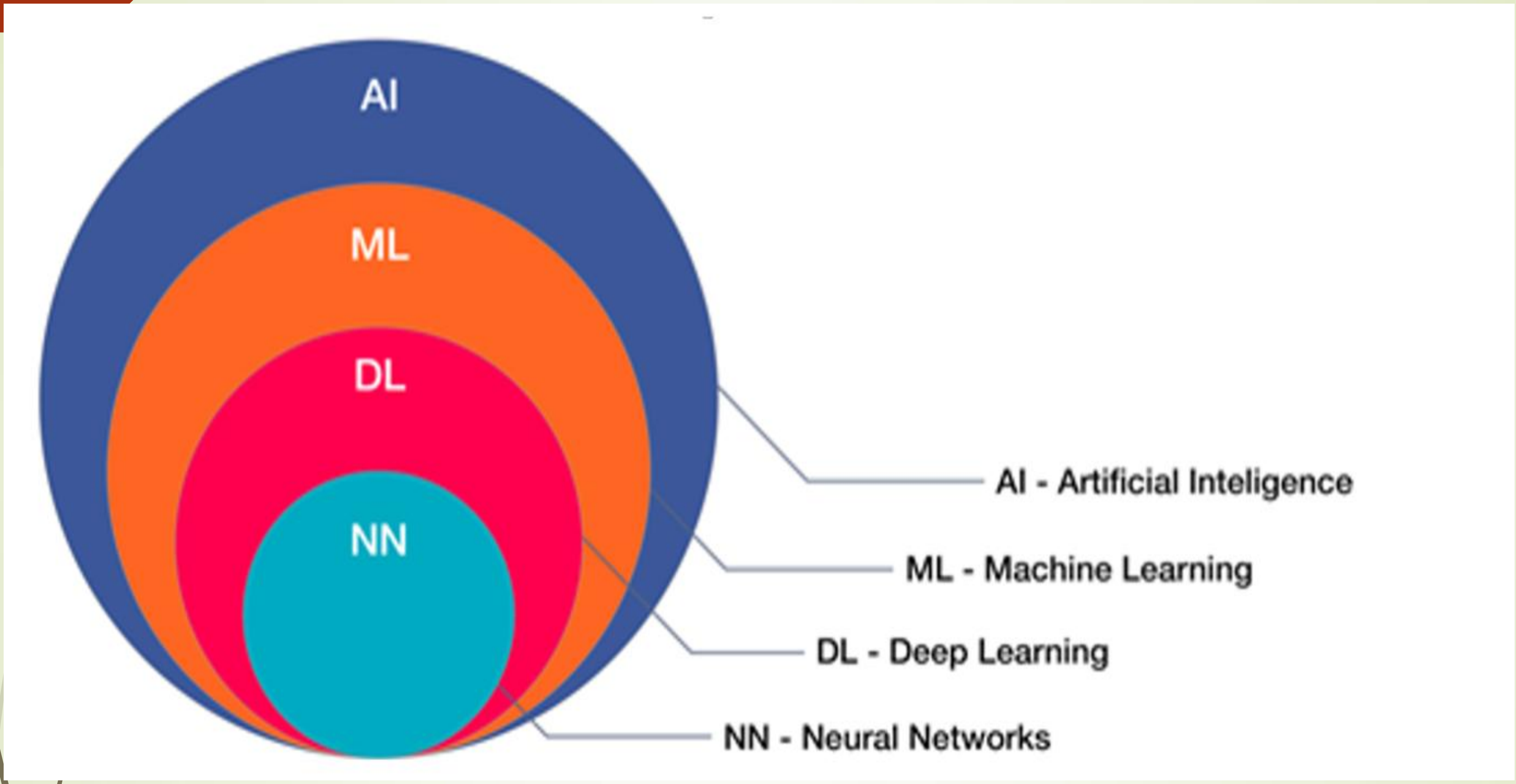
CHAPTER 5:MACHINE LEARNING

Prepared by: Afreen Banu & Ashwin R G


AI
ML
DL
NN
AI - Artificial Inteligence
ML - Machine Learning
DL - Deep Learning
NN - Neural Networks

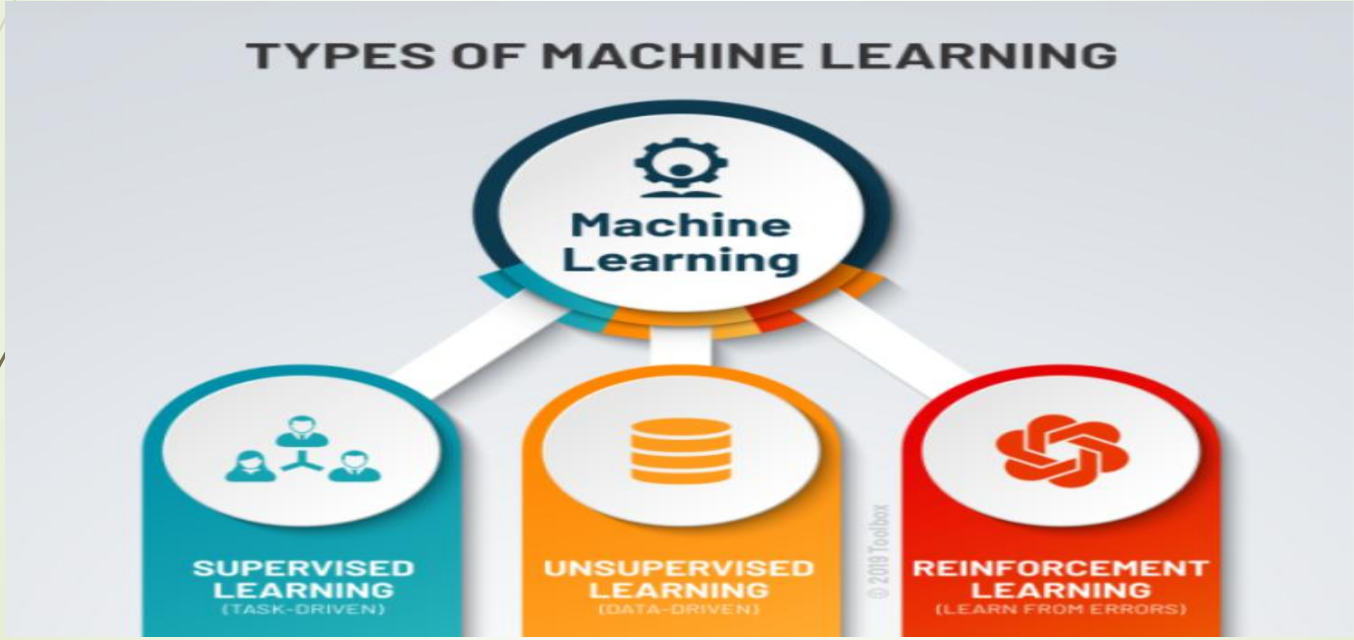
WHAT IS ML?

- Machine learning (ML) is **a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so**. Machine learning algorithms use historical data as input to predict new output values.

Features of ml

- ASSIGNMENT 1

TYPES OF MACHINE LEARNING

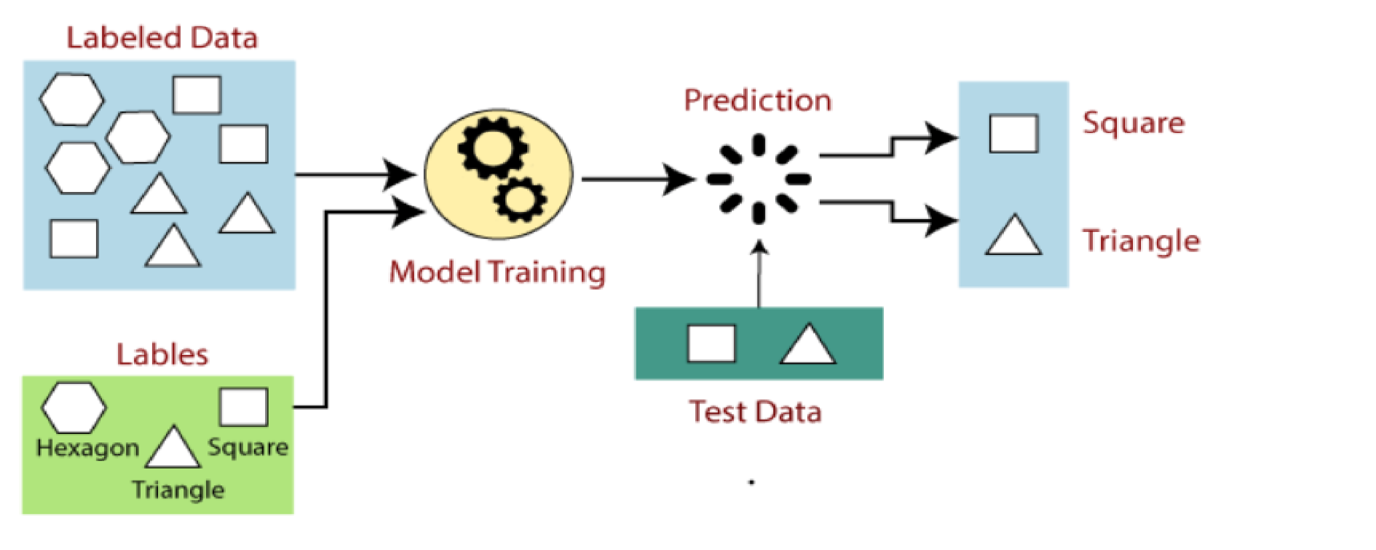


TYPES OF MACHINE LEARNING

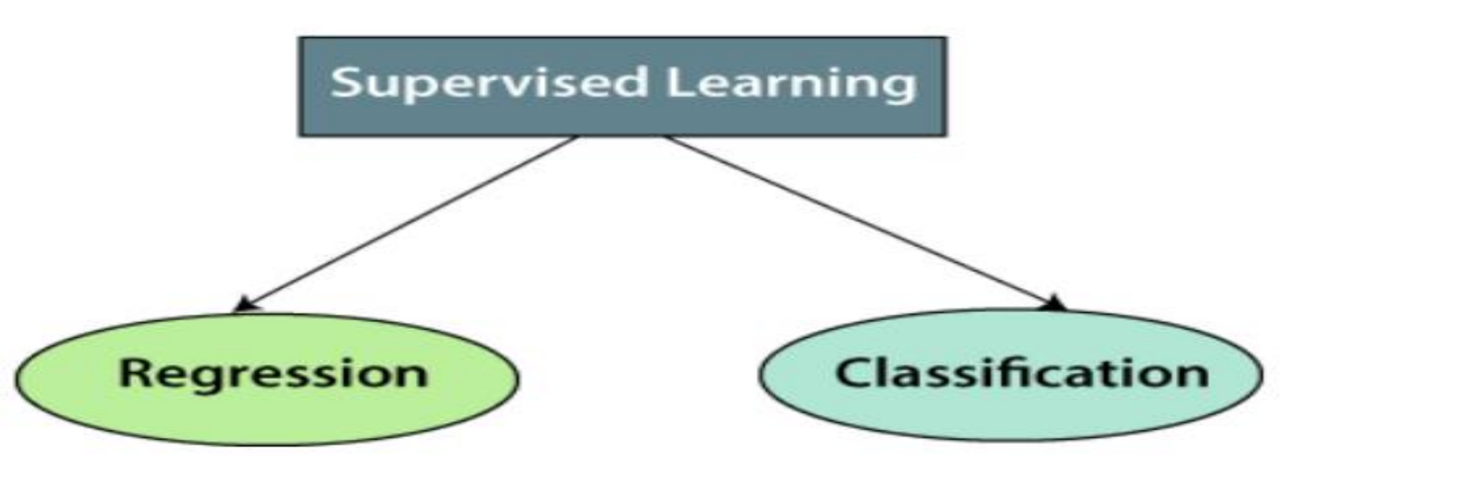
Supervised learning

- Supervised learning is the types of machine learning in which machines are trained using well "labelled" training data, and on basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output.

Working of Supervised learning



Types of supervised learning



regression

- Regression is **a technique for investigating the relationship between independent variables and a dependent variable or outcome.** It's used as a method for predictive modelling in machine learning, in which an algorithm is used to predict continuous outcomes

LINEAR REGRESSION

- Linear Regression assumes that there is a linear relationship present between dependent and independent variables. In simple words, it finds the best fitting line/plane that describes two or more variables.
- **There are 2 types of linear regression:**
- 1. Simple Linear Regression
- 2. Multiple Linear Regression

Applications of Linear Regression in Python

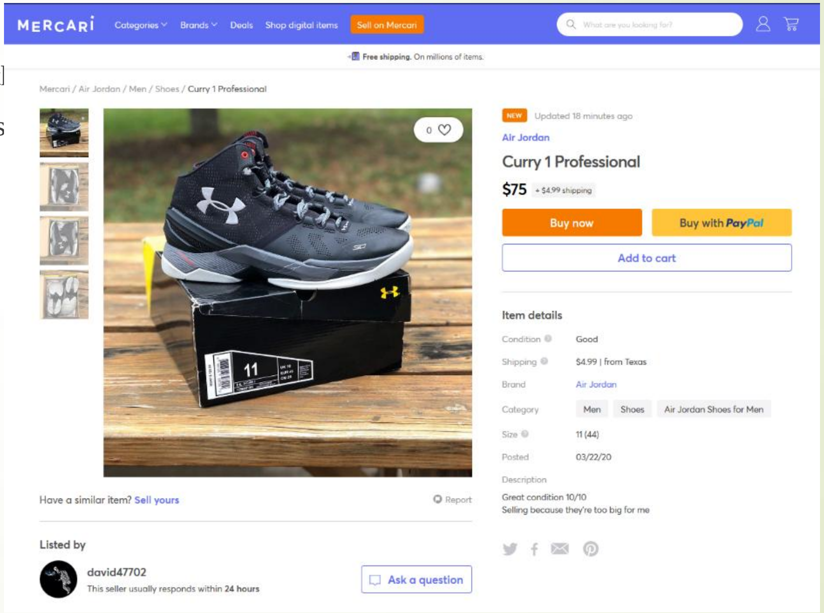
- Economic Growth

Linear regression is used to determine the e
growth of a country or a state in the upcoming
It can also be used to predict a nation's gross
product (GDP).

