Entropy is a measure of the amount of uncertainty in the dataset S.

$Entropy(A) = \sum [\, p(S|A) \cdot Entropy(S|A) \,]$

$Gain(S, A) = Entropy(S) - Entropy(A)$

Information Gain(S,A) tells us how much uncertainty in S was reduced after splitting set S on attribute A

ID3 Algorithm

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

Metrics used in ID3 Algorithm are

$$\text{Entropy}(S) = \sum - p(I) \cdot \log_2 p(I)$$

$$\text{Entropy}(A) = \sum [\, p(S|A) \cdot \text{Entropy}(S|A) \,]$$

$$\text{Gain}(S, A) = \text{Entropy}(S) - \text{Entropy}(A)$$

$$\text{Entropy}(S) = E(9,5) = -9/14 \log_2(9/14) - 5/14 \log_2(5/14) = 0.94$$

ID3 Algorithm

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

$$\text{Entropy}(S) = \sum - p(I) \cdot \log_2 p(I)$$

$$\text{Entropy}(A) = \sum [\, p(S|A) \cdot \text{Entropy}(S|A) \,]$$

$$\text{Gain}(S, A) = \text{Entropy}(S) - \text{Entropy}(A)$$

$$\text{Entropy}(S) = E(9,5) = -9/14 \log_2(9/14) - 5/14 \log_2(5/14) = 0.94$$

Now Consider the Age attribute

For Age, we have three values

1. $age_{<=30}$ (2 yes and 3 no),
2. $age_{31..40}$ (4 yes and 0 no),
3. $age_{>40}$ (3 yes and 2 no)

$$\text{Entropy}(age) = 5/14\,(-2/5 \log_2(2/5) - 3/5 \log_2(3/5)) + 4/14\,(0) + 5/14\,(-3/5 \log_2(3/5) - 2/5 \log_2(2/5))$$

$$= 5/14(0.9709) + 0 + 5/14(0.9709) = 0.6935$$

$$\text{Gain}(age) = 0.94 - 0.6935 = 0.2465$$

ID3 Algorithm

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

$Entropy(S) = \sum - p(I) \cdot \log_2 p(I)$

$Entropy(A) = \sum [\, p(S|A) \cdot Entropy(S|A) \,]$

$Gain(S, A) = Entropy(S) - Entropy(A)$

$Entropy(S) = E(9,5) = -9/14 \log_2(9/14) - 5/14 \log_2(5/14) = 0.94$

Now Consider the Income Attribute

For Income, we have three values

1. $income_{high}$ (2 yes and 2 no)
2. $income_{medium}$ (4 yes and 2 no)
3. $income_{low}$ (3 yes 1 no)

$Entropy(income) = 4/14(-2/4\log_2(2/4)-2/4\log_2(2/4)) + 6/14 (-4/6\log_2(4/6)-2/6\log_2(2/6)) + 4/14 (-3/4\log2(3/4)-1/4\log2(1/4))$

$= 4/14\ (1) + 6/14\ (0.918) + 4/14\ (0.811)$

$= 0.285714 + 0.393428 + 0.231714 = 0.9108$

$Gain(income) = 0.94 - 0.9108 = 0.0292$

ID3 Algorithm

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

$$Entropy(S) = \sum - p(I) \cdot \log_2 p(I)$$

$$Entropy(A) = \sum [\, p(S|A) \cdot Entropy(S|A) \,]$$

$$Gain(S, A) = Entropy(S) - Entropy(A)$$

$$Entropy(S) = E(9,5) = -9/14 \log_2(9/14) - 5/14 \log_2(5/14) = 0.94$$

Now Consider the Student Attribute

For Income, we have two values

1. $student_{yes}$ (6 yes and 1 no)
2. $student_{no}$ (3 yes 4 no)

$$Entropy(student) = 7/14(-6/7\log_2(6/7) - 1/7\log_2(1/7)) + 7/14(-3/7\log_2(3/7) - 4/7\log_2(4/7))$$
$$= 7/14(0.5916) + 7/14(0.9852)$$
$$= 0.2958 + 0.4926 = 0.7884$$

$$Gain(student) = 0.94 - 0.7884 = 0.1516$$

ID3 Algorithm

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

$Entropy(S) = \sum - p(I) \cdot \log_2 p(I)$

$Entropy(A) = \sum [\, p(S|A) \cdot Entropy(S|A) \,]$

$Gain(S, A) = Entropy(S) - Entropy(A)$

$Entropy(S) = E(9,5) = -9/14 \log_2(9/14) - 5/14 \log_2(5/14) = 0.94$

Now Consider the Credit_Rating Attribute

For Income, we have two values

1. credit_rating fair (6 yes and 2 no)
2. credit_rating excellent (3 yes 3 no)

$Entropy(credit\_rating) = 8/14(-6/8\log_2(6/8) - 2/8\log_2(2/8)) + 6/14(-3/6\log_2(3/6) - 3/6\log_2(3/6))$

$= 8/14(0.8112) + 6/14(1)$

$= 0.4635 + 0.4285 = 0.8920$

$Gain(credit\_rating) = 0.94 - 0.8920 = 0.0479$

ID3 Algorithm

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

$$Entropy(S) = \sum - p(I) \cdot \log_2 p(I)$$

$$Entropy(A) = \sum [\, p(S|A) \cdot Entropy(S|A) \,]$$

$$Gain(S, A) = Entropy(S) - Entropy(A)$$

$$Entropy(S) = E(9,5) = -9/14 \log_2(9/14) - 5/14 \log_2(5/14) = 0.94$$

Gain(age) = 0.94 – 0.6935 = 0.2465

Gain(income) = 0.94 – 0.9108 = 0.0292

Gain (student) = 0.94 – 0.7884 = 0.1516

Gain(credit_rating) = 0.94 – 0.8920 = 0.0479

ID3 Algorithm

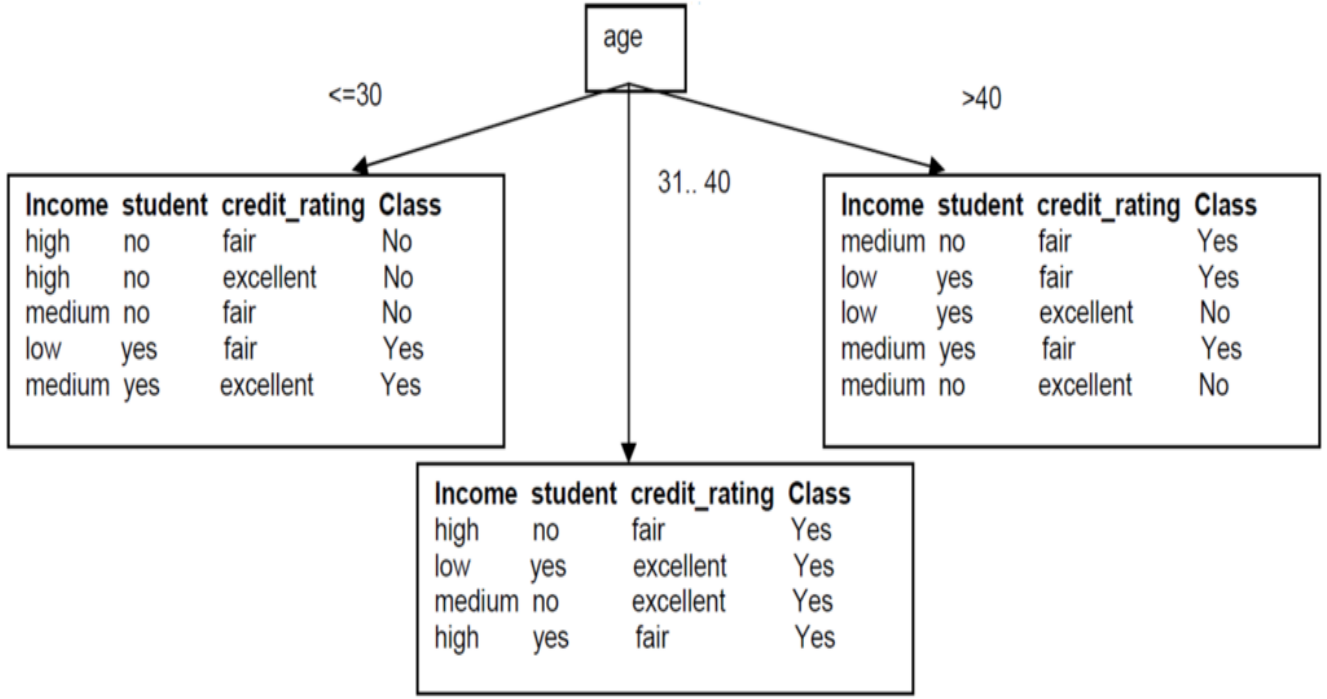
Age	Income	Student	Credit rating	Buys computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

$$Entropy(S) = \sum - p(I) \cdot \log_2 p(I)$$

$$Entropy(A) = \sum [\, p(S|A) \cdot Entropy(S|A) \,]$$

$$Gain(S, A) = Entropy(S) - Entropy(A)$$

Since Age has the highest Information Gain we start splitting the dataset using the age attribute.



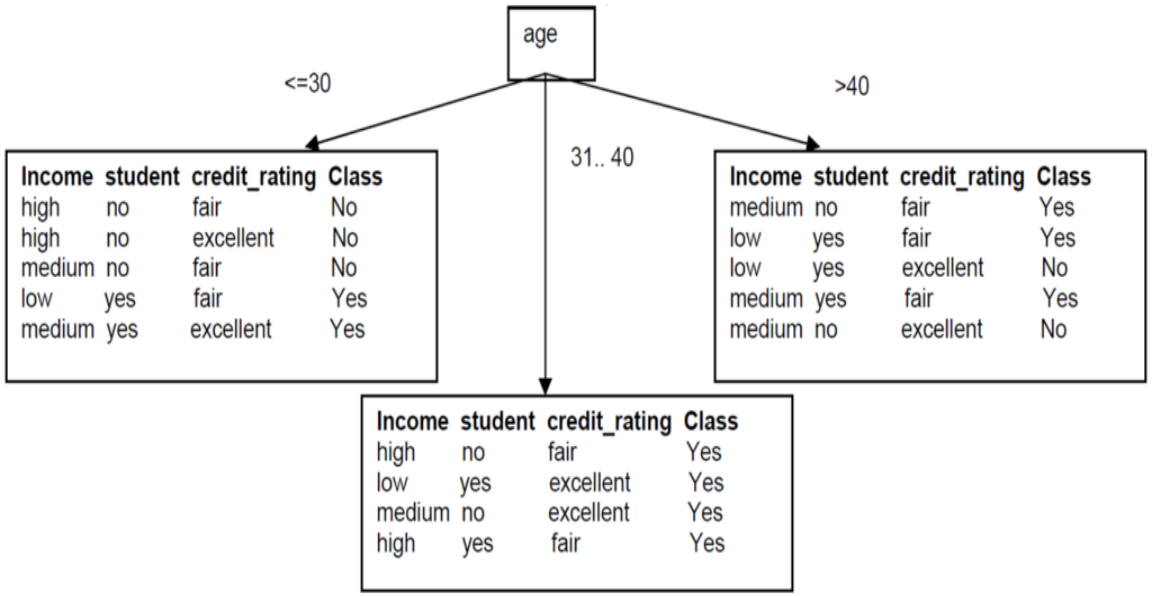
ID3 Algorithm

$$Entropy(S) = \sum - p(I) \cdot \log_2 p(I)$$

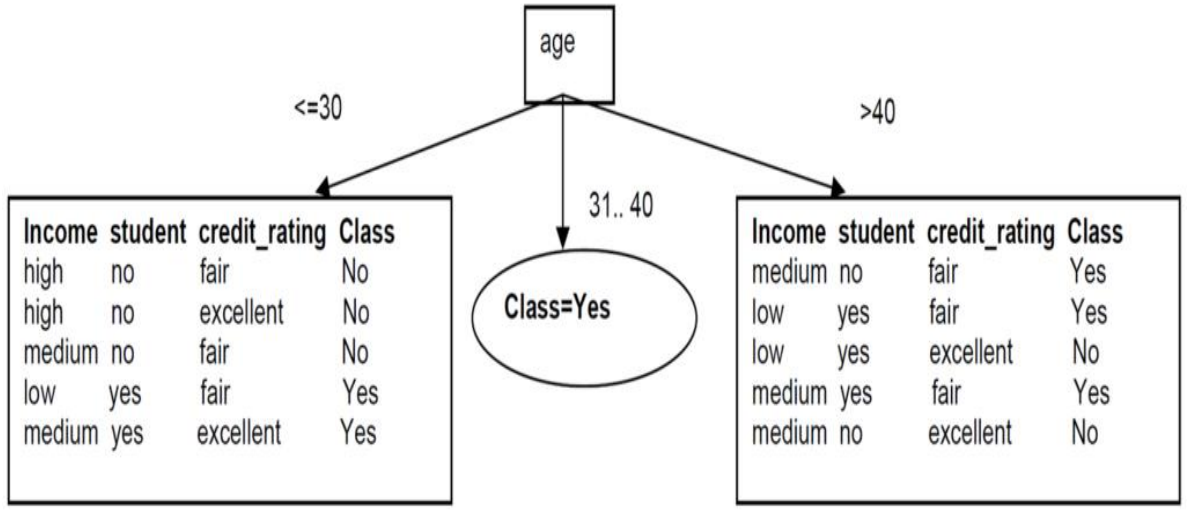
$$Entropy(A) = \sum [\, p(S|A) \cdot Entropy(S|A) \,]$$

$$Gain(S, A) = Entropy(S) - Entropy(A)$$

Since Age has the highest Information Gain we start splitting the dataset using the age attribute.



Since all records under the branch age31..40 are all of the class, Yes, we can replace the leaf with Class=Yes



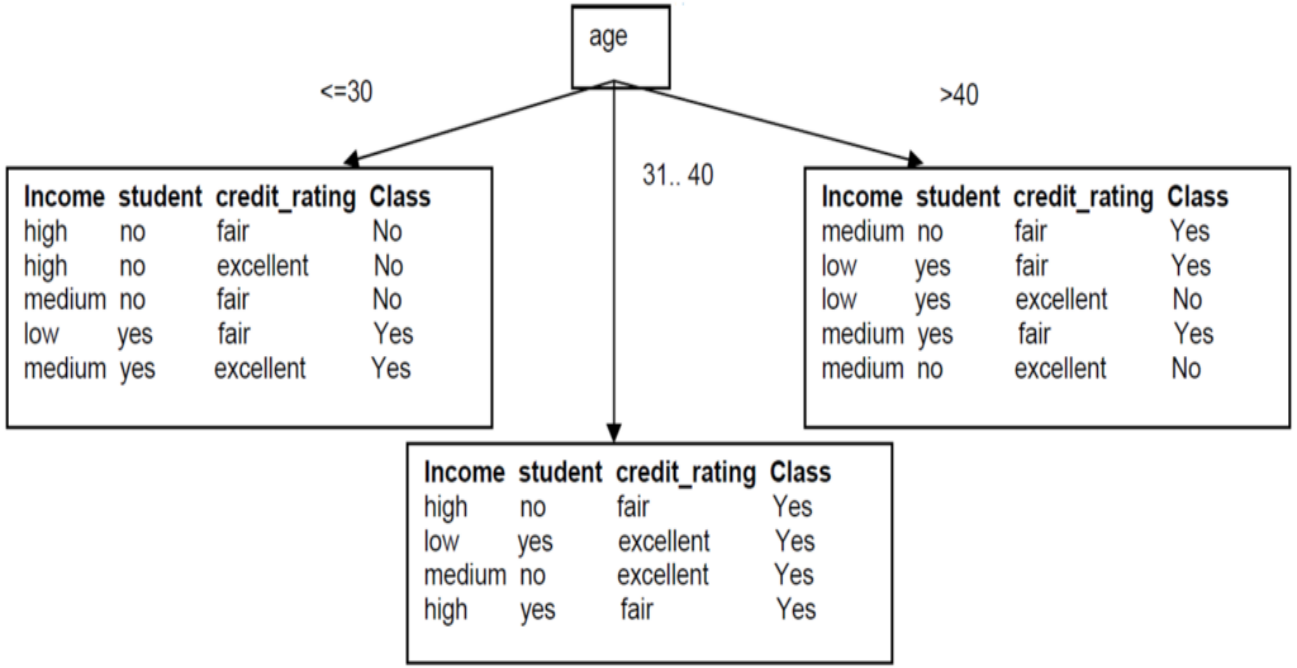
ID3 Algorithm

Now build the decision tree for the left subtree

$$\text{Entropy}(S) = \sum - p(I) \cdot \log_2 p(I)$$

$$\text{Entropy}(A) = \sum [\, p(S|A) \cdot \text{Entropy}(S|A) \,]$$

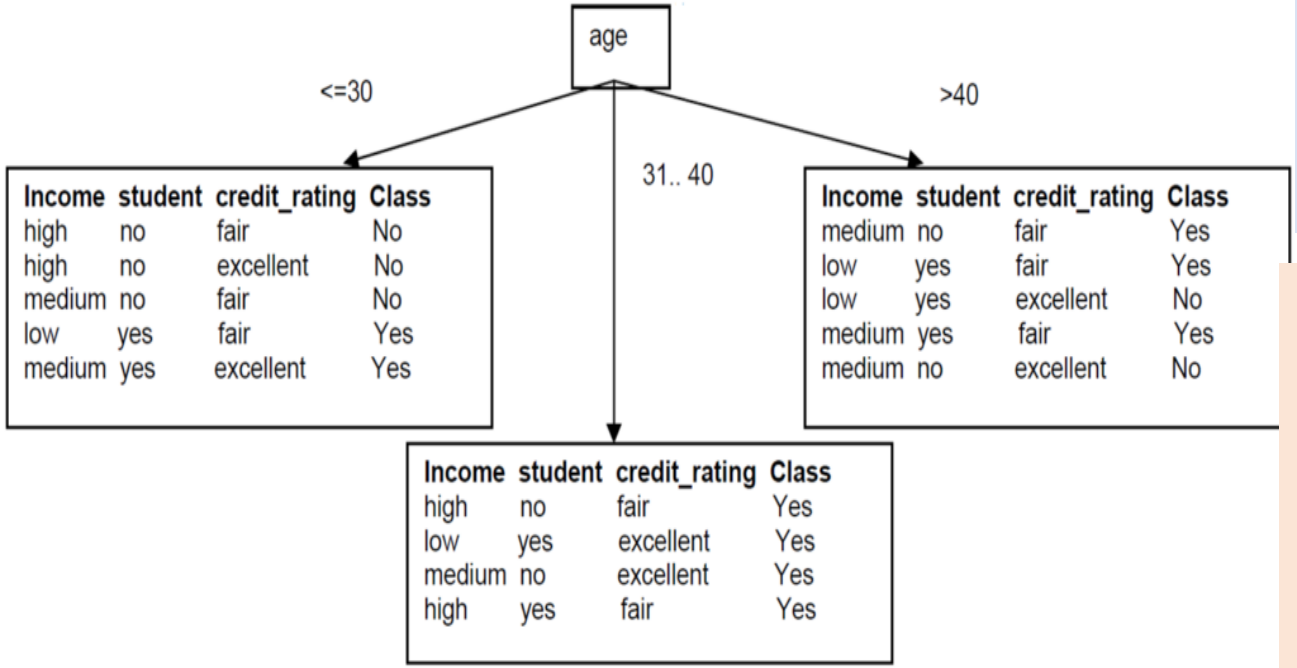
$$\text{Gain}(S, A) = \text{Entropy}(S) - \text{Entropy}(A)$$



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

ID3 Algorithm

Now build the decision tree for the left subtree



$Entropy(S) = \sum - p(I) \cdot \log_2 p(I)$

$Entropy(A) = \sum [\, p(S|A) \cdot Entropy(S|A) \,]$

$Gain(S, A) = Entropy(S) - Entropy(A)$

For branch age<=30 we still have attributes income, student, and credit_rating.
Which one should be used to split the partition?

The Entropy is $E(S_{age<=30})$
$= E(2,3) = -2/5 \log_2(2/5) - 3/5 \log_2(3/5) = 0.97$

For **Income Attribute,** we have three values

1. $income_{high}$ (0 yes and 2 no)
2. $income_{medium}$ (1 yes and 1 no)
3. $income_{low}$ (1 yes and 0 no)

$Entropy(income) = 2/5(0) + 2/5 (-1/2\log_2(1/2)-1/2\log_2(1/2)) + 1/5 (0) = 2/5 (1) = 0.4$

$Gain(income) = 0.97 - 0.4 = 0.57$

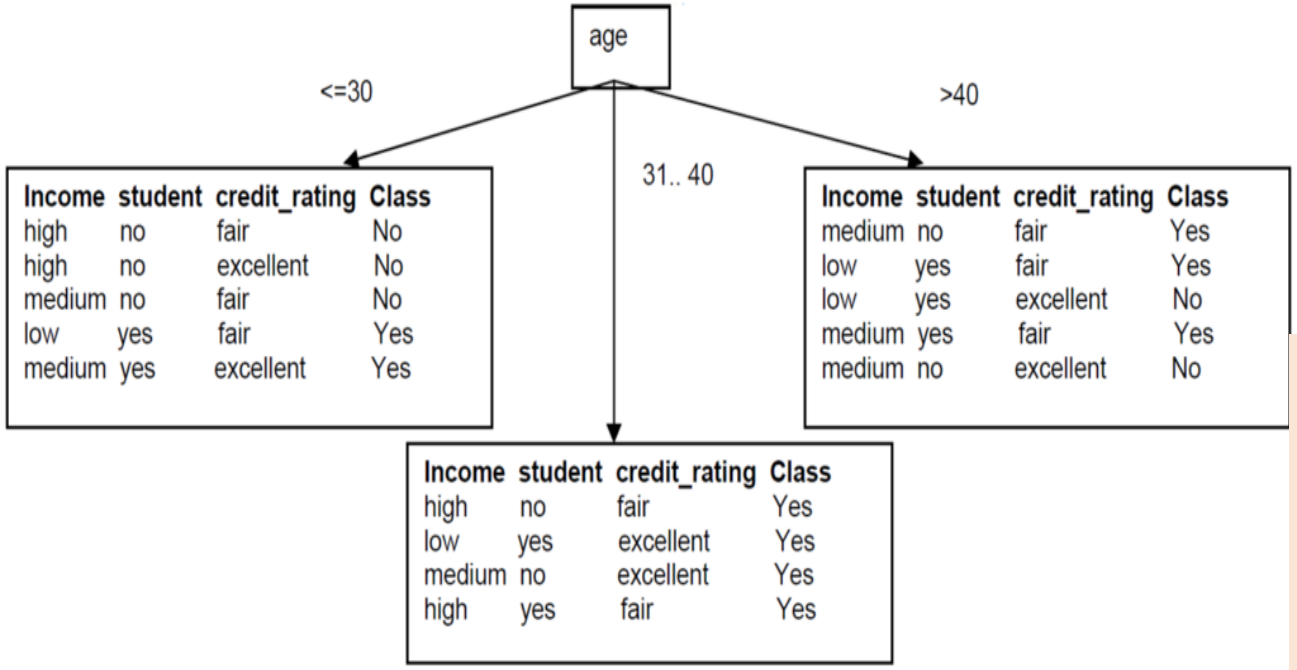
ID3 Algorithm

$Entropy(S) = \sum - p(I) \cdot \log_2 p(I)$

$Entropy(A) = \sum [\, p(S|A) \cdot Entropy(S|A) \,]$

$Gain(S, A) = Entropy(S) - Entropy(A)$

Now build the decision tree for the left subtree



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

Income	student	credit_rating	Class
high	no	fair	Yes
low	yes	excellent	Yes
medium	no	excellent	Yes
high	yes	fair	Yes

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

For branch age<=30 we still have attributes income, student, and credit_rating.
Which one should be used to split the partition?

The Entropy is $E(S_{age<=30})$
$= E(2,3) = -2/5 \log_2(2/5) - 3/5 \log_2(3/5) = 0.97$

For **Student,** we have two values

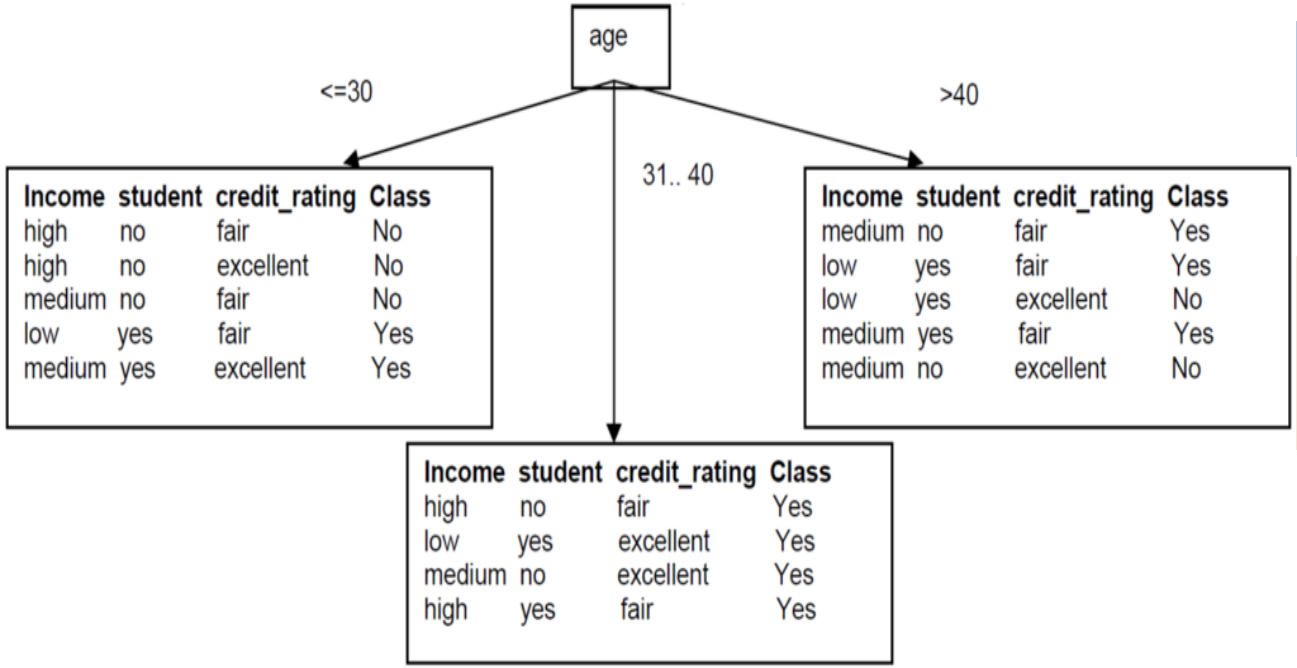
1. $student_{yes}$ (2 yes and 0 no)
2. $student_{no}$ (0 yes 3 no)

$Entropy(student) = 2/5(0) + 3/5(0) = 0$

$Gain(student) = 0.97 - 0 = 0.97$

ID3 Algorithm

Now build the decision tree for the left subtree



$Entropy(S) = \sum - p(I) \cdot \log_2 p(I)$

$Entropy(A) = \sum [\, p(S|A) \cdot Entropy(S|A) \,]$

$Gain(S, A) = Entropy(S) - Entropy(A)$

The Entropy is $E(S_{age<=30})$

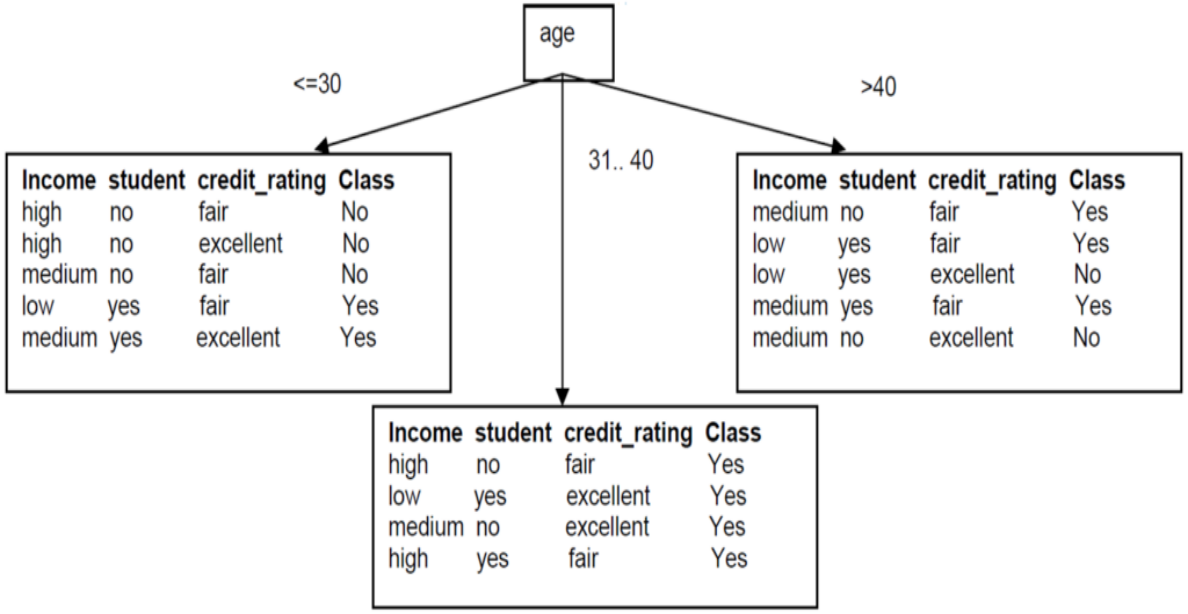
$= E(2,3) = -2/5 \log_2(2/5) - 3/5 \log_2(3/5) = 0.97$

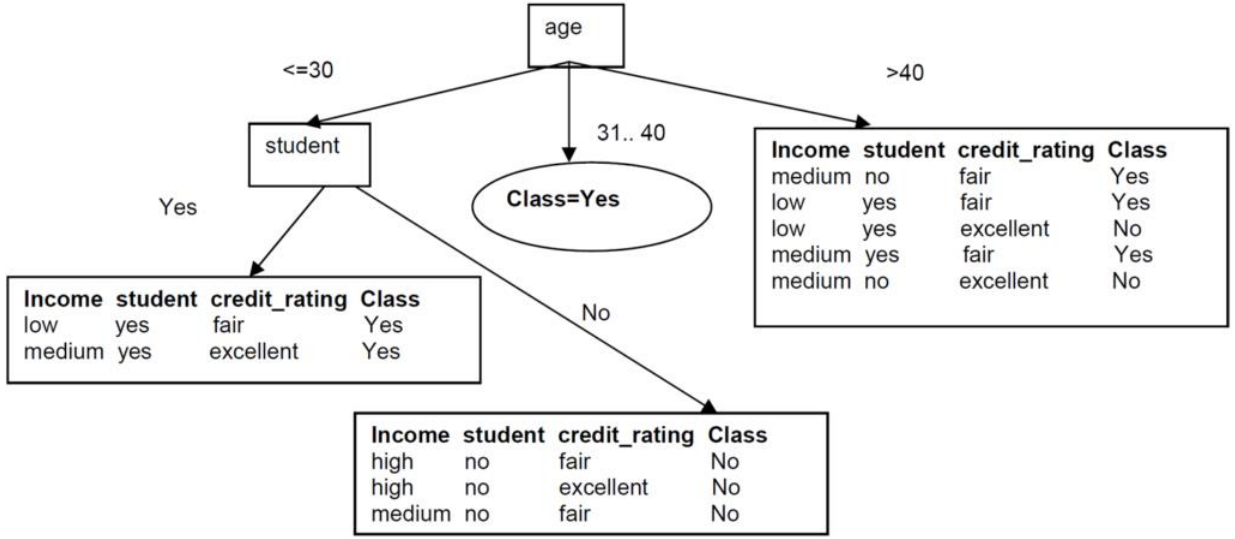
Gain(income) = 0.97 – 0.4 = 0.57

Gain (student) = 0.97 – 0 = 0.97

We can then safely split on attribute student without checking the other attributes since the information gain is maximized.

