Assignment

Q1**. What are the different types of learnings?**

Ans 1.) **Learning Problems**

- 1. Supervised Learning
- 2. Unsupervised Learning
- 3. Reinforcement Learning

Hybrid Learning Problems

- 4. Semi-Supervised Learning
- 5. Self-Supervised Learning
- 6. Multi-Instance Learning

Statistical Inference

- 7. Inductive Learning
- 8. Deductive Inference
- 9. Transductive Learning

Learning Techniques

- 10. Multi-Task Learning
- 11. Active Learning
- 12. Online Learning
- 13. Transfer Learning
- 14. Ensemble Learning

1. Supervised Learning

Supervised learning describes a class of problem that involves using a model to learn a mapping between input examples and the target variable.

- **Classification**: Supervised learning problem that involves predicting a class label.
- **Regression**: Supervised learning problem that involves predicting a numerical label.

2. Unsupervised Learning

Unsupervised learning describes a class of problems that involves using a model to describe or extract relationships in data.

- **Clustering: Unsupervised** learning problem that involves finding groups in data.
- **Density Estimation**: Unsupervised learning problem that involves summarizing the distribution of data.

3. Reinforcement Learning

Reinforcement learning describes a class of problems where an agent operates in an environment and must *learn* to operate using feedback.

Hybrid Learning Problems

The lines between unsupervised and supervised learning is blurry, and there are many hybrid approaches that draw from each field of study.

4. Semi-Supervised Learning

Semi-supervised learning is supervised learning where the training data contains very few labeled examples and a large number of unlabeled examples.

5. Self-Supervised Learning

Self-supervised learning refers to an unsupervised learning problem that is framed as a supervised learning problem in order to apply supervised learning algorithms to solve it.

6. Multi-Instance Learning

Multi-instance learning is a supervised learning problem where individual examples are unlabeled; instead, bags or groups of samples are labeled.

7. Inductive Learning

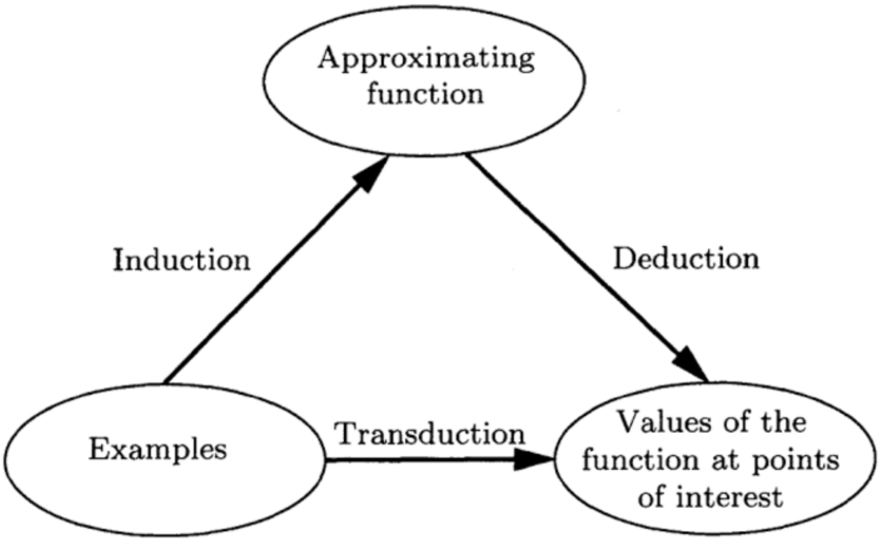
Inductive learning involves using evidence to determine the outcome.

8. Deductive Inference

Deduction or deductive inference refers to using general rules to determine specific outcomes.

9. Transductive Learning

Transduction or transductive learning is used in the field of statistical learning theory to refer to predicting specific examples given specific examples from a domain.



10. Multi-Task Learning

Multi-task learning is a type of supervised learning that involves fitting a model on one dataset that addresses multiple related problems.

11. Active Learning

Active learning is a technique where the model is able to query a human user operator during the learning process in order to resolve ambiguity during the learning process.

12. Online Learning

Online learning involves using the data available and updating the model directly before a prediction is required or after the last observation was made.

13. Transfer Learning

Transfer learning is a type of learning where a model is first trained on one task, then some or all of the model is used as the starting point for a related task.

14. Ensemble Learning

Ensemble learning is an approach where two or more modes are fit on the same data and the predictions from each model are combined.

Q2. What is the difference between supervised and unsupervised machine learning?

Ans .

Supervised learning: Supervised learning is the learning of the model where with input variable (say, x) and an output variable (say, Y) and an algorithm to map the input to the output.

That is, **Y = f(X)**

Why supervised learning?

The basic aim is to approximate the mapping function(mentioned above) so well that when there is a new input data (x) then the corresponding output variable can be predicted.

It is called supervised learning because the process of an learning(from the training dataset) can be thought of as a teacher who is supervising the entire learning process. Thus, the “learning algorithm” iteratively makes predictions on the training data and is corrected by the “teacher”, and the learning stops when the algorithm achieves an acceptable level of performance(or the desired accuracy).

Example of Supervised Learning

Suppose there is a basket which is filled with some fresh fruits, the task is to arrange the same type of fruits at one place.

Also, suppose that the fruits are apple, banana, cherry, grape.

Suppose one already knows from their previous work (or experience) that, the shape of each and every fruit present in the basket so, it is easy for them to arrange the same type of fruits in one place.

Here, the previous work is called as training data in Data Mining terminology. So, it learns the things from the training data. This is because it has a response variable which says y that if some fruit has so and so features then it is grape, and similarly for each and every fruit.

This type of information is deciphered from the data that is used to train the model.

This type of learning is called Supervised Learning.

Such problems are listed under classical Classification Tasks.

Unsupervised Learning: Unsupervised learning is where only the input data (say, X) is present and no corresponding output variable is there.

Why Unsupervised Learning?

The main aim of Unsupervised learning is to model the distribution in the data in order to learn more about the data.

It is called so, because there is no correct answer and there is no such teacher(unlike supervised learning). Algorithms are left to their own devises to discover and present the interesting structure in the data.

Example of Unsupervised Learning

Again, Suppose there is a basket and it is filled with some fresh fruits. The task is to arrange the same type of fruits at one place.

This time there is no information about those fruits beforehand, its the first time that the fruits are being seen or discovered

So how to group similar fruits without any prior knowledge about those.

First, any physical characteristic of a particular fruit is selected. Suppose color.

Then the fruits are arranged on the basis of the color. The groups will be something as shown below:

RED COLOR GROUP: apples & cherry fruits.

GREEN COLOR GROUP: bananas & grapes.

So now, take another physical character say, size, so now the groups will be something like this.

RED COLOR AND BIG SIZE: apple.

RED COLOR AND SMALL SIZE: cherry fruits.

GREEN COLOR AND BIG SIZE: bananas.

GREEN COLOR AND SMALL SIZE: grapes.

The job is done!

Here, there is no need to know or learn anything beforehand. That means, no train data and no response variable. This type of learning is known as Unsupervised Learning.