Applications of Linear Regression in Python

- Product Price

Linear regression can be used to predict what t
of a product will be in the future, whether prices
up or down.

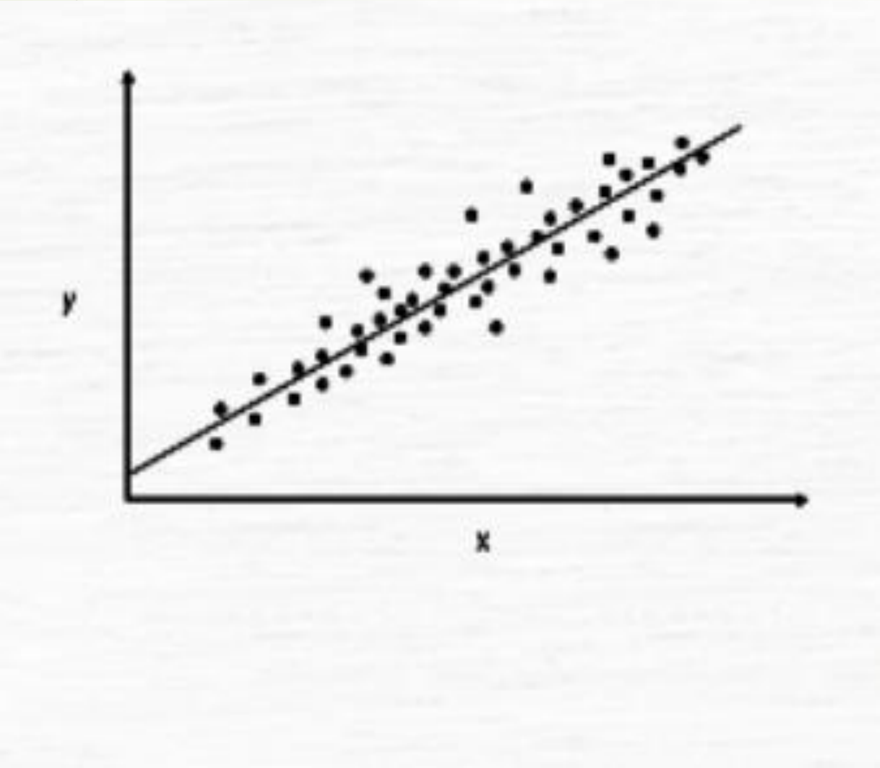


Applications of Linear Regression in Python

- Score Predictions

Linear regression can be used to predict the number of runs a baseball player will score in upcoming games based on previous performance.

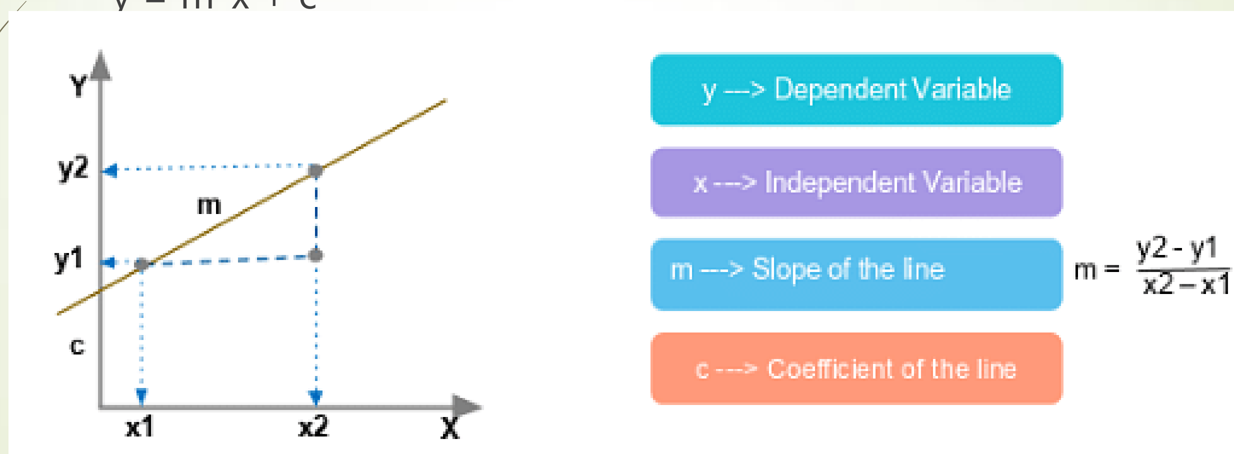
Simple Linear Regression



Regression Equation

- The simplest linear regression equation with one dependent variable and one independent variable is:

$y = m*x + c$



- ***Simple Linear Regression**: It is a type of linear regression model where there is only independent or explanatory variable.*
- **Linear Regression can be written mathematically as follows:**
- **Y = β0 + β1.X1 + β2. X2 + β3. X3 + β4. X4+ β5. X5 + β5. X6 + ϵ**
- **charges = β0 + β1.bmi + β2.age+ β3.sex + β4 .children+ β5.region + β5.smoker + ϵ**
- **charges**= response variable, generally denoted by Y
- **bmi, age, sex, children, region, smoker**=Predictor variables, denoted by X1, X2, X3 and X4 respectively
- **β0 =** Y-intercept
- **β1, β2, β3, β4, β5 =** regression coefficients
- **ϵ =** Error terms (Residuals)

Least Square Method – Finding the best fit line

Regression Line, **y = mx+c** where,

y = Dependent Variable

x= Independent Variable ; c = y-Intercept

$$m = \frac{\sum (x-\bar{x})(y-\bar{y})}{\sum (x-\bar{x})^2}$$

Problem solving

x	y	$x-\bar{x}$	$y-\bar{y}$	$(x-\bar{x})^2$	$(x-\bar{x})(y-\bar{y})$
1	3	-2	-0.6	4	1.2
2	4	-1	0.4	1	-0.4
3	2	0	-1.6	0	0
4	4	1	0.4	1	0.4
5	5	2	1.4	4	2.8

$$m = \frac{\Sigma\,(x-\bar{x})(y-\bar{y})}{\Sigma\,(x-\bar{x})^2}$$

x	y	$x-\bar{x}$	$y-\bar{y}$	$(x-\bar{x})^2$	$(x-\bar{x})(y-\bar{y})$
1	3	-2	-0.6	4	1.2
2	4	-1	0.4	1	-0.4
3	2	0	-1.6	0	0
4	4	1	0.4	1	0.4
5	5	2	1.4	4	2.8
3	3.6			$\Sigma = 10$	$\Sigma = 4$

$$m = \sum \frac{(x-\bar{x})(y-\bar{y})}{(x-\bar{x})^2} = \frac{4}{10}$$

$y = mx+c$
$3.6=0.4x3 + c$

$y = mx+c$
$3.6 - 1.2 = c$

$y = mx+c$
$c = 2.4$

$m = 0.4$
$c = 2.4$
$y = 0.4x + 2.4$

For given m = 0.4 & c = 2.4, lets predict values for y for x = {1,2,3,4,5}

$y = 0.4 \times 1 + 2.4 = 2.8$
$y = 0.4 \times 2 + 2.4 = 3.2$
$y = 0.4 \times 3 + 2.4 = 3.6$
$y = 0.4 \times 4 + 2.4 = 4.0$
$y = 0.4 \times 5 + 2.4 = 4.4$

x	y_p
1	2.8
2	3.2
3	3.6
4	4.0
5	4.4

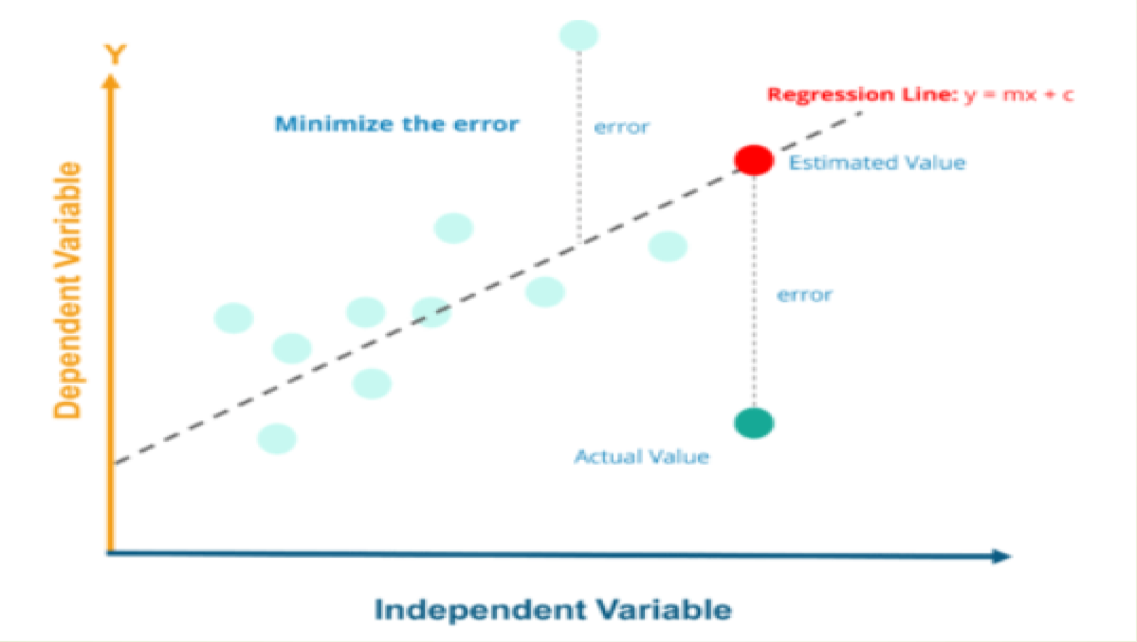
Calculation of R^2

x	y	$y - \bar{y}$
1	3	- 0.6
2	4	0.4
3	2	-1.6
4	4	0.4
5	5	1.4
[illegible]	3.6	

$$R^2 = 1 - \frac{\sum (y_p - \bar{y})^2}{\sum (y - \bar{y})^2}$$

Least Square Method – Implementation using Python

for the implementation part, i will be using a dataset consisting of head size and brain weight of different people.



MULTIPLE LINEAR REGRESSION

- **Multiple linear regression** is used to estimate the relationship between **two or more independent variables** and **one dependent variable.**

1. How strong the relationship is between two or more independent variables and one dependent variable
2. The value of the dependent variable at a certain value of the independent variables.

Multiple Linear Regression

- In simple linear regression, we have the equation:

  $y = m*x + c$

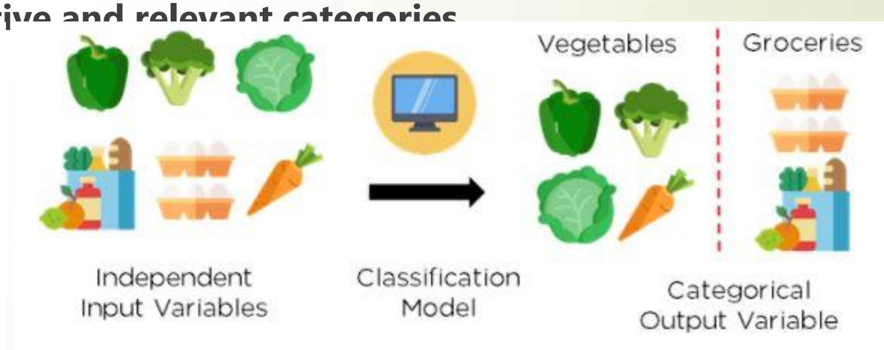
- For multiple linear regression, we have the equation:

  $y = m1x1 + m2x2 + m3x3 + \ldots\ldots + c$

- Here, we have multiple independent variables, x1, x2 and x3, and multiple slopes, m1, m2, m3 and so on.

Classification in Machine Learning

- Classification is defined as **the process of recognition**, **understanding**, and **grouping of objects** and ideas into preset categories a.k.a "sub-populations." With the help of these pre-categorized training datasets, classification in machine learning programs leverage a wide range of algorithms to classify future **datasets into respective and relevant categories.**



Learners in Classification Problems

- **Lazy Learners**

It first stores the training dataset before waiting for the test dataset to arrive. When using a lazy learner, the classification is carried out using the training dataset's most appropriate data. Less time is spent on training, but more time is spent on predictions. Some of the examples are case-based reasoning and the KNN algorithm.