ID3 Algorithm





Entropy(S) = $\sum - p(I) \cdot \log_2 p(I)$

Entropy(A)= $\sum [\, p(S|A) \cdot Entropy(S|A) \,]$

Gain(S, A) = Entropy(S) – Entropy(A)

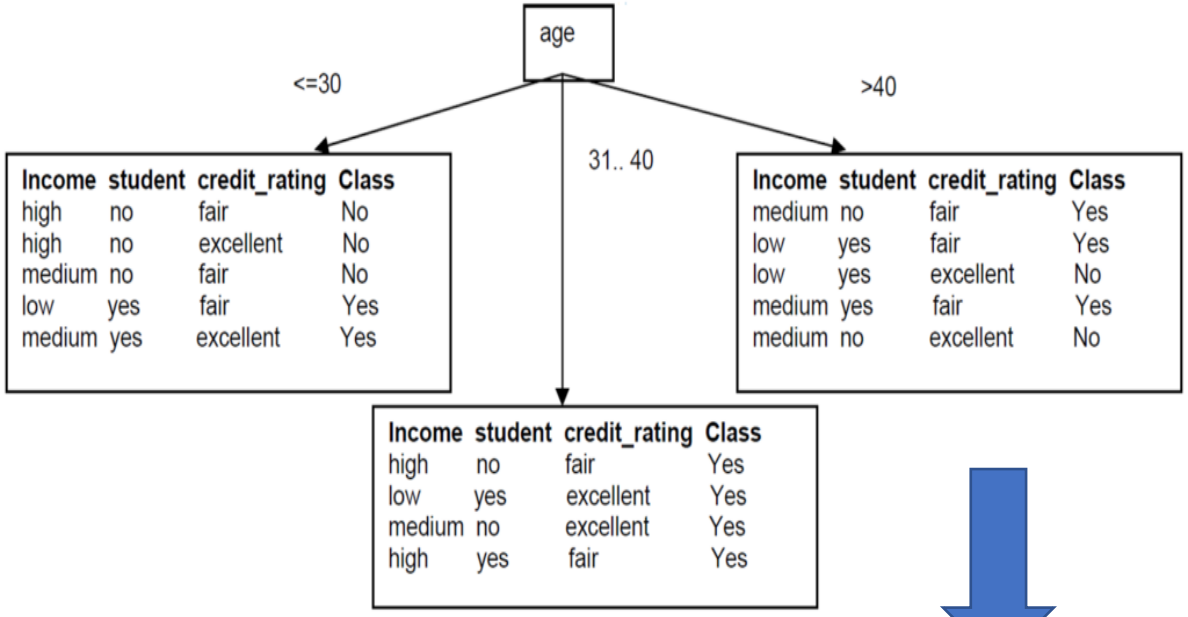
The Entropy is $E(S_{age<=30})$
= $E(2,3)= -2/5 \log_2(2/5) - 3/5 \log_2(3/5)=0.97$

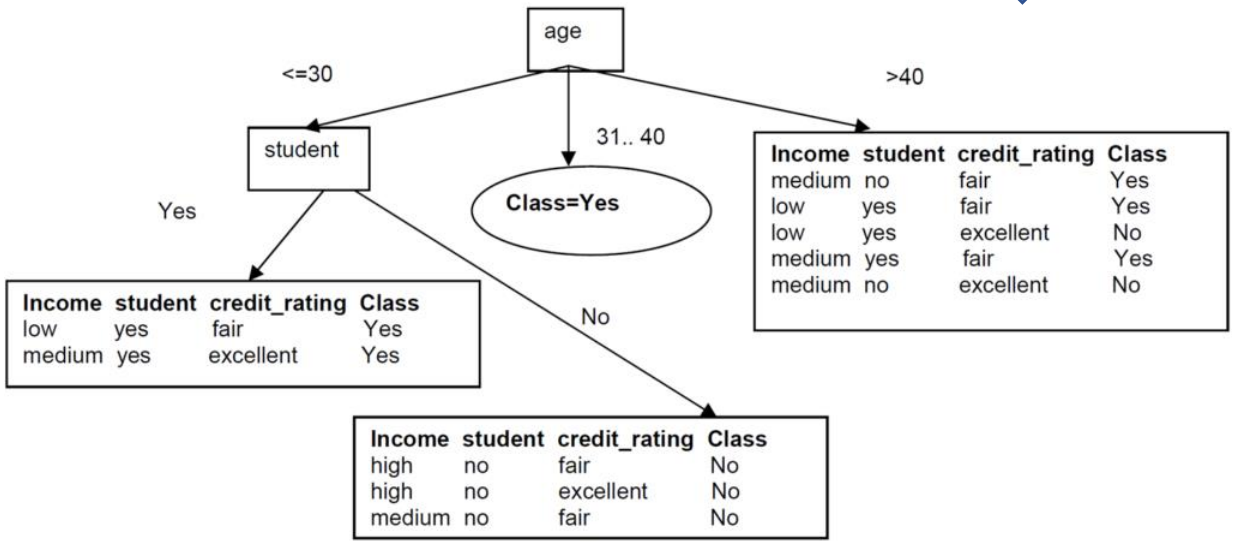
Gain(income) = 0.97 – 0.4 = 0.57

Gain (student) = 0.97 – 0 = 0.97

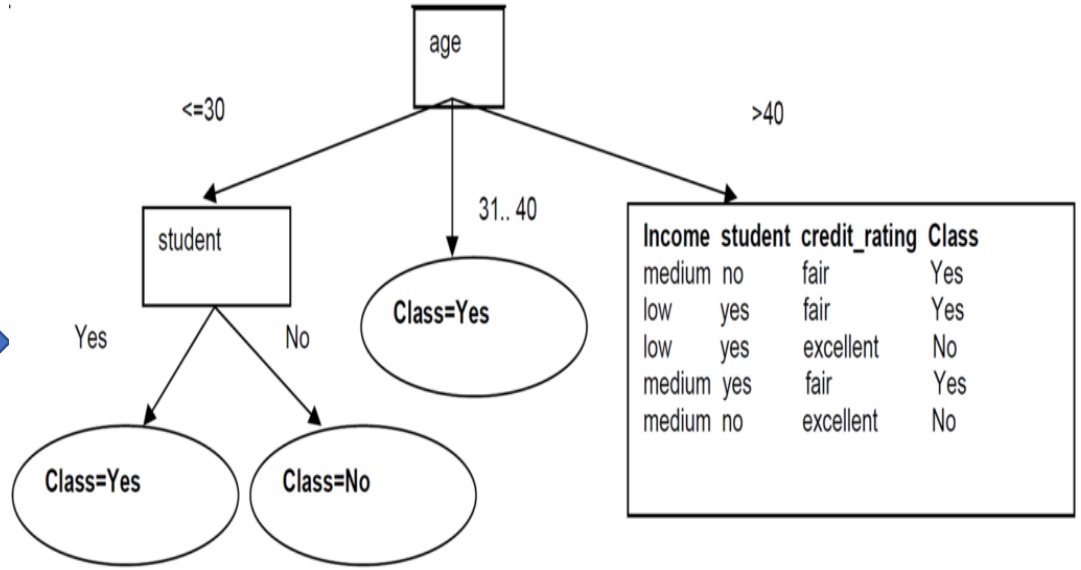
We can then safely split on attribute student without checking the other attributes since the information gain is maximized.

ID3 Algorithm



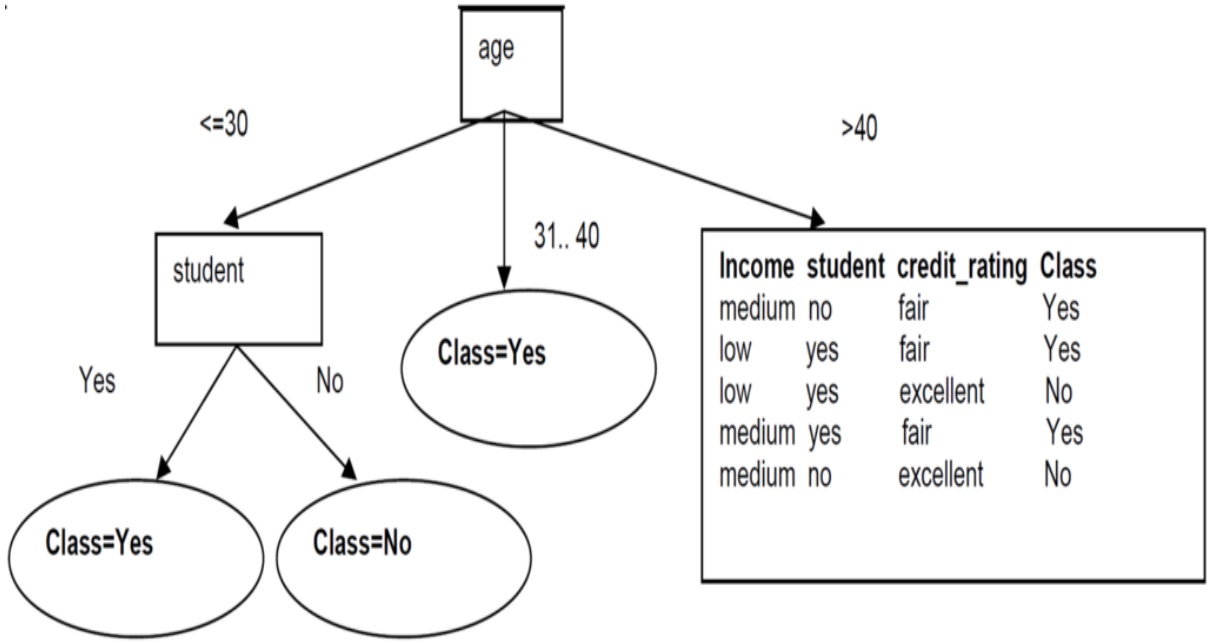


Since these two new branches are from distinct classes, we make them into leaf nodes with their respective class as label:



ID3 Algorithm

Now build the decision tree for right left subtree



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

$Entropy(S) = \sum - p(I) \cdot \log_2 p(I)$

$Entropy(A) = \sum [\, p(S|A) \cdot Entropy(S|A) \,]$

$Gain(S, A) = Entropy(S) - Entropy(A)$

$Entropy(S_{age>40}) = I(3,2) = -3/5 \log_2(3/5) - 2/5 \log_2(2/5) = 0.97$

For **Income**, we have two values

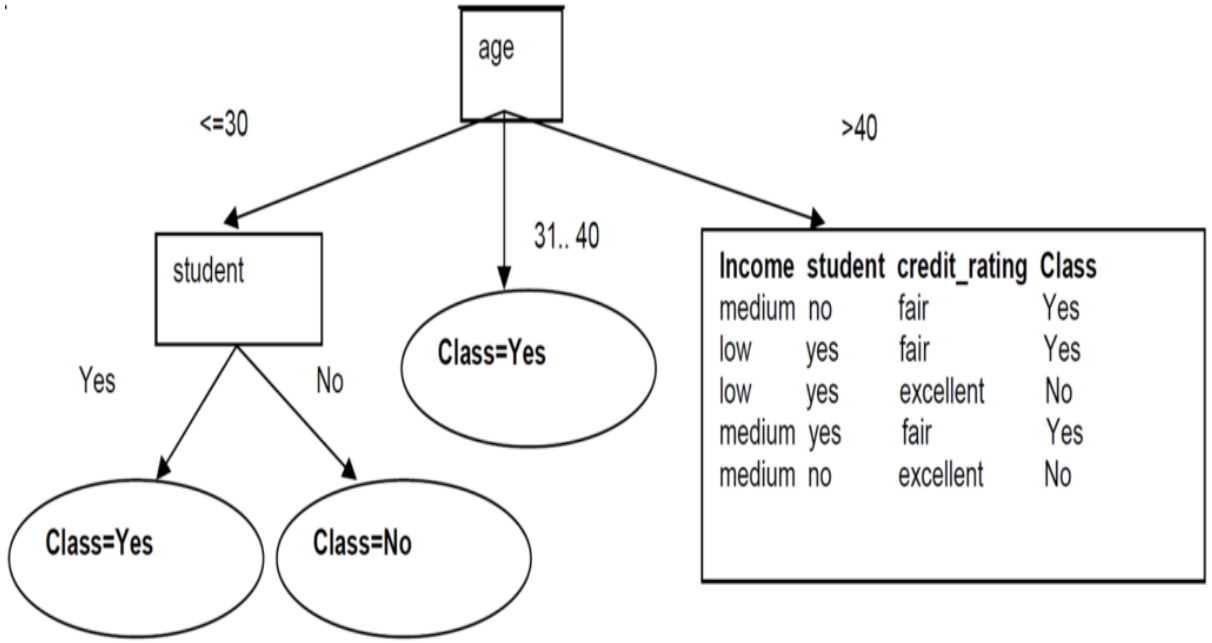
1. $income_{medium}$ (2 yes and 1 no)
2. $income_{low}$ (1 yes and 1 no)

Entropy(**income**) = 3/5(-2/3log2(2/3)-1/3log2(1/3)) + 2/5 (-1/2log2(1/2)-1/2log2(1/2))
= 3/5(0.9182)+2/5 (1) = 0.55+0. 4= 0.95

Gain(income) = 0.97 – 0.95 = 0.02

ID3 Algorithm

Now build the decision tree for right left subtree



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

$$\text{Entropy}(S) = \sum - p(I) \cdot \log_2 p(I)$$

$$\text{Entropy}(A) = \sum [\, p(S|A) \cdot \text{Entropy}(S|A) \,]$$

$$\text{Gain}(S, A) = \text{Entropy}(S) - \text{Entropy}(A)$$

$$\text{Entropy}(S_{age>40}) = I(3,2) = -3/5 \log_2(3/5) - 2/5 \log_2(2/5) = 0.97$$

For **Student**, we have two values

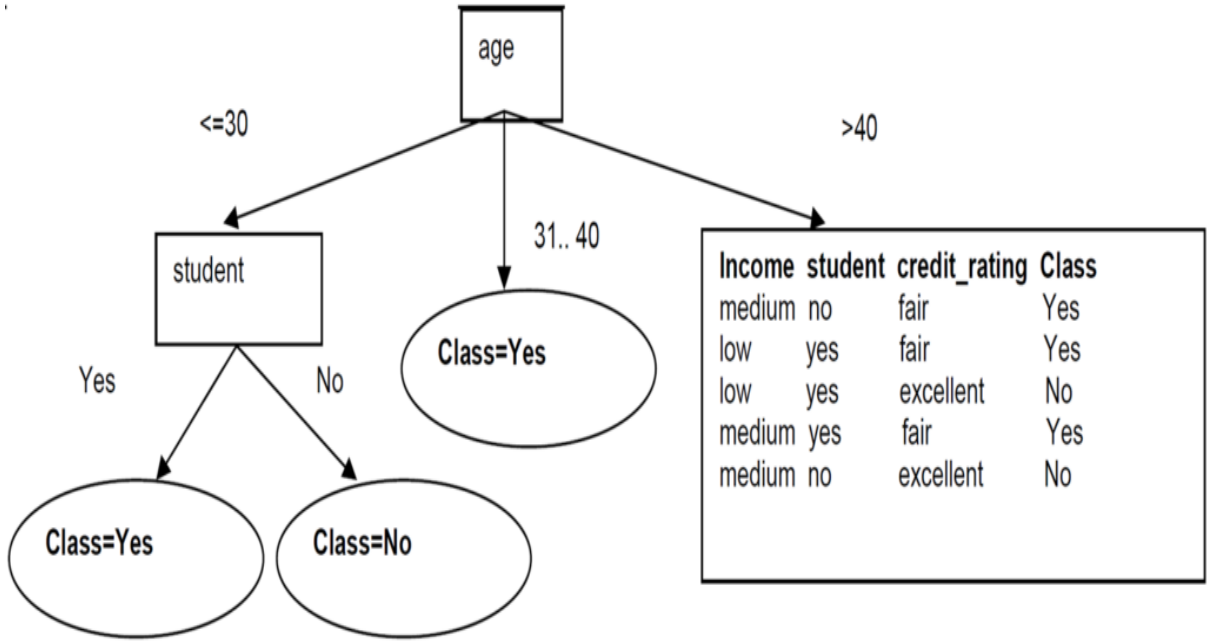
1. student_{yes} (2 yes and 1 no)
2. student_{no} (1 yes and 1 no)

$$\text{Entropy}(\textbf{student}) = 3/5(-2/3\log_2(2/3) - 1/3\log_2(1/3)) + 2/5(-1/2\log_2(1/2) - 1/2\log_2(1/2)) = 0.95$$

$$\text{Gain (student)} = 0.97 - 0.95 = 0.02$$

ID3 Algorithm

Now build the decision tree for right left subtree



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

$Entropy(S) = \sum - p(I) \cdot \log_2 p(I)$

$Entropy(A) = \sum [\, p(S|A) \cdot Entropy(S|A) \,]$

$Gain(S, A) = Entropy(S) - Entropy(A)$

$Entropy(S_{age>40}) = I(3,2) = -3/5 \log_2(3/5) - 2/5 \log_2(2/5) = 0.97$

For **Credit_Rating**, we have two values

1. credit_ratingfair (3 yes and 0 no)
2. credit_ratingexcellent (0 yes and 2 no)

$Entropy(credit\_rating) = 0$

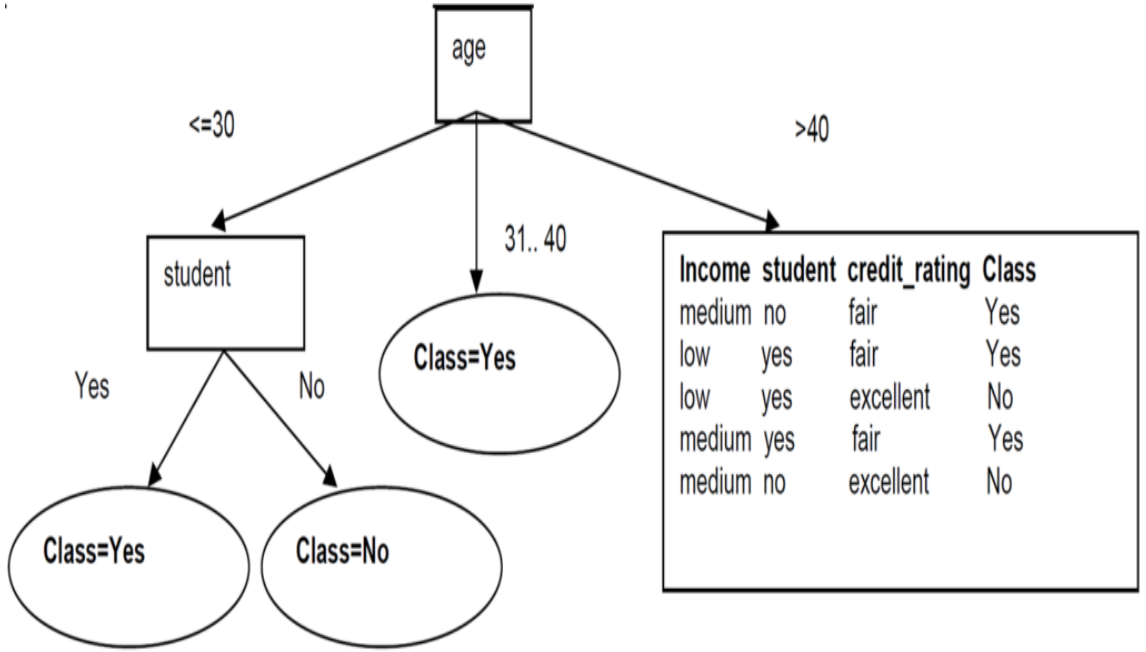
$Gain(credit\_rating) = 0.97 - 0 = 0.97$

ID3 Algorithm

Gain(income) = 0.97 – 0.95 = 0.02

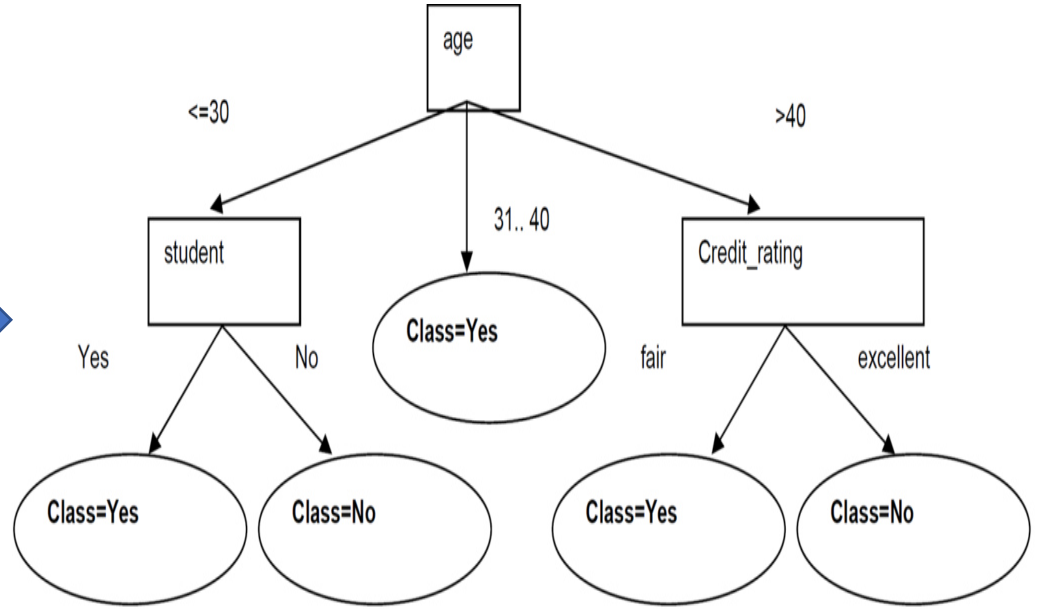
Gain (student) = 0.97 – 0.95 = 0.02

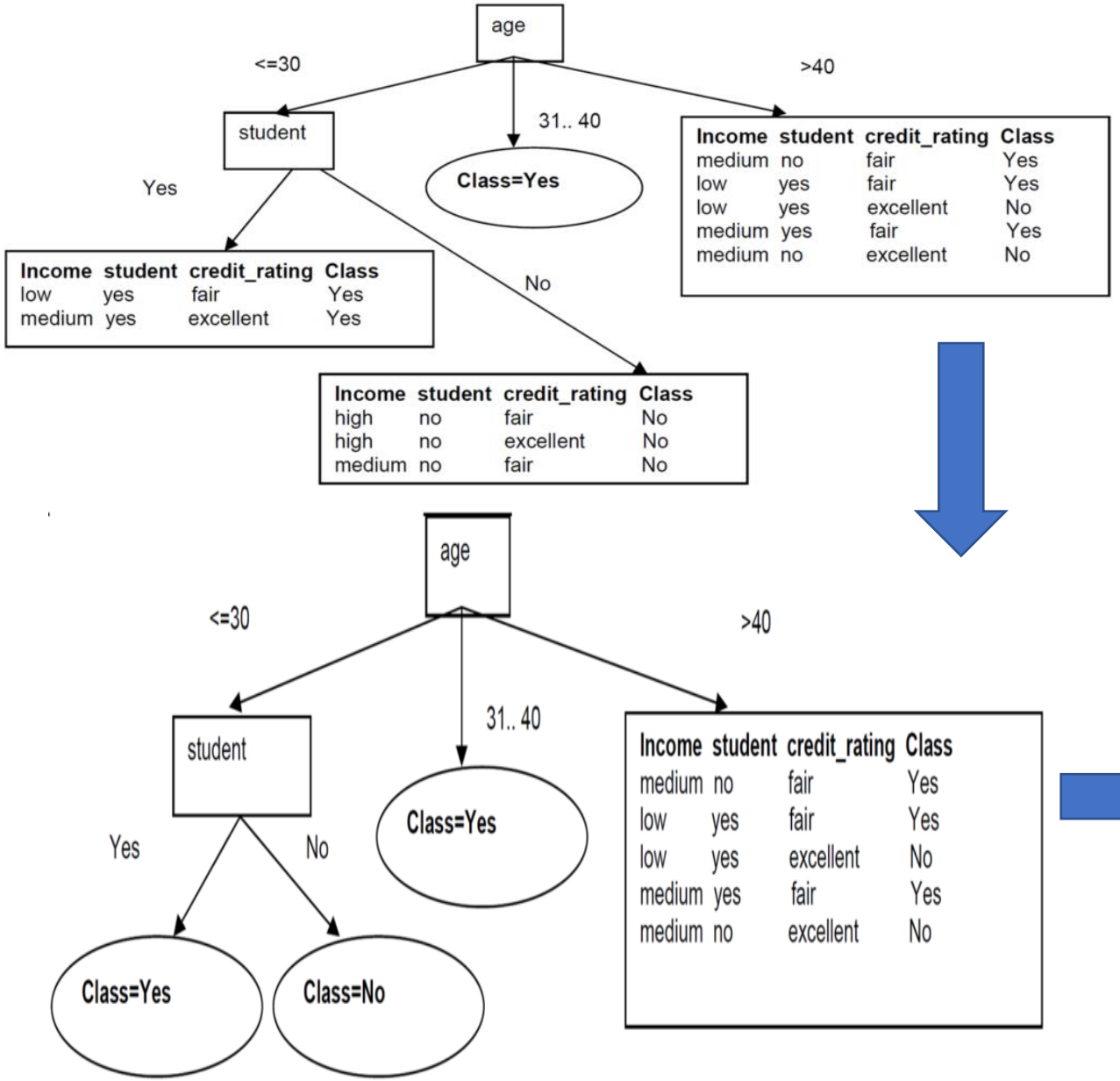
Gain(credit_rating) = 0.97 – 0 = 0.97

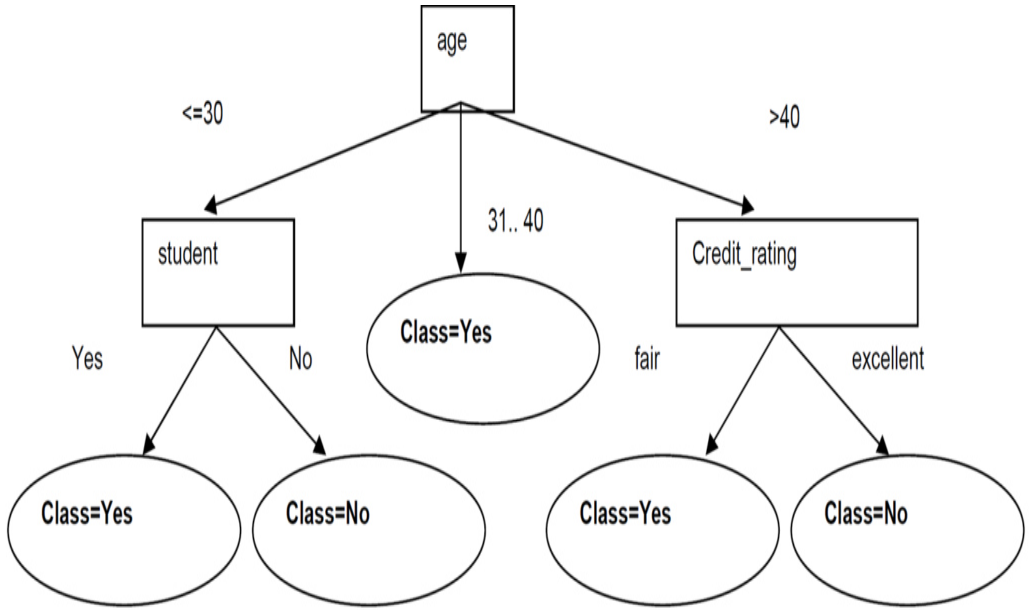


Income	student	credit_rating	Class
medium	no	fair	Yes
low	yes	fair	Yes
low	yes	excellent	No
medium	yes	fair	Yes
medium	no	excellent	No

We then split based on credit_rating.







New example: age<=30, income=medium, student=yes, credit-rating=fair

Buys_computer = yes

Majors of ID3 Algorithm

- ID3 uses a greedy approach that's why it does not guarantee an optimal solution; it can get stuck in local optimums.
- ID3 can overfit to the training data (to avoid overfitting, smaller decision trees should be preferred over larger ones).
- This algorithm usually produces small trees, but it does not always produce the smallest possible tree.
- ID3 is harder to use on continuous data (if the values of any given attribute is continuous, then there are many more places to split the data on this attribute, and searching for the best value to split by can be time consuming).

What is the disadvantage of using Information Gain for feature selection?