Difference b/w Supervised and Unsupervised Learning :

	SUPERVISED LEARNING	UNSUPERVISED LEARNING
Input Data	Uses Known and Labeled Data as input	Uses Unknown Data as input
Computational Complexity	Very Complex	Less Computational Complexity
Real Time	Uses off-line analysis	Uses Real Time Analysis of Data
Number of Classes	Number of Classes are known	Number of Classes are not known
Accuracy of Results	Accurate and Reliable Results	Moderate Accurate and Reliable Results

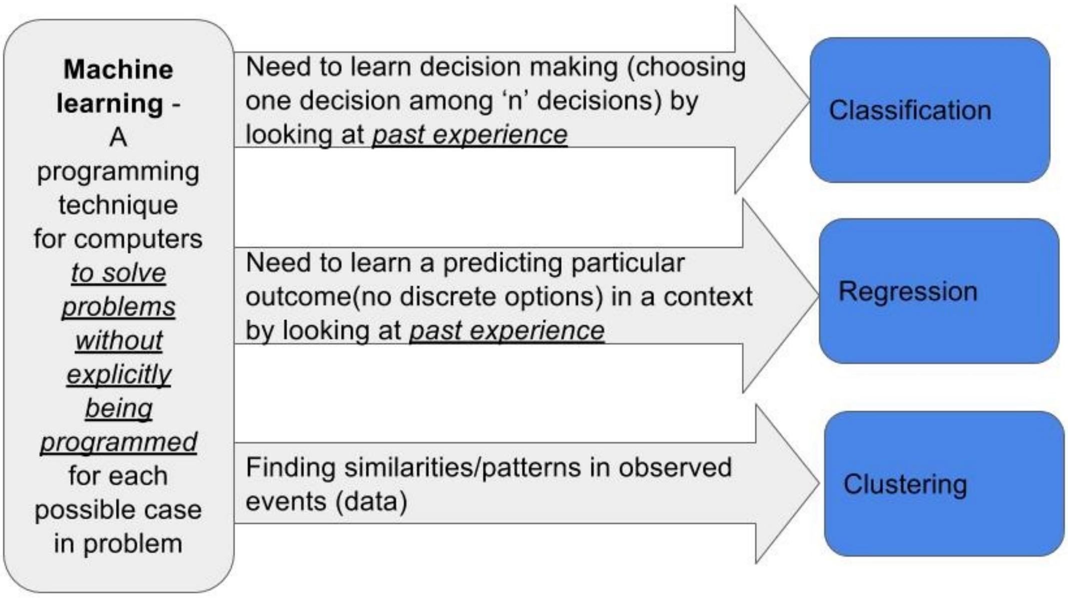
Q3. Define classification, regression and clustering?

Ans:-

Regression: It predicts continuous valued output.The Regression analysis is the statistical model which is used to predict the numeric data instead of labels. It can also identify the distribution trends based on the available data or historic data. Predicting a person’s income from their age, education is example of regression task.

Classification: It predicts discrete number of values. In classification the data is categorized under different labels according to some parameters and then the labels are predicted for the data. Classifying emails as either spam or not spam is example of classification problem.

Clustering: Clustering is the task of partitioning the dataset into groups, called clusters.The goal is to split up the data in such a way that points within single cluster are very similar and points in different clusters are different. It determines grouping among unlabeled data.



Machine learning - A programming technique for computers ***to solve problems without explicitly being programmed*** for each possible case in problem

Need to learn decision making (choosing one decision among ‘n’ decisions) by looking at ***past experience***

Classification

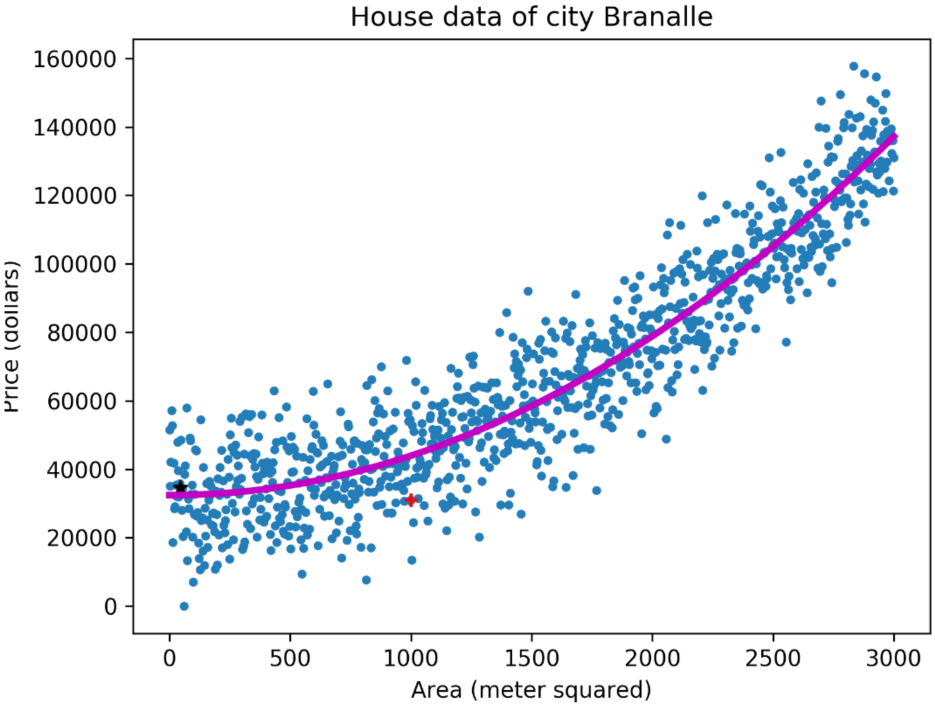
Need to learn a predicting particular outcome(no discrete options) in a context by looking at ***past experience***

Regression

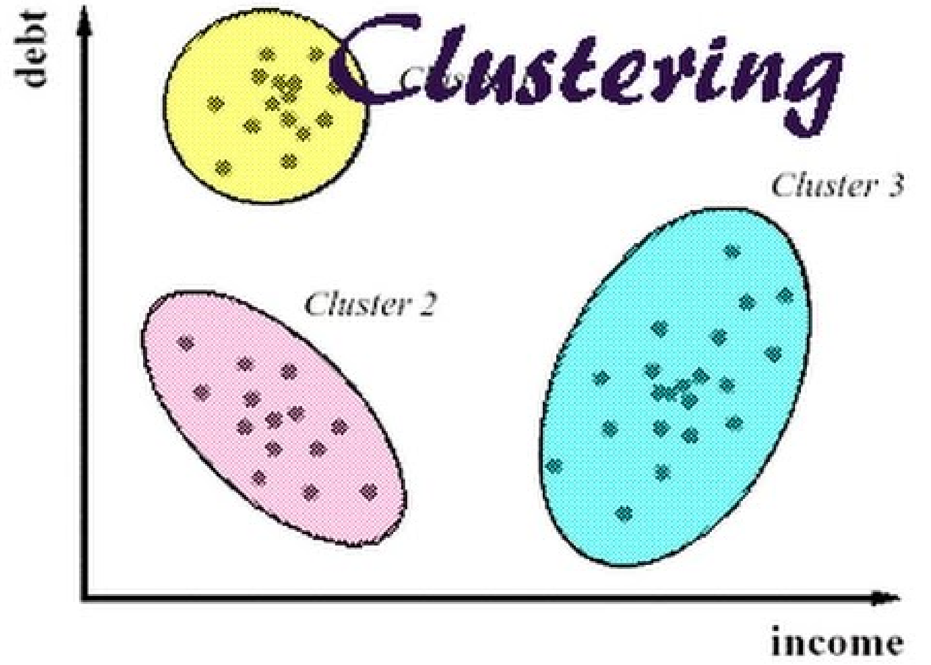
Finding similarities/patterns in observed events (data)

Clustering

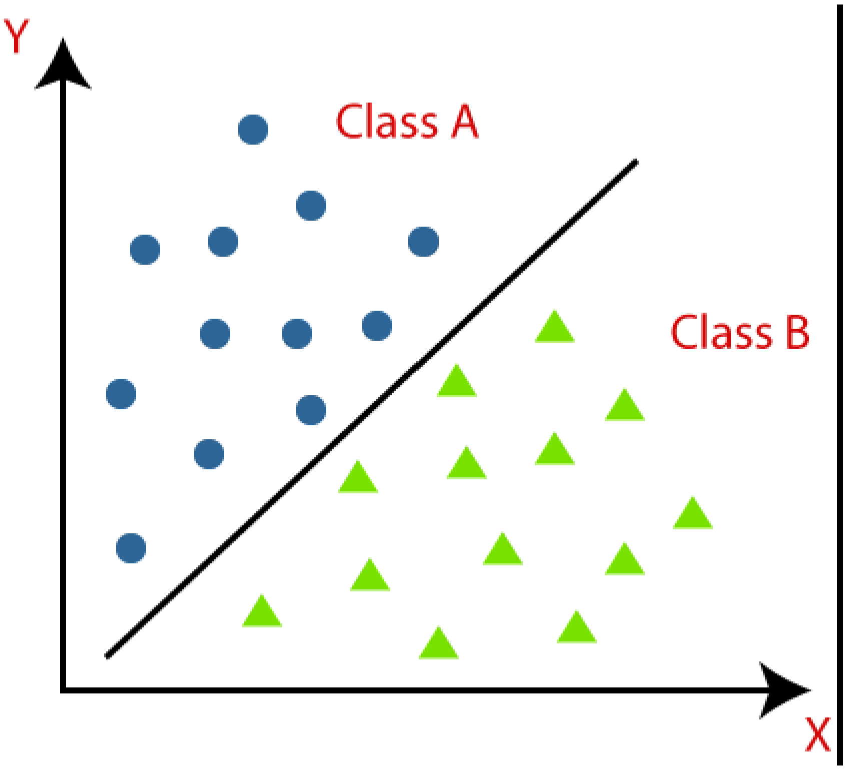
Regression:-



Clustering:-



Classification:-



Q4. What are the advantages and disadvantages of the decision tree?
Ans:-

Decision Tree is a very popular machine learning algorithm. Decision Tree solves the problem of machine learning by transforming the data into tree representation. Each internal node of the tree representation denotes an attribute and each leaf node denotes a class label.