Learners in Classification Problems

- **Eager Learners**

Before obtaining a test dataset, eager learners build a classification model using a training dataset. They spend more time studying and less time predicting. Some of the examples are ANN, naive Bayes, and Decision trees.

Classification

The Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations on the basis of training data. In Classification, a program learns from the given dataset or observations and then classifies new observation into a number of classes or groups. Such as, **Yes or No, 0 or 1, Spam or Not Spam, cat or dog,**

4 Types Of Classification Tasks In Machine Learning

1. Binary Classification
2. Multi-Class Classification
3. Multi-Label Classification
4. Imbalanced Classification

Binary Classification

Those classification jobs with only two class labels are referred to as binary classification.

Examples comprise -

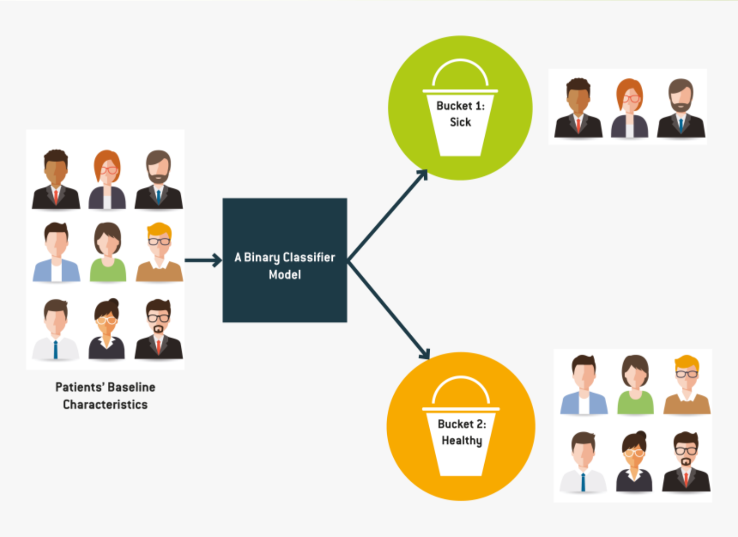
- Prediction of conversion (buy or not).
- Churn forecast (churn or not).
- Detection of spam email (spam or not).
- Binary classification problems often require two classes, one representing the normal state and the other representing the aberrant state.

For instance, the normal condition is "not spam," while the abnormal state is "spam." Another illustration is when a task involving a medical test has a normal condition of "cancer not identified" and an abnormal state of "cancer detected." Class label 0 is given to the class in the normal state, whereas class label 1 is given to the class in the abnormal condition. A model that forecasts a Bernoulli probability distribution for each case is frequently used to represent a binary classification task.

Binary Classification

The following are well-known binary classification algorithms:

- Logistic Regression
- Support Vector Machines
- Simple Bayes
- Decision Trees



Multi-Class Classification

Multi-class labels are used in classification tasks referred to as multi-class classification.

Examples comprise -

- Categorization of faces.
- Classifying plant species.
- Character recognition using optical.

The multi-class classification does not have the idea of normal and abnormal outcomes, in contrast to binary classification. Instead, instances are grouped into one of several well-known classes. In some cases, the number of class labels could be rather high. In a facial recognition system, for instance, a model might predict that a shot belongs to one of thousands or tens of thousands of faces.

Text translation models and other problems involving word prediction could be categorized as a particular case of multi-class classification. Each word in the sequence of words to be predicted requires a multi-class classification, where the vocabulary size determines the number of possible classes that may be predicted and may range from tens of thousands to hundreds of thousands of words.

Popular Classification Algorithms

- Decision tree
- Naive Bayes
- K-Nearest Neighbors
- Support Vector Machines
- Logistic Regression
- K means

Multi-Class Classification

For multi-class classification, many binary classification techniques are applicable.

The following well-known algorithms can be used for multi-class classification:

- Progressive Boosting
- Choice trees
- Nearest K Neighbors
- Rough Forest / Random Forest
- Simple Bayes

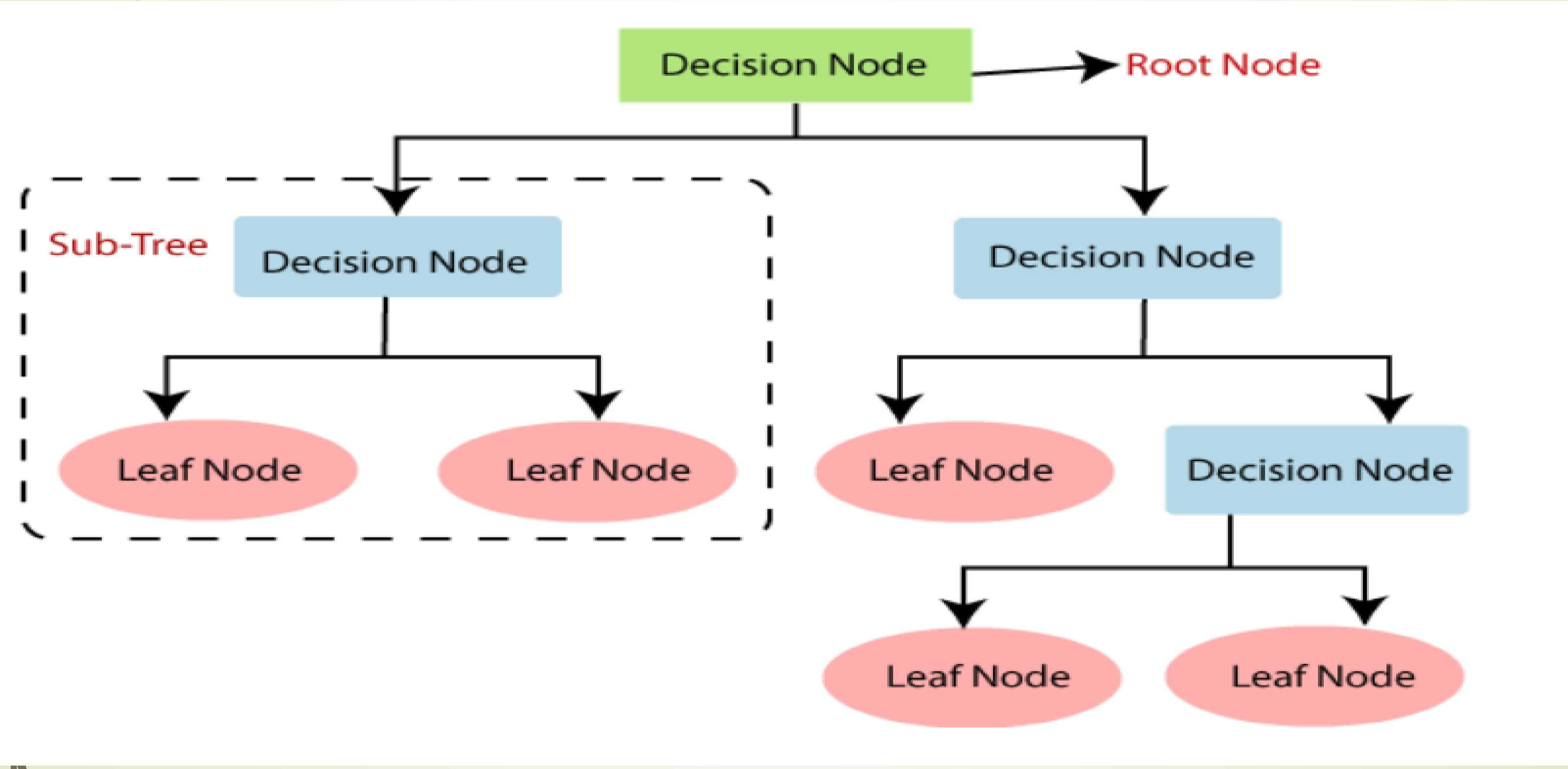
Multi-class problems can be solved using algorithms created for binary classification.

Decision tree

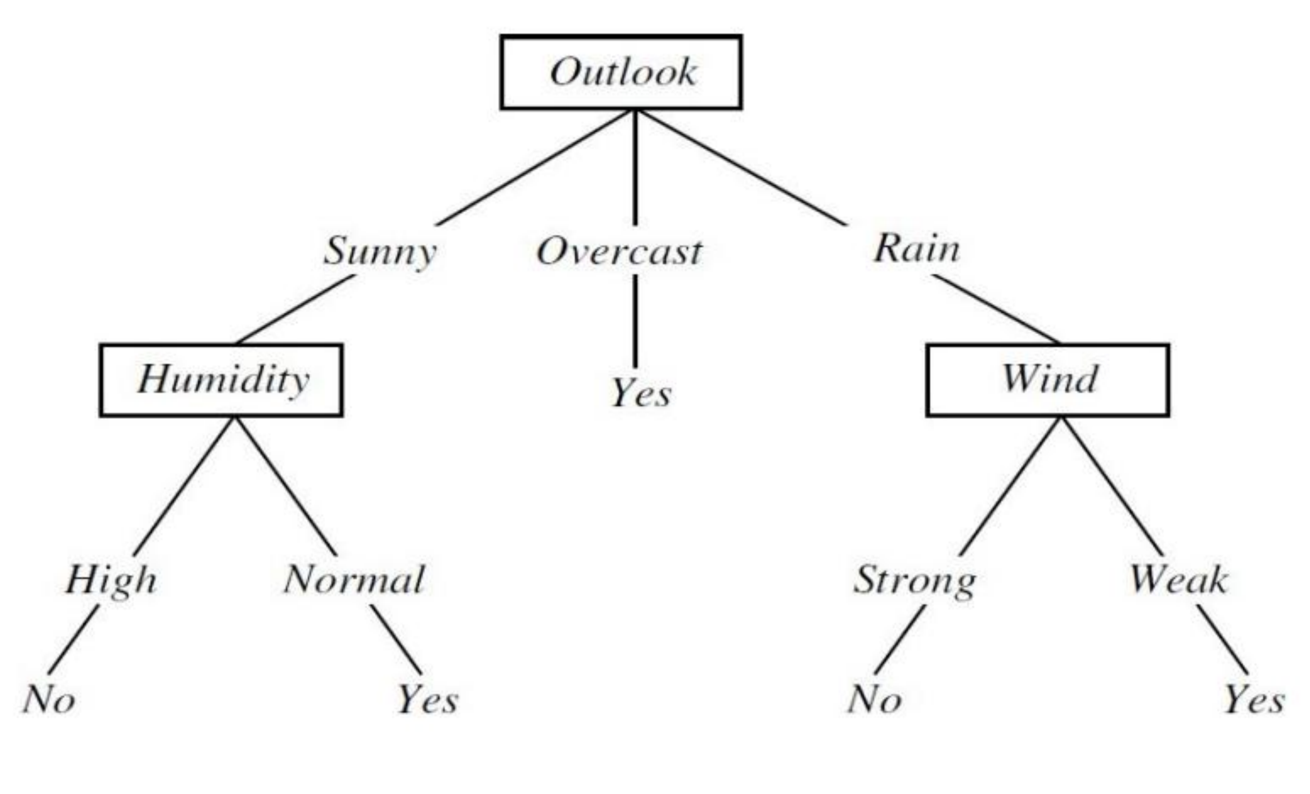
Key Terms

- **Root Node** - It represents the entire population or sample and this further gets divided into two or more homogeneous sets.
- **Splitting** - It is a process of dividing a node into two or more sub-nodes.
- **Decision Node** - When a sub-node splits into further sub-nodes, then it is called a decision node.
- **Leaf/ Terminal Node** - Nodes do not split is called Leaf or Terminal node..
- **Branch / Sub-Tree** - A subsection of the entire tree is called branch or sub-tree.
- **Parent and Child Node** - A node, which is divided into sub-nodes is called the parent node of sub-nodes whereas sub-nodes are the child of the parent node.
- **Entropy** - A decision tree is built top-down from a root node and involves partitioning the data into subsets that contain instances with similar values (homogeneous). ID 3 algorithm uses entropy to calculate the homogeneity of a sample. If the sample is completely homogeneous the entropy is zero and if the sample is equally divided it has an entropy of one.
- **Information Gain** - The information gain is based on the decrease in entropy after a dataset is split on an attribute. Constructing a decision tree is all about finding an attribute that returns the highest information gain (i.e., the most homogeneous branches).
- **Internal Node** - An internal node (also known as an inner node, inode for short, or branch node) is any node of a tree that has child nodes
- **Branch** - The lines connecting elements are called "branches".

decision t r e e repre se nt ation



- tree to decide whether a person can play tennis or not



- To define information gain, we begin by defining a measure called entropy. *Entropy measures the impurity of a collection of examples.*
- Given a collection S, containing positive and negative examples of some target concept, the entropy of S relative to this Boolean classification is

$$Entropy\,(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

Where,

p_+ is the proportion of positive examples in S

p_- is the proportion of negative examples in S.