- Natural bias of information gain: it favors attributes with many possible values.
- Consider the attribute *Date* in the *PlayTennis* example.
  - *Date* would have the highest information gain since it perfectly separates the training data.
- It would be selected at the root resulting in a very broad tree
- Very good on the training, this tree would perform poorly in predicting unknown instances. **Overfitting**.

Majors of ID3 Algorithm

Algorithm	Splitting Criteria of algorithm	Attribute types Managed by algorithm	Pruning Strategy of algorithm	Outlier Detection	Missing values	Invented By
ID3	Information Gain	Manages only Categorical value	No pruning is done	No pruning is done	Do not Manages missing values.	invented by Ross Quinlan

Advantages and Disadvantages of ID3 Algorithm

Advantages
Inexpensive to construct
Extremely fast at classifying unknown records Easy to interpret for small-sized trees.
Robust to noise (especially when methods to avoid over-fitting are employed)
Can easily handle redundant or irrelevant attributes (unless the attributes are interacting)

Disadvantages
The space of possible decision trees is exponentially large. Greedy approaches are often unable to find the best tree.
Does not take into account interactions between attributes
Each decision boundary involves only a single attribute

C4.5

- C4.5 is an improvement over ID3 and was developed in the early 1990s.
- It works with continuous and discrete values.
- It works with missing values by marking as ‘?’,but these missing attribute values are not considered in the calculations.
- It uses the information gain Ratio metric and adds the ability to handle missing values and continuous features.
- C4.5 also adds pruning, a technique for reducing the size of the tree and avoiding overfitting.

C4.5

- The C4.5 algorithm requires a larger training set for better accuracy.
- ID3 is more biased towards attributes with larger values.
- For example, if there is an attribute called 'Register No' for students it would be unique for every student and will have distinct value for every data instance resulting in more values for the attribute.
- Hence, every instance belongs to a category and would have higher Information Gain than other attributes.
- To overcome this bias issue, C4.5 uses a purity measure Gain ratio to identify the best split attribute.

Decision Tree C4.5 Algorithm – Construction

1. First, Compute Entropy _Info for the whole training dataset based on the target attribute.

$$\text{Entropy\_Info}(T) = -\sum_{i=1}^{m} P_i \log_2 P_i$$

2. Next, for each of the attribute in the training dataset Compute

Entropy_Info,

$$\text{Entropy\_Info}(T, A) = \sum_{i=1}^{v} \frac{|A_i|}{|T|} \times \text{Entropy\_Info}\ (A_i)$$

Info_Gain,

$$\text{Information\_Gain}(A) = \text{Entropy\_Info}(T) - \text{Entropy\_Info}(T, A)$$

Split_Info

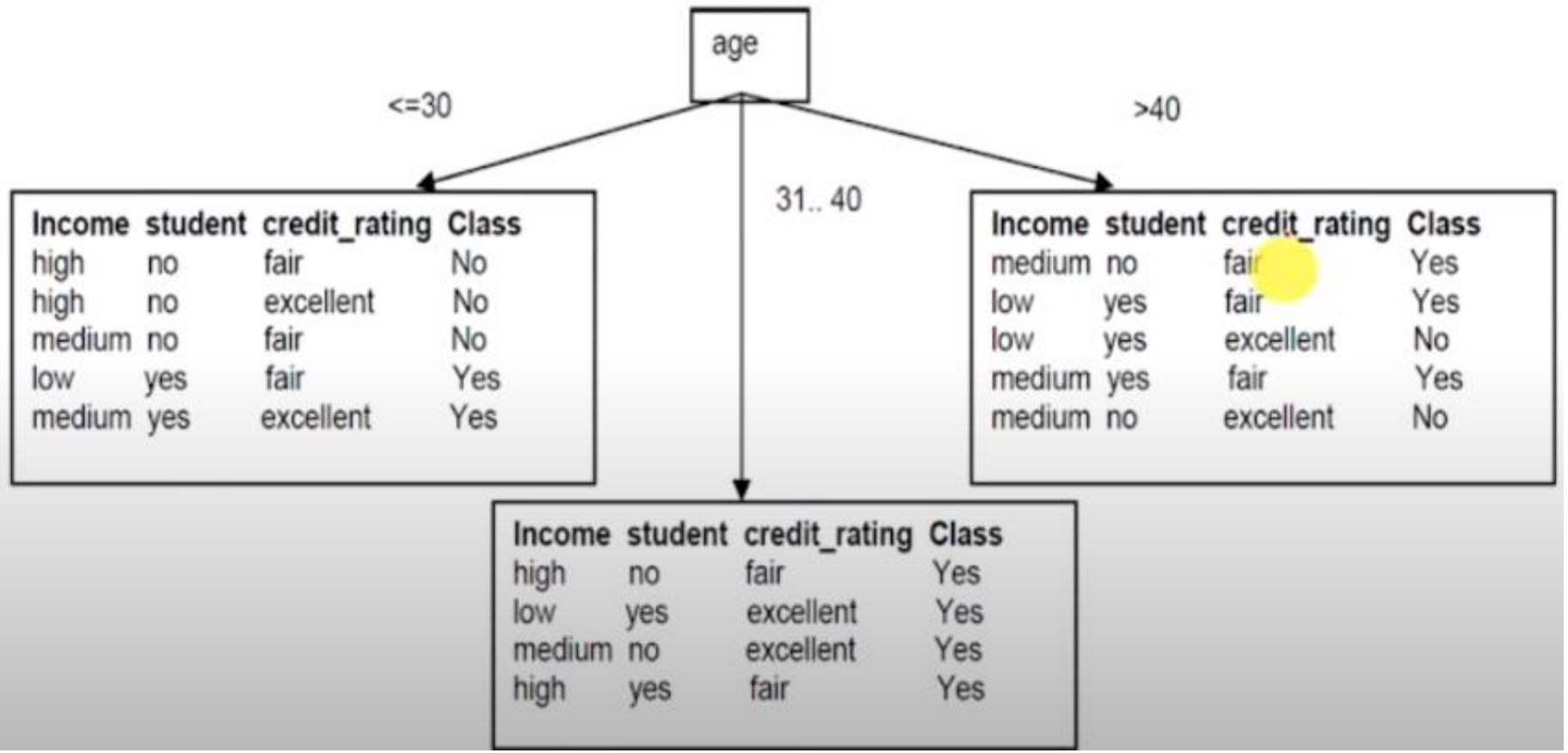
$$\text{Split\_Info}(T, A) = -\sum_{i=1}^{v} \frac{|A_i|}{|T|} \times \log_2 \frac{|A_i|}{|T|}$$

Gain_Ratio

$$\text{Gain\_Ratio}(A) = \frac{\text{Info\_Gain}(A)}{\text{Split\_Info}(T, A)}$$

3. Choose the attribute for which Gain_Ratio is maximum as the best split attribute.

4. The best split attribute is placed as the root node.



5. The root node is branched into subtrees with each subtree as an outcome of the test condition of the root node attribute. Accordingly, the training dataset is also split into subsets.
6. Recursively apply the same operation for the subset of the training set with the remaining attributes until a leaf node is derived or no more training instances are available in the subset.

C4.5 Example for discrete value

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

Step 1: Calculate the Class_Entropy for the target class 'Job Offer'.

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

Example

Step 1: Calculate the Class_Entropy for the target class 'Job Offer'.

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

$$\text{Entropy\_Info}(T) = -\sum_{i=1}^{m} P_i \log_2 P_i$$

$$\text{Entropy\_Info}(\text{Target Attribute} = \text{Job Offer}) = \text{Entropy\_Info}(7, 3) =$$

$$= -\left[\frac{7}{10}\log_2\frac{7}{10} + \frac{3}{10}\log_2\frac{3}{10}\right]$$

$$= (-0.3599 + -0.5208)$$

$$= 0.8807$$

Example

Step 2: Calculate the Entropy _Info, Gain(Info_Gain), Split_Info, Gain_Ratio for each of the attribute in the training dataset.

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

Example

Step 2: Calculate the Entropy _Info, Gain(Info_Gain), Split_Info, Gain_Ratio for each of the attribute in the training dataset.

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

$$\text{Entropy\_Info}(T, A) = \sum_{i=1}^{v} \frac{|A_i|}{|T|} \times \text{Entropy\_Info}\ (A_i)$$

Example

Step 2: Calculate the Entropy _Info, Gain(Info_Gain), Split_Info, Gain_Ratio for each of the attribute in the training dataset.

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

$$\text{Entropy\_Info}(T, A) = \sum_{i=1}^{v} \frac{|A_i|}{|T|} \times \text{Entropy\_Info}\ (A_i)$$

CGPA:

$$\text{Entropy Info}(T, \text{CGPA})$$

$$= \frac{4}{10}\left[-\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4}\right] + \frac{4}{10}\left[-\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4}\right] + \frac{2}{10}\left[-\frac{0}{2}\log_2\frac{0}{2} - \frac{2}{2}\log_2\frac{2}{2}\right]$$

$$= \frac{4}{10}(0.3111 + 0.4997) + 0 + 0$$

$$= 0.3243$$

Example

Step 2: Calculate the Entropy _Info, Gain(Info_Gain), Split_Info, Gain_Ratio for each of the attribute in the training dataset.

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

$$\text{Information\_Gain}(A) = \text{Entropy\_Info}(T) - \text{Entropy\_Info}(T, A)$$

$$\text{Gain(CGPA)} = 0.8807 - 0.3243 = 0.5564$$

$$\text{Split\_Info}(T, A) = -\sum_{i=1}^{v} \frac{|A_i|}{|T|} \times \log_2 \frac{|A_i|}{|T|}$$

$$\text{Split\_Info}(T, \text{CGPA}) = -\frac{4}{10}\log_2\frac{4}{10} - \frac{4}{10}\log_2\frac{4}{10} - \frac{2}{10}\log_2\frac{2}{10}$$

$$= 0.5285 + 0.5285 + 0.4641 = 1.5211$$

$$\text{Gain Ratio(CGPA)} = \frac{0.5564}{1.5211} = 0.3658$$

Example

Step 2: Calculate the Entropy _Info, Gain(Info_Gain), Split_Info, Gain_Ratio for each of the attribute in the training dataset.

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

$$\text{Entropy\_Info}(T, A) = \sum_{i=1}^{v} \frac{|A_i|}{|T|} \times \text{Entropy\_Info}\ (A_i)$$

Interactiveness:

$$\text{Entropy Info}(T, \text{Interactiveness})$$

$$= \frac{6}{10}\left[-\frac{5}{6}\log_2\frac{5}{6} - \frac{1}{6}\log_2\frac{1}{6}\right] + \frac{4}{10}\left[-\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4}\right]$$

$$= \frac{6}{10}(0.2191 + 0.4306) + \frac{4}{10}(0.4997 + 0.4997)$$

$$= 0.3898 + 0.3998 = 0.7896$$

Example

Step 2: Calculate the Entropy _Info, Gain(Info_Gain), Split_Info, Gain_Ratio for each of the attribute in the training dataset.

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

$$\text{Information\_Gain}(A) = \text{Entropy\_Info}(T) - \text{Entropy\_Info}(T, A)$$

$$\text{Gain(Interactiveness)} = 0.8807 - 0.7896 = 0.0911$$

$$\text{Split\_Info}(T, A) = -\sum_{i=1}^{v} \frac{|A_i|}{|T|} \times \log_2 \frac{|A_i|}{|T|}$$

$$\text{Split\_Info}(T, \text{Interactiveness}) = -\frac{6}{10}\log_2\frac{6}{10} - \frac{4}{10}\log_2\frac{4}{10} = 0.9704$$

$$\text{Gain\_Ratio(Interactiveness)} = \frac{0.0911}{0.9704} = 0.0939$$

Example

Step 2: Calculate the Entropy _Info, Gain(Info_Gain), Split_Info, Gain_Ratio for each of the attribute in the training dataset.

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

$$\text{Information\_Gain}(A) = \text{Entropy\_Info}(T) - \text{Entropy\_Info}(T, A)$$

$$\text{Gain(Practical Knowledge)} = 0.8807 - 0.6361 = 0.2448$$

$$\text{Split\_Info}(T, A) = -\sum_{i=1}^{v} \frac{|A_i|}{|T|} \times \log_2 \frac{|A_i|}{|T|}$$

$$\text{Split\_Info}(T, \text{Practical Knowledge}) = -\frac{2}{10}\log_2\frac{2}{10} - \frac{5}{10}\log_2\frac{5}{10} - \frac{3}{10}\log_2\frac{3}{10}$$

$$= 1.4853$$

$$\text{Gain\_Ratio(Practical Knowledge)} = \frac{0.2448}{1.4853} = 0.1648$$

Example

Step 2: Calculate the Entropy _Info, Gain(Info_Gain), Split_Info, Gain_Ratio for each of the attribute in the training dataset.

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

$$\text{Entropy\_Info}(T, A) = \sum_{i=1}^{v} \frac{|A_i|}{|T|} \times \text{Entropy\_Info}\ (A_i)$$

Communication Skills:

Entropy_Info(T, Communication Skills)

$$= \frac{5}{10}\left[-\frac{4}{5}\log_2\frac{4}{5} - \frac{1}{5}\log_2\frac{1}{5}\right] + \frac{3}{10}\left[-\frac{3}{3}\log_2\frac{3}{3} - \frac{0}{3}\log_2\frac{0}{3}\right] + \frac{2}{10}\left[-\frac{0}{2}\log_2\frac{0}{2} - \frac{2}{2}\log_2\frac{2}{2}\right]$$

$$= \frac{5}{10}(0.5280 + 0.3897) + \frac{3}{10}(0) + \frac{2}{10}(0)$$

$$= 0.3609$$

Example

Step 2: Calculate the Entropy _Info, Gain(Info_Gain), Split_Info, Gain_Ratio for each of the attribute in the training dataset.

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

$$\text{Information\_Gain}(A) = \text{Entropy\_Info}(T) - \text{Entropy\_Info}(T, A)$$

$$\text{Gain(Communication Skills)} = 0.8813 - 0.36096 = 0.5202$$

$$\text{Split\_Info}(T, A) = -\sum_{i=1}^{v} \frac{|A_i|}{|T|} \times \log_2 \frac{|A_i|}{|T|}$$

$$\text{Split\_Info}(T, \text{Communication Skills})$$

$$= -\frac{5}{10}\log_2\frac{5}{10} - \frac{3}{10}\log_2\frac{3}{10} - \frac{2}{10}\log_2\frac{2}{10}$$

$$= 1.4853$$

$$\text{Gain\_Ratio(Communication Skills)} = \frac{0.5202}{1.4853} = 0.3502$$

Example

Step 2: Calculate the Entropy _Info, Gain(Info_Gain), Split_Info, Gain_Ratio for each of the attribute in the training dataset.

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

Gain_Ratio

Attribute	Gain_Ratio
CGPA	0.3658
INTERACTIVENESS	0.0939
PRACTICAL KNOWLEDGE	0.1648
COMMUNICATION SKILLS	0.3502

Example

Step 2: Calculate the Entropy _Info, Ga

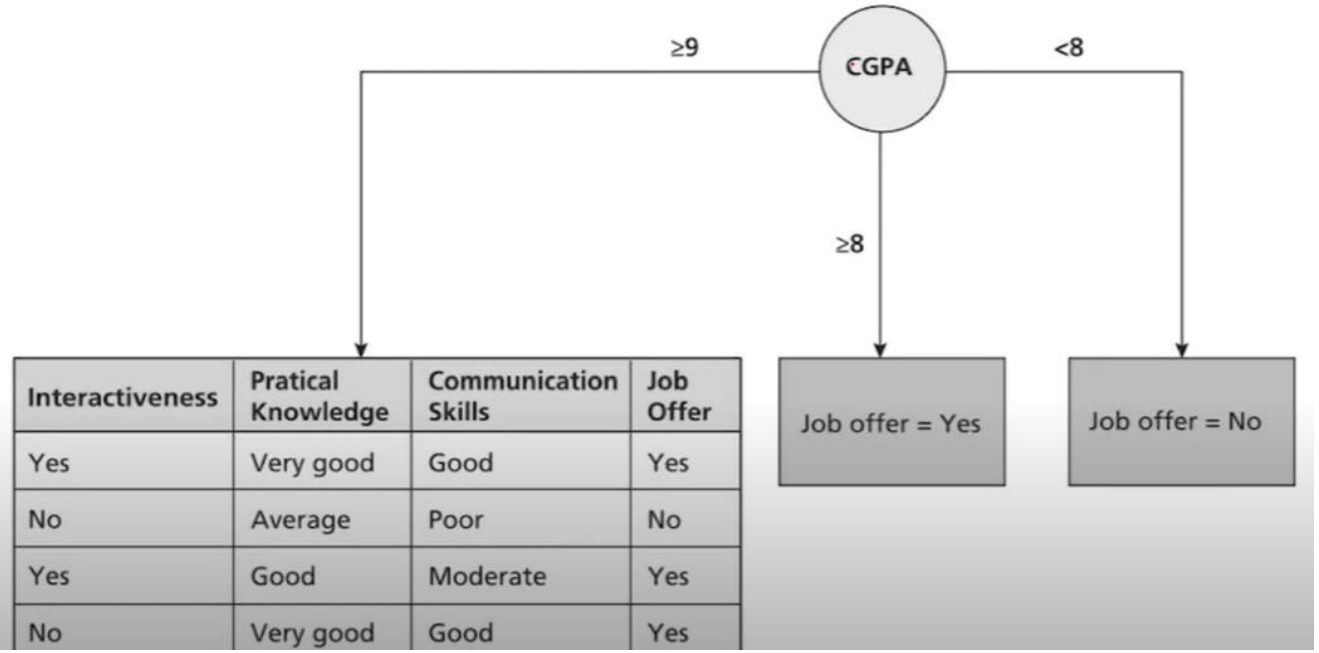
each

aset.

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

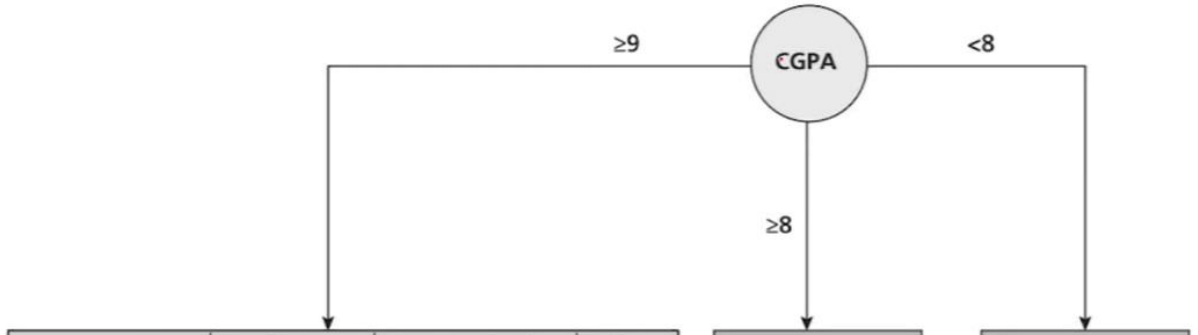
Gain_Ratio

Attribute	Gain_Ratio
CGPA	0.3658
INTERACTIVENESS	0.0939
PRACTICAL KNOWLEDGE	0.1648
COMMUNICATION SKILLS	0.3502



Interactiveness	Pratical Knowledge	Communication Skills	Job Offer
Yes	Very good	Good	Yes
No	Average	Poor	No
Yes	Good	Moderate	Yes
No	Very good	Good	Yes

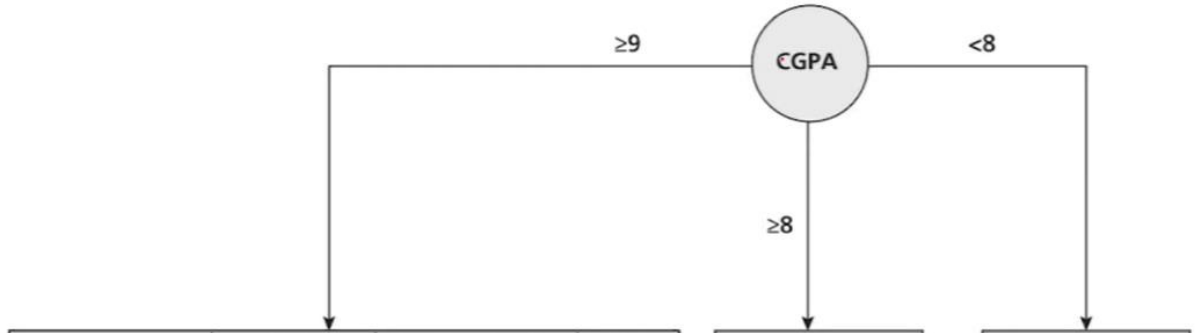
Example



Interactiveness	Pratical Knowledge	Communication Skills	Job Offer
Yes	Very good	Good	Yes
No	Average	Poor	No
Yes	Good	Moderate	Yes
No	Very good	Good	Yes

Step 1: Calculate the Class_Entropy for the target class 'Job Offer'.

Inter active	Practical Knowledge	Comm Skills	Job Offer
Yes	Very good	Good	Yes
No	Average	Poor	No
Yes	Good	Moderate	Yes
No	Very good	Good	Yes



Interactiveness	Pratical Knowledge	Communication Skills	Job Offer
Yes	Very good	Good	Yes
No	Average	Poor	No
Yes	Good	Moderate	Yes
No	Very good	Good	Yes

Step 1: Calculate the Class_Entropy for the target class 'Job Offer'.

Inter active	Practical Knowledge	Comm Skills	Job Offer
Yes	Very good	Good	Yes
No	Average	Poor	No
Yes	Good	Moderate	Yes
No	Very good	Good	Yes

$$\text{Entropy\_Info}(T) = -\sum_{i=1}^{m} P_i \log_2 P_i$$

$$\text{Entropy\_Info}(\text{Target Class} = \text{Job Offer}) = -\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4}$$

$$= 0.3112 + 0.5$$

$$= 0.8112 \checkmark$$

Step 2: Calculate the Entropy _Info, Gain(Info_Gain), Split_Info, Gain_Ratio for each of the attribute in the training dataset.

Inter active	Practical Knowledge	Comm Skills	Job Offer
Yes	Very good	Good	Yes
No	Average	Poor	No
Yes	Good	Moderate	Yes
No	Very good	Good	Yes

Interactiveness:

$$\text{Entropy\_Info}(T, \text{Interactiveness})$$

$$= \frac{2}{4}\left[-\frac{2}{2}\log_2\frac{2}{2} - \frac{0}{2}\log_2\frac{0}{2}\right] + \frac{2}{4}\left[-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}\right]$$

$$= 0 + 0.4997$$

$$\text{Gain(Interactiveness)} = 0.8108 - 0.4997 = 0.3111$$

$$\text{Split\_Info}(T, \text{Interactiveness}) = -\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4} = 0.5 + 0.5 = 1$$

$$\text{Gain\_Ratio(Interactiveness)} = \frac{\text{Gain(Interactiveness)}}{\text{Split\_Info}(T, \text{Interactiveness})}$$

$$= \frac{0.3112}{1} = 0.3112$$

Step 2: Calculate the Entropy _Info, Gain(Info_Gain), Split_Info, Gain_Ratio for each of the attribute in the training dataset.

Inter active	Practical Knowledge	Comm Skills	Job Offer
Yes	Very good	Good	Yes
No	Average	Poor	No
Yes	Good	Moderate	Yes
No	Very good	Good	Yes

Practical Knowledge:

Entropy_Info(T, Practical Knowledge)

$$= \frac{2}{4}\left[-\frac{2}{2}\log_2\frac{2}{2} - \frac{0}{2}\log_2\frac{0}{2}\right] + \frac{1}{4}\left[-\frac{0}{1}\log_2\frac{0}{1} - \frac{1}{1}\log_2\frac{1}{1}\right] + \frac{1}{4}\left[-\frac{1}{1}\log_2\frac{1}{1} - \frac{0}{1}\log_2\frac{0}{1}\right] = 0$$

Gain(Practical Knowledge) $= 0.8108$

$$\text{Split\_Info}(T, \text{Practical Knowledge}) = -\frac{2}{4}\log_2\frac{2}{4} - \frac{1}{4}\log_2\frac{1}{4} - \frac{1}{4}\log_2\frac{1}{4} = 1.5$$

$$\text{Gain\_Ratio}(\text{Practical Knowledge}) = \frac{0.8108}{1.5} = 0.5408$$

Step 2: Calculate the Entropy _Info, Gain(Info_Gain), Split_Info, Gain_Ratio for each of the attribute in the training dataset.

Inter active	Practical Knowledge	Comm Skills	Job Offer
Yes	Very good	Good	Yes
No	Average	Poor	No
Yes	Good	Moderate	Yes
No	Very good	Good	Yes

Communication Skills:

$$\text{Entropy\_Info}(T, \text{Communication Skills}) = \frac{2}{4}\left[-\frac{2}{2}\log_2\frac{2}{2} - \frac{0}{2}\log_2\frac{0}{2}\right] + \frac{1}{4}\left[-\frac{0}{1}\log_2\frac{0}{1} - \frac{1}{1}\log_2\frac{1}{1}\right] + \frac{1}{4}\left[-\frac{1}{1}\log_2\frac{1}{1} - \frac{0}{1}\log_2\frac{0}{1}\right] = 0$$

$$\text{Gain(Communication Skills)} = 0.8108$$

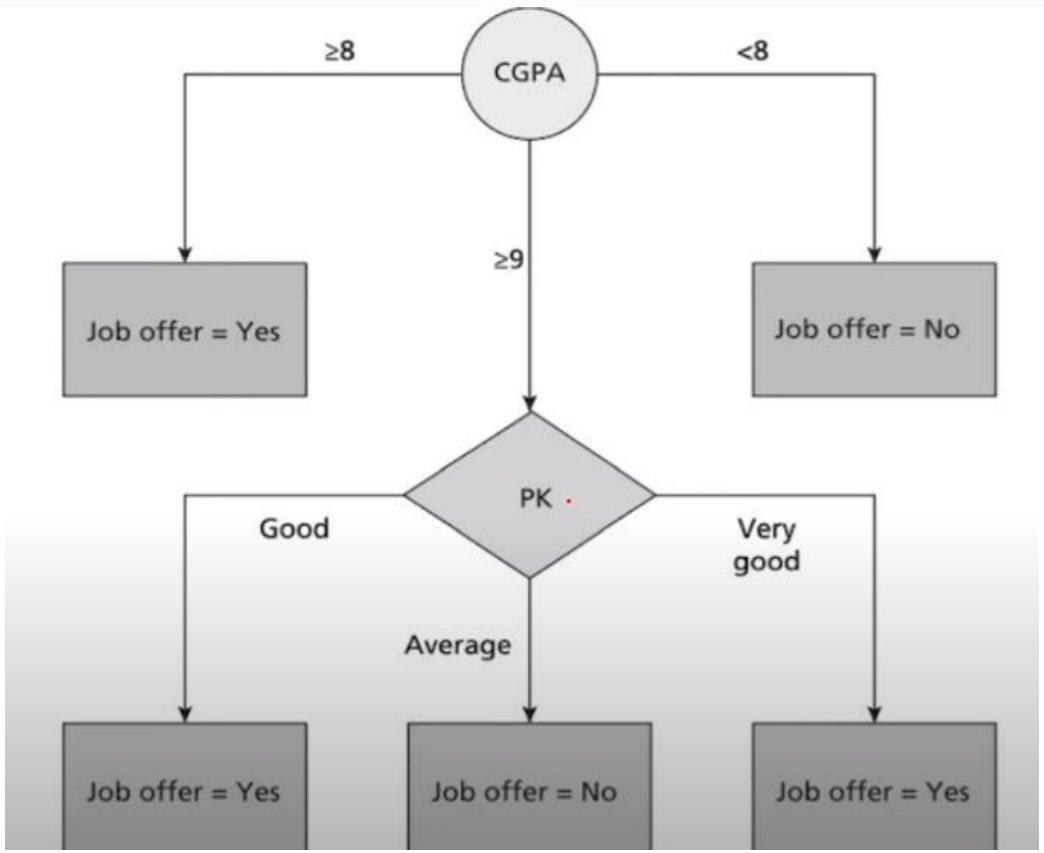
$$\text{Split\_Info}(T, \text{Communication Skills}) = -\frac{2}{4}\log_2\frac{2}{4} - \frac{1}{4}\log_2\frac{1}{4} - \frac{1}{4}\log_2\frac{1}{4} = 1.5$$

$$\text{Gain\_Ratio(Communication Skills)} = \frac{0.8108}{1.5} = 0.5408$$

Inter active	Practical Knowledge	Comm Skills	Job Offer
Yes	Very good	Good	Yes
No	Average	Poor	No
Yes	Good	Moderate	Yes
No	Very good	Good	Yes

Gain-Ratio

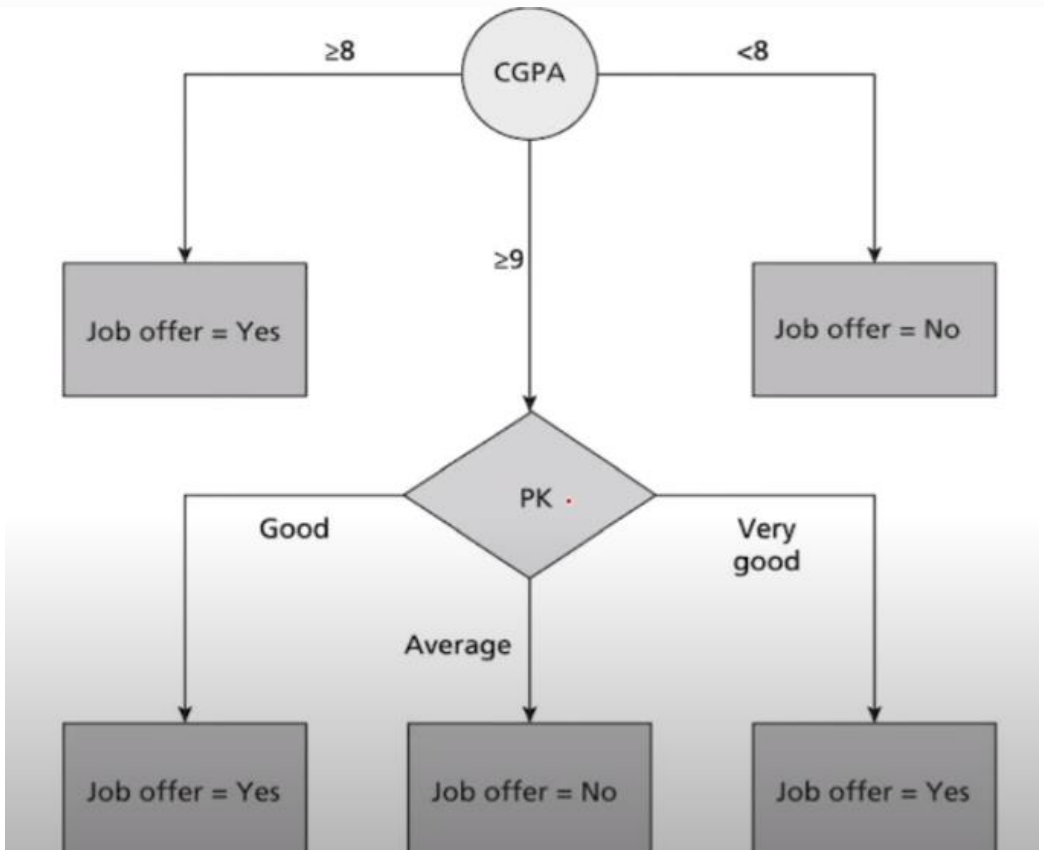
Attributes	Gain_Ratio
Interactiveness	0.3112
Practical Knowledge	0.5408
Communication Skills	0.5408



Inter active	Practical Knowledge	Comm Skills	Job Offer
Yes	Very good	Good	Yes
No	Average	Poor	No
Yes	Good	Moderate	Yes
No	Very good	Good	Yes

Gain-Ratio

Attributes	Gain_Ratio
Interactiveness	0.3112
Practical Knowledge	0.5408
Communication Skills	0.5408



Dealing with Continuous-Valued Attributes

Example. Temperature in the PlayTennis example

- Sort the examples according to Temperature

Temperature	40	48	60	72	80	90
PlayTennis	No	No	Yes	Yes	Yes	No

Temperature	40	48	60	72	80	90
PlayTennis	No	No	Yes	Yes	Yes	No

Dealing with Continuous-Valued Attributes

Example. Temperature in the PlayTennis example

- Sort the examples according to Temperature

Temperature	40	48	60	72	80	90
PlayTennis	No	No	Yes	Yes	Yes	No

Temperature	40	48	60	72	80	90
PlayTennis	No	No	Yes	Yes	Yes	No

Determine candidate thresholds by averaging consecutive values where there is a change in classification: **(48+60)/2=54** and **(80+90)/2=85**

Dealing with Continuous-Valued Attributes

Temperature	40	48	60	72	80	90
PlayTennis	No	No	Yes	Yes	Yes	No

Determine candidate thresholds by averaging consecutive values where there is a change in classification: **(48+60)/2=54** and **(80+90)/2=85**

Evaluate information gain of candidate thresholds (attributes) $\text{Temperature}_{>54}$ and $\text{Temperature}_{>85}$. Then Select the threshold based on the information gain.