Decision tree algorithm can be used to solve both regression and classification problems.

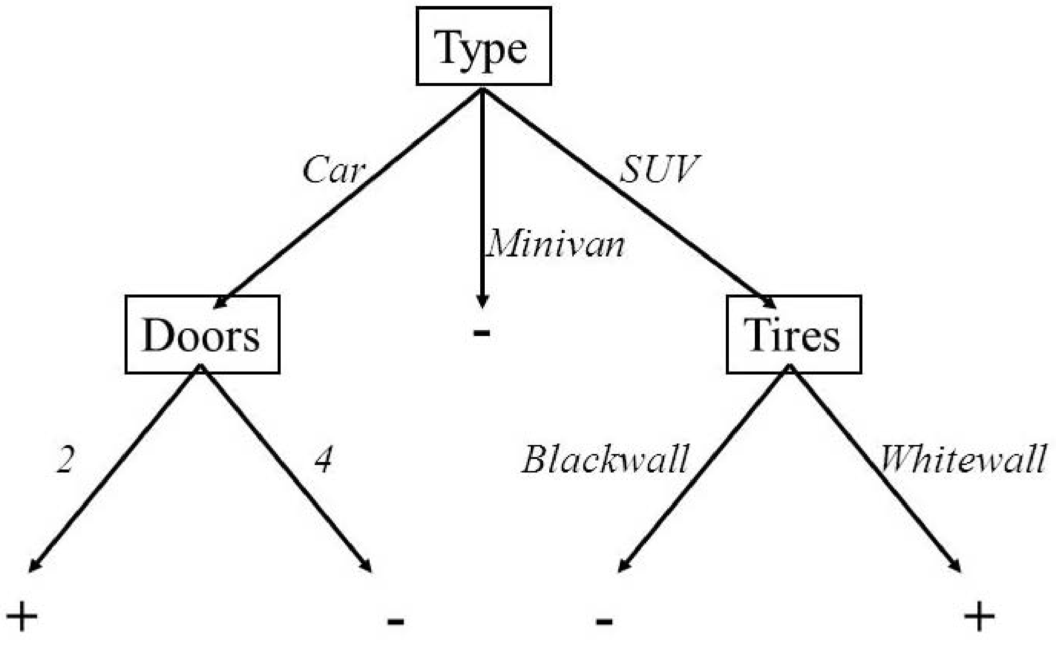
Advantages:-

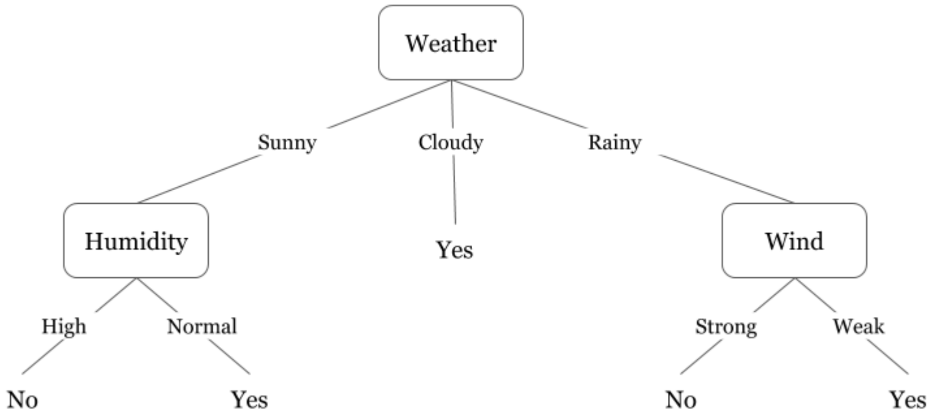
1. Compared to other algorithms decision trees requires less effort for data preparation during pre-processing.
2. A decision tree does not require normalization of data.
3. A decision tree does not require scaling of data as well.
4. Missing values in the data also does NOT affect the process of building decision tree to any considerable extent.
5. A Decision trees model is very intuitive and easy to explain to technical teams as well as stakeholders.

Disadvantages:-

1. A small change in the data can cause a large change in the structure of the decision tree causing instability.
2. For a Decision tree sometimes calculation can go far more complex compared to other algorithms.
3. Decision tree often involves higher time to train the model.
4. Decision tree training is relatively expensive as complexity and time taken is more.
5. Decision Tree algorithm is inadequate for applying regression and predicting continuous values.

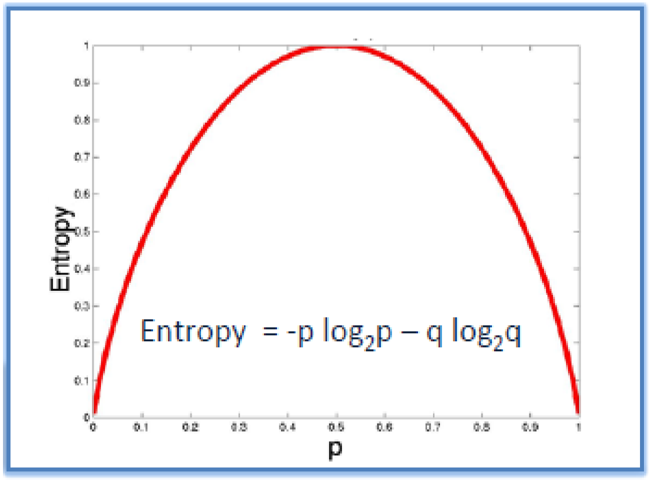
A Decision Tree





5**.What is entropy? What is the formula for entropy?**

ANSWER 5:
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). ID3 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 then it has entropy of one.

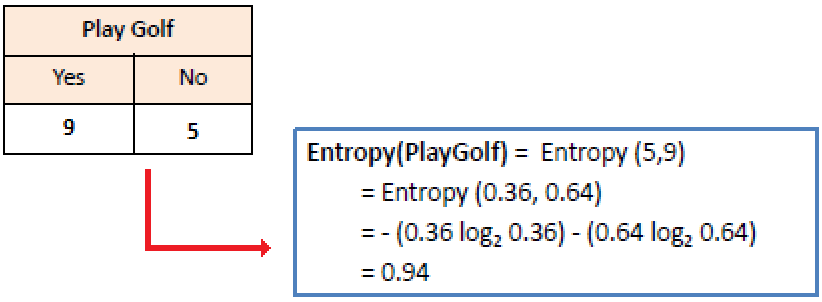


$$\text{Entropy} = -0.5\log_2 0.5 - 0.5\log_2 0.5 = 1$$

To build a decision tree, we need to calculate two types of entropy using frequency tables as follows:

a) Entropy using the frequency table of one attribute:

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

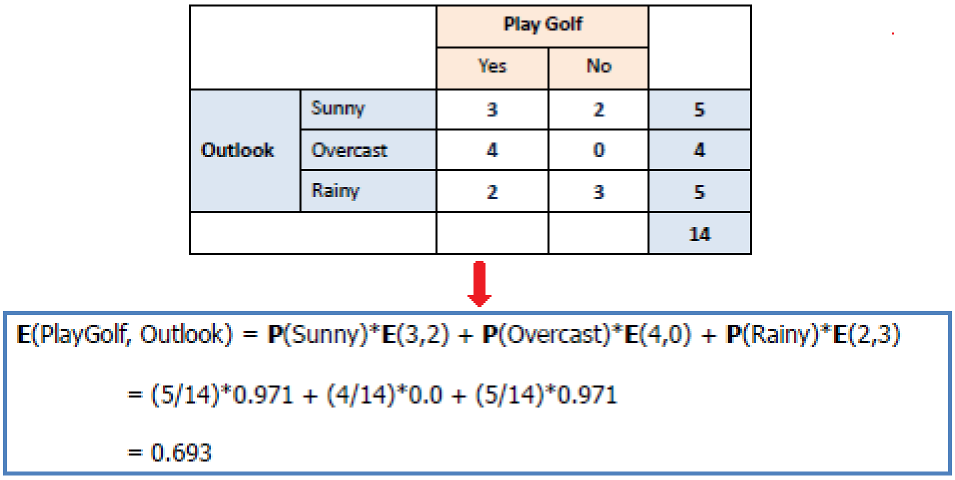


Play Golf	
Yes	No
9	5

Entropy(PlayGolf) = Entropy (5,9)
= Entropy (0.36, 0.64)
= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)
= 0.94

b) Entropy using the frequency table of two attributes:

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



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

E(PlayGolf, Outlook) = **P**(Sunny)***E**(3,2) + **P**(Overcast)***E**(4,0) + **P**(Rainy)***E**(2,3)

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

= 0.693

6.What is high entropy, low entropy and information gain?
ANSWER6:

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

Step 1: Calculate entropy of the target.

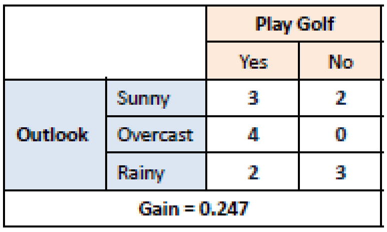
$$\textbf{Entropy(PlayGolf)} = \text{Entropy}(5,9)$$

$$= \text{Entropy}(0.36, 0.64)$$

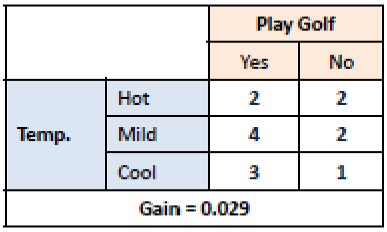
$$= -(0.36 \log_2 0.36) - (0.64 \log_2 0.64)$$

$$= 0.94$$

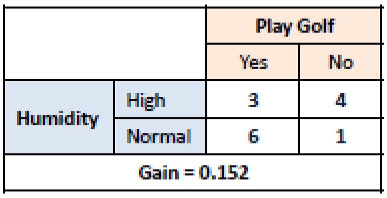
Step 2: The dataset is then split on the different attributes. The entropy for each branch is calculated. Then it is added proportionally, to get total entropy for the split. The resulting entropy is subtracted from the entropy before the split. The result is the Information Gain, or decrease in entropy.



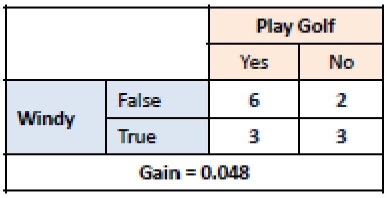
		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			



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



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

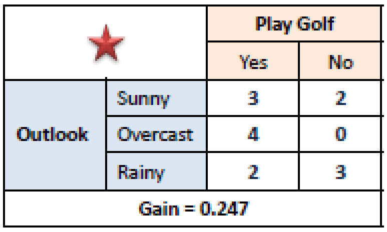


		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Gain = 0.048			

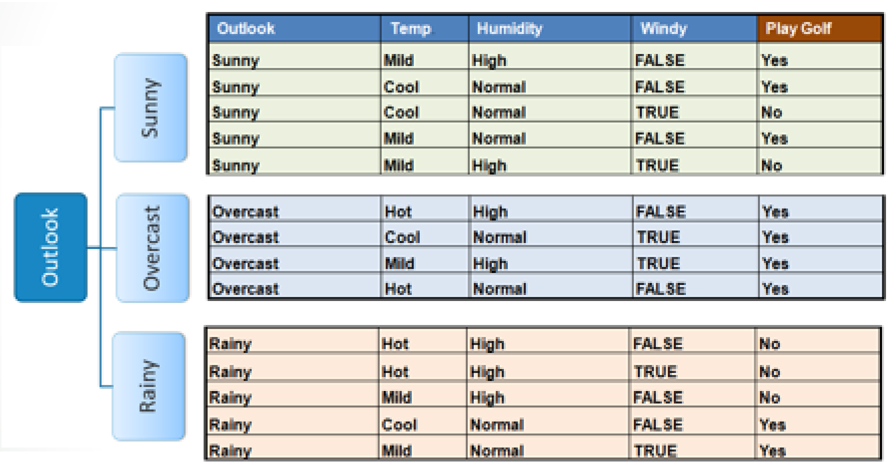
$$Gain(T,X) = Entropy(T) - Entropy(T,X)$$

$$\boxed{\begin{aligned} &\mathbf{G}(\text{PlayGolf, Outlook}) = \mathbf{E}(\text{PlayGolf}) - \mathbf{E}(\text{PlayGolf, Outlook}) \\ &\qquad = 0.940 - 0.693 = 0.247 \end{aligned}}$$

Step 3: Choose attribute with the largest information gain as the decision node, divide the dataset by its branches and repeat the same process on every branch.

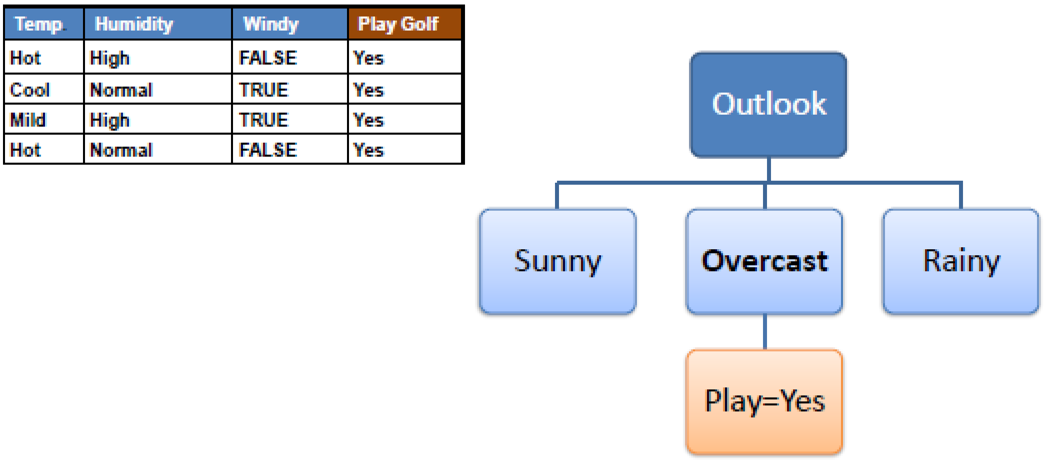


★		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			



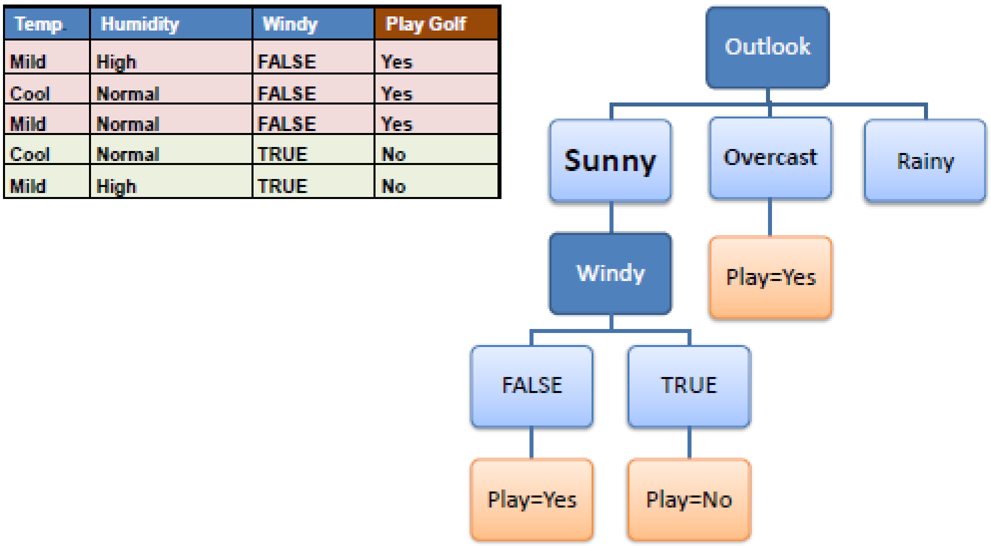
Outlook	Temp	Humidity	Windy	Play Golf
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Sunny	Mild	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Overcast	Cool	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes

Step 4a: A branch with entropy of 0 is a leaf node.