- The entropy is 0 if all members of S belong to the same class
- The entropy is 1 when the collection contains an equal number of positive and negative examples
- If the collection contains unequal numbers of positive and negative examples, the entropy is between 0 and 1

INFORMATION GAIN MEASURES THE EXPECTED REDUCTION IN ENTROPY

- ***Information gain,*** is the expected reduction in entropy caused by partitioning the examples according to this attribute.
- The information gain, Gain(S, A) of an attribute A, relative to a collection of examples S, is defined as

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

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	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

Attribute: Outlook

$Values\ (Outlook) = Sunny, Overcast, Rain$

$S = [9+, 5-]$ $\quad Entropy(S) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.94$

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

$S_{Overcast} \leftarrow [4+, 0-]$ $\quad Entropy(S_{Overcast}) = -\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4} = 0$

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

$$Gain\ (S, Outlook) = Entropy(S) - \sum_{v \in \{Sunny, Overcast, Rain\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, Outlook)$$

$$= Entropy(S) - \frac{5}{14} Entropy(S_{Sunny}) - \frac{4}{14} Entropy(S_{Overcast})$$

$$- \frac{5}{14} Entropy(S_{Rain})$$

$$Gain(S, Outlook) = 0.94 - \frac{5}{14} 0.971 - \frac{4}{14} 0 - \frac{5}{14} 0.971 = 0.2464$$

Day	Outlook	Temp	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

Attribute: Temp

$Values\ (Temp) = Hot, Mild, Cool$

$S = [9+, 5-]$ $\quad Entropy(S) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.94$

$S_{Hot} \leftarrow [2+, 2-]$ $\quad Entropy(S_{Hot}) = -\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4} = 1.0$

$S_{Mild} \leftarrow [4+, 2-]$ $\quad Entropy(S_{Mild}) = -\frac{4}{6}\log_2\frac{4}{6} - \frac{2}{6}\log_2\frac{2}{6} = 0.9183$

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

$$Gain\ (S, Temp) = Entropy(S) - \sum_{v \in \{Hot, Mild, Cool\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, Temp)$$

$$= Entropy(S) - \frac{4}{14} Entropy(S_{Hot}) - \frac{6}{14} Entropy(S_{Mild})$$

$$- \frac{4}{14} Entropy(S_{Cool})$$

$$Gain(S, Temp) = 0.94 - \frac{4}{14} 1.0 - \frac{6}{14} 0.9183 - \frac{4}{14} 0.8113 = 0.0289$$

Day	Outlook	Temp	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

Attribute: Humidity

$Values\ (Humidity) = High, Normal$

$S = [9+, 5-]$ $\quad Entropy(S) = -\frac{9}{14} log_2 \frac{9}{14} - \frac{5}{14} log_2 \frac{5}{14} = 0.94$

$S_{High} \leftarrow [3+, 4-]$ $\quad Entropy(S_{High}) = -\frac{3}{7} log_2 \frac{3}{7} - \frac{4}{7} log_2 \frac{4}{7} = 0.9852$

$S_{Normal} \leftarrow [6+, 1-]$ $\quad Entropy(S_{Normal}) = -\frac{6}{7} log_2 \frac{6}{7} - \frac{1}{7} log_2 \frac{1}{7} = 0.5916$

$$Gain\ (S, Humidity) = Entropy(S) - \sum_{v \in \{High, Normal\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, Humidity)$$

$$= Entropy(S) - \frac{7}{14} Entropy(S_{High}) - \frac{7}{14} Entropy(S_{Normal})$$

$$Gain(S, Humidity) = 0.94 - \frac{7}{14} 0.9852 - \frac{7}{14} 0.5916 = 0.1516$$

Day	Outlook	Temp	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

Attribute: Wind

$Values\ (Wind) = Strong, Weak$

$S = [9+, 5-]$ $\quad Entropy(S) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.94$

$S_{Strong} \leftarrow [3+, 3-]$ $\quad Entropy(S_{Strong}) = 1.0$

$S_{Weak} \leftarrow [6+, 2-]$ $\quad Entropy(S_{Weak}) = -\frac{6}{8}\log_2\frac{6}{8} - \frac{2}{8}\log_2\frac{2}{8} = 0.8113$

$$Gain\ (S, Wind) = Entropy(S) - \sum_{v \in \{Strong, Weak\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, Wind) = Entropy(S) - \frac{6}{14} Entropy(S_{Strong}) - \frac{8}{14} Entropy(S_{Weak})$$

$$Gain(S, Wind) = 0.94 - \frac{6}{14}\, 1.0 - \frac{8}{14}\, 0.8113 = 0.0478$$

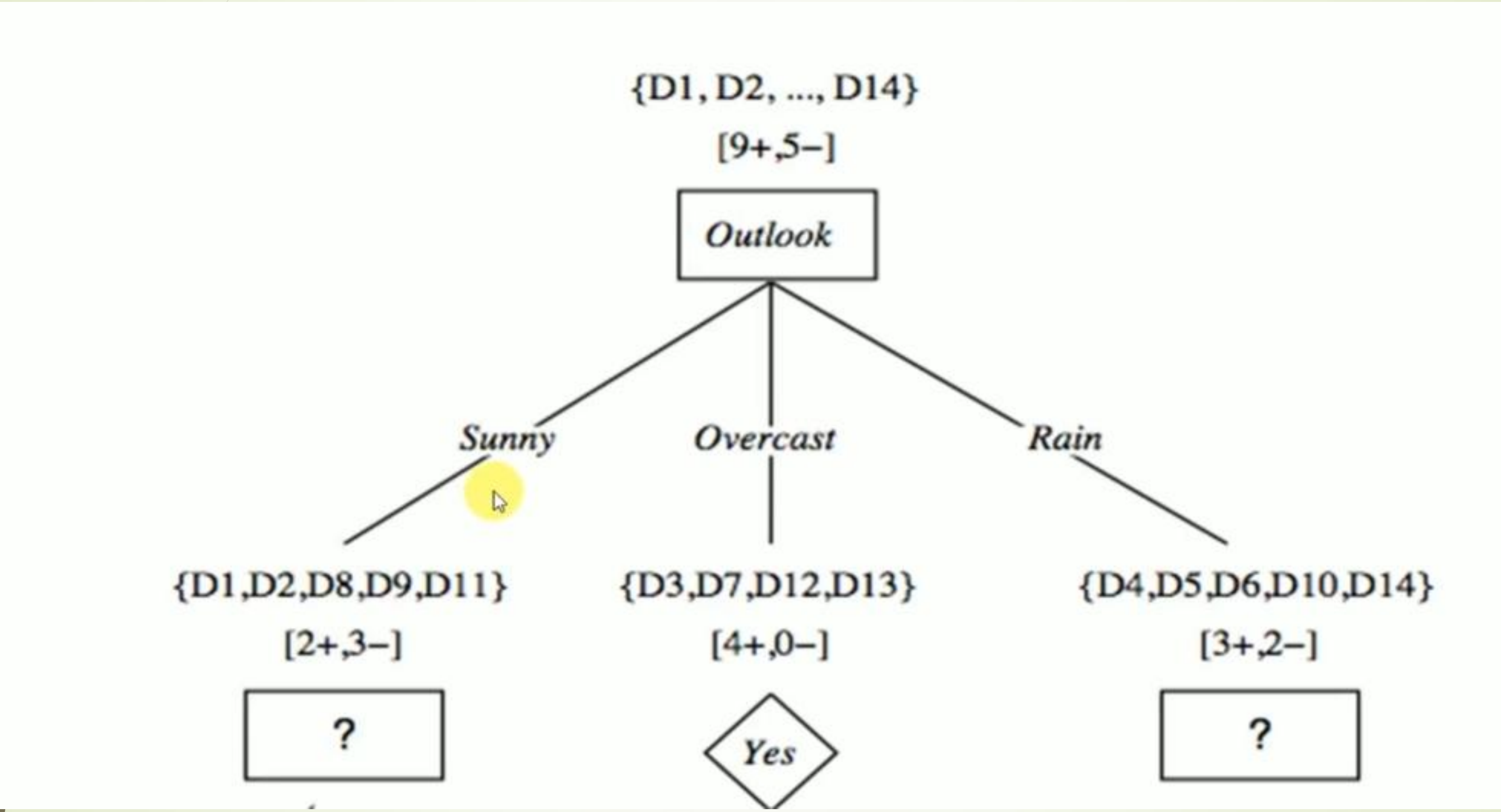
Day	Outlook	Temp	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	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

$$Gain(S, Outlook) = 0.2464$$

$$Gain(S, Temp) = 0.0289$$

$$Gain(S, Humidity) = 0.1516$$

$$Gain(S, Wind) = 0.0478$$



{D1, D2, ..., D14}
[9+,5−]
Outlook
Sunny
Overcast
Rain
{D1,D2,D8,D9,D11}
[2+,3−]
?
{D3,D7,D12,D13}
[4+,0−]
Yes
{D4,D5,D6,D10,D14}
[3+,2−]
?

Day	Temp	Humidity	Wind	Play Tennis
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

Attribute: Temp

$Values\ (Temp) = Hot, Mild, Cool$

$S_{Sunny} = [2+, 3-]$ $\quad Entropy(S_{Sunny}) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5} = 0.97$

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

$S_{Mild} \leftarrow [1+, 1-]$ $\quad Entropy(S_{Mild}) = 1.0$

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

$$Gain\ (S_{Sunny}, Temp) = Entropy(S) - \sum_{v \in \{Hot, Mild, Cool\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Sunny}, Temp)$$

$$= Entropy(S) - \frac{2}{5}Entropy(S_{Hot}) - \frac{2}{5}Entropy(S_{Mild})$$

$$- \frac{1}{5}Entropy(S_{Cool})$$

$$Gain(S_{sunny}, Temp) = 0.97 - \frac{2}{5}0.0 - \frac{2}{5}1 - \frac{1}{5}0.0 = 0.570$$

Day	Temp	Humidity	Wind	Play Tennis
Dl	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
Dl1	Mild	Normal	Strong	Yes

Attribute: Wind

$Values\ (Wind) = Strong, Weak$

$S_{Sunny} = [2+, 3-]$ $\quad$ $Entropy(S) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5} = 0.97$

$S_{Strong} \leftarrow [1+, 1-]$ $\quad$ $Entropy(S_{Strong}) = 1.0$

$S_{Weak} \leftarrow [1+, 2-]$ $\quad$ $Entropy(S_{Weak}) = -\frac{1}{3}log_2\frac{1}{3} - \frac{2}{3}log_2\frac{2}{3} = 0.9183$

$$Gain\ (S_{Sunny}, Wind) = Entropy(S) - \sum_{v \in \{Strong, Weak\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Sunny}, Wind) = Entropy(S) - \frac{2}{5}Entropy(S_{Strong}) - \frac{3}{5}Entropy(S_{Weak})$$

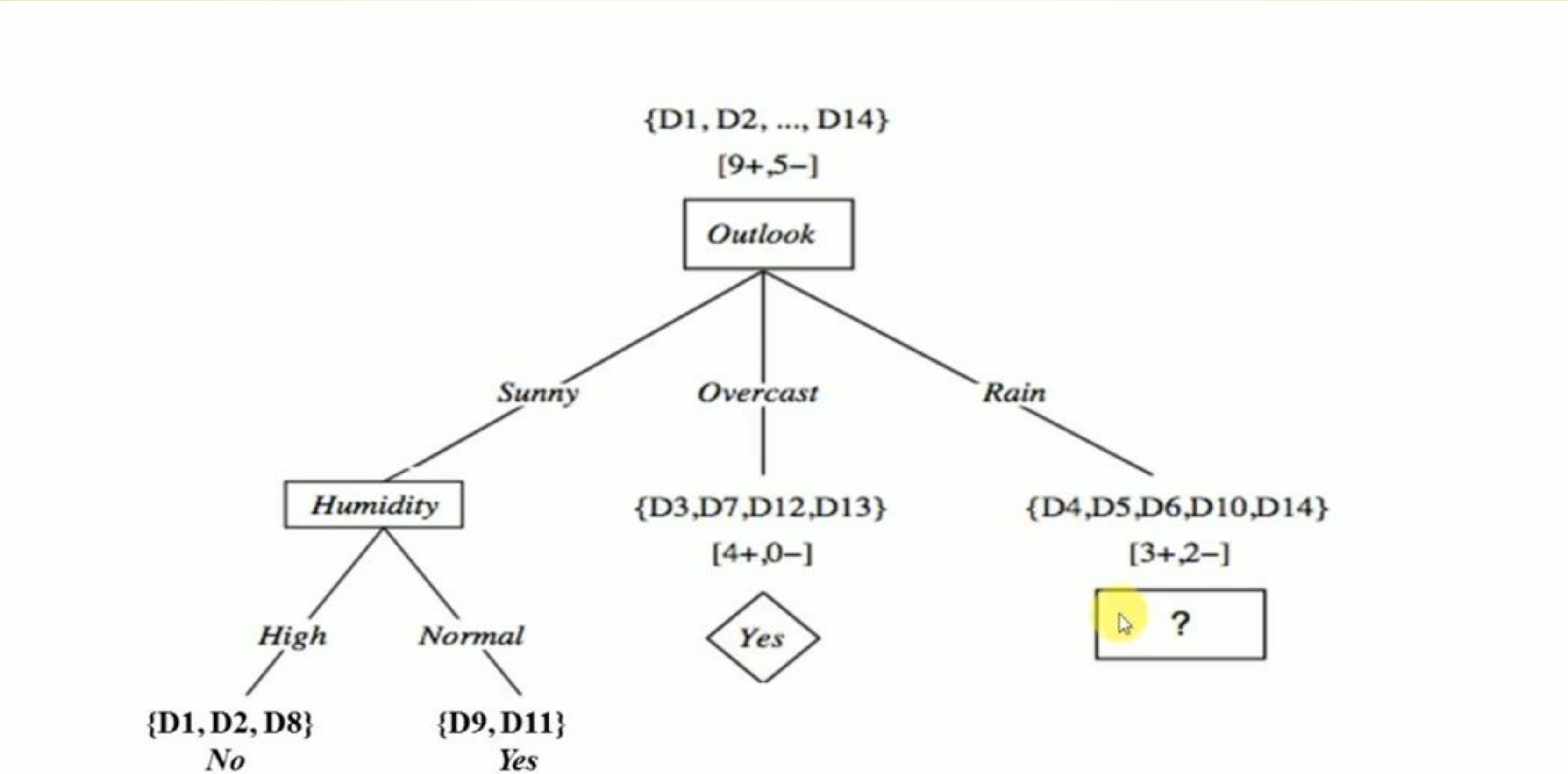
$$Gain(S_{sunny}, Wind) = 0.97 - \frac{2}{5}1.0 - \frac{3}{5}0.918 = 0.0192$$