Dealing with Continuous-Valued Attributes

Temperature	**40**	**48**	**60**	**72**	**80**	**90**
PlayTennis	**No**	**No**	**Yes**	**Yes**	**Yes**	**No**

Information gain of $Temperature_{>54}$

$$Values\ (Temp_{>54}) = <54, >54$$

$S = [3+, 3-]$ $\quad Entropy(S) = -\frac{3}{6} log_2 \frac{3}{6} - \frac{3}{6} log_2 \frac{3}{6} = 1.0$

$S_{<54} \leftarrow [0+, 2-]$ $\quad Entropy(S_{<54}) = 0.0$

$S_{>54} \leftarrow [3+, 1-]$ $\quad Entropy(S_{>54}) = -\frac{3}{4} log_2 \frac{3}{4} - \frac{1}{4} log_2 \frac{1}{4} = 0.8113$

$$Gain\ (S, Temp_{>54}) = Entropy(S) - \sum_{v \in \{<54, >54\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, Temp_{>54}) = Entropy(S) - \frac{2}{6} Entropy(S_{<54}) - \frac{4}{6} Entropy(S_{>54})$$

$$Gain(S, Temp_{>54}) = 1.0 - \frac{2}{6}\ 0.0 - \frac{4}{6}\ 0.8113 = 0.4591$$

Dealing with Continuous-Valued Attributes

Temperature	40	48	60	72	80	90
PlayTennis	No	No	Yes	Yes	Yes	No

Information gain of $Temperature_{>85}$

$Values\ (Temp_{>85}) = <85, >85$

$S = [3+, 3-]$ $\quad Entropy(S) = -\frac{3}{6} log_2 \frac{3}{6} - \frac{3}{6} log_2 \frac{3}{6} = 1.0$

$S_{<85} \leftarrow [3+, 2-]$ $\quad Entropy(S_{<85}) = -\frac{3}{5} log_2 \frac{3}{5} - \frac{2}{5} log_2 \frac{2}{5} = 0.971$

$S_{>85} \leftarrow [0+, 1-]$ $\quad Entropy(S_{>85}) = 0.0$

$$Gain\ (S, Temp_{>85}) = Entropy(S) - \sum_{v \in \{<85, >85\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, Temp_{>85}) = Entropy(S) - \frac{5}{6} Entropy(S_{<85}) - \frac{1}{6} Entropy(S_{>85})$$

$$Gain(S, Temp_{>85}) = 1.0 - \frac{5}{6}\ 0.971 - \frac{1}{6}\ 0.0 = 0.1908$$

Dealing with Continuous-Valued Attributes

Temperature	40	48	60	72	80	90
PlayTennis	No	No	Yes	Yes	Yes	No

$$Gain(S, Temp_{>54}) = 0.4591$$

$$Gain(S, Temp_{>85}) = 0.1908$$

How C4.5 Algorithm Handling Continuous Features

ID	color	root	sound	texture	umbilicus	surface	density	sugar	ripe
1	green	curly	muffled	clear	hollow	hard	0.697	0.460	true
2	dark	curly	dull	clear	hollow	hard	0.774	0.376	true
3	dark	curly	muffled	clear	hollow	hard	0.634	0.264	true
4	green	curly	dull	clear	hollow	hard	0.608	0.318	true
5	light	curly	muffled	clear	hollow	hard	0.556	0.215	true
6	green	slightly curly	muffled	clear	slightly hollow	soft	0.403	0.237	true
7	dark	slightly curly	muffled	slightly blurry	slightly hollow	soft	0.481	0.149	true
8	dark	slightly curly	muffled	clear	slightly hollow	hard	0.437	0.211	true
9	dark	slightly curly	dull	slightly blurry	slightly hollow	hard	0.666	0.091	false
10	green	straight	crisp	clear	flat	soft	0.243	0.267	false
11	light	straight	crisp	blurry	flat	hard	0.245	0.057	false
12	light	curly	muffled	blurry	flat	soft	0.343	0.099	false
13	green	slightly curly	muffled	slightly blurry	hollow	hard	0.639	0.161	false
14	light	slightly curly	dull	slightly blurry	hollow	hard	0.657	0.198	false
15	dark	slightly curly	muffled	clear	slightly hollow	soft	0.360	0.370	false
16	light	curly	muffled	blurry	flat	hard	0.593	0.042	false
17	green	curly	dull	slightly blurry	slightly hollow	hard	0.719	0.103	false

How C4.5 Algorithm Handling Continuous Features

ID	color	root	sound	texture	umbilicus	surface	density	sugar	ripe
1	green	curly	muffled	clear	hollow	hard	0.697	0.460	true
2	dark	curly	dull	clear	hollow	hard	0.774	0.376	true
3	dark	curly	muffled	clear	hollow	hard	0.634	0.264	true
4	green	curly	dull	clear	hollow	hard	0.608	0.318	true
5	light	curly	muffled	clear	hollow	hard	0.556	0.215	true
6	green	slightly curly	muffled	clear	slightly hollow	soft	0.403	0.237	true
7	dark	slightly curly	muffled	slightly blurry	slightly hollow	soft	0.481	0.149	true
8	dark	slightly curly	muffled	clear	slightly hollow	hard	0.437	0.211	true
9	dark	slightly curly	dull	slightly blurry	slightly hollow	hard	0.666	0.091	false
10	green	straight	crisp	clear	flat	soft	0.243	0.267	false
11	light	straight	crisp	blurry	flat	hard	0.245	0.057	false
12	light	curly	muffled	blurry	flat	soft	0.343	0.099	false
13	green	slightly curly	muffled	slightly blurry	hollow	hard	0.639	0.161	false
14	light	slightly curly	dull	slightly blurry	hollow	hard	0.657	0.198	false
15	dark	slightly curly	muffled	clear	slightly hollow	soft	0.360	0.370	false
16	light	curly	muffled	blurry	flat	hard	0.593	0.042	false
17	green	curly	dull	slightly blurry	slightly hollow	hard	0.719	0.103	false

- Given a data set *D* and a continuous feature *a*, suppose *n* values of *a* are observed in *D*, and we sort these values in ascending order, denoted by $\{a^1, a^2, \ldots, a^n\}$.
- With *a* split point *t*, *D* is partitioned into the subsets D_t^- and D_t^+, where D_t^- includes the samples with the value of *a* not greater than *t*, and D_t^+ includes the samples with the value of *a* greater than *t*.
- For adjacent feature values a^i and a^i+1, the partitions are identical for choosing any *t* in the interval [a^i , a^i+1).
- As a result, for continuous feature *a*, there are *n−1* elements in the following set of candidate split points:

$$T_a = \left\{ \frac{a^i + a^{i+1}}{2} \mid 1 \leqslant i \leqslant n-1 \right\}$$

How C4.5 Algorithm Handling Continuous Features

$$T_a = \left\{ \frac{a^i + a^{i+1}}{2} \mid 1 \leqslant i \leqslant n-1 \right\}$$

- where the midpoint is used as the candidate split point for the interval [a^i , a^i+1).
- Then, the split points are examined in the same way as discrete features, and the optimal split points are selected for splitting nodes.
- We can modify the information gain equation from this

$$\text{Gain}(D, a) = \text{Ent}(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} \text{Ent}(D^v)$$

$$\begin{aligned} \text{Gain}(D, a) &= \max_{t \in T_a} \text{Gain}(D, a, t) \\ &= \max_{t \in T_a} \text{Ent}(D) - \sum_{\lambda \in \{-,+\}} \frac{|D_t^\lambda|}{|D|} \text{Ent}(D_t^\lambda) \end{aligned}$$

where Gain(*D, a, t*) is the information gain of bi-partitioning *D* by *t*, and the split point with the largest Gain(*D, a, t*) is selected.

- watermelon data set.

ID	color	root	sound	texture	umbilicus	surface	density	sugar	ripe
1	green	curly	muffled	clear	hollow	hard	0.697	0.460	true
2	dark	curly	dull	clear	hollow	hard	0.774	0.376	true
3	dark	curly	muffled	clear	hollow	hard	0.634	0.264	true
4	green	curly	dull	clear	hollow	hard	0.608	0.318	true
5	light	curly	muffled	clear	hollow	hard	0.556	0.215	true
6	green	slightly curly	muffled	clear	slightly hollow	soft	0.403	0.237	true
7	dark	slightly curly	muffled	slightly blurry	slightly hollow	soft	0.481	0.149	true
8	dark	slightly curly	muffled	clear	slightly hollow	hard	0.437	0.211	true
9	dark	slightly curly	dull	slightly blurry	slightly hollow	hard	0.666	0.091	false
10	green	straight	crisp	clear	flat	soft	0.243	0.267	false
11	light	straight	crisp	blurry	flat	hard	0.245	0.057	false
12	light	curly	muffled	blurry	flat	soft	0.343	0.099	false
13	green	slightly curly	muffled	slightly blurry	hollow	hard	0.639	0.161	false
14	light	slightly curly	dull	slightly blurry	hollow	hard	0.657	0.198	false
15	dark	slightly curly	muffled	clear	slightly hollow	soft	0.360	0.370	false
16	light	curly	muffled	blurry	flat	hard	0.593	0.042	false
17	green	curly	dull	slightly blurry	slightly hollow	hard	0.719	0.103	false

$$\text{Gain}(D, \text{color}) = 0.109; \quad \text{Gain}(D, \text{root}) = 0.143;$$
$$\text{Gain}(D, \text{sound}) = 0.141; \quad \text{Gain}(D, \text{texture}) = 0.381;$$
$$\text{Gain}(D, \text{umbilicus}) = 0.289; \quad \text{Gain}(D, \text{surface}) = 0.006;$$
$$\text{Gain}(D, \text{density}) = 0.262; \quad \text{Gain}(D, \text{sugar}) = 0.349.$$

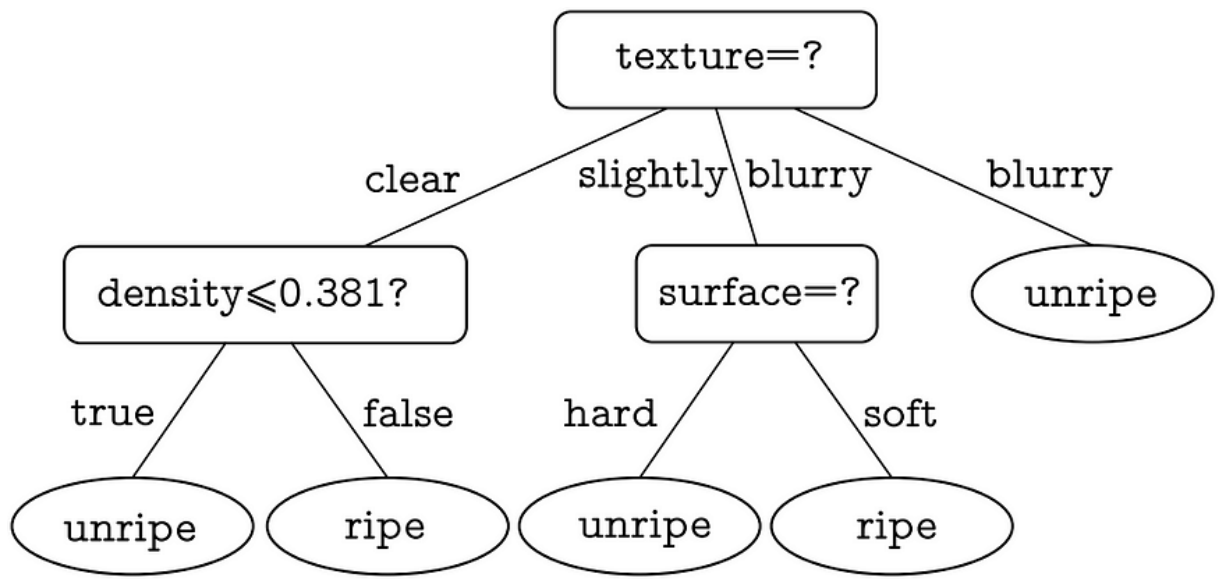
- At the beginning, all 17 training samples have different density values. According to the split equation above, the candidate split point set includes 16 values:
- T_density = {0.244, 0.294, 0.351, 0.381, 0.420, 0.459, 0.518, 0.574, 0.600, 0.621, 0.636, 0.648, 0.661, 0.681, 0.708, 0.746}.
- According to the modified information gain equation above, the information gain of density is 0.262, and the corresponding split point is 0.381.

- For the feature sugar, its candidate split point set includes 16 values:
- T_sugar = {0.049, 0.074, 0.095, 0.101, 0.126, 0.155, 0.179, 0.204, 0.213, 0.226, 0.250, 0.265, 0.292, 0.344, 0.373, 0.418}.
- Similarly, the information gain of sugar is 0.349 and the corresponding split point is 0.126.
- Combining the results, the information gains of features in table above are

ID	color	root	sound	texture	umbilicus	surface	density	sugar	ripe
1	green	curly	muffled	clear	hollow	hard	0.697	0.460	true
2	dark	curly	dull	clear	hollow	hard	0.774	0.376	true
3	dark	curly	muffled	clear	hollow	hard	0.634	0.264	true
4	green	curly	dull	clear	hollow	hard	0.608	0.318	true
5	light	curly	muffled	clear	hollow	hard	0.556	0.215	true
6	green	slightly curly	muffled	clear	slightly hollow	soft	0.403	0.237	true
7	dark	slightly curly	muffled	slightly blurry	slightly hollow	soft	0.481	0.149	true
8	dark	slightly curly	muffled	clear	slightly hollow	hard	0.437	0.211	true
9	dark	slightly curly	dull	slightly blurry	slightly hollow	hard	0.666	0.091	false
10	green	straight	crisp	clear	flat	soft	0.243	0.267	false
11	light	straight	crisp	blurry	flat	hard	0.245	0.057	false
12	light	curly	muffled	blurry	flat	soft	0.343	0.099	false
13	green	slightly curly	muffled	slightly blurry	hollow	hard	0.639	0.161	false
14	light	slightly curly	dull	slightly blurry	hollow	hard	0.657	0.198	false
15	dark	slightly curly	muffled	clear	slightly hollow	soft	0.360	0.370	false
16	light	curly	muffled	blurry	flat	hard	0.593	0.042	false
17	green	curly	dull	slightly blurry	slightly hollow	hard	0.719	0.103	false

$$\text{Gain}(D, \text{color}) = 0.109; \quad \text{Gain}(D, \text{root}) = 0.143;$$
$$\text{Gain}(D, \text{sound}) = 0.141; \quad \text{Gain}(D, \text{texture}) = 0.381;$$
$$\text{Gain}(D, \text{umbilicus}) = 0.289; \quad \text{Gain}(D, \text{surface}) = 0.006;$$
$$\text{Gain}(D, \text{density}) = 0.262; \quad \text{Gain}(D, \text{sugar}) = 0.349.$$

Since splitting by texture has the largest information gain, it is selected as the splitting feature for the root node.
The splitting process proceeds recursively, and the final decision tree is shown in figure below:



C4.5 Example

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	85	85	Weak	No
2	Sunny	80	90	Strong	No
3	Overcast	83	78	Weak	Yes
4	Rain	70	96	Weak	Yes
5	Rain	68	80	Weak	Yes
6	Rain	65	70	Strong	No
7	Overcast	64	65	Strong	Yes
8	Sunny	72	95	Weak	No
9	Sunny	69	70	Weak	Yes
10	Rain	75	80	Weak	Yes
11	Sunny	75	70	Strong	Yes
12	Overcast	72	90	Strong	Yes
13	Overcast	81	75	Weak	Yes
14	Rain	71	80	Strong	No

C4.5 Example

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	85	85	Weak	No
2	Sunny	80	90	Strong	No
3	Overcast	83	78	Weak	Yes
4	Rain	70	96	Weak	Yes
5	Rain	68	80	Weak	Yes
6	Rain	65	70	Strong	No
7	Overcast	64	65	Strong	Yes
8	Sunny	72	95	Weak	No
9	Sunny	69	70	Weak	Yes
10	Rain	75	80	Weak	Yes
11	Sunny	75	70	Strong	Yes
12	Overcast	72	90	Strong	Yes
13	Overcast	81	75	Weak	Yes
14	Rain	71	80	Strong	No

Entropy(Decision) = ∑ — p(I) . log p(I) = — p(Yes) . log p(Yes) — p(No) . log2(No) = — (9/14) . log(9/14) — (5/14) . log(5/14) = 0.940

Here, we need to calculate gain ratios instead of gains.

GainRatio(A) = Gain(A) / SplitInfo(A)

SplitInfo(A) = -∑ |Dj|/|D| x log|Dj|/|D|

C4.5 Example

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	85	85	Weak	No
2	Sunny	80	90	Strong	No
3	Overcast	83	78	Weak	Yes
4	Rain	70	96	Weak	Yes
5	Rain	68	80	Weak	Yes
6	Rain	65	70	Strong	No
7	Overcast	64	65	Strong	Yes
8	Sunny	72	95	Weak	No
9	Sunny	69	70	Weak	Yes
10	Rain	75	80	Weak	Yes
11	Sunny	75	70	Strong	Yes
12	Overcast	72	90	Strong	Yes
13	Overcast	81	75	Weak	Yes
14	Rain	71	80	Strong	No

Entropy(Decision) = ∑ — p(I) . log p(I) =
— p(Yes) . log p(Yes) — p(No) . log2(No) =
— (9/14) . log(9/14) — (5/14) . log(5/14) =
0.940
Here, we need to calculate gain ratios instead of gains.
GainRatio(A) = Gain(A) / SplitInfo(A)
SplitInfo(A) = -∑ |Dj|/|D| x log|Dj|/|D|

Let's calculate for Wind Attribute:
Gain(Decision, Wind) = Entropy(Decision) — ∑ (p(Decision|Wind) . Entropy(Decision|Wind))

Gain(Decision, Wind) = Entropy(Decision) — [p(Decision|Wind=Weak) . Entropy(Decision|Wind=Weak)] + [p(Decision|Wind=Strong) . Entropy(Decision|Wind=Strong)]

Entropy(Decision|Wind=Weak) = — p(No) . logp(No) — p(Yes) . logp(Yes) = — (2/8) . log(2/8) — (6/8) . log(6/8) = 0.811
Entropy(Decision|Wind=Strong) = — (3/6) . log(3/6) — (3/6) . log(3/6) = 1
Gain(Decision, Wind) = 0.940 — (8/14).(0.811) — (6/14).(1) = 0.940–0.463–0.428 = 0.049

There are 8 decisions for weak wind, and 6 decisions for strong wind.
SplitInfo(Decision, Wind) = -(8/14).log(8/14) — (6/14).log(6/14) = 0.461 + 0.524 = 0.985

GainRatio(Decision, Wind) = Gain(Decision, Wind) / SplitInfo(Decision, Wind) = 0.049 / 0.985 = 0.049

C4.5 Example

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	85	85	Weak	No
2	Sunny	80	90	Strong	No
3	Overcast	83	78	Weak	Yes
4	Rain	70	96	Weak	Yes
5	Rain	68	80	Weak	Yes
6	Rain	65	70	Strong	No
7	Overcast	64	65	Strong	Yes
8	Sunny	72	95	Weak	No
9	Sunny	69	70	Weak	Yes
10	Rain	75	80	Weak	Yes
11	Sunny	75	70	Strong	Yes
12	Overcast	72	90	Strong	Yes
13	Overcast	81	75	Weak	Yes
14	Rain	71	80	Strong	No

Similarly, for wind, Outlook, Humidity<> 80 and Temperature <>83

let's consider gain as splitting criterion and request you to please follow same steps with Gain Ratio.

Attribute	Gain	Gain Ratio
Wind	0.049	0.049
Outlook	0.246	0.155
Humidity <> 80	0.101	0.107
Temperature <> 83	0.113	0.305

If we will use gain, then outlook will be the root node because it has the highest gain value.

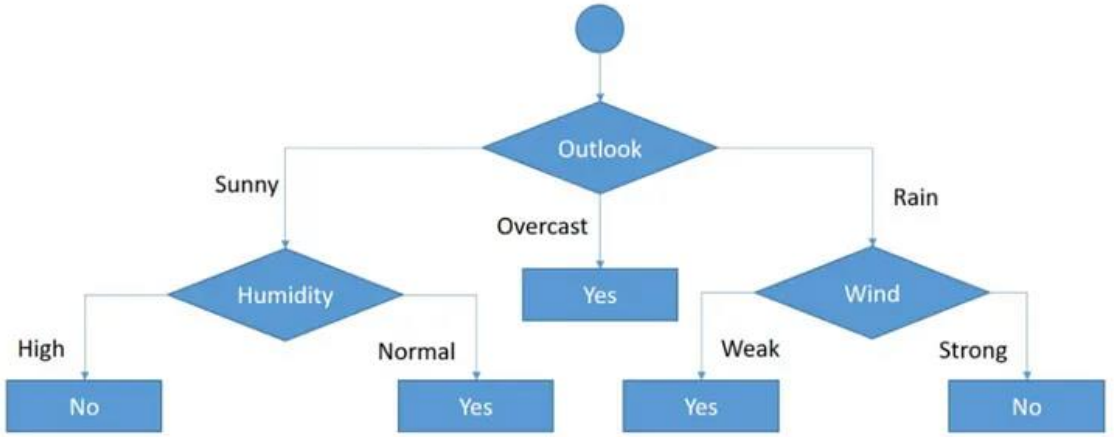
Performs similar steps for all attributes over outlook and the resultant tree looks like:

C4.5 Example

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	85	85	Weak	No
2	Sunny	80	90	Strong	No
3	Overcast	83	78	Weak	Yes
4	Rain	70	96	Weak	Yes
5	Rain	68	80	Weak	Yes
6	Rain	65	70	Strong	No
7	Overcast	64	65	Strong	Yes
8	Sunny	72	95	Weak	No
9	Sunny	69	70	Weak	Yes
10	Rain	75	80	Weak	Yes
11	Sunny	75	70	Strong	Yes
12	Overcast	72	90	Strong	Yes
13	Overcast	81	75	Weak	Yes
14	Rain	71	80	Strong	No

Attribute	Gain	Gain Ratio
Wind	0.049	0.049
Outlook	0.246	0.155
Humidity <> 80	0.101	0.107
Temperature <> 83	0.113	0.305

If we will use gain, then outlook will be the root node because it has the highest gain value.
Performs similar steps for all attributes over outlook and the resultant tree looks like:



Improvements in C4.5 over ID3

- · Handling both continuous and discrete
- · Handling training data with missing attribute values
- · Handling attributes with differing costs.
- · Pruning trees after creation

Limitations:

The limitations of C4. 5 is **its information entropy**, it gives poor results for larger distinct attributes.

Algorithm	Splitting Criteria of algorithm	Attribute types Managed by algorithm	Pruning Strategy of algorithm	Outlier Detection	Missing values	Invented By
ID3	Information Gain	Manages only Categorical value	No pruning is done	No pruning is done	Do not Manages missing values.	invented by Ross Quinlan
C4.5	Gain Ratio	Manages both Categorical and Numeric value	Error Based pruning is used	Error Based pruning is used	Manages missing values.	developed by Ross Quinlan

CART (Classification and Regression Trees)

- CART is a newer decision tree algorithm that was developed in the late 1980s.
- CART can be used for both categorical and continuous target variables.
- CART uses the Gini impurity metric to determine the best feature to split on at each node of the tree.
- Mathematically, we can write Gini Impurity as follows:

$$Gini = 1 - \sum_{i=1}^{n} (p_i)^2$$

- where p_i is the probability of an object being classified to a particular class.

Step 1: Calculate the Gini_Index for the dataset. The target attribute 'Job Offer' has 7 instances as Yes and 3 instances as No.

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

Step 1: Calculate the Gini_Index for the dataset. The target attribute 'Job Offer' has 7 instances as Yes and 3 instances as No.

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

$$\text{Gini\_Index}(T) = 1 - \sum_{i=1}^{m} P_i^2$$

$$\text{Gini\_Index}(T) = 1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2$$

$$= 1 - 0.49 - 0.09$$

$$= 1 - 0.58$$

$$\text{Gini\_Index}(T) = 0.42$$

Step 2: Compute Gini_Index for each of the attribute and each of the subset in the attribute.

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

Categories of CGPA

CGPA	Job Offer = Yes	Job Offer = No
≥9	3	1
≥8	4	0
<8	0	2

Step 2: Compute Gini_Index for each of the attribute and each of the subset in the attribute.

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

Categories of CGPA

CGPA	Job Offer = Yes	Job Offer = No
≥9	3	1
≥8	4	0
<8	0	2

$$\text{Gini\_Index}(T) = 1 - \sum_{i=1}^{m} P_i^2$$

$$\text{Gini\_Index}(T, A) = \frac{|S_1|}{|T|}\text{Gini}(S_1) + \frac{|S_2|}{|T|}\text{Gini}(S_2)$$

Step 2: Compute Gini_Index for each of the attribute and each of the subset in the attribute.

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

Categories of CGPA

CGPA	Job Offer = Yes	Job Offer = No
≥9	3	1
≥8	4	0
<8	0	2

$$\text{Gini\_Index}(T) = 1 - \sum_{i=1}^{m} P_i^2$$

$$\text{Gini\_Index}(T, A) = \frac{|S_1|}{|T|}\text{Gini}(S_1) + \frac{|S_2|}{|T|}\text{Gini}(S_2)$$

{}, {>=9}, {>=8}, {>8}, {>=9,>=8}, {>=9, >8} ,

{>=8, >8} and {>=9, >=8, >8}

Step 2: Compute Gini_Index for each of the attribute and each of the subset in the attribute.

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

{}, {>=9}, {>=8}, {>8}, {>=9,>=8}, {>=9, >8} ,

{>=8, >8} and {>=9, >=8, >8}

Categories of CGPA

CGPA	Job Offer = Yes	Job Offer = No
≥9	3	1
≥8	4	0
<8	0	2

$$\text{Gini\_Index}(T) = 1 - \sum_{i=1}^{m} P_i^2$$

$$\text{Gini\_Index}(T, A) = \frac{|S_1|}{|T|}\text{Gini}(S_1) + \frac{|S_2|}{|T|}\text{Gini}(S_2)$$

$$\begin{aligned}\text{Gini\_Index}(T, \text{CGPA} \in \{\geq 9, \geq 8\}) &= 1 - (7/8)^2 - (1/8)^2 \\ &= 1 - 0.7806 \\ &= 0.2194\end{aligned}$$

$$\begin{aligned}\text{Gini\_Index}(T, \text{CGPA} \in \{<8\}) &= 1 - (0/2)^2 - (2/2)^2 \\ &= 1 - 1 \\ &= 0\end{aligned}$$

$$\begin{aligned}\text{Gini\_Index}(T, \text{CGPA} \in \{(\geq 9, \geq 8), <8\} &= (8/10) \times 0.2194 + (2/10) \times 0 \\ &= 0.17552\end{aligned}$$

Step 2: Compute Gini_Index for each of the attribute and each of the subset in the attribute.