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

Step 4b: A branch with entropy more than 0 needs further splitting.



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

Step 5: The ID3 algorithm is run recursively on the non-leaf branches, until all data is classified.

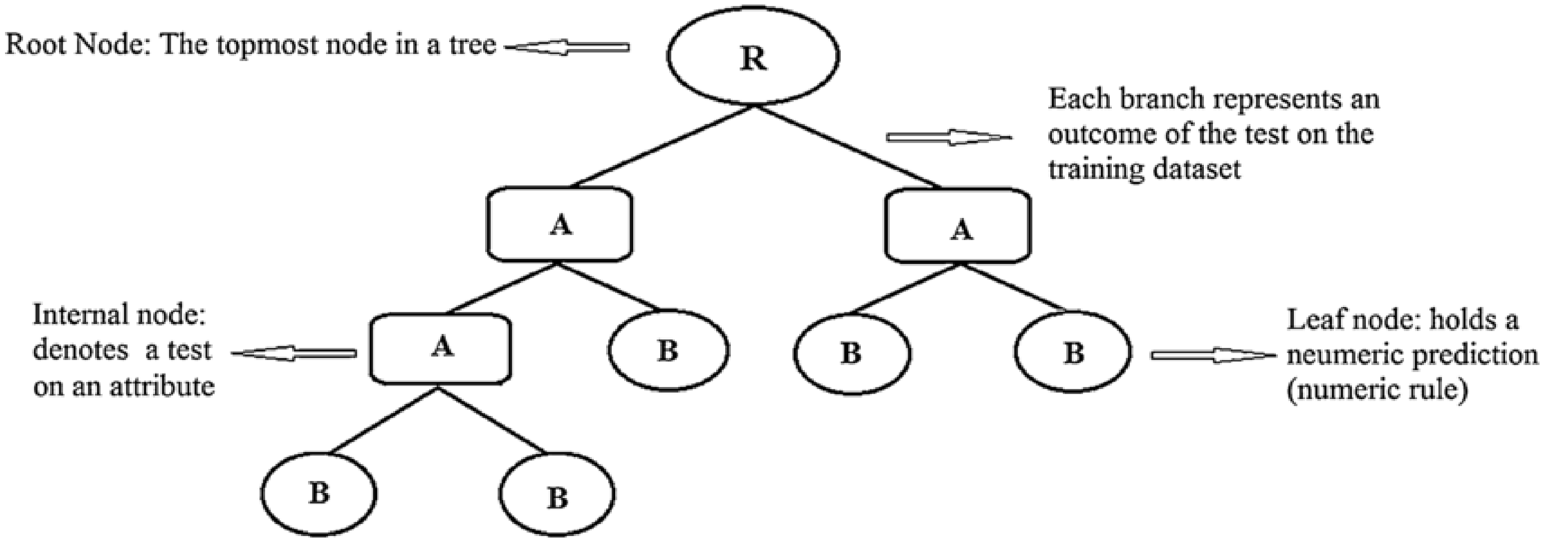
7 . How does the decision tree work, explain with an example?

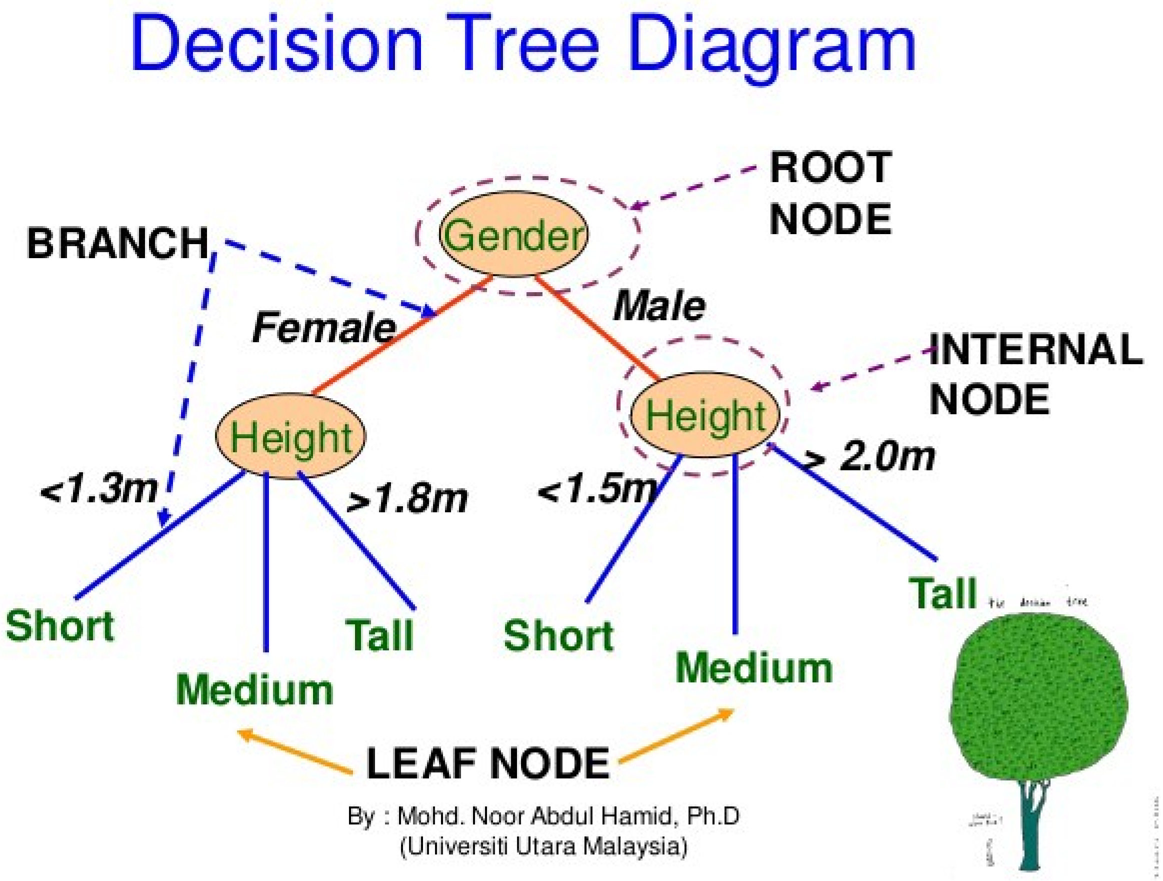
Ans.-

Decision tree is a type of supervised learning algorithm (having a pre-defined target variable) that is mostly used in classification problems. It works for both categorical and continuous input and output variables. In this technique, we split the population or sample into two or more homogeneous sets (or sub-populations) based on most significant splitter / differentiator in input variables.

Let’s say we have a sample of 30 students with three variables *Gender* (Boy/ Girl), *Class* (IX/ X) and *Height* (5 to 6 ft). 15 out of these 30 play cricket in leisure time. Now, we want to create a model to predict who will play cricket during leisure period? In this problem, we need to segregate students who play cricket in their leisure time based on highly significant input variable among all three.

This is where decision tree helps, it will segregate the students based on all values of three variable and identify the variable, which creates the best homogeneous sets of students (which are heterogeneous to each other). In the snapshot below, you can see that variable Gender is able to identify best homogeneous sets compared to the other two variables.