Day	Temp	Humidity	Wind	Play Tennis
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

$$Gain(S_{sunny}, Temp) = 0.570$$

$$Gain(S_{sunny}, Humidity) = 0.97$$

$$Gain(S_{sunny}, Wind) = 0.0192$$



{D1, D2, ..., D14}
[9+,5−]
Outlook
Sunny
Overcast
Rain
Humidity
{D3,D7,D12,D13}
[4+,0−]
{D4,D5,D6,D10,D14}
[3+,2−]
High
Normal
Yes
?
{D1, D2, D8}
No
{D9, D11}
Yes

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

Attribute: Temp

$Values\ (Temp) = Hot, Mild, Cool$

$S_{Rain} = [3+, 2-]$

$Entropy(S_{Sunny}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.97$

$S_{Hot} \leftarrow [0+, 0-]$

$Entropy(S_{Hot}) = 0.0$

$S_{Mild} \leftarrow [2+, 1-]$

$Entropy(S_{Mild}) = -\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3} = 0.9183$

$S_{Cool} \leftarrow [1+, 1-]$

$Entropy(S_{Cool}) = 1.0$

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

Attribute: Temp

$Values\ (Temp) = Hot, Mild, Cool$

$S_{Rain} = [3+, 2-]$ $\quad Entropy(S_{Sunny}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.97$

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

$S_{Mild} \leftarrow [2+, 1-]$ $\quad Entropy(S_{Mild}) = -\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3} = 0.9183$

$S_{Cool} \leftarrow [1+, 1-]$ $\quad Entropy(S_{Cool}) = 1.0$

$$Gain\ (S_{Rain}, Temp) = Entropy(S) - \sum_{v \in \{Hot, Mild, Cool\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Rain}, Temp)$$

$$= Entropy(S) - \frac{0}{5}Entropy(S_{Hot}) - \frac{3}{5}Entropy(S_{Mild})$$

$$- \frac{2}{5}Entropy(S_{Cool})$$

$$Gain(S_{Rain}, Temp) = 0.97 - \frac{0}{5}0.0 - \frac{3}{5}0.918 - \frac{2}{5}1.0 = 0.0192$$

Day	Temp	Humidity	Wind	Play Tennis
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
Dl0	Mild	Normal	Weak	Yes
Dl4	Mild	High	Strong	No

Attribute: Humidity

$Values\ (Humidity) = High, Normal$

$S_{Rain} = [3+, 2-]$ $\quad$ $Entropy(S_{Sunny}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.97$

$S_{High} \leftarrow [1+, 1-]$ $\quad$ $Entropy(S_{High}) = 1.0$

$S_{Normal} \leftarrow [2+, 1-]$ $\quad$ $Entropy(S_{Normal}) = -\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3} = 0.9183$

$$Gain\ (S_{Rain}, Humidity) = Entropy(S) - \sum_{v \in \{High, Normal\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Rain}, Humidity) = Entropy(S) - \frac{2}{5}Entropy(S_{High}) - \frac{3}{5}Entropy(S_{Normal})$$

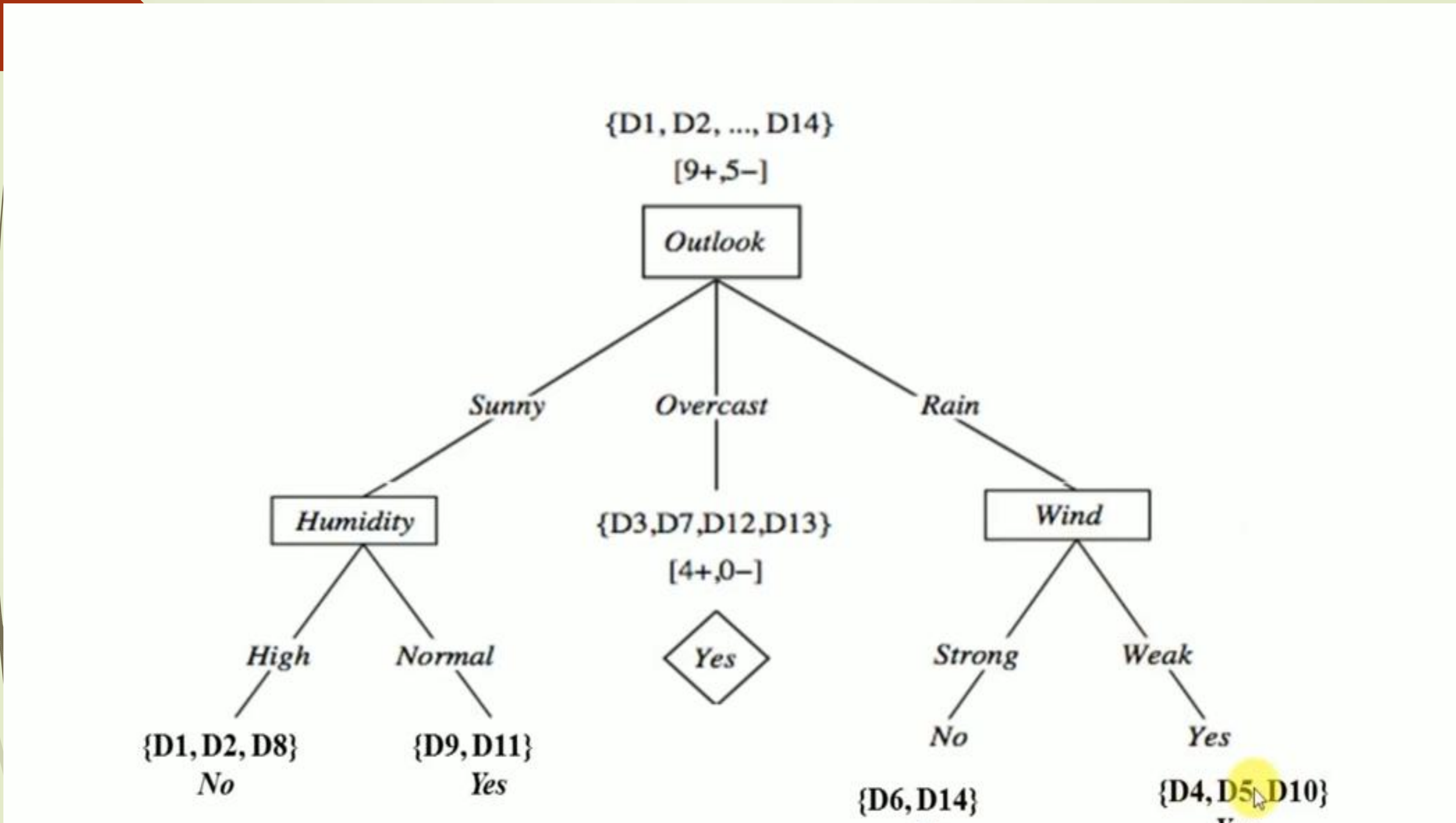
$$Gain(S_{Rain}, Humidity) = 0.97 - \frac{2}{5}\ 1.0 - \frac{3}{5}0.918 = 0.0192$$

Day	Temp	Humidity	Wind	Play Tennis
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
Dl0	Mild	Normal	Weak	Yes
Dl4	Mild	High	Strong	No

$$Gain(S_{Rain}, Temp) = 0.0192$$

$$Gain(S_{Rain}, Humidity) = 0.0192$$

$$Gain(S_{Rain}, Wind) = 0.97$$



{D1, D2, ..., D14}
[9+,5−]
Outlook
Sunny
Overcast
Rain
Humidity
{D3,D7,D12,D13}
[4+,0−]
Wind
High
Normal
Yes
Strong
Weak
{D1, D2, D8}
No
{D9, D11}
Yes
No
Yes
{D6, D14}
{D4, D5, D10}

NAÏVE BAYES

- Let's start with a basic introduction to the Bayes theorem, named after Thomas Bayes from the 1700s. The Naive Bayes classifier works on the principle of conditional probability, as given by the Bayes theorem.
- If we toss two coins and look at all the different possibilities, we have the sample space as:{HH, HT, TH, TT}
  - The probability of getting two heads = 1/4
  - The probability of at least one tail = 3/4
  - The probability of the second coin being head given the first coin is tail = 1/2
  - The probability of getting two heads given the first coin is a head = 1/2
- The Bayes theorem gives us the conditional probability of event A, given that event B has occurred.

- According to Bayes theorem:

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

where:

P(A|B) = Conditional Probability of A given B

P(B|A) = Conditional Probability of A given B

P(A) = Probability of event A

P(B) = Probability of event A

Example 1

Day	Outlook	Temp	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

$$P(PlayTennis = yes) = 9/14 = .64$$

$$P(PlayTennis = no) = 5/14 = .36$$

Outlook	Y	N
sunny	2/9	3/5
overcast	4/9	0
rain	3/9	2/5
Tempreature		
hot	2/9	2/5
mild	4/9	2/5
cool	3/9	1/5

Humidity	Y	N
high	3/9	4/5
normal	6/9	1/5

Windy		
Strong	3/9	3/5
Weak	6/9	2/5

$$\langle Outlook = sunny, Temperature = cool, Humidity = high, Wind = strong \rangle$$

$$v_{NB} = \underset{v_j \in \{yes, no\}}{\operatorname{argmax}} P(v_j) \prod_i P(a_i|v_j)$$

$$= \underset{v_j \in \{yes, no\}}{\operatorname{argmax}} P(v_j) \quad P(Outlook = sunny|v_j) P(Temperature = cool|v_j)$$

$$\cdot P(Humidity = high|v_j) P(Wind = strong|v_j)$$

$$v_{NB}(yes) = P(yes)\ P(sunny|yes)\ P(cool|yes)\ P(high|yes)\ P(strong|yes) = .0053$$

$$v_{NB}(no) = P(no)\ P(sunny|no)\ P(cool|no)\ P(high|no)\ P(strong|no) = .0206$$

$$v_{NB}(yes) = \frac{v_{NB}(yes)}{v_{NB}(yes) + v_{NB}(no)} = 0.205 \qquad v_{NB}(no) = \frac{v_{NB}(no)}{v_{NB}(yes) + v_{NB}(no)} = 0.795$$

Assignment 1, 2

1. Problems in Decision tree
2. Advantages and disadvantages of naïve bayes classifier

MULTI CLASS CLASSIFICATION

For multi-class classification, many binary classification techniques are applicable.

The following well-known algorithms can be used for multi-class classification:

- Progressive Boosting
- Choice trees
- Nearest K Neighbors
- Rough Forest / Random Forest
- Simple Bayes

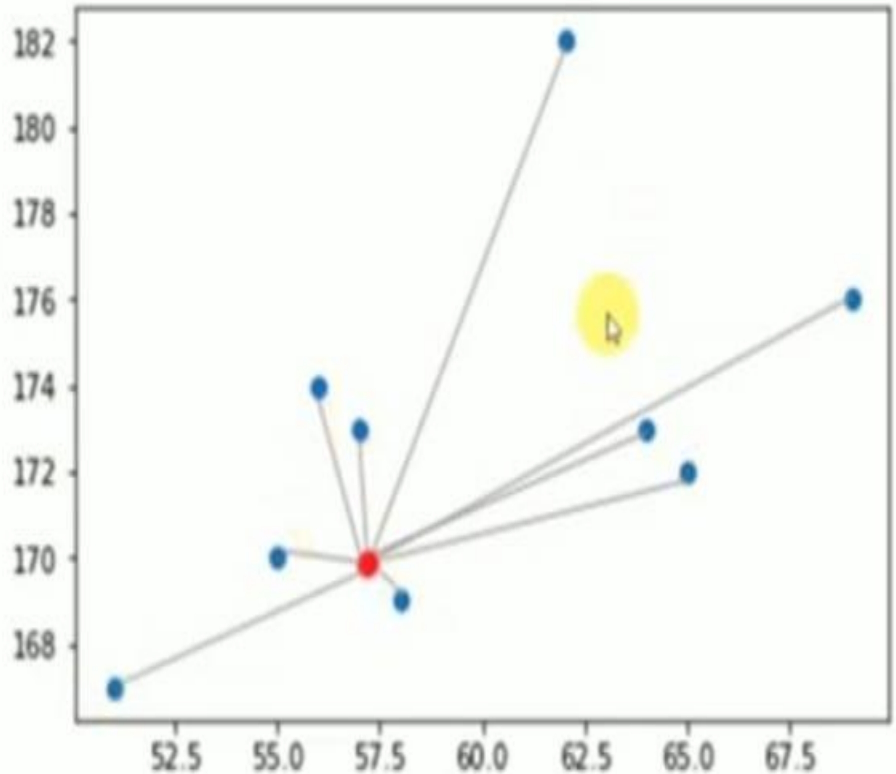
Multi-class problems can be solved using algorithms created for binary classification.