Categories of CGPA

CGPA	Job Offer = Yes	Job Offer = No
≥9	3	1
≥8	4	0
<8	0	2

$$\text{Gini\_Index}(T) = 1 - \sum_{i=1}^{m} P_i^2$$

$$\text{Gini\_Index}(T, A) = \frac{|S_1|}{|T|}\text{Gini}(S_1) + \frac{|S_2|}{|T|}\text{Gini}(S_2)$$

$$\begin{aligned}
\text{Gini_Index}(T, \text{CGPA} \in \{\geq 9, \geq 8\}) &= 1 - (7/8)^2 - (1/8)^2 \\
&= 1 - 0.7806 \\
&= 0.2194 \\
\text{Gini_Index}(T, \text{CGPA} \in \{<8\}) &= 1 - (0/2)^2 - (2/2)^2 \\
&= 1 - 1 \\
&= 0 \\
\text{Gini_Index}(T, \text{CGPA} \in \{(\geq 9, \geq 8), <8\} &= (8/10) \times 0.2194 + (2/10) \times 0 \\
&= 0.17552 \\
\text{Gini_Index}(T, \text{CGPA} \in \{\geq 9, <8\}) &= 1 - (3/6)^2 - (3/6)^2 \\
&= 1 - 0.5 = 0.5 \\
\text{Gini_Index}(T, \text{CGPA} \in \{\geq 8\}) &= 1 - (4/4)^2 - (0/4)^2 \\
&= 1 - 1 = 0 \\
\text{Gini_Index}(T, \text{CGPA} \in \{(\geq 9, <8), \geq 8\}) &= (6/10) \times 0.5 + (4/10) \times 0 \\
&= 0.3
\end{aligned}$$

Step 2: Compute Gini_Index for each of the attribute and each of the subset in the attribute.

Categories of CGPA

CGPA	Job Offer = Yes	Job Offer = No
≥9	3	1
≥8	4	0
<8	0	2

$$\text{Gini\_Index}(T, \text{CGPA} \in \{\geq 8, <8\}) = 1 - (4/6)^2 - (2/6)^2$$
$$= 1 - 0.555$$
$$= 0.445$$
$$\text{Gini\_Index}(T, \text{CGPA} \in \{\geq 9\}) = 1 - (3/4)^2 - (1/4)^2$$
$$= 1 - 0.625$$
$$= 0.375$$
$$\text{Gini\_Index}(T, \text{CGPA} \in \{(\geq 8, <8), \geq 9\}) = (6/10) \times 0.445 + (4/10) \times 0.375$$
$$= 0.417$$

Step 2: Compute Gini_Index for each of the attribute and each of the subset in the attribute.

Categories of CGPA

CGPA	Job Offer = Yes	Job Offer = No
≥9	3	1
≥8	4	0
<8	0	2

$$\text{Gini\_Index}(T, \text{CGPA} \in \{\geq 8, <8\}) = 1 - (4/6)^2 - (2/6)^2$$
$$= 1 - 0.555$$
$$= 0.445$$
$$\text{Gini\_Index}(T, \text{CGPA} \in \{\geq 9\}) = 1 - (3/4)^2 - (1/4)^2$$
$$= 1 - 0.625$$
$$= 0.375$$
$$\text{Gini\_Index}(T, \text{CGPA} \in \{(\geq 8, <8), \geq 9\}) = (6/10) \times 0.445 + (4/10) \times 0.375$$
$$= 0.417$$

Categories of CGPA

CGPA	Job Offer = Yes	Job Offer = No
≥9	3	1
≥8	4	0
<8	0	2

Gini_Index of CGPA

Subsets		Gini_Index
(≥9, ≥8)	<8	0.1755
(≥9, <8)	≥8	0.3
(≥8, <8)	≥9	0.417

Step 2: Compute Gini_Index for each of the attribute and each of the subset in the attribute.

Categories of CGPA

CGPA	Job Offer = Yes	Job Offer = No
≥9	3	1
≥8	4	0
<8	0	2

Gini_Index of CGPA

Subsets		Gini_Index
(≥9, ≥8)	<8	0.1755
(≥9, <8)	≥8	0.3
(≥8, <8)	≥9	0.417

Step 3: Choose the best splitting subset which has **minimum Gini_Index** for an attribute.

The subset **CGPA = {(>=9, >=8), <8}** has the lowest Gini_Index value as 0.1755 is chosen as the best splitting subset.

Step 2: Compute Gini_Index for each of the attribute and each of the subset in the attribute.

Categories of CGPA

CGPA	Job Offer = Yes	Job Offer = No
≥9	3	1
≥8	4	0
<8	0	2

Gini_Index of CGPA

Subsets		Gini_Index
(≥9, ≥8)	<8	0.1755
(≥9, <8)	≥8	0.3
(≥8, <8)	≥9	0.417

Step 3: Choose the best splitting subset which has **minimum Gini_Index** for an attribute.

The subset **CGPA = {(>=9, >=8), <8}** has the lowest Gini_Index value as 0.1755 is chosen as the best splitting subset.

Step 4: Compute **ΔGini or best splitting subset** of that attribute.

$$\Delta\text{Gini(CGPA)} = \text{Gini}(T) - \text{Gini}(T, \text{CGPA})$$
$$= 0.42 - 0.1755$$
$$= 0.2445$$

$$\text{Gini\_Index}(T) = 1 - \sum_{i=1}^{m} P_i^2$$

$$\text{Gini\_Index}(T) = 1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2$$
$$= 1 - 0.49 - 0.09$$
$$= 1 - 0.58$$
$$\text{Gini\_Index}(T) = 0.42$$

Repeat the same process for the remaining attributes in the dataset such as for Interactiveness, Practical Knowledge, and Communication Skills.

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

Categories for Interactiveness

Interactiveness	Job Offer = Yes	Job Offer = No
Yes	5	1
No	2	2

Repeat the same process for the remaining attributes in the dataset such as for Interactiveness, Practical Knowledge, and Communication Skills.

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

Categories for Interactiveness

Interactiveness	Job Offer = Yes	Job Offer = No
Yes	5	1
No	2	2

$$\text{Gini\_Index}(T, \text{Interactiveness} \in \{\text{Yes}\}) = 1 - \left(\frac{5}{6}\right)^2 - \left(\frac{1}{6}\right)^2$$
$$= 1 - 0.72$$
$$= 0.28$$

$$\text{Gini\_Index}(T, \text{Interactiveness} \in \{\text{No}\}) = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2$$
$$= 1 - 0.5$$
$$= 0.5$$

$$\text{Gini\_Index}(T, \text{Interactiveness} \in \{\text{Yes, No}\}) = \frac{6}{10}(0.28) + \frac{4}{10}(0.5)$$
$$= 0.168 + 0.2$$
$$= 0.368$$

$$\Delta\text{Gini}(\text{Interactiveness}) = \text{Gini}(T) - \text{Gini}(T, \text{Interactiveness})$$
$$= 0.42 - 0.368$$
$$= 0.052$$

Repeat the same process for the remaining attributes in the dataset such as for Interactiveness, Practical Knowledge, and Communication Skills.

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

Categories for Practical Knowledge

Practical Knowledge	Job Offer = Yes	Job Offer = No
Very Good	2	0
Good	4	1
Average	1	2

Repeat the same process for the remaining attributes in the dataset such as for Interactiveness, Practical Knowledge, and Communication Skills.

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

Categories for Practical Knowledge

Practical Knowledge	Job Offer = Yes	Job Offer = No
Very Good	2	0
Good	4	1
Average	1	2

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Very Good, Good}\}) = 1 - \left(\frac{6}{7}\right)^2 - \left(\frac{1}{7}\right)^2$$

$$= 0.7544$$

$$= 0.2456$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Average}\}) = 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2$$

$$= 1 - 0.555 = 0.445$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Very Good, Good}\}, \text{Average}) = \left(\frac{7}{10}\right)^2 \times 0.2456 + \left(\frac{3}{10}\right) \times 0.445$$

$$= 0.3054$$

Repeat the same process for the remaining attributes in the dataset such as for Interactiveness, Practical Knowledge, and Communication Skills.

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

Categories for Practical Knowledge

Practical Knowledge	Job Offer = Yes	Job Offer = No
Very Good	2	0
Good	4	1
Average	1	2

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Very Good, Average}\}) = 1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2$$
$$= 1 - 0.52$$
$$= 0.48$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Good}\}) = 1 - \left(\frac{4}{5}\right)^2 - \left(\frac{1}{5}\right)^2$$
$$= 1 - 0.68$$
$$= 0.32$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Very Good, Average}\}, \text{Good})$$
$$= \left(\frac{5}{10}\right) \times 0.48 + \left(\frac{5}{10}\right) \times 0.32 = 0.40$$

Repeat the same process for the remaining attributes in the dataset such as for Interactiveness, Practical Knowledge, and Communication Skills.

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

Categories for Practical Knowledge

Practical Knowledge	Job Offer = Yes	Job Offer = No
Very Good	2	0
Good	4	1
Average	1	2

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Good, Average}\}, \text{Very Good}) = \left(\frac{8}{10}\right) \times 0.4688 + \left(\frac{2}{10}\right) \times 0 = 0.3750$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Good, Average}\}) = 1 - \left(\frac{5}{8}\right)^2 - \left(\frac{3}{8}\right)^2 = 1 - 0.5312 = 0.4688$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Very Good}\}) = 1 - \left(\frac{2}{2}\right)^2 - \left(\frac{0}{2}\right)^2 = 1 - 1 = 0$$

Repeat the same process for the remaining attributes in the dataset such as for Interactiveness, Practical Knowledge, and Communication Skills.

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

Categories for Practical Knowledge

Practical Knowledge	Job Offer = Yes	Job Offer = No
Very Good	2	0
Good	4	1
Average	1	2

Categories for Practical Knowledge

Practical Knowledge	Job Offer = Yes	Job Offer = No
Very Good	2	0
Good	4	1
Average	1	2

Gini_Index for Practical Knowledge

Subsets		Gini_Index
(Very Good, Good)	Average	0.3054
(Very Good, Average)	Good	0.40
(Good, Average)	Very Good	0.3750

$$\Delta\text{Gini}(\text{Practical Knowledge}) = \text{Gini}(T) - \text{Gini}(T, \text{Practical Knowledge})$$
$$= 0.42 - 0.3054 = 0.1146$$

Categories for Communication Skills

Communication Skills	Job Offer = Yes	Job Offer = No
Good	4	1
Moderate	3	0
Poor	0	2

$$\text{Gini\_Index}(T, \text{Communication Skills} \in \{\text{Good, Moderate}\}) = 1 - \left(\frac{7}{8}\right)^2 - \left(\frac{1}{8}\right)^2$$

$$= 1 - 0.7806$$

$$= 0.2194$$

$$\text{Gini\_Index}(T, \text{Communication Skills} \in \{\text{Poor}\}) = 1 - \left(\frac{2}{2}\right)^2 - \left(\frac{0}{2}\right)^2$$

$$= 1 - 1 = 0$$

$$\text{Gini\_Index}(T, \text{Communication Skills} \in \{\text{Good, Moderate}\}, \text{Poor})$$

$$= \left(\frac{8}{10}\right) \times 0.2194 + \left(\frac{2}{10}\right) \times 0 = 0.1755$$

Categories for Communication Skills

Communication Skills	Job Offer = Yes	Job Offer = No
Good	4	1
Moderate	3	0
Poor	0	2

$$\text{Gini\_Index}(T, \text{Communication Skills} \in \{\text{Good, Poor}\}) = 1 - \left(\frac{4}{7}\right)^2 - \left(\frac{3}{7}\right)^2$$
$$= 1 - 0.5101$$
$$= 0.4899$$

$$\text{Gini\_Index}(T, \text{Communication Skills} \in \{\text{Moderate}\}) = 1 - \left(\frac{3}{3}\right)^2 - \left(\frac{0}{3}\right)^2$$
$$= 1 - 1 = 0$$

$$\text{Gini\_Index}(T, \text{Communication Skills} \in \{\text{Good, Poor}\}, \text{Moderate})$$
$$= \left(\frac{7}{10}\right) \times 0.4899 + \left(\frac{3}{10}\right) \times 0 = 0.3429$$

Categories for Communication Skills

Communication Skills	Job Offer = Yes	Job Offer = No
Good	4	1
Moderate	3	0
Poor	0	2

$$\text{Gini\_Index}(T, \text{Communication Skills} \in \{\text{Moderate, Poor}\}, \text{Good}) = \left(\frac{5}{10}\right)^2 \times 048 + \left(\frac{5}{10}\right)^2 \times 0.32 = 0.40$$

$$\text{Gini\_Index}(T, \text{Communication Skills} \in \{\text{Moderate, Poor}\}) = 1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2 = 1 - 0.52 = 0.48$$

$$\text{Gini\_Index}(T, \text{Communication Skills} \in \{\text{Good}\}) = 1 - \left(\frac{4}{5}\right)^2 - \left(\frac{1}{5}\right)^2 = 1 - 0.68 = 0.32$$

Categories for Communication Skills

Communication Skills	Job Offer = Yes	Job Offer = No
Good	4	1
Moderate	3	0
Poor	0	2

Gini-Index for Subsets of Communication Skills

Subsets		Gini_Index
(Good, Moderate)	Poor	0.1755
(Good, Poor)	Moderate	0.3429
(Moderate, Poor)	Good	0.40

$$\Delta\text{Gini}(\text{Communication Skills}) = \text{Gini}(T) - \text{Gini}(T, \text{Communication Skills})$$
$$= 0.42 - 0.1755$$
$$= 0.2445$$

Gini_Index and ΔGini for all Attributes

Attribute	Gini_Index	ΔGini
CGPA	0.1755	0.2445 ✓
Interactiveness	0.368	0.052
Practical knowledge	0.3054	0.1146
Communication Skills	0.1755	0.2445 ✓

Categories for Communication Skills

Communication Skills	Job Offer = Yes	Job Offer = No
Good	4	1
Moderate	3	0
Poor	0	2

Gini-Index for Subsets of Communication Skills

Subsets		Gini_Index
(Good, Moderate)	Poor	0.1755
(Good, Poor)	Moderate	0.3429
(Moderate, Poor)	Good	0.40

$$\Delta\text{Gini}(\text{Communication Skills}) = \text{Gini}(T) - \text{Gini}(T, \text{Communication Skills})$$
$$= 0.42 - 0.1755$$
$$= 0.2445$$

Gini_Index and ΔGini for all Attributes

Attribute	Gini_Index	ΔGini
CGPA	0.1755	0.2445 ✓
Interactiveness	0.368	0.052
Practical knowledge	0.3054	0.1146
Communication Skills	0.1755	0.2445 ✓

Subset **CGPA = {(>=9, >=8), <8}**

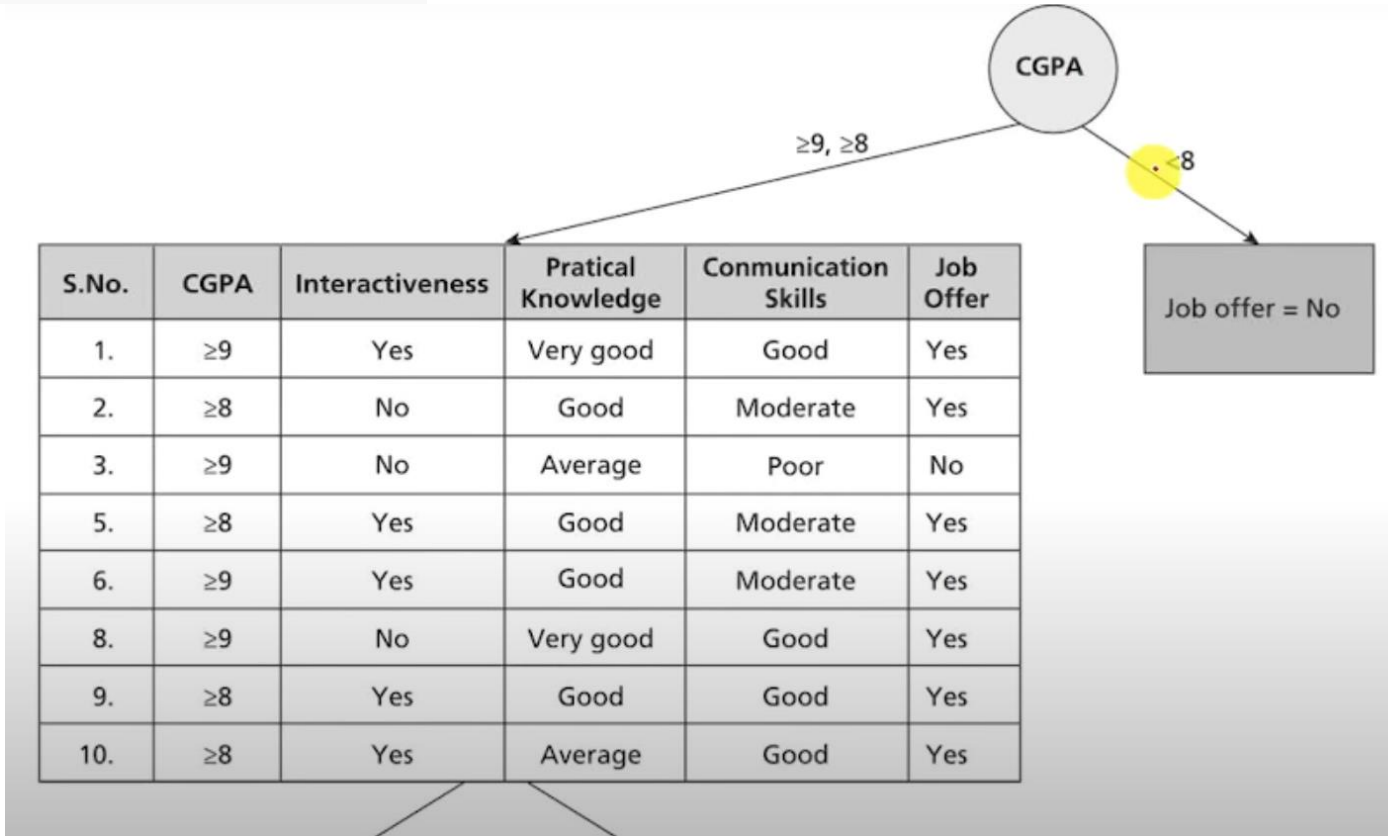
is the best splitting subset

Gini_Index and ΔGini for all Attributes

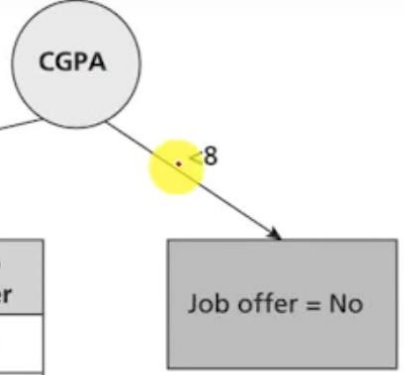
Attribute	Gini_Index	ΔGini
CGPA	0.1755	0.2445 ✓
Interactiveness	0.368	0.052
Practical knowledge	0.3054	0.1146
Communication Skills	0.1755	0.2445 ✓

Subset CGPA = {(>=9, >=8), <8}

is the best splitting subset



S.No.	CGPA	Interactiveness	Pratical Knowledge	Conmunication Skills	Job Offer
1.	≥9	Yes	Very good	Good	Yes
2.	≥8	No	Good	Moderate	Yes
3.	≥9	No	Average	Poor	No
5.	≥8	Yes	Good	Moderate	Yes
6.	≥9	Yes	Good	Moderate	Yes
8.	≥9	No	Very good	Good	Yes
9.	≥8	Yes	Good	Good	Yes
10.	≥8	Yes	Average	Good	Yes



S.No.	CGPA	Interactiveness	Pratical Knowledge	Conmunication Skills	Job Offer
1.	≥9	Yes	Very good	Good	Yes
2.	≥8	No	Good	Moderate	Yes
3.	≥9	No	Average	Poor	No
5.	≥8	Yes	Good	Moderate	Yes
6.	≥9	Yes	Good	Moderate	Yes
8.	≥9	No	Very good	Good	Yes
9.	≥8	Yes	Good	Good	Yes
10.	≥8	Yes	Average	Good	Yes

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

$$\text{Gini\_Index}(T) = 1 - \left(\frac{7}{8}\right)^2 - \left(\frac{1}{8}\right)^2$$
$$= 1 - 0.766 - 0.0156$$
$$= 1 - 0.58$$
$$\text{Gini\_Index}(T) = 0.2184$$

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

$$\text{Gini\_Index}(T) = 1 - \left(\frac{7}{8}\right)^2 - \left(\frac{1}{8}\right)^2$$
$$= 1 - 0.766 - 0.0156$$
$$= 1 - 0.58$$
$$\text{Gini\_Index}(T) = 0.2184$$

Interactiveness	Job Offer = Yes	Job Offer = No
Yes	5	0
No	2	1

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

$$\text{Gini\_Index}(T) = 1 - \left(\frac{7}{8}\right)^2 - \left(\frac{1}{8}\right)^2$$
$$= 1 - 0.766 - 0.0156$$
$$= 1 - 0.58$$
$$\text{Gini\_Index}(T) = 0.2184$$

Interactiveness	Job Offer = Yes	Job Offer = No
Yes	5	0
No	2	1

$$\text{Gini\_Index}(T, \text{Interactiveness} \in \{\text{Yes}\}) = 1 - \left(\frac{5}{5}\right)^2 - \left(\frac{0}{5}\right)^2$$
$$= 1 - 1 = 0$$

$$\text{Gini\_Index}(T, \text{Interactiveness} \in \{\text{No}\}) = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2$$
$$= 1 - 0.44 - 0.111 = 0.449$$

$$\text{Gini\_Index}(T, \text{Interactiveness} \in \{\text{Yes, No}\}) = \left(\frac{7}{8}\right) \times 0 + \left(\frac{1}{8}\right) \times 0.449$$
$$= 0.056$$

$$\Delta\text{Gini}(\text{Interactiveness}) =$$
$$\text{Gini}(T) - \text{Gini}(T, \text{Interactiveness})$$
$$= 0.2184 - 0.056 = 0.1624$$

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

$$\text{Gini\_Index}(T) = 1 - \left(\frac{7}{8}\right)^2 - \left(\frac{1}{8}\right)^2$$
$$= 1 - 0.766 - 0.0156$$
$$= 1 - 0.58$$
$$\text{Gini\_Index}(T) = 0.2184$$

Gini_Index for Subsets of Practical Knowledge

Subsets		Gini_Index
(Very Good, Good)	Average	0.125
(Very Good, Average)	Good	0.1875
(Good, Average)	Very Good	0.2085

$$\Delta\text{Gini}(\text{Practical Knowledge}) = \text{Gini}(T) - \text{Gini}(T, \text{Practical Knowledge})$$
$$= 0.2184 - 0.125$$
$$= 0.0934$$

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

Gini_Index for Subsets of Practical Knowledge

Subsets		Gini_Index
(Very Good, Good)	Average	0.125
(Very Good, Average)	Good	0.1875
(Good, Average)	Very Good	0.2085

$$\Delta\text{Gini}(\text{Practical Knowledge}) = \text{Gini}(T) - \text{Gini}(T, \text{Practical Knowledge})$$
$$= 0.2184 - 0.125$$
$$= 0.0934$$

Gini_Index for Subsets of Communication Skills

Subsets		Gini_Index
(Good, Moderate)	Poor	0
(Good, Poor)	Moderate	0.2
(Moderate, Poor)	Good	0.1875

$$\Delta\text{Gini}(\text{Communication Skills}) = \text{Gini}(T) - \text{Gini}(T, \text{Communication Skills})$$
$$= 0.2184 - 0 = 0.2184$$

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

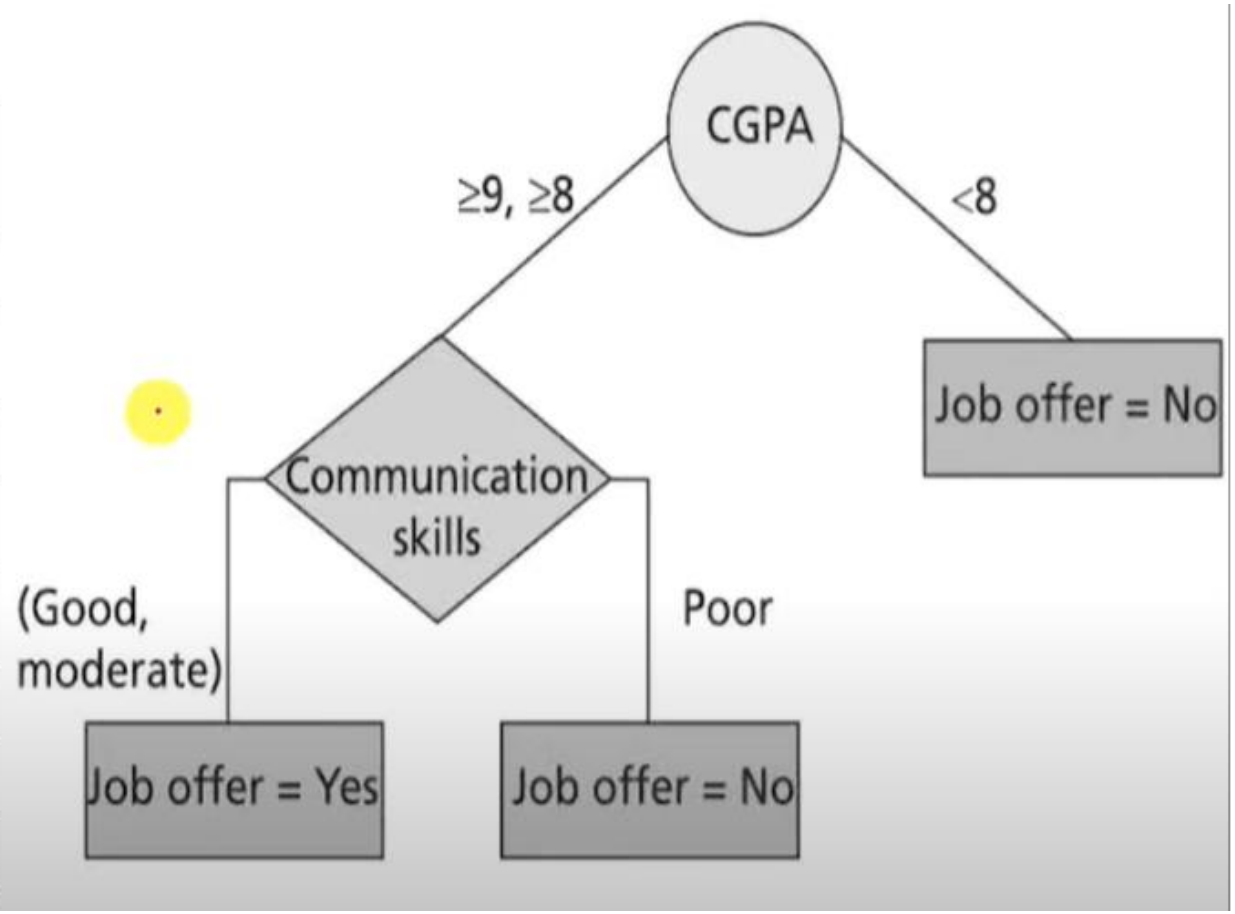
Gini_Index and ΔGini Values for All Attributes

Attribute	Gini_Index	ΔGini
Interactiveness	0.056	0.1624
Practical knowledge	0.125	0.0934
Communication Skills	0	0.2184

Gini_Index and ΔGini Values for All Attributes

Attribute	Gini_Index	ΔGini
Interactiveness	0.056	0.1624
Practical knowledge	0.125	0.0934
Communication Skills	0	0.2184

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes



CART (Classification and Regression Trees)

- CART is a newer decision tree algorithm that was developed in the late 1980s.
- CART can be used for both categorical and continuous target variables.
- CART uses the Gini impurity metric to determine the best feature to split on at each node of the tree.
- Mathematically, we can write Gini Impurity as follows:

$$Gini = 1 - \sum_{i=1}^{n} (p_i)^2$$

- where p_i is the probability of an object being classified to a particular class.

From this one can calculate the average of Gini Index by

$$Gini(S, A) = \sum_{i}^{n} \frac{absS_i}{absS} * Gini(S_i) \qquad (4)$$

- Advantages of Gini Index; it doesn't require computer to compute logarithmic functions, which are computationally intensive.
- Gini Index is Minimized instead of Maximized.

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

PlayTennis: training examples

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

Outlook

Outlook is a nominal feature. It can be sunny, overcast or rain. I will summarize the final decisions for outlook feature.

Outlook	Yes	No	Number of instances
Sunny	2	3	5
Overcast	4	0	4
Rain	3	2	5

Gini(Outlook=Sunny) = $1 - (2/5)^2 - (3/5)^2 = 1 - 0.16 - 0.36 = 0.48$

Gini(Outlook=Overcast) = $1 - (4/4)^2 - (0/4)^2 = 0$

Gini(Outlook=Rain) = $1 - (3/5)^2 - (2/5)^2 = 1 - 0.36 - 0.16 = 0.48$

Then, we will calculate weighted sum of gini indexes for outlook feature.

Gini(Outlook) = $(5/14) \times 0.48 + (4/14) \times 0 + (5/14) \times 0.48 = 0.171 + 0 + 0.171 = 0.342$

PlayTennis: training examples

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

Temperature

Similarly, temperature is a nominal feature and it could have 3 different values: Cool, Hot and Mild. Let's summarize decisions for temperature feature.

Temperature	Yes	No	Number of instances
Hot	2	2	4
Cool	3	1	4
Mild	4	2	6

Gini(Temp=Hot) = $1 - (2/4)^2 - (2/4)^2 = 0.5$

Gini(Temp=Cool) = $1 - (3/4)^2 - (1/4)^2 = 1 - 0.5625 - 0.0625 = 0.375$

Gini(Temp=Mild) = $1 - (4/6)^2 - (2/6)^2 = 1 - 0.444 - 0.111 = 0.445$

We'll calculate weighted sum of gini index for temperature feature

Gini(Temp) = $(4/14) \times 0.5 + (4/14) \times 0.375 + (6/14) \times 0.445 = 0.142 + 0.107 + 0.190 = 0.439$

PlayTennis: training examples

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

Humidity

Humidity is a binary class feature. It can be high or normal.

Humidity	Yes	No	Number of instances
High	3	4	7
Normal	6	1	7

Gini(Humidity=High) = $1 - (3/7)^2 - (4/7)^2 = 1 - 0.183 - 0.326 = 0.489$

Gini(Humidity=Normal) = $1 - (6/7)^2 - (1/7)^2 = 1 - 0.734 - 0.02 = 0.244$

Weighted sum for humidity feature will be calculated next

Gini(Humidity) = (7/14) x 0.489 + (7/14) x 0.244 = 0.367

PlayTennis: training examples

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

Wind

Wind is a binary class similar to humidity. It can be weak and strong.