Root Node: The topmost node in a tree
R
Each branch represents an outcome of the test on the training dataset
A
A
Internal node: denotes a test on an attribute
A
B
B
B
Leaf node: holds a neumeric prediction (numeric rule)
B
B



Decision tree identifies the most significant variable and its value that gives best homogeneous sets of population. To identify the variable and the split, decision tree uses various algorithms.

Types of Decision Trees

Types of decision tree is based on the type of target variable we have. It can be of two types:

1. **Categorical Variable Decision Tree:** Decision Tree which has categorical target variable then it called as categorical variable decision tree. E.g.:- In above scenario of student problem, where the target variable was “Student will play cricket or not” i.e. YES or NO.

2. **Continuous Variable Decision Tree:** Decision Tree has continuous target variable then it is called as Continuous Variable Decision Tree.

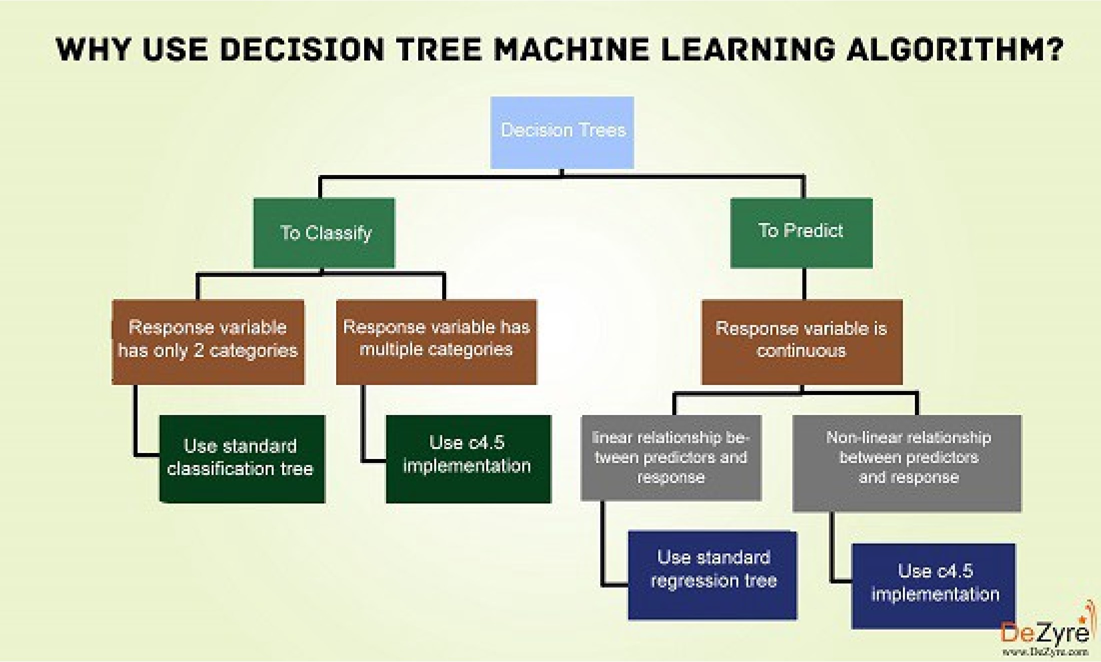
E.g.:- Let’s say we have a problem to predict whether a customer will pay his renewal premium with an insurance company (yes/ no). Here we know that income of customer is a significant variable but insurance company does not have income details for all customers. Now, as we know this is an important variable, then we can build a decision tree to predict customer income based on occupation, product and various other variables. In this case, we are predicting values for continuous variable.

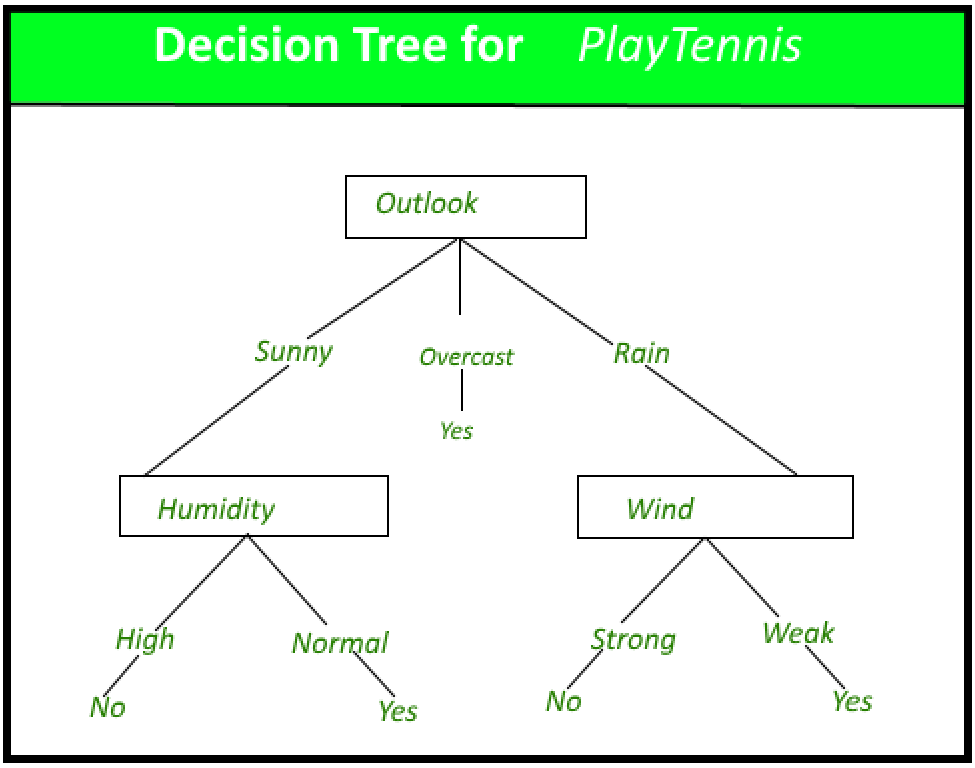
The decision tree algorithm tries to solve the problem, by using tree representation. Each internal node of the tree corresponds to an attribute, and each leaf node corresponds to a class label.

Assumptions while creating Decision Tree

Some of the assumptions we make while using Decision tree:

- At the beginning, the whole training set is considered as the **root.**
- Feature values are preferred to be categorical. If the values are continuous then they are discretized prior to building the model.
- Records are **distributed recursively** on the basis of attribute values.
- Order to placing attributes as root or internal node of the tree is done by using some statistical approach.


WHY USE DECISION TREE MACHINE LEARNING ALGORITHM?
Decision Trees
To Classify
To Predict
Response variable has only 2 categories
Response variable has multiple categories
Response variable is continuous
Use standard classification tree
Use c4.5 implementation
linear relationship between predictors and response
Non-linear relationship between predictors and response
Use standard regression tree
Use c4.5 implementation
DeZyre
www.DeZyre.com



Advantages of Decision Tree:

1. **Easy to Understand**: Decision tree output is very easy to understand even for people from non-analytical background. It does not require any statistical knowledge to read and interpret them. Its graphical representation is very intuitive and users can easily relate their hypothesis.

2. **Useful in Data exploration:** Decision tree is one of the fastest way to identify most significant variables and relation between two or more variables. With the help of decision trees, we can create new variables / features that has better power to predict target variable. It can also be used in data exploration stage.

3. Decision trees require relatively **little effort from users for data preparation**.

4. **Less data cleaning required:** It requires less data cleaning compared to some other modeling techniques. It is not influenced by outliers and missing values to a fair degree.

Disadvantages of Decision Tree:

1. **Over fitting:** Decision-tree learners can create over-complex trees that do not generalize the data well. This is called overfitting. Over fitting is one of

the most practical difficulty for decision tree models. This problem gets solved by setting constraints on model parameters and pruning.

2. **Not fit for continuous variables**: While working with continuous numerical variables, decision tree loses information, when it categorizes variables in different categories.

3. Decision trees can be unstable because small variations in the data might result in a completely different tree being generated. This is called **variance**, which needs to be lowered by methods like **bagging** and **boosting**.

4. *Greedy* algorithms cannot guarantee to return the globally optimal decision tree. This can be mitigated by training multiple trees, where the features and samples are randomly sampled with replacement.

5. Decision tree learners create *biased* trees if some classes dominate. It is therefore recommended to balance the data set prior to fitting with the decision tree.