KNN

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on 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.

Height (CM)	Weight (KG)	Class
167	51	Underweight
182	62	Normal
176	69	Normal
173	64	Normal
172	65	Normal
174	56	Underweight
169	58	Normal
173	57	Normal
170	55	Normal
170	57	?

Height (CM)	Weight (KG)	Class
167	51	Underweight
182	62	Normal
176	69	Normal
173	64	Normal
172	65	Normal
174	56	Underweight
169	58	Normal
173	57	Normal
170	55	Normal
170	57	?



Height (CM)	Weight (KG)	Class
167	51	Underweight
182	62	Normal
176	69	Normal
173	64	Normal
172	65	Normal
174	56	Underweight
169	58	Normal
173	57	Normal
170	55	Normal
170	57	?

THE DISTANCE FORMULA

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Height (CM)	Weight (KG)	Class	Distance	Rank
169	58	Normal	1.4	1
170	55	Normal	2	2
173	57	Normal	3	3
174	56	Underweight	4.1	4
167	51	Underweight	6.7	5
173	64	Normal	7.6	6
172	65	Normal	8.2	7
182	62	Normal	13	8
176	69	Normal	13.4	9
170	57	?		

Height (CM)	Weight (KG)	Class	Distance	Rank
169	58	Normal ✓	1.4	1 ✓
170	55	Normal ✓	2	2 ✓
173	57	Normal ✓	3	3 ✓
174	56	Underweight ✓	4.1	4 ✓
167	51	Underweight ✓	6.7	5 ✓
173	64	Normal	7.6	6
172	65	Normal	8.2	7
182	62	Normal	13	8
176	69	Normal	13.4	9
170	57	?		

- If K=1, Normal
- If K=2, Normal
- If K=3, Normal
- If K=4, Normal
- If K=5, Normal

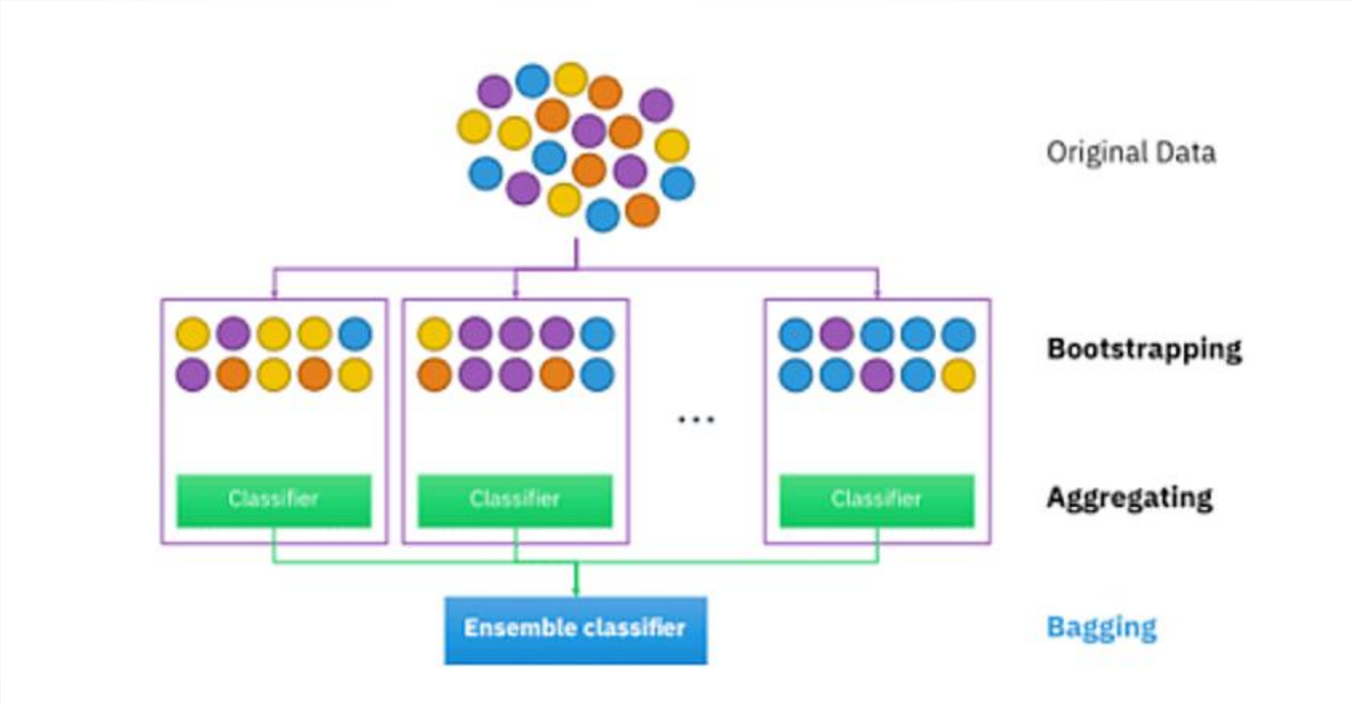
Random Forest Algorithm

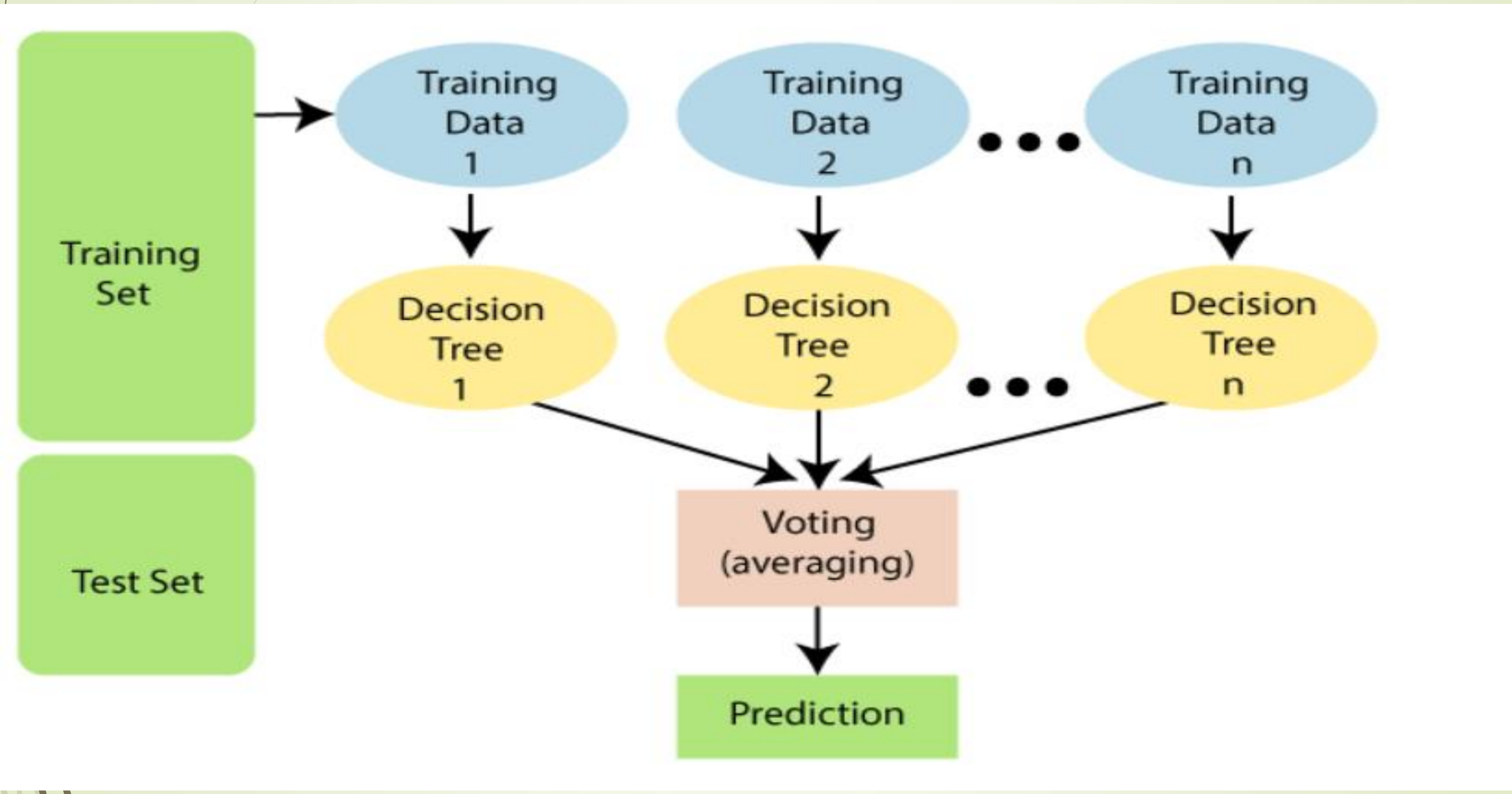
- ***"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.“***
- **The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.**

Bagging

- From the principle mentioned above, we can understand Random forest uses the Bagging code. Now, let us understand this concept in detail. Bagging is also known as Bootstrap Aggregation used by random forest. The process begins with any original random data. After arranging, it is organised into samples known as Bootstrap Sample. This process is known as Bootstrapping.Further, the models are trained individually, yielding different results known as Aggregation. In the last step, all the results are combined, and the generated output is based on majority voting. This step is known as Bagging and is done using an Ensemble Classifier.

Bagging


Original Data
Bootstrapping
...
Classifier
Classifier
Classifier
Aggregating
Ensemble classifier
Bagging


Training Set
Test Set
Training Data 1
Training Data 2
Training Data n
Decision Tree 1
Decision Tree 2
Decision Tree n
Voting (averaging)
Prediction

Assignment 3, 4

- Features of Random forest
- Difference between random forest and decision trees

UNSUPERVISED LEARNING

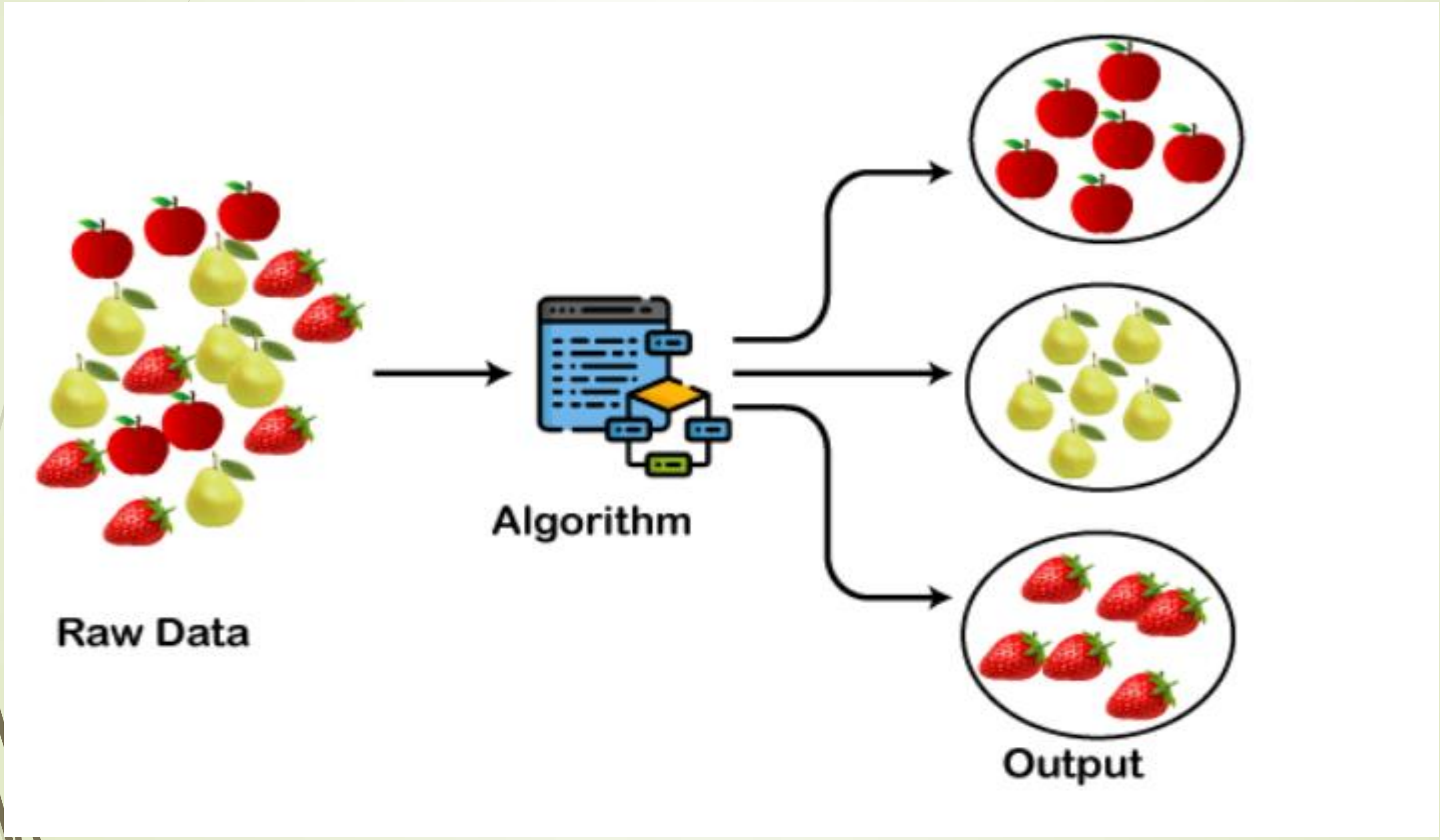
No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself

TYPES

- Clustering
- Association

clustering

- **Clustering** is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them.