Wind	Yes	No	Number of instances
Weak	6	2	8
Strong	3	3	6

Gini(Wind=Weak) = $1 - (6/8)^2 - (2/8)^2 = 1 - 0.5625 - 0.062 = 0.375$

Gini(Wind=Strong) = $1 - (3/6)^2 - (3/6)^2 = 1 - 0.25 - 0.25 = 0.5$

Gini(Wind) = $(8/14) \times 0.375 + (6/14) \times 0.5 = 0.428$

PlayTennis: training examples

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

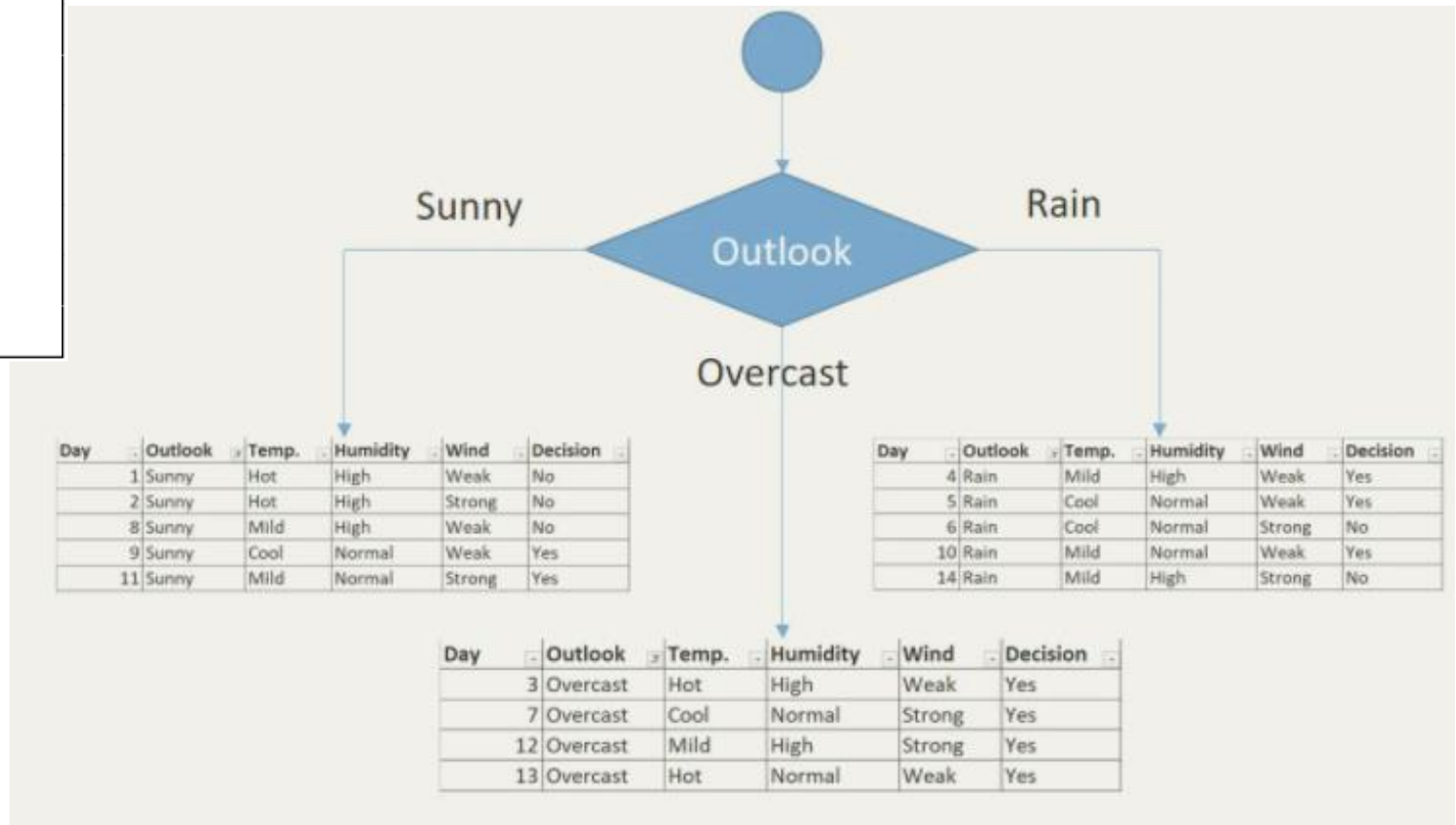
Time to decide

We've calculated gini index values for each feature. The winner will be outlook feature because its cost is the lowest.

Feature	Gini index
Outlook	0.342
Temperature	0.439
Humidity	0.367
Wind	0.428

PlayTennis: training examples

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



Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Day	Outlook	Temp.	Humidity	Wind	Decision
3	Overcast	Hot	High	Weak	Yes
7	Overcast	Cool	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes

We will apply same principles to those sub datasets in the following steps.

Focus on the sub dataset for sunny outlook. We need to find the gini index scores for temperature, humidity and wind features respectively.

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Gini of temperature for sunny outlook

Temperature	Yes	No	Number of instances
Hot	0	2	2
Cool	1	0	1
Mild	1	1	2

Gini(Outlook=Sunny and Temp.=Hot) = $1 - (0/2)^2 - (2/2)^2 = 0$

Gini(Outlook=Sunny and Temp.=Cool) = $1 - (1/1)^2 - (0/1)^2 = 0$

Gini(Outlook=Sunny and Temp.=Mild) = $1 - (1/2)^2 - (1/2)^2 = 1 - 0.25 - 0.25 = 0.5$

Gini(Outlook=Sunny and Temp.) = $(2/5)x0 + (1/5)x0 + (2/5)x0.5 = 0.2$

Gini of humidity for sunny outlook

Humidity	Yes	No	Number of instances
High	0	3	3
Normal	2	0	2

Gini(Outlook=Sunny and Humidity=High) = $1 - (0/3)^2 - (3/3)^2 = 0$

Gini(Outlook=Sunny and Humidity=Normal) = $1 - (2/2)^2 - (0/2)^2 = 0$

Gini(Outlook=Sunny and Humidity) = $(3/5)x0 + (2/5)x0 = 0$

Gini of wind for sunny outlook

Wind	Yes	No	Number of instances
Weak	1	2	3
Strong	1	1	2

Gini(Outlook=Sunny and Wind=Weak) = $1 - (1/3)^2 - (2/3)^2 = 0.266$

Gini(Outlook=Sunny and Wind=Strong) = $1- (1/2)^2 - (1/2)^2 = 0.2$

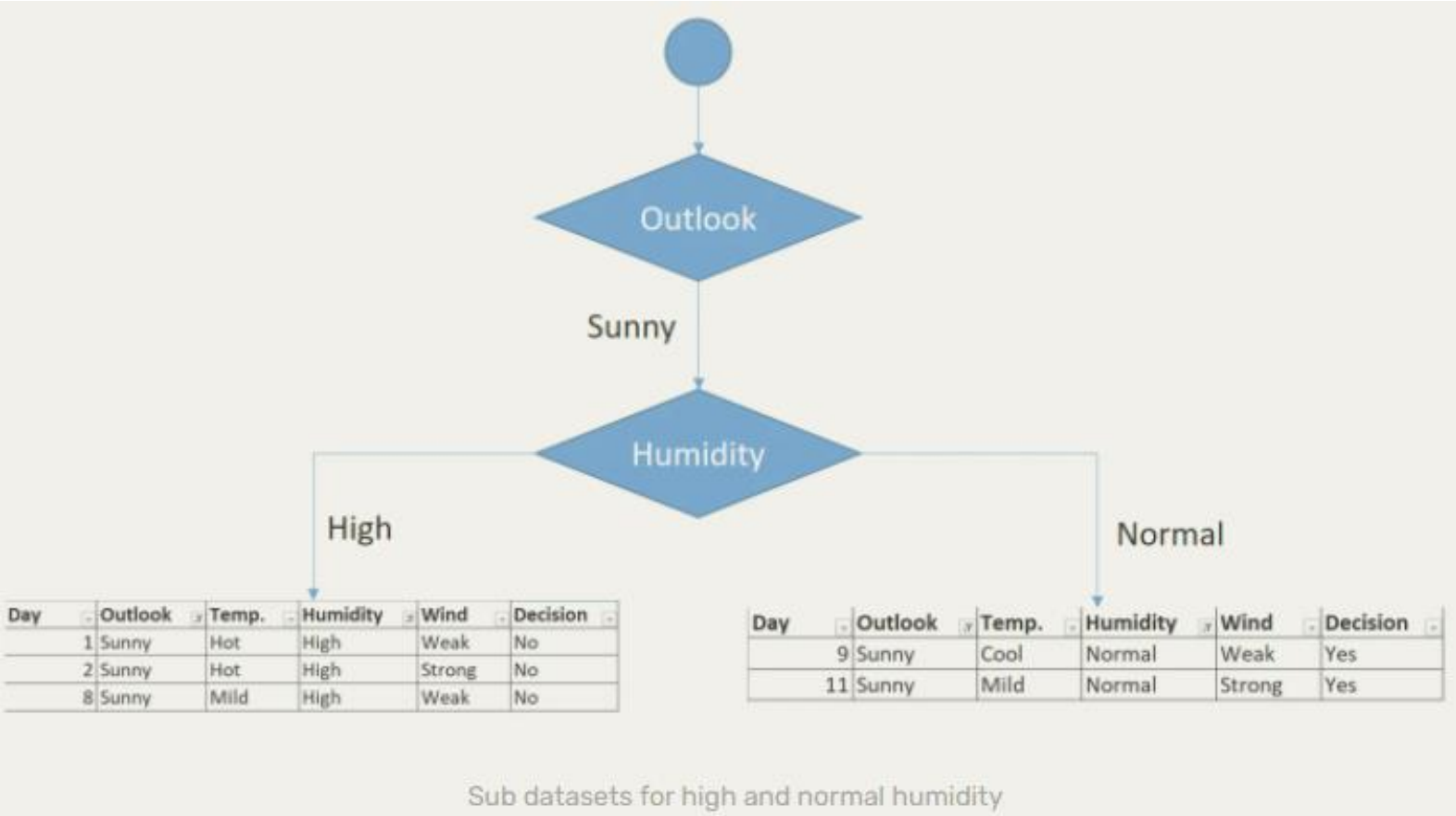
Gini(Outlook=Sunny and Wind) = $(3/5)x0.266 + (2/5)x0.2 = 0.466$

Decision for sunny outlook

We've calculated gini index scores for feature when outlook is sunny. The winner is humidity because it has the lowest value.

Feature	Gini index
Temperature	0.2
Humidity	0
Wind	0.466

We'll put humidity check at the extension of sunny outlook.

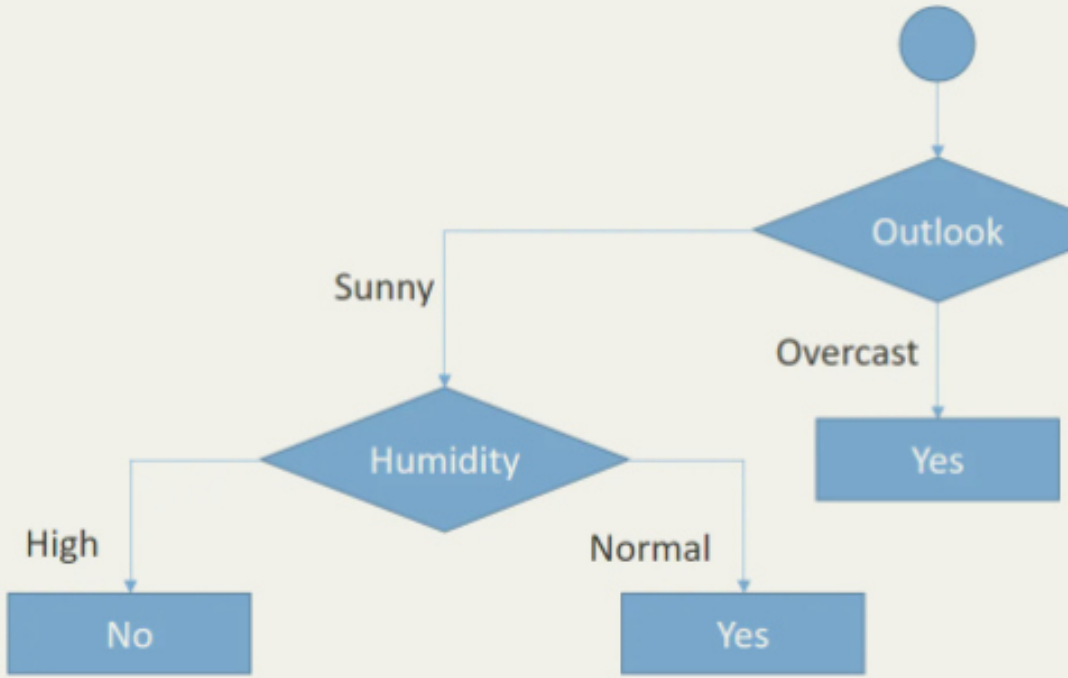


Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No

Day	Outlook	Temp.	Humidity	Wind	Decision
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Sub datasets for high and normal humidity

As seen, decision is always no for high humidity and sunny outlook. On the other hand, decision will always be yes for normal humidity and sunny outlook. This branch is over.



Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Decisions for high and normal humidity

Now, we need to focus on rain outlook.

Rain outlook

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

We'll calculate gini index scores for temperature, humidity and wind features when outlook is rain.

Gini of temprature for rain outlook

Temperature	Yes	No	Number of instances
Cool	1	1	2
Mild	2	1	3

Gini(Outlook=Rain and Temp.=Cool) = $1 - (1/2)^2 - (1/2)^2 = 0.5$

Gini(Outlook=Rain and Temp.=Mild) = $1 - (2/3)^2 - (1/3)^2 = 0.444$

Gini(Outlook=Rain and Temp.) = $(2/5)\times0.5 + (3/5)\times0.444 = 0.466$

Gini of humidity for rain outlook

Humidity	Yes	No	Number of instances
High	1	1	2
Normal	2	1	3

Gini(Outlook=Rain and Humidity=High) = $1 - (1/2)^2 - (1/2)^2 = 0.5$

Gini(Outlook=Rain and Humidity=Normal) = $1 - (2/3)^2 - (1/3)^2 = 0.444$

Gini(Outlook=Rain and Humidity) = $(2/5)\times0.5 + (3/5)\times0.444 = 0.466$

Gini of wind for rain outlook

Wind	Yes	No	Number of instances
Weak	3	0	3
Strong	0	2	2

Gini(Outlook=Rain and Wind=Weak) = $1 - (3/3)^2 - (0/3)^2 = 0$

Gini(Outlook=Rain and Wind=Strong) = $1 - (0/2)^2 - (2/2)^2 = 0$

Gini(Outlook=Rain and Wind) = $(3/5)x0 + (2/5)x0 = 0$

Decision for rain outlook

The winner is wind feature for rain outlook because it has the minimum gini index score in features.

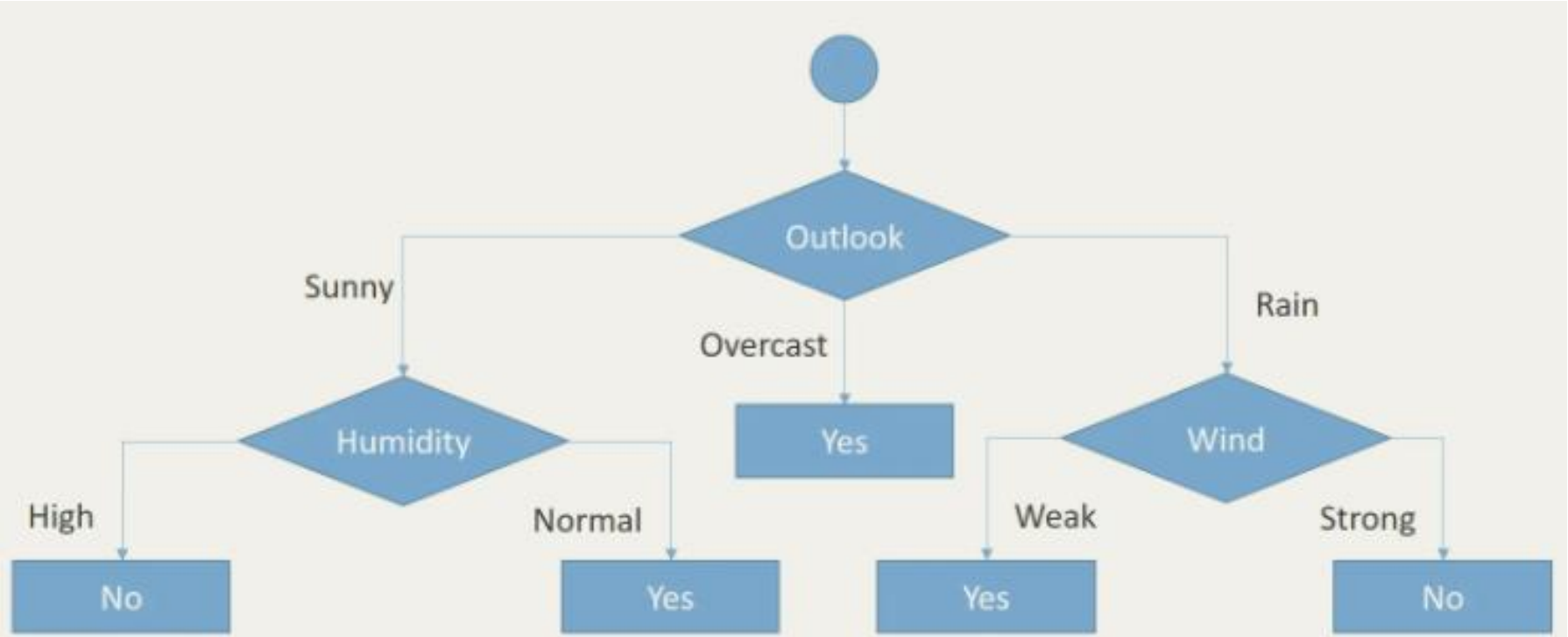
Feature	Gini index
Temperature	0.466
Humidity	0.466
Wind	0

Put the wind feature for rain outlook branch and monitor the new sub data sets.

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes

Day	Outlook	Temp.	Humidity	Wind	Decision
6	Rain	Cool	Normal	Strong	No
14	Rain	Mild	High	Strong	No

As seen, decision is always yes when wind is weak. On the other hand, decision is always no if wind is strong. This means that this branch is over.



Final form of the decision tree built by CART algorithm

Algorithm	Splitting Criteria of algorithm	Attribute types Managed by algorithm	Pruning Strategy of algorithm	Outlier Detection	Missing values	Invented By
C4.5	Gain Ratio	Manages both Categorical and Numeric value	Error Based pruning is used	Error Based pruning is used	Manages missing values.	developed by Ross Quinlan
ID3	Information Gain	Manages only Categorical value	No pruning is done	No pruning is done	Do not Manages missing values.	invented by Ross Quinlan
CART	Gini Index	Manages both Categorical and Numeric value	Cost-Complexity pruning is used	Cost-Complexity pruning is used	Manages missing values.	first published by Leo Breiman in 1984

ISSUES IN DECISION TREE LEARNING

1. Overfitting the Data
2. Incorporating Continuous valued attributes
3. Handling training examples with missing attribute values
4. Handling attributes with different costs
5. Alternative measures for selecting attributes

Issue 1 Overfitting the Data

- Over-fitting is nothing but the model runs accurately on the given training data so much that it would be inaccurate in predicting the outcomes of the untrained data.
- In decision trees, over-fitting occurs when the tree is designed so as to perfectly fit all samples in the training data set.
- Thus it ends up with branches with strict rules of sparse data.
- Thus this effects the accuracy when predicting samples that are not part of the training set.

Prevent overfitting/Determine how deeply to grow the decision tree

- 1. we stop splitting the tree at some point;
  - we need to introduce two hyperparameters for training like maximum depth of the tree and minimum size of a leaf.
- 2. we generate a complete tree first, and then get rid of some branches called as **pruning**.
  - In pruning, you trim off the branches of the tree, i.e., remove the decision nodes starting from the leaf node such that the overall accuracy is not disturbed.
  - This is done by segregating the actual training set into two sets: training data set, D and validation data set, V.
  - Prepare the decision tree using the segregated training data set, D.
  - Then continue trimming the tree accordingly to optimize the accuracy of the validation data set, V.

Issue 2 Continues Valued attributes

- Define new discrete valued attributes that partition the continuous attribute value into a discrete set of intervals

Temperature:	40	48	60	72	80	90
PlayTennis:	No	No	Yes	Yes	Yes	No

- Find a set of thresholds midway Between different target values of the attribute : $Temperature_{>54}$ and $Temperature_{>85}$

- Pick a threshold, *c*, that produces the greatest information gain : $temperature_{>54}$

Issue 3 Unknown/Missing Attribute Values

- What if some examples missing values of *A*?
- Use training example anyway, sort through tree
  - If node *n* tests *A*, assign most common value of *A* among other examples sorted to node *n*
  - Assign most common value of *A* among other examples with same target value
  - Assign probability p_i to each possible value v_i of *A*
    - Assign fraction p_i of example too each descendant in tree
- Classify new examples in same fashion

Issue 4 Attributes with Many Values

- Problem
  - If attribute has many values, *Gain* will select it
  - Imagine using *Date = Oct_13_2004* as attribute

- One approach: use *GainRatio* instead

$$GainRatio(S,A) \equiv \frac{Gain(S,A)}{SplitInfomation(S,A)}$$

$$SplitInfomation(S,A) \equiv -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

where S_i is subset of S for which A has value v_i

Issue 5 Attributes with Costs

- Use low-cost attributes where possible, relying on high-cost attributes only when needed to produce reliable classficiations
- Tan and Schlimmer (1990) $\frac{Gain^2(S,A)}{Cost(A)}$
- Nunez (1988) $\frac{2^{Gain(S,A)}-1}{(Cost(A)+1)^w}$

where $w \in [0, 1]$ determines importance of cost

Issue 5 Alternative Measures for selecting attributes

- There are a lot of alternatives to entropy and information gain.
- Two of them are Gain Ratio and Gini Index.
- Gain ratio is a modification to information gain.
- To compute Gain ratio we use two parameters:
  - number of branches created and size of the branch.
- First calculate the information gain then compute the intrinsic Information with the function below.

$$IntI(S,A) = -\sum_{i \to 0}^{n} \frac{|S_i|}{S} * \log\left(\frac{|S_i|}{S}\right) \qquad (1)$$

Now we can calculate the information Gain as follows.

$$GR(S,A) = \frac{Gain(S,A)}{IntI(S,A)} \qquad (2)$$

Issue 5 Alternative Measures for selecting attributes

- The Gini Index is more concerned with the impurity of the attribute.
- Gini Index is one of the most popular alternative to Entropy as well.
- It is widely used in Classification and Regression Trees (CART).
- To calculate impurity:

$$Gini(S) = 1 - \sum_{i}^{n} p_i^2 \qquad (3)$$

From this one can calculate the average of Gini Index by

$$Gini(S, A) = \sum_{i}^{n} \frac{absS_i}{absS} * Gini(S_i) \qquad (4)$$

- Advantages of Gini Index; it doesn't require computer to compute logarithmic functions, which are computationally intensive.
- Gini Index is Minimized instead of Maximized.

Bayesian classification

- Naïve Bayes algorithm is a supervised learning algorithm, which is based on **Bayes theorem** and used for solving classification problems.
- It is mainly used in *text classification* that includes a high-dimensional training dataset.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- **It is a probabilistic classifier, which means it predicts on the basis of the probability of an object**.
- Some popular examples of Naïve Bayes Algorithm are **spam filtration, Sentimental analysis, and classifying articles**.

Why is it called Naïve Bayes?

- The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:
- **Naïve**: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
- **Bayes**: It is called Bayes because it depends on the principle of Bayes' Theorem.

Why is it called Naïve Bayes?

- Bayes' Theorem:
- Bayes' theorem is also known as **Bayes' Rule** or **Bayes' law**, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.
- The formula for Bayes' theorem is given as:

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

Where,

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

P(B) is Marginal Probability: Probability of Evidence.

Working of Naïve Bayes' Classifier:

- Suppose we have a dataset of **weather conditions** and corresponding target variable "**Play**". So using this dataset we need to decide that whether we should play or not on a particular day according to the weather conditions. So to solve this problem, we need to follow the below steps:
- Convert the given dataset into frequency tables.
- Generate Likelihood table by finding the probabilities of given features.
- Now, use Bayes theorem to calculate the posterior probability.
- **Problem**: If the weather is sunny, then the Player should play or not?
- **Solution**: To solve this, first consider the below dataset:

Working of Naïve Bayes' Classifier:

	Outlook	Play
0	Rainy	Yes
1	Sunny	Yes
2	Overcast	Yes
3	Overcast	Yes
4	Sunny	No
5	Rainy	Yes
6	Sunny	Yes
7	Overcast	Yes
8	Rainy	No
9	Sunny	No
10	Sunny	Yes
11	Rainy	No
12	Overcast	Yes
13	Overcast	Yes

Frequency table for the Weather Conditions:

Weather	Yes	No
Overcast	5	0
Rainy	2	2
Sunny	3	2
Total	10	5

Likelihood table weather condition:

Weather	No	Yes	
Overcast	0	5	5/14= 0.35
Rainy	2	2	4/14=0.29
Sunny	2	3	5/14=0.35
All	4/14=0.29	10/14=0.71	

Working of Naïve Bayes' Classifier:

Applying Bayes'theorem:

P(Yes|Sunny)= P(Sunny|Yes)*P(Yes)/P(Sunny)
P(Sunny|Yes)= 3/10= 0.3
P(Sunny)= 0.35
P(Yes)=0.71

So P(Yes|Sunny) = 0.3*0.71/0.35= **0.60**
P(No|Sunny)= P(Sunny|No)*P(No)/P(Sunny)
P(Sunny|NO)= 2/4=0.5

P(No)= 0.29
P(Sunny)= 0.35

So P(No|Sunny)= 0.5*0.29/0.35 = **0.41**
So as we can see from the above calculation that **P(Yes|Sunny)>P(No|Sunny)**

Hence on a Sunny day, Player can play the game.

Frequency table for the Weather Conditions:

Weather	Yes	No
Overcast	5	0
Rainy	2	2
Sunny	3	2
Total	10	5

Likelihood table weather condition:

Weather	No	Yes	
Overcast	0	5	5/14= 0.35
Rainy	2	2	4/14=0.29
Sunny	2	3	5/14=0.35
All	4/14=0.29	10/14=0.71	

- **Advantages of Naïve Bayes Classifier:**

- Naïve Bayes is one of the fast and easy ML algorithms to predict a class of datasets.
- It can be used for Binary as well as Multi-class Classifications.
- It performs well in Multi-class predictions as compared to the other Algorithms.
- It is the most popular choice for **text classification problems**
- **Disadvantages of Naïve Bayes Classifier:**
- Naive Bayes assumes that all features are independent or unrelated, so it cannot learn the relationship between features.

Applications of Naïve Bayes Classifier:

- It is used for **Credit Scoring**.
- It is used in **medical data classification**.
- It can be used in **real-time predictions** because Naïve Bayes Classifier is an eager learner.
- It is used in Text classification such as **Spam filtering** and **Sentiment analysis**.

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:

$P(\text{Play}=Yes) = 9/14$ (prior probability) $P(\text{Play}=No) = 5/14$

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

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

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

$$= = 0.0053$$

- Map rule:

$$P(x' \mid \text{No}) = P(\text{Outlook=Sunny/No}) *$$
$$P(\text{Temperature=Cool/No}) *$$
$$P(\text{Humidity=High/No}) *$$
$$P(\text{Wind=Strong/No}) *$$
$$P(\text{No})$$

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

$$= 0.0206$$

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'

K-Nearest Neighbor(KNN) Classifier Algorithm

Introduction

- K-Nearest Neighbor is a Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.