1. **What is clustering and in what situations is it applied?**

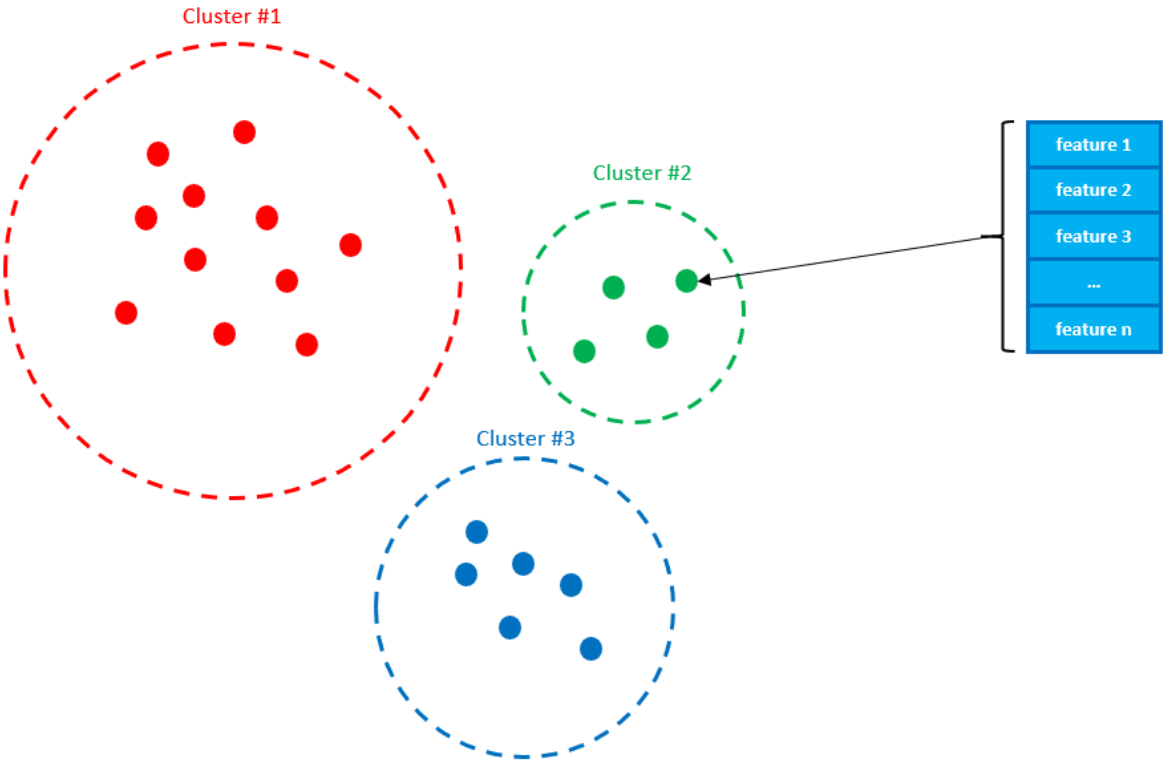
Ans-

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.

Clustering algorithms are a powerful technique for machine learning on unsupervised data. The most common algorithms in machine learning are hierarchical clustering and K-Means clustering.

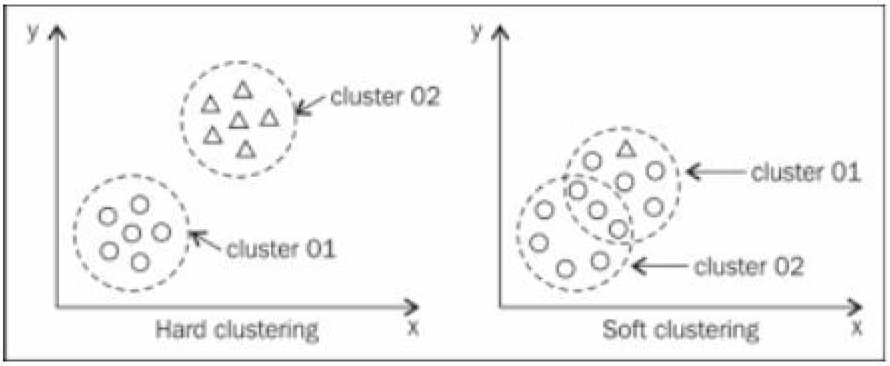
Why Clustering ?

Clustering is very much important as it determines the intrinsic grouping among the unlabeled data present. There are no criteria for a good clustering. It depends on the user, what is the criteria they may use which satisfy their need. For instance, we could be interested in finding representatives for homogeneous groups (data reduction), in finding “natural clusters” and describe their unknown properties (“natural” data types), in finding useful and suitable groupings (“useful” data classes) or in finding unusual data objects (outlier detection). This algorithm must make some assumptions which constitute the similarity of points and each assumption make different and equally valid clusters



Types of Clustering

Clustering can be divided into two subgroups :



- **Hard clustering:** It is about grouping the data items such that each piece is only assigned to one cluster. As an instance, we want the algorithm to read all of the tweets and determine if a tweet is a positive or negative tweet.

- **Soft Clustering**: Sometimes, we don’t need a binary answer. Soft clustering is about grouping the data items such that an object can exist in multiple clusters.

Applications of Clustering

We can find clustering useful in the following areas:

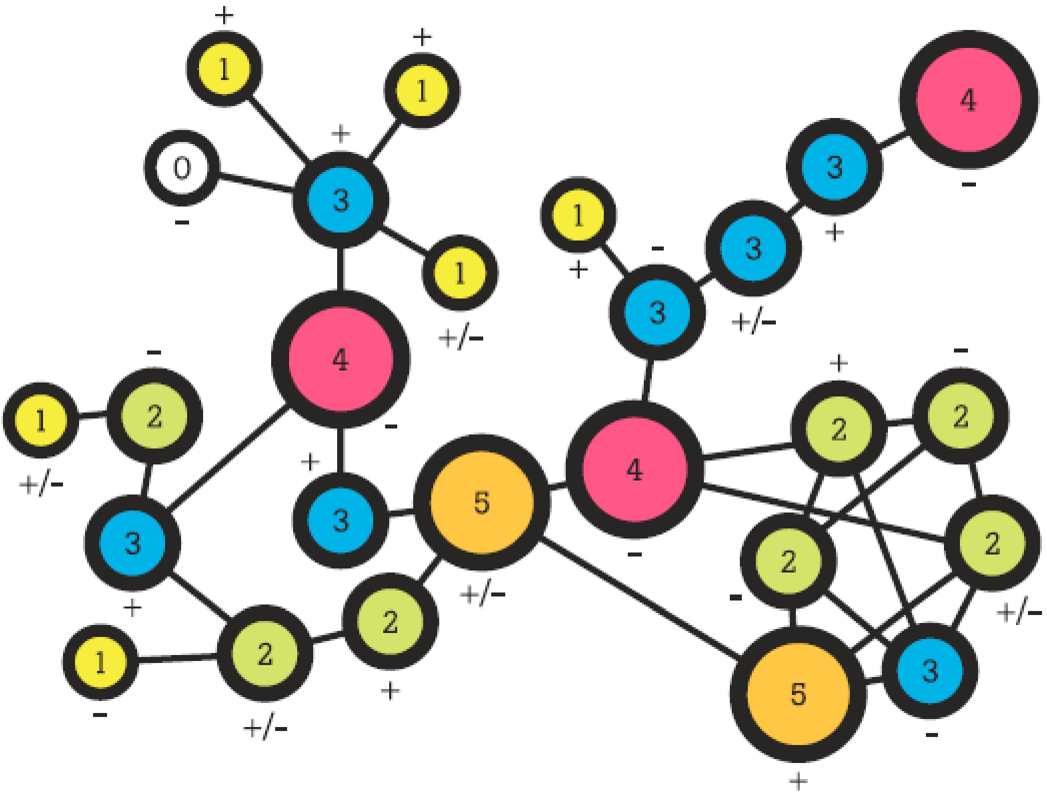
Customer Segmentation: Subdivision of customers into groups/segments such that each customer segment consists of customers with similar market characteristics — pricing, loyalty, spending behaviours etc. Some of the segmentation variables could be, e.g., the number of items bought on sale, avg transaction value, the total number of transactions.