Raw Data
Algorithm
Output

The clustering technique can be widely used in various tasks. Some most common uses of this technique are:

- Market Segmentation
- Statistical data analysis
- Social network analysis
- Image segmentation
- Anomaly detection, etc.

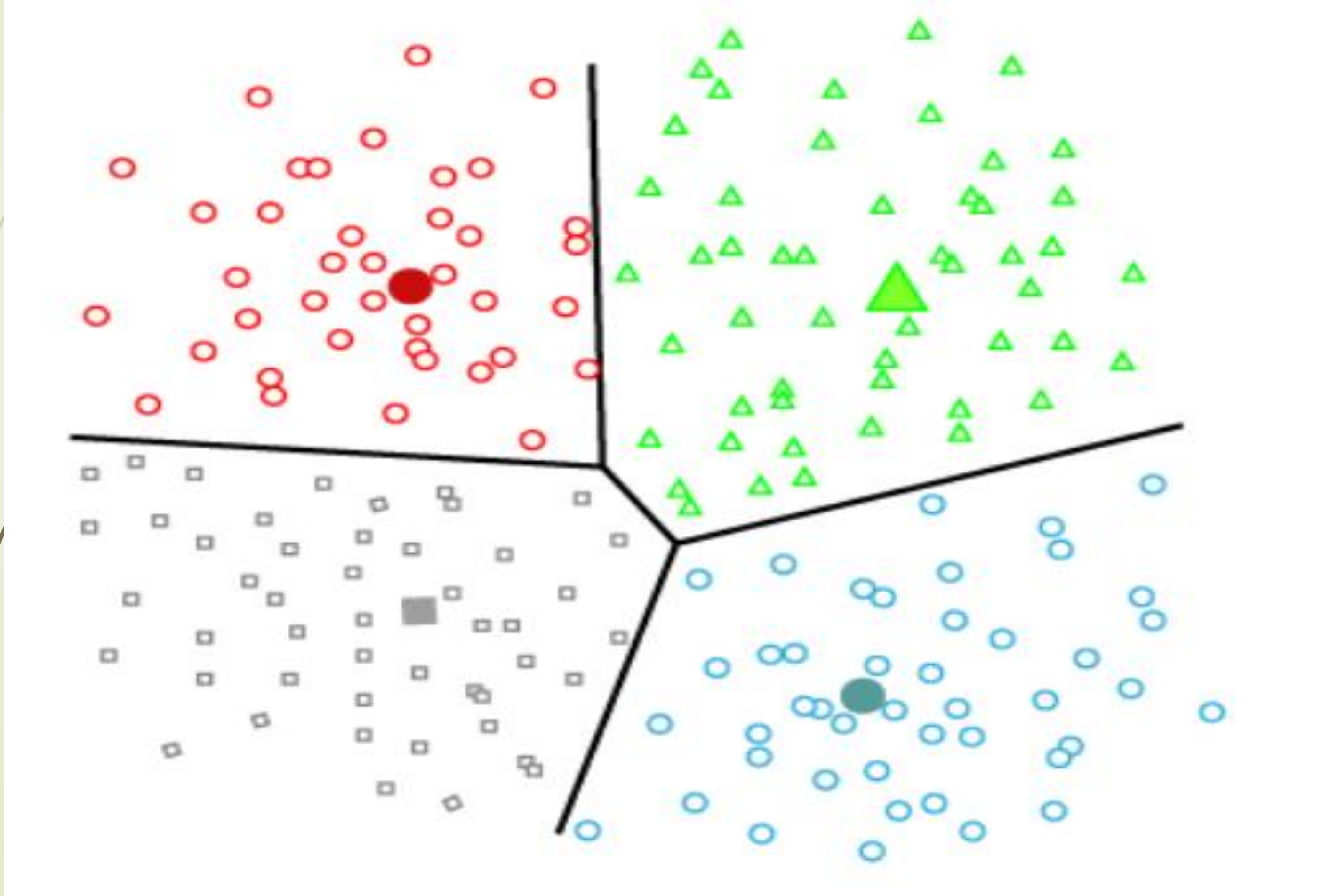
Apart from these general usages, it is used by the **Amazon** in its recommendation system to provide the recommendations as per the past search of products. **Netflix** also uses this technique to recommend the movies and web-series to its users as per the watch history.

Types of Clustering Methods

The clustering methods are broadly divided into **Hard clustering** (datapoint belongs to only one group) and **Soft Clustering** (data points can belong to another group also).

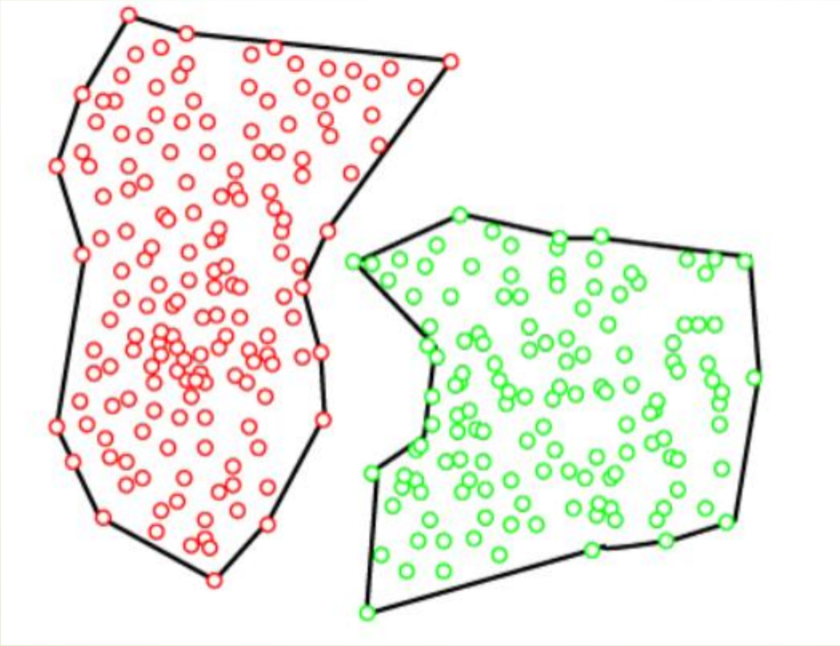
1. **Partitioning Clustering**
2. **Density-Based Clustering**
3. **Distribution Model-Based Clustering**
4. **Hierarchical Clustering**
5. **Fuzzy Clustering**

- Partitioning Clustering
- It is a type of clustering that divides the data into non-hierarchical groups. It is also known as the **centroid-based method**. The most common example of partitioning clustering is the **K-Means Clustering algorithm**.
- In this type, the dataset is divided into a set of k groups, where K is used to define the number of pre-defined groups. The cluster center is created in such a way that the distance between the data points of one cluster is minimum as compared to another cluster centroid.



Density-Based Clustering

The density-based clustering method connects the highly-dense areas into clusters, and the arbitrarily shaped distributions are formed as long as the dense region can be connected.

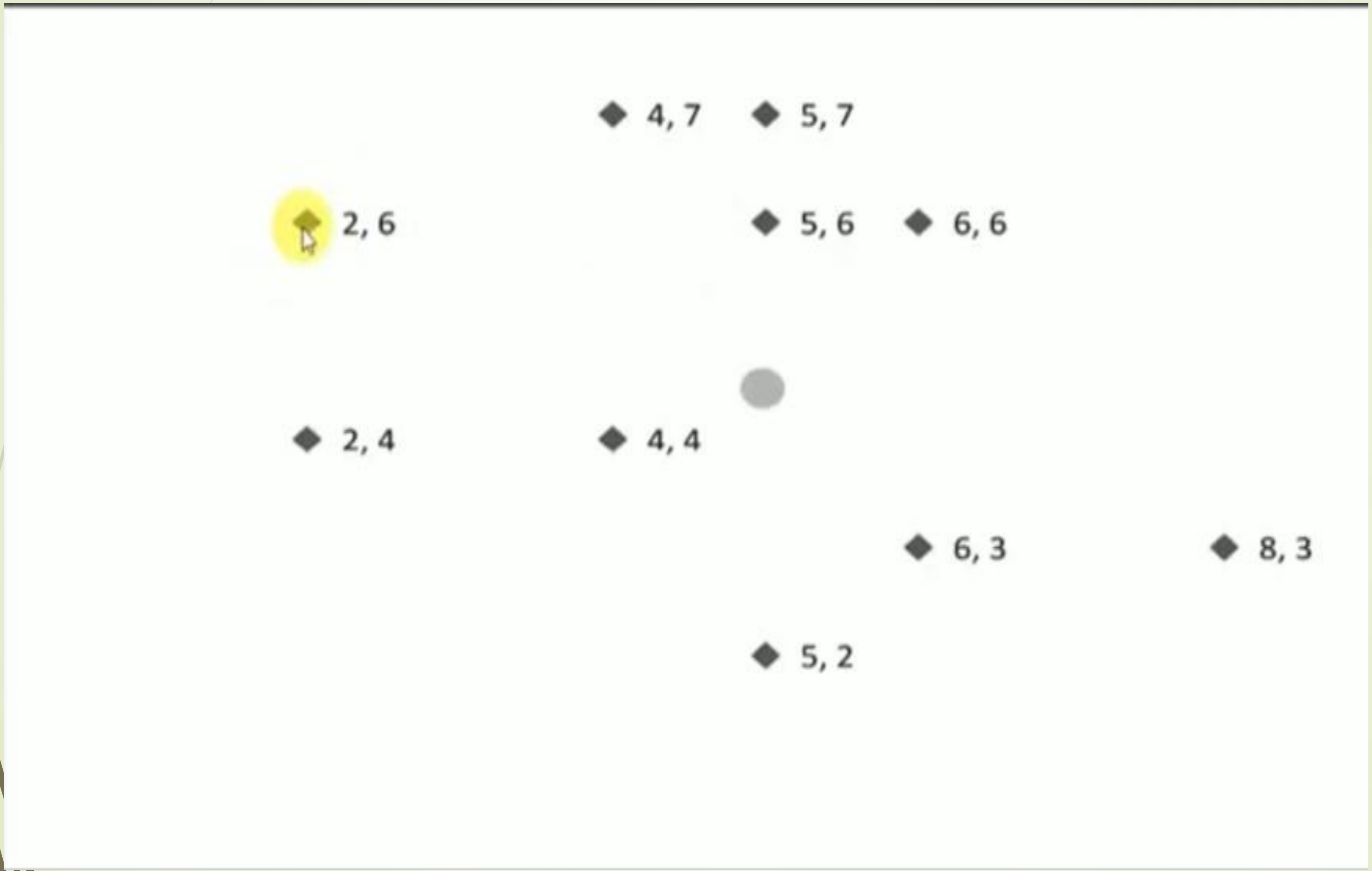


Algorithm 1 k-means algorithm

```
Specify the number k of clusters to assign.
Randomly initialize k centroids.
repeat
    expectation: Assign each point to its closest centroid.
    maximization: Compute the new centroid (mean) of each cluster.
until The centroid positions do not change.
```

Example of k means

X	Y
2	4
2	6
5	6
4	7
8	3
6	6
5	2
5	7
6	3
4	4



4, 7
5, 7
2, 6
5, 6
6, 6
2, 4
4, 4
6, 3
8, 3
5, 2

Iteration - 1

C1 - Seed Point1 – (1, 5)
C2 - Seed Point2 – (4, 1)
C3 - Seed Point3 – (8, 4)

$$D = \sqrt{((x_2 - x_1)^2 + (y_2 - y_1)^2)}$$

C1 – Centroid – (2.66, 5.66)
C2 – Centroid – (4.5, 3)
C3 – Centroid – (6, 5)

		Distance to			Cluster
X	Y	(1, 5)	(4, 1)	(8, 4)	Number
2	4	1.41	3.61	6.00	C1
2	6	1.41	5.39	6.32	C1
5	6	4.12	5.10	3.61	C3
4	7	3.61	6.00	5.00	C1
8	3	7.28	4.47	1.00	C3
6	6	5.10	5.39	2.83	C3
5	2	5.00	1.41	3.61	C2
5	7	4.47	6.08	4.24	C3
6	3	5.39	2.83	2.24	C3
4	4	3.16	3.00	4.00	C2

Iteration - 2

C1 – Centroid – (2.66, 5.66)
C2 – Centroid – (4.5, 3)
C3 – Centroid – (6, 5)

C1 – Centroid – (2.66, 5.66)
C2 – Centroid – (5, 3)
C3 – Centroid – (6, 5.5)