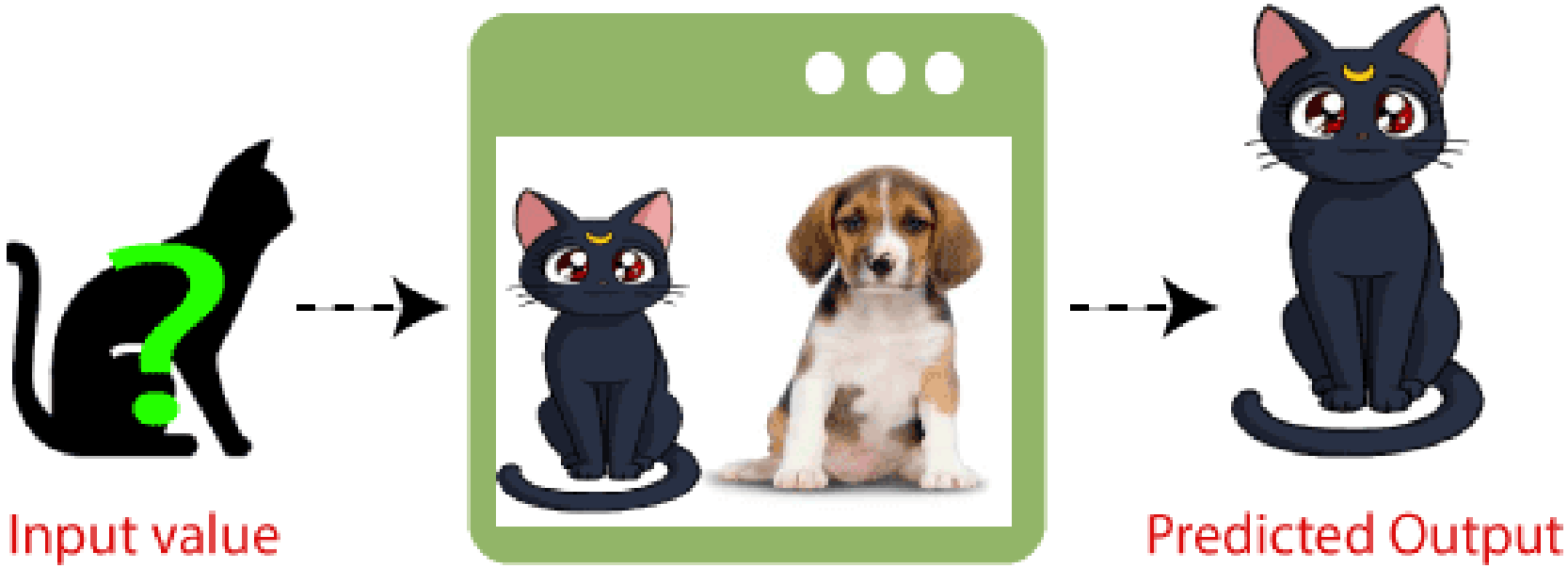
Introduction

- 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 algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
- **Example:**

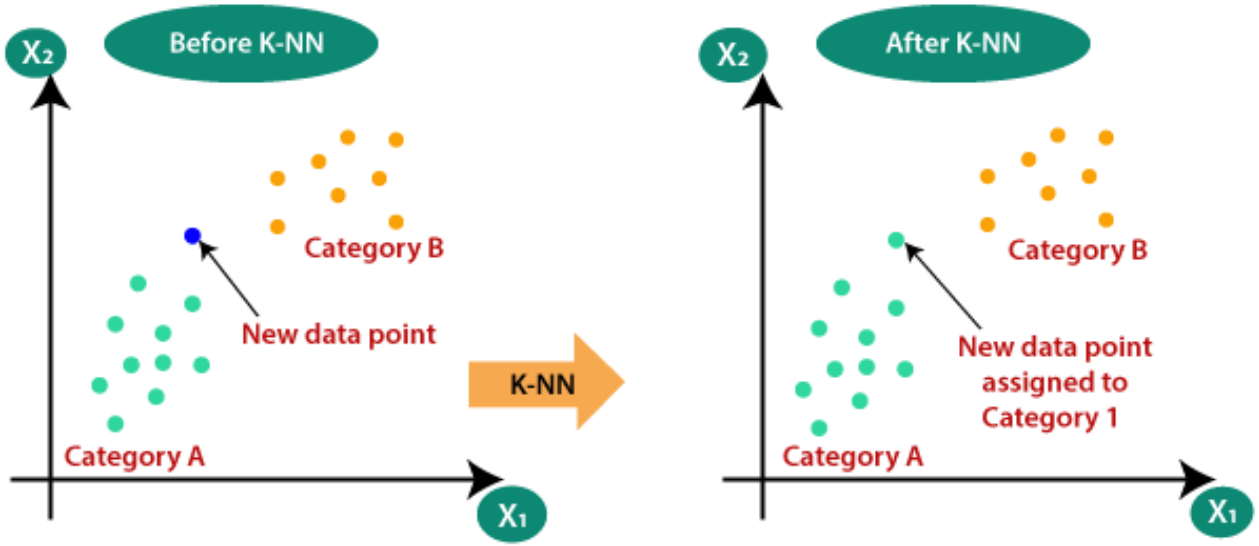
Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

•

KNN Classifier



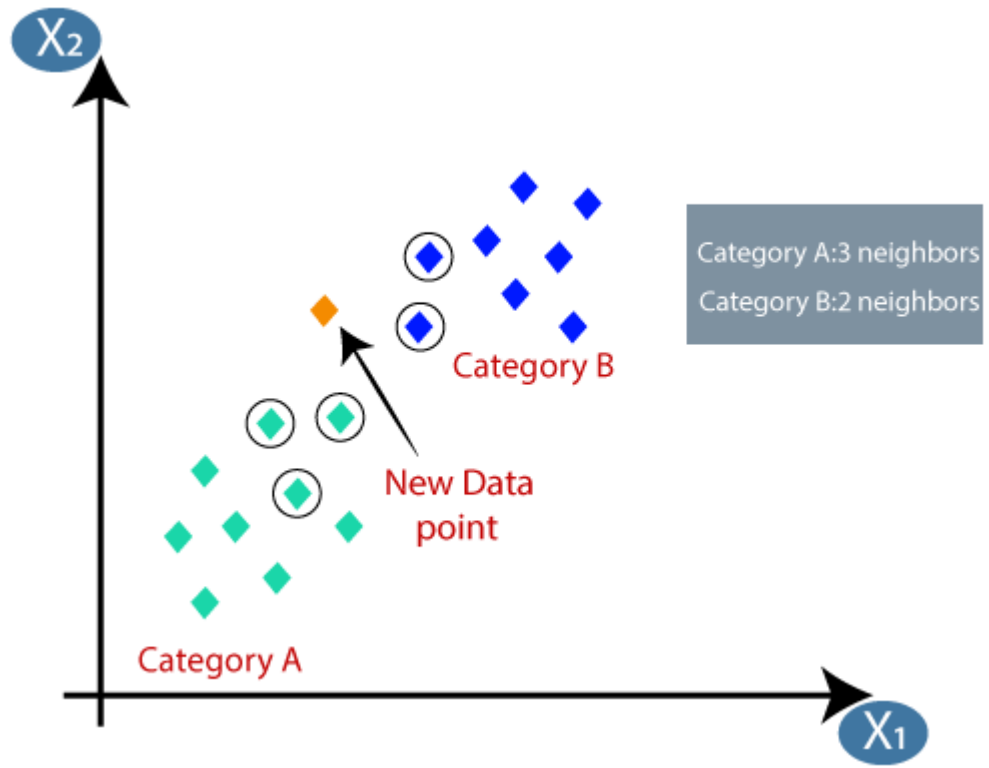
Why do we need a K-NN Algorithm?



- Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories.
- To solve this type of problem, we need a K-NN algorithm.
- With the help of K-NN, we can easily identify the category or class of a particular dataset.

How does K-NN work?

- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.



•As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.

How to select the value of K in the K-NN Algorithm?

- There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
- A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
- Large values for K are good, but it may find some difficulties

Advantages/ Disadvantages

Advantages of KNN Algorithm:

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

Disadvantages of KNN Algorithm:

- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.

Example KNN

Sepal Length	Sepal Width	Species
5.3	3.7	Setosa
5.1	3.8	Setosa
7.2	3.0	Virginica
5.4	3.4	Setosa
5.1	3.3	Setosa
5.4	3.9	Setosa
7.4	2.8	Virginica
6.1	2.8	Verscicolor
7.3	2.9	Virginica
6.0	2.7	Verscicolor
5.8	2.8	Virginica
6.3	2.3	Verscicolor
5.1	2.5	Verscicolor
6.3	2.5	Verscicolor
5.5	2.4	Verscicolor

Find the class label for given instance using KNN with K=5

Sepal Length	Sepal Width	Species
5.2	3.1	?

Sepal Length	Sepal Width	Species	Distance
5.3	3.7	Setosa	0.608
5.1	3.8	Setosa	0.707
7.2	3.0	Virginica	2.002
5.4	3.4	Setosa	0.36
5.1	3.3	Setosa	0.22
5.4	3.9	Setosa	0.82
7.4	2.8	Virginica	2.22
6.1	2.8	Verscicolor	0.94
7.3	2.9	Virginica	2.1
6.0	2.7	Verscicolor	0.89
5.8	2.8	Virginica	0.67
6.3	2.3	Verscicolor	1.36
5.1	2.5	Verscicolor	0.60
6.3	2.5	Verscicolor	1.25
5.5	2.4	Verscicolor	0.75

Step 1: Find distance

$$\text{Distance (Sepal Length, Sepal Width)} = \sqrt{(x-a)^2 + (y-b)^2}$$

$$\text{Distance (Sepal Length, Sepal Width)} = \sqrt{(5.2-5.3)^2 + (3.1-3.7)^2}$$

$$\text{Distance (Sepal Length, Sepal Width)} = 0.608$$

Sepal Length	Sepal Width	Species	Distance
5.3	3.7	Setosa	0.608

Sepal Length	Sepal Width	Species	Distance	Rank
5.3	3.7	Setosa	0.608	3
5.1	3.8	Setosa	0.707	6
7.2	3.0	Virginica	2.002	13
5.4	3.4	Setosa	0.36	2
5.1	3.3	Setosa	0.22	1
5.4	3.9	Setosa	0.82	8
7.4	2.8	Virginica	2.22	15
6.1	2.8	Verscicolor	0.94	10
7.3	2.9	Virginica	2.1	14
6.0	2.7	Verscicolor	0.89	9
5.8	2.8	Virginica	0.67	5
6.3	2.3	Verscicolor	1.36	12
5.1	2.5	Verscicolor	0.60	4
6.3	2.5	Verscicolor	1.25	11
5.5	2.4	Verscicolor	0.75	7

Step 2: Find Rank

Step 3: Find nearest neighbours to assign class

If k = 1 – Setosa

If k = 2 – Setosa

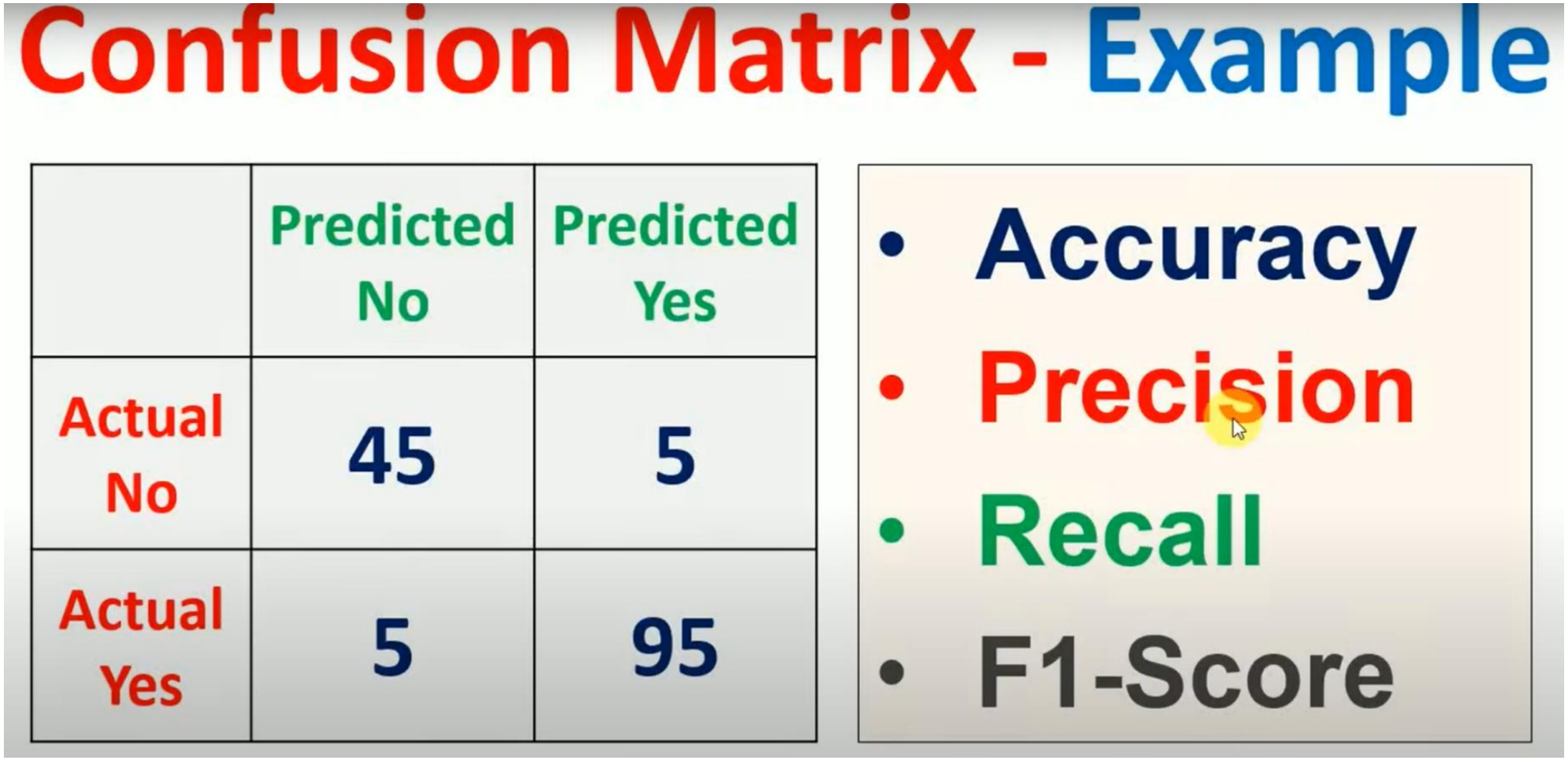
If k = 5 – Setosa

Evaluating Machine Learning algorithms

Evaluating Machine Learning algorithms

- **Classification Metrics:**
  - Accuracy
  - Precision
  - Recall
  - F1 Score
  - Confusion Matrix
- **Regression Metrics:**
  - Mean Absolute Error (MAE)
  - Mean Squared Error (MSE)
  - Root Mean Squared Error (RMSE)
  - Relative MSE
  - Coefficient of Variation(CV)

Evaluating Machine Learning algorithms



Classification Metrics

Problem Definition – Confusion Matrix

- Consider the confusion matrix given below for a binary classifier predicting the presence of a disease
- The classifier made a total of 150 predictions Out of those 150 cases, the classifier predicted "yes" 100 times, and "no" 50 times.

	Predicted No	Predicted Yes
Actual No	45	5
Actual Yes	5	95

Classification Metrics

Problem Definition – Confusion Matrix

- Consider the confusion matrix given below for a binary classifier predicting the presence of a disease
- The classifier made a total of 150 predictions Out of those 150 cases, the classifier predicted "yes" 100 times, and "no" 50 times.
- In reality, 100 patients in the sample have the disease, and 50 patients do not.

	Predicted No	Predicted Yes
Actual No	45	5
Actual Yes	5	95

Classification Metrics

Problem Definition – Confusion Matrix

- Consider the confusion matrix given below for a binary classifier predicting the presence of a disease
- The classifier made a total of 150 predictions Out of those 150 cases, the classifier predicted "yes" 100 times, and "no" 50 times.
- In reality, 100 patients in the sample have the disease, and 50 patients do not.

	Predicted No	Predicted Yes
Actual No	45	5
Actual Yes	5	95

	Predicted No	Predicted Yes
Actual No	TN = 45	FP = 5
Actual Yes	FN = 5	T.P = 95

Classification Metrics

- **Accuracy:** Overall, how often is the classifier correct?
- $Accuraccy = \frac{TN+TP}{TN+FP+FN+TP}$

$$= \frac{45+95}{150} = 93.33\%$$

	Predicted No	Predicted Yes
Actual No	TN = 45	FP = 5
Actual Yes	FN = 5	TP = 95

Classification Metrics

- **Misclassification Rate:** Overall, how often is it wrong?
- $Missclassification\ Rate = \frac{FN+FP}{TN+FP+FN+TP}$

$$= \frac{5+5}{150} = 6.67\%$$

	Predicted No	Predicted Yes
Actual No	TN = 45	FP = 5
Actual Yes	FN = 5	TP = 95

Classification Metrics

- **True Positive Rate:** When it's actually yes, how often does it predict yes?
- also known as "Sensitivity" or "Recall"
- $True\ Positive\ rate = \frac{TP}{Actual\ Yes}$

$$= \frac{95}{100} = 95\%$$

	Predicted No	Predicted Yes
Actual No	TN = 45	FP = 5
Actual Yes	FN = 5	TP = 95

Classification Metrics

- **False Positive Rate:** When it's actually no, how often does it predict yes?

	Predicted No	**Predicted Yes**
Actual No	TN = 45	FP = 5
Actual Yes	FN = 5	TP = 95

- $False\ Positive\ rate = \frac{FP}{Actual\ No}$

$$= \frac{5}{50} = 10\%$$

Classification Metrics

- **True Negative Rate:** When it's actually no, how often does it predict no?
- also known as "Specificity"

	Predicted No	Predicted Yes
Actual No	TN = 45	FP = 5
Actual Yes	FN = 5	TP = 95

- $True\ Negative\ rate = \frac{TN}{Actual\ No}$

$$= \frac{45}{50} = 90\%$$

Classification Metrics

- **Precision:** When it predicts yes, how often is it correct?

- $Precision = \frac{TP}{Predicted\ Yes}$

$$= \frac{95}{100} = 95\%$$

	Predicted No	Predicted Yes
Actual No	TN = 45	FP = 5
Actual Yes	FN = 5	TP = 95

Classification Metrics

- **Prevalence:** How often does the yes condition actually occur in our sample?

	Predicted No	Predicted Yes
Actual No	TN = 45	FP = 5
Actual Yes	FN = 5	TP = 95

- $Prevalence = \frac{Actual\ Yes}{Total}$

$$= \frac{100}{150} = 66.67\%$$

Regression Metrics

- **Linear Regression Metrics**

- Consider the following training set Table for predicting the sales of the items.
- Consider two fresh items I6 and I7, whose actual values are 80 and 75, respectively.
- A regression model predicts the values of the items I6 and I7 as 75 and 85, respectively.
- Find MAE, MSE, RMSE, RelMSE and CV.

x_i Items	y_j (Sales)
I1	80
I2	90
I3	100
I4	110
I5	120

Test Items	Actual Value	Predicted Value
I6	80	75
I7	75	85

What are the Metrics used to Evaluate the performance of Regression Models in ML

- **Linear Regression Metrics**

- **Mean Absolute Error (MAE)**

$$MAE = \frac{1}{n}\sum_{i=0}^{n-1}\left|y_i - \hat{y}_i\right|$$

$$MAE = \frac{1}{2} \times \left(\left|80 - 75\right| + \left|75 - 85\right|\right) = \frac{15}{2} = 7.5$$

x_i Items	y_j (Sales)
I1	80
I2	90
I3	100
I4	110
I5	120

Test Items	Actual Value	Predicted Value
I6	80	75
I7	75	85

What are the Metrics used to Evaluate the performance of Regression Models in ML

- **Linear Regression Metrics**

- **Mean Squared Error (MSE)**

$$\text{MSE} = \frac{1}{n}\sum_{i=0}^{n-1}(y_i - \hat{y}_i)^2$$

$$\text{MSE} = \frac{1}{2} \times \left(|80 - 75|^2 + |75 - 85|^2\right) = \frac{125}{2} = 62.5$$

x_i Items	y_j (Sales)
I1	80
I2	90
I3	100
I4	110
I5	120

Test Items	Actual Value	Predicted Value
I6	80	75
I7	75	85

What are the Metrics used to Evaluate the performance of Regression Models in ML

- **Linear Regression Metrics**

  - **Mean Squared Error (MSE)**

$$\text{MSE} = \frac{1}{n}\sum_{i=0}^{n-1}(y_i - \hat{y}_i)^2$$

$$\text{MSE} = \frac{1}{2} \times \left(|80 - 75|^2 + |75 - 85|^2\right) = \frac{125}{2} = 62.5$$

x_i Items	y_j (Sales)
I1	80
I2	90
I3	100
I4	110
I5	120

Test Items	Actual Value	Predicted Value
I6	80	75
I7	75	85

  - **Root Mean Square Error (RMSE)**

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{62.5} = 7.91$$

x_i Items	y_j (Sales)
I1	80
I2	90
I3	100
I4	110
I5	120

Test Items	Actual Value	Predicted Value
I6	80	75
I7	75	85

What are the Metrics used to Evaluate the performance of Regression Models in ML

- **Linear Regression Metrics**

- **Relative MSE**

$$\text{RelMSE} = \frac{\sum_{i=0}^{n-1}(y_i - \hat{y}_i)^2}{\sum_{i=0}^{n-1}(y_i - \bar{y}_i)^2}$$

- For finding **RelMSE** and **CV**, the training table should be used to find the average of y.

x_i Items	y_j (Sales)
I1	80
I2	90
I3	100
I4	110
I5	120

Test Items	Actual Value	Predicted Value
I6	80	75
I7	75	85

What are the Metrics used to Evaluate the performance of Regression Models in ML

- **Linear Regression Metrics**

- **Relative MSE**

$$\text{RelMSE} = \frac{\sum_{i=0}^{n-1}(y_i - \hat{y}_i)^2}{\sum_{i=0}^{n-1}(y_i - \bar{y}_i)^2}$$

- For finding **RelMSE** and **CV**, the training table should be used to find the average of y.

x_i Items	y_j (Sales)
I1	80
I2	90
I3	100
I4	110
I5	120

Test Items	Actual Value	Predicted Value
I6	80	75
I7	75	85

- **Relative MSE**

$$\text{RelMSE} = \frac{\sum_{i=0}^{n-1}(y_i - \hat{y}_i)^2}{\sum_{i=0}^{n-1}(y_i - \bar{y}_i)^2}$$

- For finding **RelMSE** and **CV**, the training table should be used to find the average of y.

x_i Items	y_j (Sales)
I1	80
I2	90
I3	100
I4	110
I5	120

The average of y is $\frac{80 + 90 + 100 + 110 + 120}{5} = \frac{500}{5} = 100.$

$$\text{RelMSE} = \frac{(80 - 75)^2 + (75 - 85)^2}{(80 - 100)^2 + (75 - 100)^2} = \frac{125}{1025} = 0.1219$$

Test Items	Actual Value	Predicted Value
I6	80	75
I7	75	85

What are the Metrics used to Evaluate the performance of Regression Models in ML

- **Linear Regression Metrics**

- **Relative MSE**

$$\text{RelMSE} = \frac{\sum_{i=0}^{n-1}(y_i - \hat{y}_i)^2}{\sum_{i=0}^{n-1}(y_i - \bar{y}_i)^2}$$

- For finding **RelMSE** and **CV**, the training table should be used to find the average of y.

x_i Items	y_j (Sales)
I1	80
I2	90
I3	100
I4	110
I5	120

The average of y is $\frac{80+90+100+110+120}{5} = \frac{500}{5} = 100.$

Test Items	Actual Value	Predicted Value
I6	80	75
I7	75	85

$$\text{RelMSE} = \frac{(80-75)^2 + (75-85)^2}{(80-100)^2 + (75-100)^2} = \frac{125}{1025} = 0.1219$$

- **Coefficient of Variation**

$$CV = \frac{\text{RMSE}}{\bar{y}}$$

$$CV = \frac{\sqrt{62.5}}{100} = 0.08$$

x_i Items	y_j (Sales)
I1	80
I2	90
I3	100
I4	110
I5	120

Test Items	Actual Value	Predicted Value
I6	80	75
I7	75	85

- Machine Learning Model Evaluation - GeeksforGeeks

Model selection in machine learning

- Model selection in machine learning is the process of choosing the best machine learning algorithm and its hyper parameters for a particular task or problem.
- It's a critical step in building effective and accurate machine learning models.

Model selection in machine learning

- **Understand the Problem:** Begin by gaining a deep understanding of the problem you're trying to solve and the goals of your machine learning project. Consider the type of problem (classification, regression, clustering, etc.) and the nature of the data.
- **Data Preprocessing:** Prepare your data by handling missing values, encoding categorical variables, scaling features, and splitting the data into training and testing sets or using cross-validation.

Model selection in machine learning

- **Select a Variety of Models:** Choose a diverse set of machine learning algorithms to experiment with. This may include linear models (e.g., Linear Regression, Logistic Regression), tree-based models (e.g., Decision Trees, Random Forests, Gradient Boosting), support vector machines, k-Nearest Neighbors, neural networks, and clustering algorithms, depending on your problem.
- **Baseline Model:** Start with a simple model or a default configuration of a model as your baseline. This serves as a point of reference for model comparisons.

Model selection in machine learning

- **Evaluation Metrics:** Define the evaluation metrics that are appropriate for your problem. Common metrics include accuracy, precision, recall, F1 score, mean squared error (MSE), root mean squared error (RMSE), etc. The choice of metric depends on the type of problem (classification, regression, clustering) and the project goals.
- **Model Training and Hyperparameter Tuning:** Train each model on the training data and tune their hyperparameters. You can use techniques like grid search, random search, or Bayesian optimization to find the best hyperparameters for each model.

Model selection in machine learning

- **Cross-Validation:** Use cross-validation techniques (e.g., k-fold cross-validation) to estimate how well each model generalizes to unseen data. This helps reduce the risk of overfitting and provides a more robust assessment of model performance.
- **Model Comparison:** Compare the performance of different models based on your chosen evaluation metrics. You can create a table or visualization to easily compare their results.
- **Ensemble Methods:** Consider using ensemble methods like bagging (e.g., Random Forests) or boosting (e.g., AdaBoost, Gradient Boosting) to combine multiple models. Ensembles often yield better performance than individual models.

Model selection in machine learning

- **Overfitting and Underfitting:** Pay attention to signs of overfitting or underfitting. An overfit model performs well on the training data but poorly on the test data, while an underfit model performs poorly on both. Choose models that strike a good balance.
- **Domain Knowledge:** Incorporate domain knowledge and insights into the model selection process. Certain algorithms may be more suitable based on the specific characteristics of the problem.
- **Interpretability:** Consider the interpretability of the models. Some models, like decision trees or linear regression, are more interpretable than complex models like deep neural networks.

Model selection in machine learning

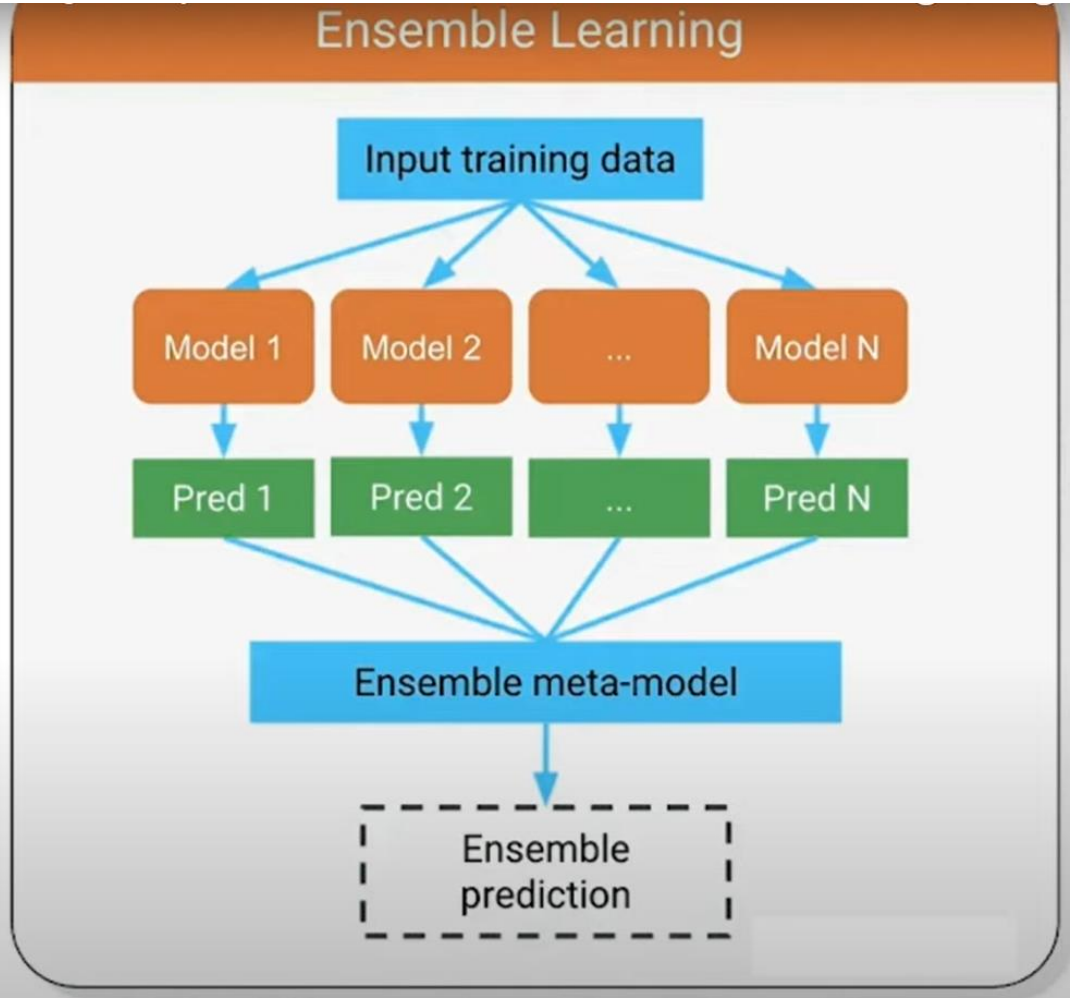
- **Regularization:** Apply regularization techniques, such as L1 or L2 regularization, to prevent overfitting in models that exhibit it.
- **Bias and Fairness:** Assess models for bias and fairness concerns, especially in applications with ethical considerations. Ensure that the chosen model does not discriminate against certain groups.
- **Final Model Selection:** Based on your experimentation and model comparisons, select the model that performs the best according to your evaluation metrics and project goals.

Model selection in machine learning

- **Test on Holdout Data:** Evaluate your final model on a separate holdout dataset that it has never seen before to estimate its real-world performance.
- **Documentation:** Document the chosen model, its hyperparameters, and the reasoning behind your decision. This documentation is crucial for reproducibility and future reference.
- **Deployment and Monitoring:** If the model will be used in a real-world application, deploy it and continuously monitor its performance. Be prepared to re-evaluate and update the model as new data becomes available.

Ensemble Learning

Machine Learning



Ensemble Learning in Machine Learning

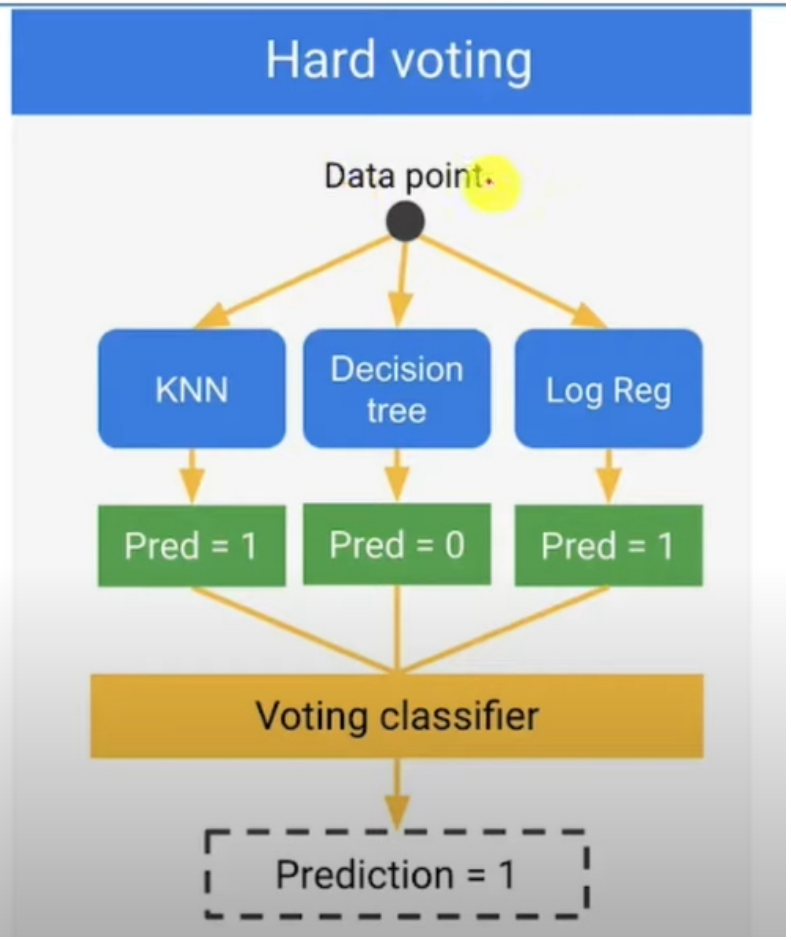
- Ensemble learning is a supervised learning technique used in machine learning to improve overall performance by combining the predictions from multiple models.