Creating NewsFeeds*:* K-Means can be used to cluster articles by their similarity — it can separate documents into disjoint clusters.

Cloud Computing Environment**:** Clustered storage to increase performance, capacity, or reliability — clustering distributes workloads to each server, manages the transfer of workloads between servers, and provides access to all files from any server regardless of the physical location of the data.

Environmental risks: K-means can be used to analyse environmental risk in an area — environmental risk zoning of a chemical industrial area.

Pattern Recognition in images: For example, to automatically detect infected fruits or for segmentation of blood cells for leukaemia detection.



Social network analysis

Trend detection in dynamic data – Clustering can also be used for trend detection in dynamic data by making various clusters of similar trends.

Social network analysis − Clustering can be used in social network analysis. The examples are generating sequences in images, videos or audios and this approach is used in various fields.

Biological data analysis − Clustering can also be used to make clusters of images, videos; hence, it can successfully be used in biological data analysis.

Q-2 What are the different types of clustering?
Ans:

Connectivity Model -

Connectivity-based clustering, also known as *hierarchical clustering*, is based on the core idea of objects being more related to nearby objects than to objects farther away. These algorithms connect "objects" to form "clusters" based on their distance. A cluster can be described largely by the maximum distance needed to connect parts of the cluster. At different distances, different clusters will form, which can be represented using a dendrogram, which explains where the common name "hierarchical clustering" comes from: these algorithms do not provide a single partitioning of the data set, but instead provide an extensive hierarchy of clusters that merge with each other at certain distances. In a dendrogram, the y-axis marks the distance at which the clusters merge, while the objects are placed along the x-axis such that the clusters don't mix.

Connectivity-based clustering is a whole family of methods that differ by the way distances are computed. Apart from the usual choice of distance functions, the user also needs to decide on the linkage criterion (since a cluster consists of multiple objects, there are multiple candidates to compute the distance) to use. Popular choices are known as single-linkage clustering (the minimum of object distances), complete linkage clustering (the maximum of object distances), and UPGMA or WPGMA ("Unweighted or Weighted Pair Group Method with Arithmetic Mean", also known as average linkage clustering). Furthermore, hierarchical clustering can be agglomerative (starting with single elements and aggregating them into clusters) or divisive (starting with the complete data set and dividing it into partitions).

Centroid Model:

In centroid-based clustering, clusters are represented by a central vector, which may not necessarily be a member of the data set. When the number of clusters is fixed to *k*, *k*-means clustering gives a formal definition as an optimization problem: find the *k* cluster centers and assign the objects to the nearest cluster center, such that the squared distances from the cluster are minimized.

The optimization problem itself is known to be NP-hard, and thus the common approach is to search only for approximate solutions. A particularly well known approximate method is Lloyd's algorithm often just referred to as "*k-means algorithm*" (although another algorithm introduced this name). It does however only find a local optimum, and is commonly run multiple times with different random initializations. Variations of *k*-means often include such optimizations as choosing the best of multiple runs, but also restricting the centroids to members of the data set (*k*-medoids), choosing medians (*k*-medians clustering), choosing the initial centers less randomly (*k*-means++) or allowing a fuzzy cluster assignment (fuzzy c-means).

Q-3 What is the K means algorithm?Give an example and explain how does it work?

Ans:

k-means clustering is a method of vector quantization, originally from signal processing, that aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. It is popular for cluster analysis in data mining. *k*-means clustering minimizes within-cluster variances (squared Euclidean distances), but not regular Euclidean distances, which would be the more difficult Weber problem: the mean optimizes squared errors, whereas only the geometric median minimizes Euclidean distances. For instance, Better Euclidean solutions can be found using k-medians and k-medoids.

Algorithm:

The most common algorithm uses an iterative refinement technique. Due to its ubiquity, it is often called "the *k*-means algorithm"; it is also referred to as Lloyd's algorithm, particularly in the computer science community. It is sometimes also referred to as "naive *k*-means", because there exist much faster alternatives.

Given an initial set of *k* means $m_1^{(1)},\ldots,m_k^{(1)}$ (see below), the algorithm proceeds by alternating between two steps:

Assignment step: Assign each observation to the cluster with the nearest mean: that with the least squared Euclidean distance. (Mathematically, this means partitioning the observations according to the Voronoi diagram generated by the means.)

$$S_i^{(t)} = \big\{ x_p : \big\| x_p - m_i^{(t)} \big\|^2 \le \big\| x_p - m_j^{(t)} \big\|^2 \ \forall j, 1 \le j \le k \big\},$$

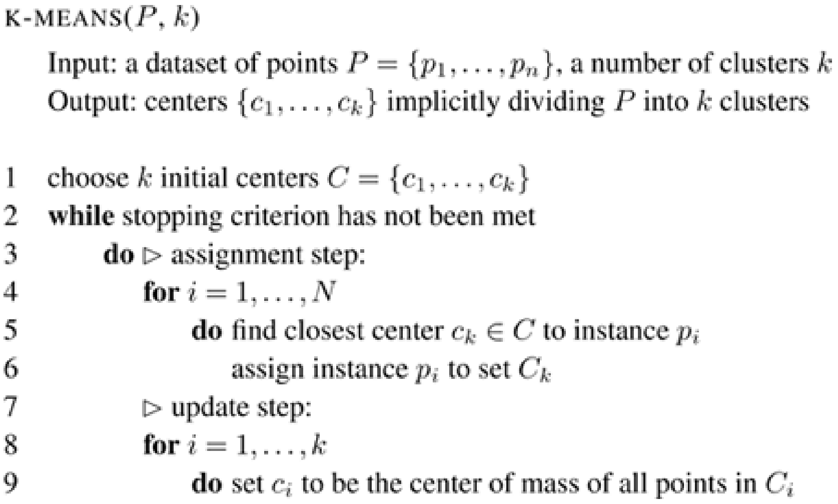
where each is assigned to exactly one ,even if it could be assigned to two or more of them.

Update step: Recalculate means (centroids) for observations assigned to each cluster.

$$m_i^{(t+1)} = \frac{1}{\left| S_i^{(t)} \right|} \sum_{x_j \in S_i^{(t)}} x_j$$

The algorithm has converged when the assignments no longer change. The algorithm does not guarantee to find the optimum.

Pseudocode:



```
K-MEANS($P$, $k$)
    Input: a dataset of points $P = \{p_1, \ldots, p_n\}$, a number of clusters $k$
    Output: centers $\{c_1, \ldots, c_k\}$ implicitly dividing $P$ into $k$ clusters

    choose $k$ initial centers $C = \{c_1, \ldots, c_k\}$
    while stopping criterion has not been met
        do ▷ assignment step:
            for $i = 1, \ldots, N$
                do find closest center $c_k \in C$ to instance $p_i$
                    assign instance $p_i$ to set $C_k$
            ▷ update step:
            for $i = 1, \ldots, k$
                do set $c_i$ to be the center of mass of all points in $C_i$
```

Example:

kmeans algorithm is very popular and used in a variety of applications such as market segmentation, document clustering, image segmentation and image compression, etc.

1.What is the advantages of using Naïve Bayes Classifier?

Ans.)

- Naive Bayes classifiers is a machine learning algorithm.
- Simplicity: easy to understand and implement.
- Light to train: no complicated optimisation required.
- Easily updateable if new training data is received.
- Small memory footprint.
- Although the independence assumption may seem sometimes unreasonable, its performance is usually good, even for those cases.
- It not only the prediction but also the degree of certainty, which can be very useful.
- Naive Bayes is particularly useful for Natural Language Processing.

2) What are the application of Naïve Bayes Classifier?

Ans.)

- It works well on small datasets. For most of the practical applications it hardly fits.
- NB has high bais and low variance. Hence it makes its application limited. Having said this there are no regularization or hyperparmeters tuning involved here to adjust the bias thing.

- Determining whether a given (text) document corresponds to one or more categories. In the text case, the features used might be the presence or absence of key words.

- To mark an email as spam, or not spam .

- Classify a news article about technology, politics, or sports.

- Check a piece of text expressing positive emotions, or negative emotions.

- Also used for face recognition softwares.