		Distance to			Cluster
X	Y	(2.66, 5.66)	(4.5, 3)	(6, 5)	Number
2	4	1.79	2.69	4.12	C1
2	6	0.74	3.91	4.12	C1
5	6	2.36	3.04	1.41	C3
4	7	1.90	4.03	2.83	C1
8	3	5.97	3.5	2.83	C3
6	6	3.36	3.35	1	C3
5	2	4.34	1.12	3.16	C2
5	7	2.70	4.03	2.24	C3
6	3	4.27	1.5	2	C2
4	4	2.13	1.12	2.24	C2

Iteration - 3

C1 – Centroid – (2.66, 5.66)
C2 – Centroid – (5, 3)
C3 – Centroid – (6, 5.5)

C1 – Centroid – (2.66, 5.66)
C2 – Centroid – (5.75, 3)
C3 – Centroid – (5.33, 6.33)

		Distance to			Cluster
X	Y	(2.66, 5.66)	(5, 3)	(6, 5.5)	Number
2	4	1.79	3.16	4.27	C1
2	6	0.74	4.24	4.03	C1
5	6	2.36	3.00	1.12	C3
4	7	1.90	4.12	2.50	C1
8	3	5.97	3.00	3.20	C2
6	6	3.36	3.16	0.50	C3
5	2	4.34	1.00	3.64	C2
5	7	2.70	4.00	1.80	C3
6	3	4.27	1.00	2.50	C2
4	4	2.13	1.41	2.50	C2

Iteration - 5

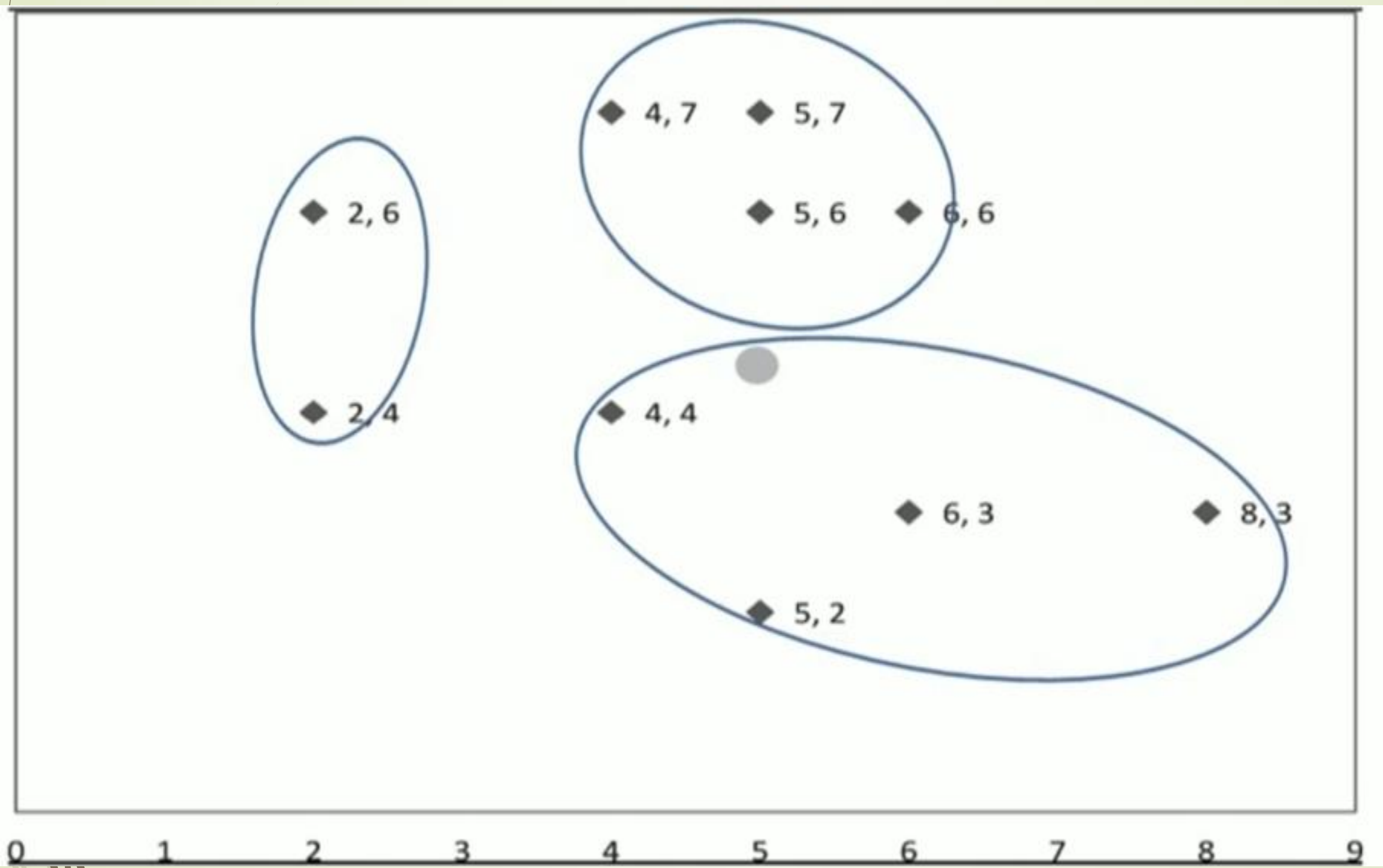
C1 – Centroid – (2, 5)

C2 – Centroid – (5.75, 3)

C3 – Centroid – (5, 6.5)

No movement of data Points Hence these are the final positions

		Distance to			Cluster
X	Y	(2, 5)	(5.75, 3)	(5, 6.5)	Number
2	4	1.00	3.88	3.91	C1
2	6	1.00	4.80	3.04	C1
5	6	3.16	3.09	0.50	C3
4	7	2.83	4.37	1.12	C3
8	3	6.32	2.25	4.61	C2
6	6	4.12	3.01	1.12	C3
5	2	4.24	1.25	4.50	C2
5	7	3.61	4.07	0.50	C3
6	3	4.47	0.25	3.64	C2
4	4	2.24	2.02	2.69	C2



4, 7
5, 7
2, 6
5, 6
6, 6
2, 4
4, 4
6, 3
8, 3
5, 2
0
1
2
3
4
5
6
7
8
9

Pros n cons

- Simple
- Flexible
- Suitable in large datasets
- Efficient
- Accuracy

Cons

- Handle numerical data only
- Specifying clusters earlier
- Prediction issue

Assignment 5 n 6

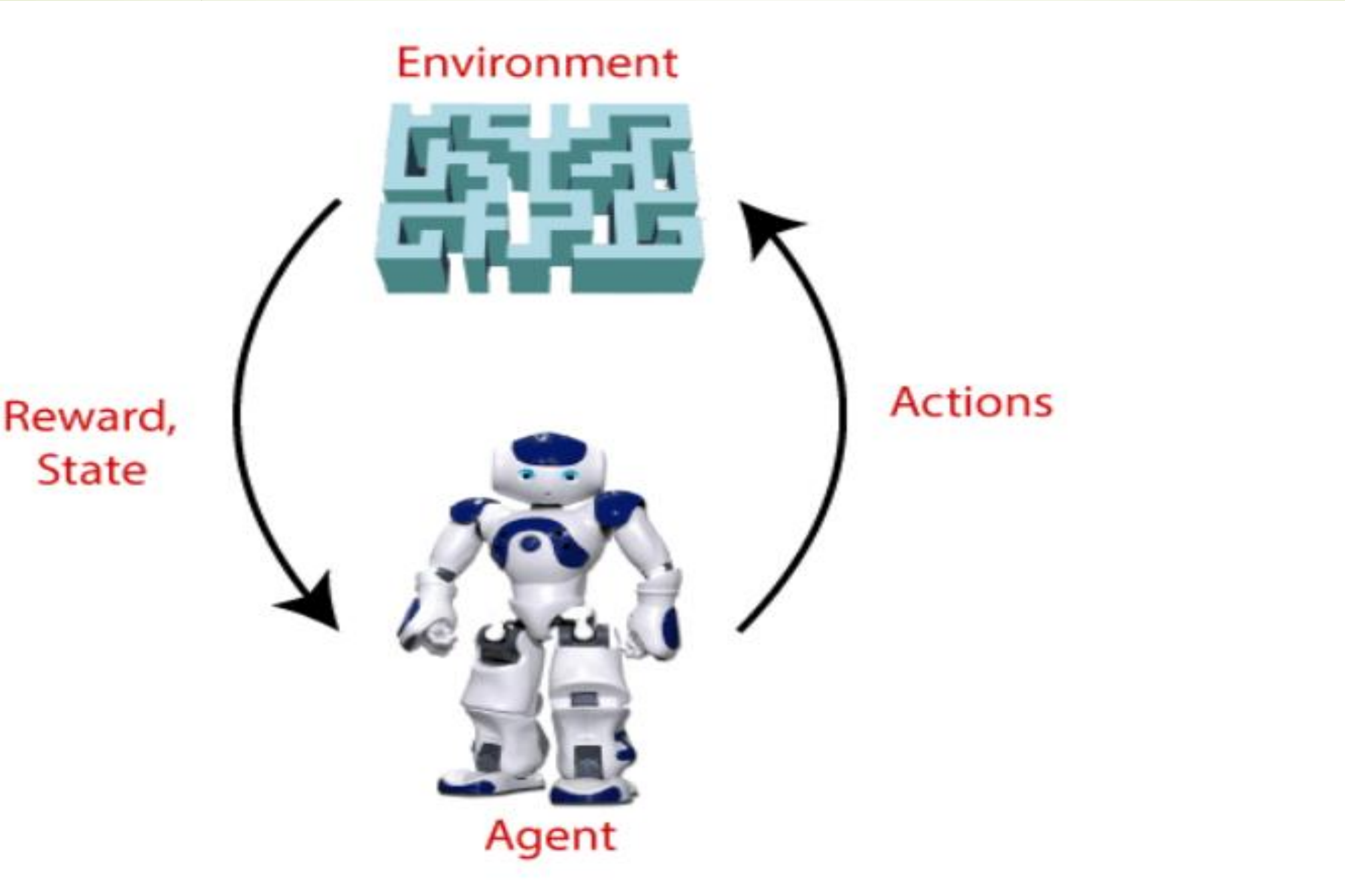
- Difference between supervised and unsupervised
- Real world applications of k means

Reinforcement learning

Reinforcement Learning is a feedback-based Machine learning technique in which an agent learns to behave in an environment by performing the actions and seeing the results of actions. For each good action, the agent gets positive feedback, and for each bad action, the agent gets negative feedback or penalty.

- In Reinforcement Learning, the agent learns automatically using feedbacks without any labeled data
- RL solves a specific type of problem where decision making is sequential, and the goal is long-term, such as **game-playing, robotics**, etc
- ***"Reinforcement learning is a type of machine learning method where an intelligent agent (computer program) interacts with the environment and learns to act within that."***.

Example of RL



- **Example:** Suppose there is an AI agent present within a maze environment, and his goal is to find the diamond. The agent interacts with the environment by performing some actions, and based on those actions, the state of the agent gets changed, and it also receives a reward or penalty as feedback.
- The agent continues doing these three things (**take action, change state/remain in the same state, and get feedback**), and by doing these actions, he learns and explores the environment.
- The agent learns that what actions lead to positive feedback or rewards and what actions lead to negative feedback penalty. As a positive reward, the agent gets a positive point, and as a penalty, it gets a negative point.

Key Features of Reinforcement Learning

- In RL, the agent is not instructed about the environment and what actions need to be taken.
- It is based on the hit and trial process.
- The agent takes the next action and changes states according to the feedback of the previous action.
- The agent may get a delayed reward.

Terms used in Reinforcement Learning

- **Agent():** An entity that can perceive/explore the environment and act upon it.
- **Environment():** A situation in which an agent is present or surrounded by. In RL, we assume the stochastic environment, which means it is random in nature.
- **Action():** Actions are the moves taken by an agent within the environment.
- **State():** State is a situation returned by the environment after each action taken by the agent.
- **Reward():** A feedback returned to the agent from the environment to evaluate the action of the agent.
- **Policy():** Policy is a strategy applied by the agent for the next action based on the current state.
- **Value():** represent how good is a state for an agent to be in. .
- Q-value(): It is mostly similar to the value, but it takes one additional parameter as a current action (a).

Supervised Learning	Unsupervised Learning	Reinforcement Learning
Input data is labelled.	Input data is not labelled.	Input data is not predefined.
Learn pattern of inputs and their labels.	Divide data into classes.	Find the best reward between a start and an end state.
Finds a mapping equation on input data and its labels.	Finds similar features in input data to classify it into classes.	Maximizes reward by assessing the results of state-action pairs
Model is built and trained prior to testing.	Model is built and trained prior to testing.	The model is trained and tested simultaneously.
Deal with regression and classification problems.	Deals with clustering and associative rule mining problems.	Deals with exploration and exploitation problems.
Decision trees, linear regression, K-nearest neighbors	K-means clustering, k-medoids clustering, agglomerative clustering	Q-learning, SARSA, Deep Q Network
Image detection, Population growth prediction	Customer segmentation, feature elicitation, targeted marketing, etc	Drive-less cars, self-navigating vacuum cleaners, etc