Ensemble Learning - Types of Ensemble Methods

- Voting (Averaging)
- Bootstrap aggregation (bagging)
- Random Forests
- Boosting
- Stacked Generalization (Blending)

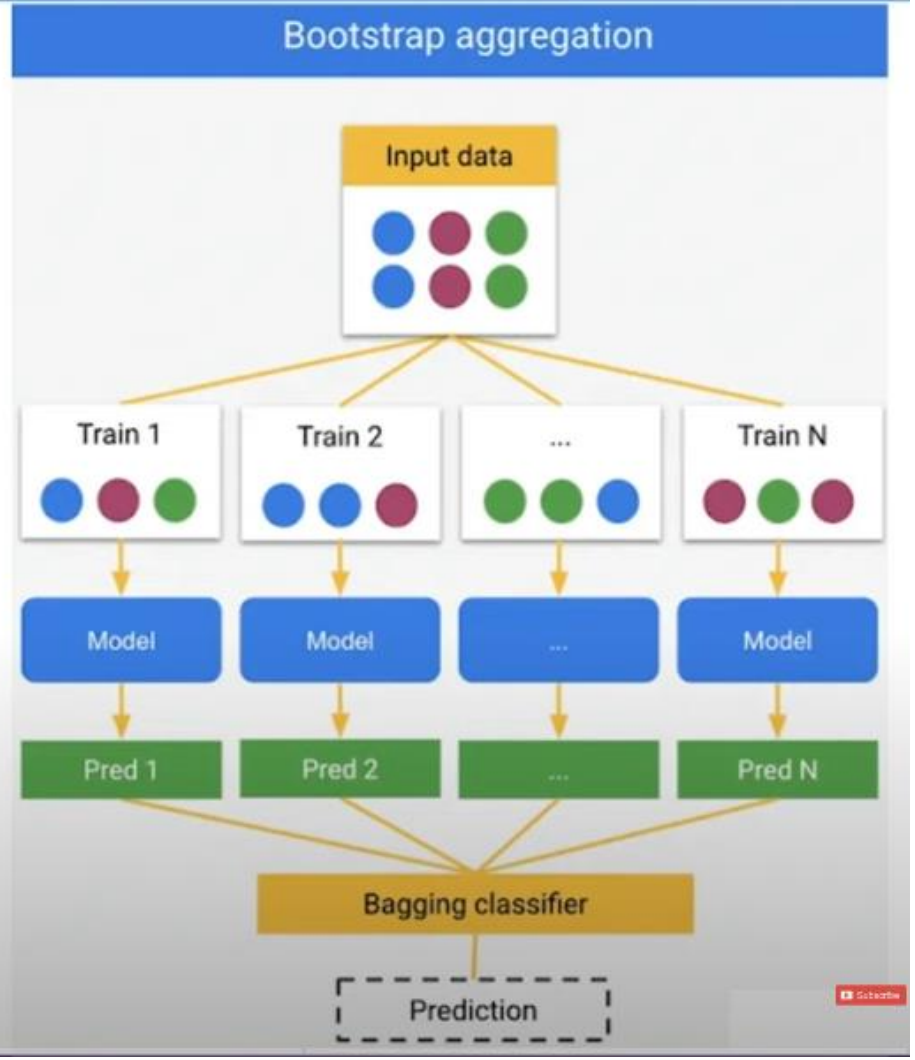
Ensemble Learning – Voting (Averaging)

- Voting is an ensemble machine learning algorithm that involves making a prediction that is the average (regression) or the sum (classification) of multiple machine learning models.



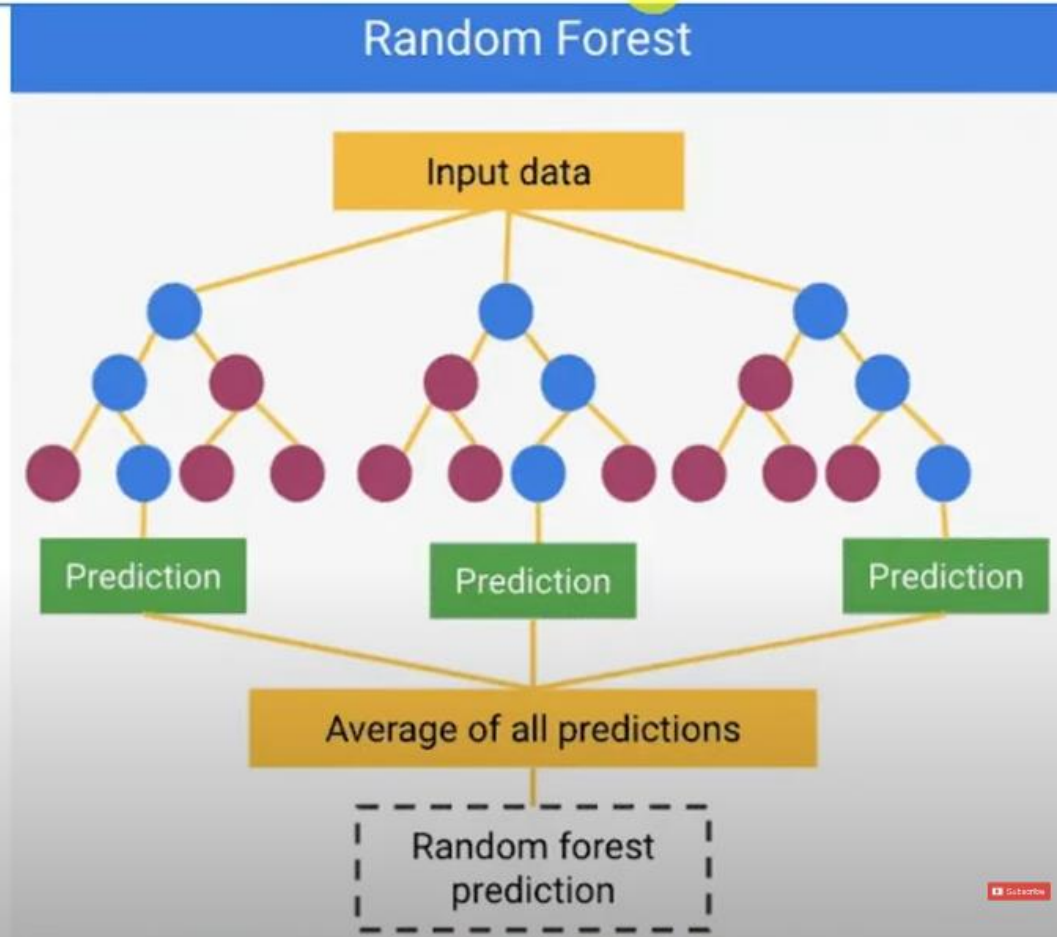
Ensemble Learning – Bootstrap aggregation (bagging)

- Bootstrap Aggregating, also known as bagging, is a machine learning ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms like classification and regression.
- It decreases the variance and helps to avoid overfitting.
- It is usually applied to decision tree methods.
- Bagging is a special case of the model averaging approach.



Ensemble Learning – Random Forest

- Random forest is a commonly-used machine learning algorithm.
- A random forest is an ensemble learning method where multiple decision trees are constructed and then they are merged to get a more accurate prediction.



Ensemble Learning – Boosting

- Boosting is an ensemble modeling technique that attempts to build a strong classifier from the number of weak classifiers.
- It is done by building a model by using weak models in series.
- Firstly, a model is built from the training data.

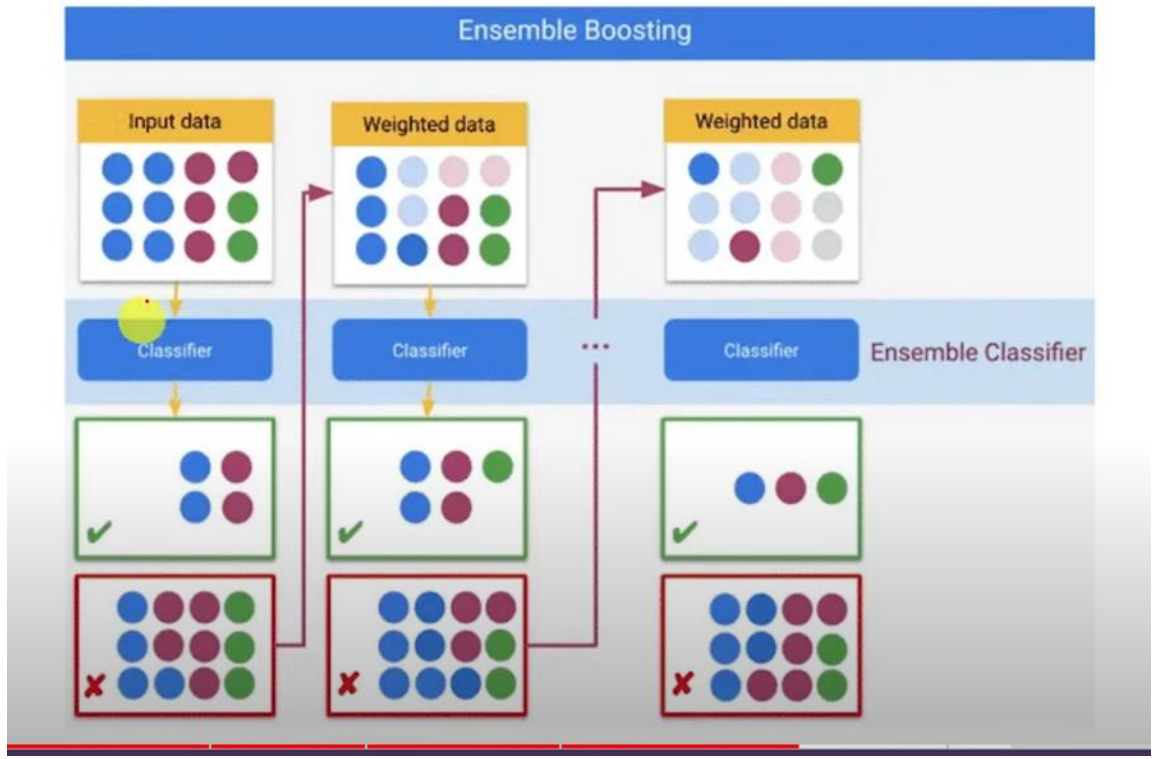
Ensemble Learning – Boosting

- Boosting is an ensemble modeling technique that attempts to build a strong classifier from the number of weak classifiers.
- It is done by building a model by using weak models in series.
- Firstly, a model is built from the training data.
- Then the second model is built which tries to correct the errors present in the first model.
- This procedure is continued and models are added until either the complete training data set is predicted correctly or the maximum number of models is added.

Ensemble Learning – Boosting

- Boosting is an ensemble modeling technique that attempts to build a strong classifier from the number of weak classifiers.
- It is done by building a model by using weak models in series.
- Firstly, a model is built from the training data.
- Then the second model is built which tries to correct the errors present in the first model.
- This procedure is continued and models are added until either the complete training data set is predicted correctly or the maximum number of models is added.

Ensemble Learning – Boosting



Ensemble Learning – Stacked Generalization (Blending)

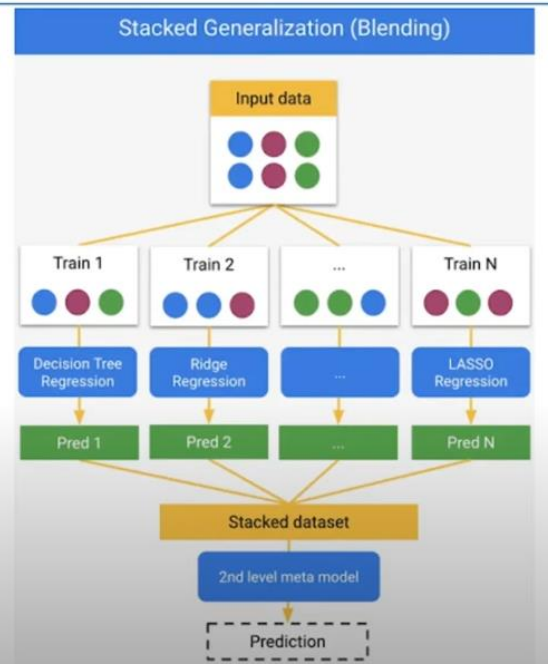
- Stacking, Blending and and Stacked Generalization are all the same thing with different names. It is a kind of ensemble learning.
- In traditional ensemble learning, we have multiple classifiers trying to fit to a training set to approximate the target function.
- Since each classifier will have its own output, we will need to find a combining mechanism to combine the results.
- This can be through voting (majority wins), weighted voting (some classifier has more authority than the others), averaging the results, etc.

Ensemble Learning – Stacked Generalization (Blending)

- Stacking, Blending and and Stacked Generalization are all the same thing with different names. It is a kind of ensemble learning.
- In traditional ensemble learning, we have multiple classifiers trying to fit to a training set to approximate the target function.
- Since each classifier will have its own output, we will need to find a combining mechanism to combine the results.
- This can be through voting (majority wins), weighted voting (some classifier has more authority than the others), averaging the results, etc.

Ensemble Learning – Stacked Generalization (Blending)

- In stacking, the combining mechanism is that the output of the classifiers (Level 0 classifiers) will be used as training data for another classifier (Level 1 classifier) to approximate the same target function.
- Basically, you let the Level 1 classifier to figure out the combining mechanism.

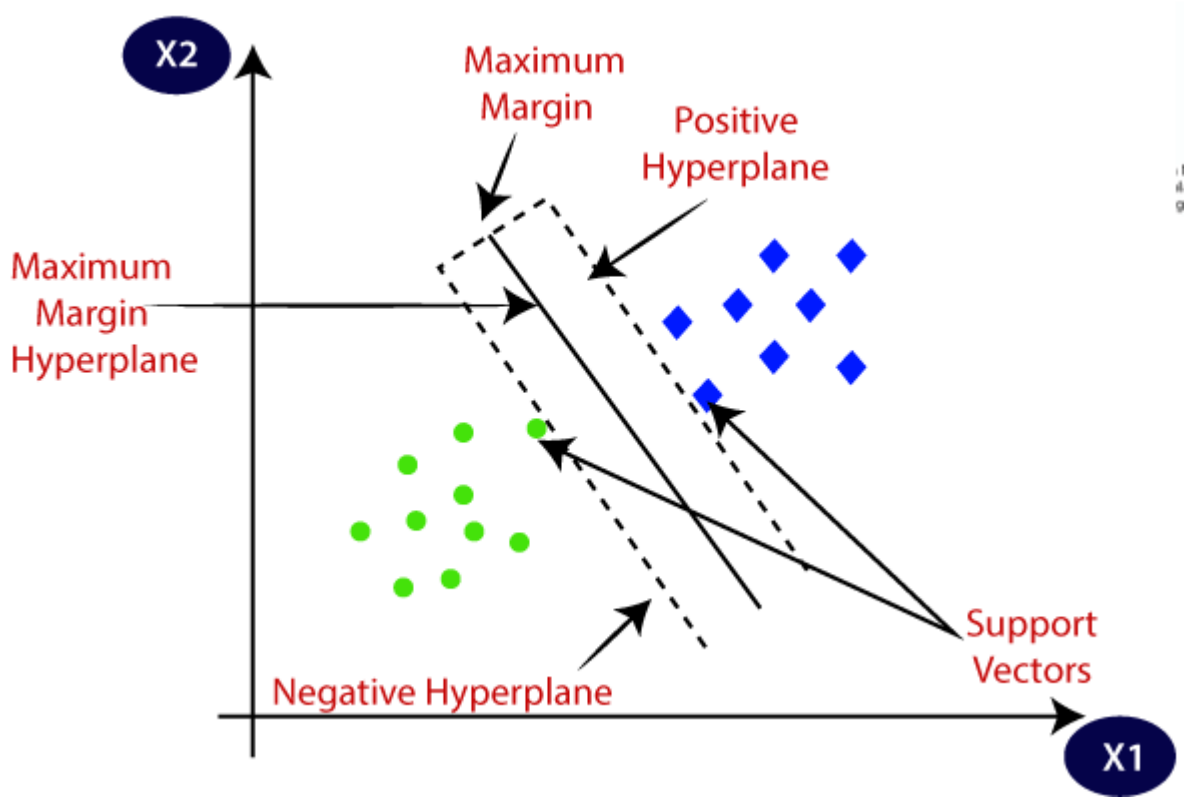


What is the Support Vector Machine?

- “Support Vector Machine” (SVM) is a supervised machine learning algorithm that can be used for both classification or regression challenges.
- However, it is mostly used in classification problems.
- In the SVM algorithm, we plot each data item as a point in n-dimensional space with the value of each feature being the value of a particular coordinate.
- Then, we perform classification by finding the hyper-plane that differentiates the two classes very well.

Support Vector Machine Algorithm

- The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a **hyperplane.**
- SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.


X2
Maximum
Margin
Positive
Hyperplane
Maximum
Margin
Hyperplane
Support
Vectors
Negative Hyperplane
X1

Hyperplane and Support Vectors in the SVM algorithm:

- **Hyperplane:** There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.
- The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features, then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.
- We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

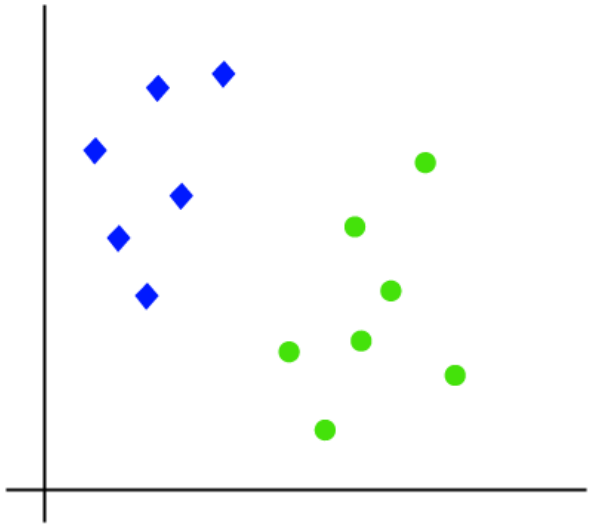
Hyperplane and Support Vectors in the SVM algorithm:

- **Support Vectors:**

The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

How does SVM works?

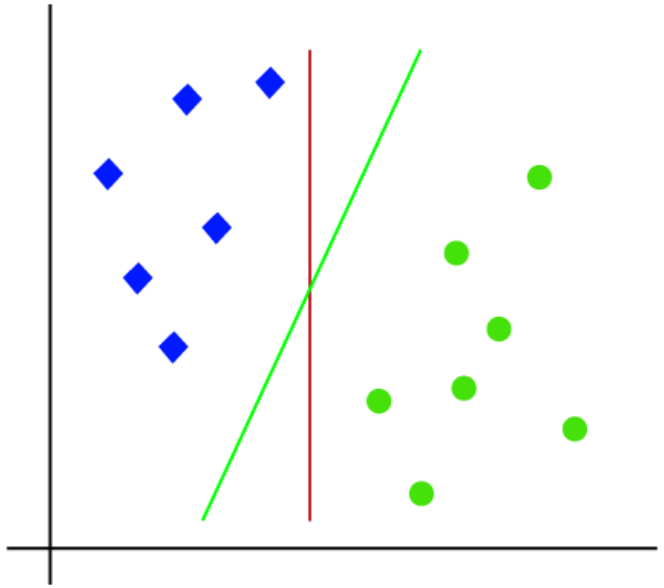
- The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the new pair(x1, x2) of coordinates in either green or blue



How does SVM works?

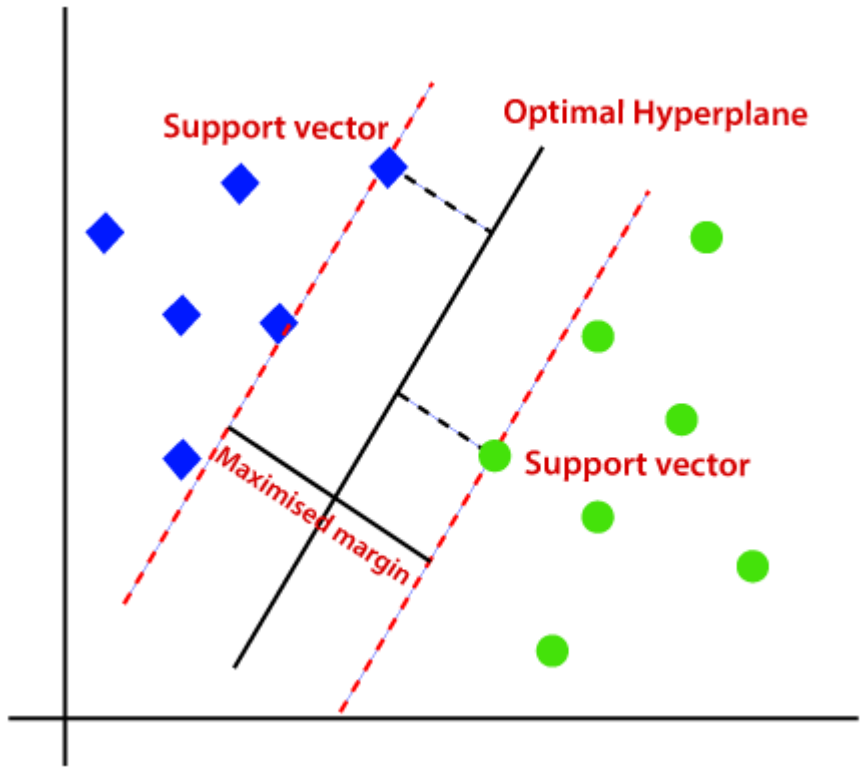
So as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes.

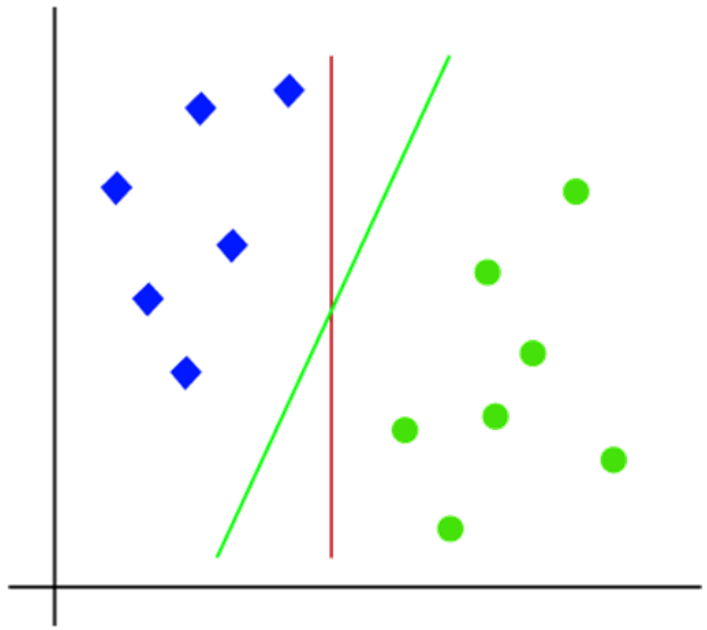
SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a **hyperplane**



How does SVM works?

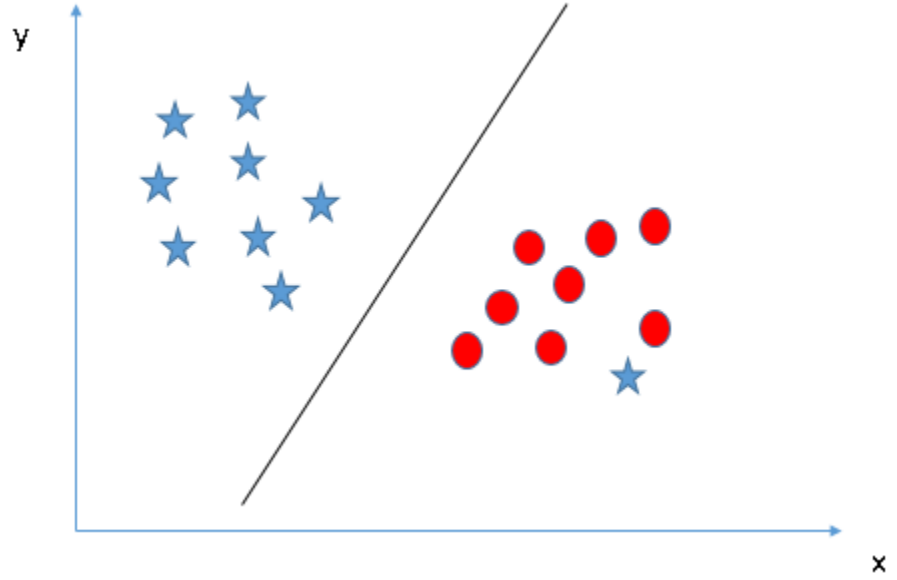
- Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a **hyperplane**.
- SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors.
- The distance between the vectors and the hyperplane is called as **margin**. And the goal of SVM is to maximize this margin.
- The **hyperplane** with maximum margin is called the **optimal hyperplane**.


Support vector
Optimal Hyperplane
Maximised margin
Support vector



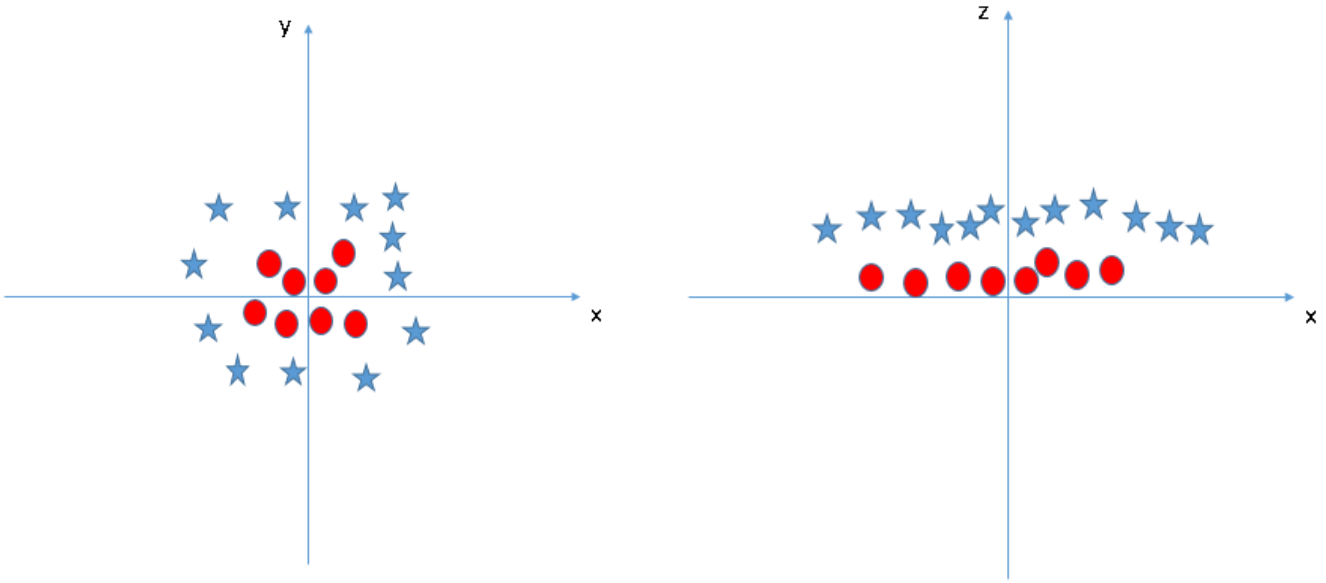
How does SVM works?

It unable to segregate the two classes using a straight line, as one of the stars lies in the territory of other(circle) class as an outlier. The SVM algorithm has a feature to ignore outliers and find the hyper-plane that has the maximum margin. Hence, we can say, SVM classification is robust to outliers.



How does SVM works?

- In the scenario below, we can't have linear hyper-plane between the two classes, so how does SVM classify these two classes? SVM can solve this problem by introducing additional feature. Here, we will add a new feature z=x^2+y^2. Now, let's plot the data points on axis x and z:



In above plot, points to consider are:

•All values for z would be positive always because z is the squared sum of both x and y

•In the original plot, red circles appear close to the origin of x and y axes, leading to lower value of z and star relatively away from the origin result to higher value of z.

SVM Kernel

- The SVM kernel is a function that takes low dimensional input space and transforms it to a higher dimensional space i.e. it converts not separable problem to separable problem.
- It is mostly useful in non-linear separation problem. Simply put, it does some extremely complex data transformations, then finds out the process to separate the data based on the labels or outputs you've defined.

Support Vector Machines (Kernels)

- **Kernel Function** is a method used to take data as input and transform into the required form of processing data.
- “Kernel” is used due to set of mathematical functions used in Support Vector Machine provides the window to manipulate the data.
- So, Kernel Function generally transforms the training set of data so that a non-linear decision surface is able to transformed to a linear equation in a higher number of dimension spaces.
- Basically, It returns the inner product between two points in a standard feature dimension.

Types of SVM kernels:

- **Polynomial Kernel**
- **Sigmoid Kernel**
- **Gaussian Kernel Radial Basis Function (RBF)**