Machine Learning

Unit 3

Prof. Priti Srinivas Sajja

P G Department of Computer Science,

Sardar Patel University, Vallabh Vidyanagar, Gujarat, India

About the Speaker

- **Name: Dr. Priti Srinivas Sajja**
- Email : priti@pritisajja.info
- Mobile : +91 98249 26020
- **URL: http://pritisajja.info**
- Academic qualifications : Ph. D in Computer Science
- **Thesis Title: Knowledge-Based Systems for Socio-Economic Rural Development**
- Subject area of specialization : Artificial Intelligence
- Publications : 219 in Books, Book Chapters, Journals, Proceedings, etc.

Unit 3 : Un-supervised Learning

- Introduction to Clustering
- Self Organizing map/Kohenon neural network
- ~~K nearest neighborhood~~
- K-means and its variations
- Applications of unsupervised learning
- Introduction to hybrid learning

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 than those in other groups.
- In simple words, the **aim is to segregate groups with similar traits and assign them into clusters**.
- Let's understand this with an example. Suppose, you are the head of a rental store and wish to understand **preferences of your costumers** to scale up your business. Is it possible for you to look at details of each costumer and devise a unique business strategy for each one of them? Definitely not. But, what you can do is to **cluster all of your costumers into say 10 groups based on their purchasing habits and use a separate strategy** for costumers in each of these 10 groups. And this is what we call clustering.

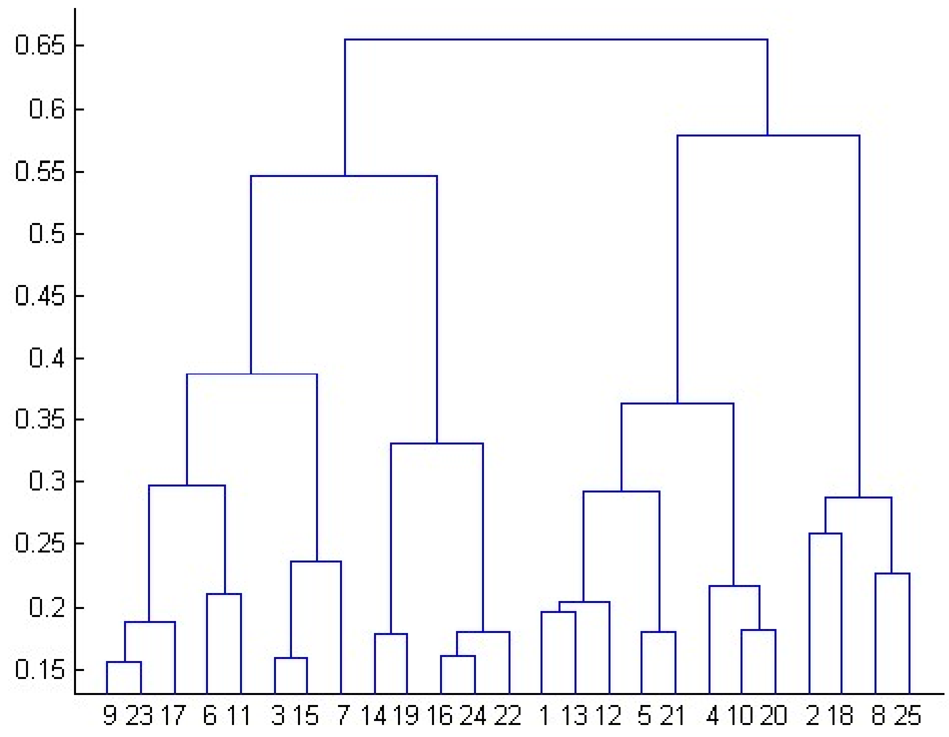
https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering/

Types of Clustering

- Broadly, clustering can be divided into two subgroups :
- **Hard Clustering:** In hard clustering, **each data point either belongs to a cluster completely or not**. For example, in the above example each customer is put into one group out of the 10 groups.
- **Soft Clustering:** In soft clustering, **instead of putting each data point into a separate cluster, a probability or likelihood of that data point to be in those clusters is assigned.** For example, from the above scenario each costumer is assigned a probability to be in either of 10 clusters of the retail store.

Other Types of Clustering

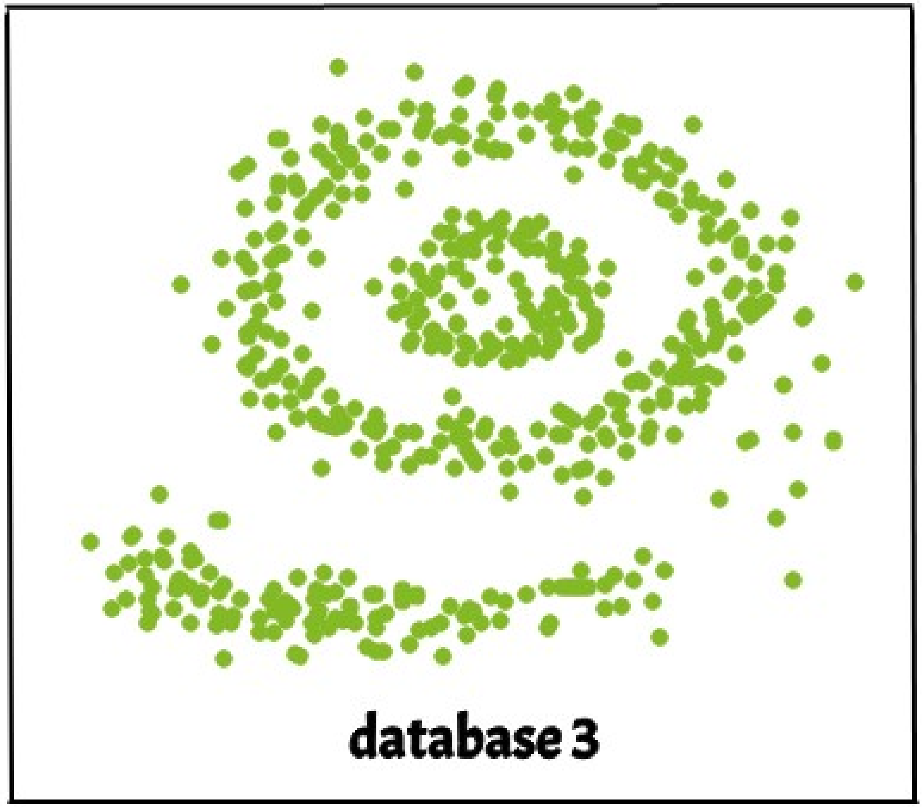
- **Hierarchical Clustering**
- Hierarchical clustering, as the name suggests is an algorithm that builds hierarchy of clusters. This algorithm starts with all the data points assigned to a cluster of their own. Then two nearest clusters are merged into the same cluster. In the end, this algorithm terminates when there is only a single cluster left.
- Most common type is Agglomerative Clustering
- The results of hierarchical clustering can be shown using dendrogram. The dendrogram can be interpreted as:



See Video Here: https://makeagif.com/i/DkJOLy

Other Types of Clustering

- **Density Based Clustering**
- Real life data my contain noise and may be in arbitrary shape.
- In this case density based clustering can be used.



Self Organizing Map

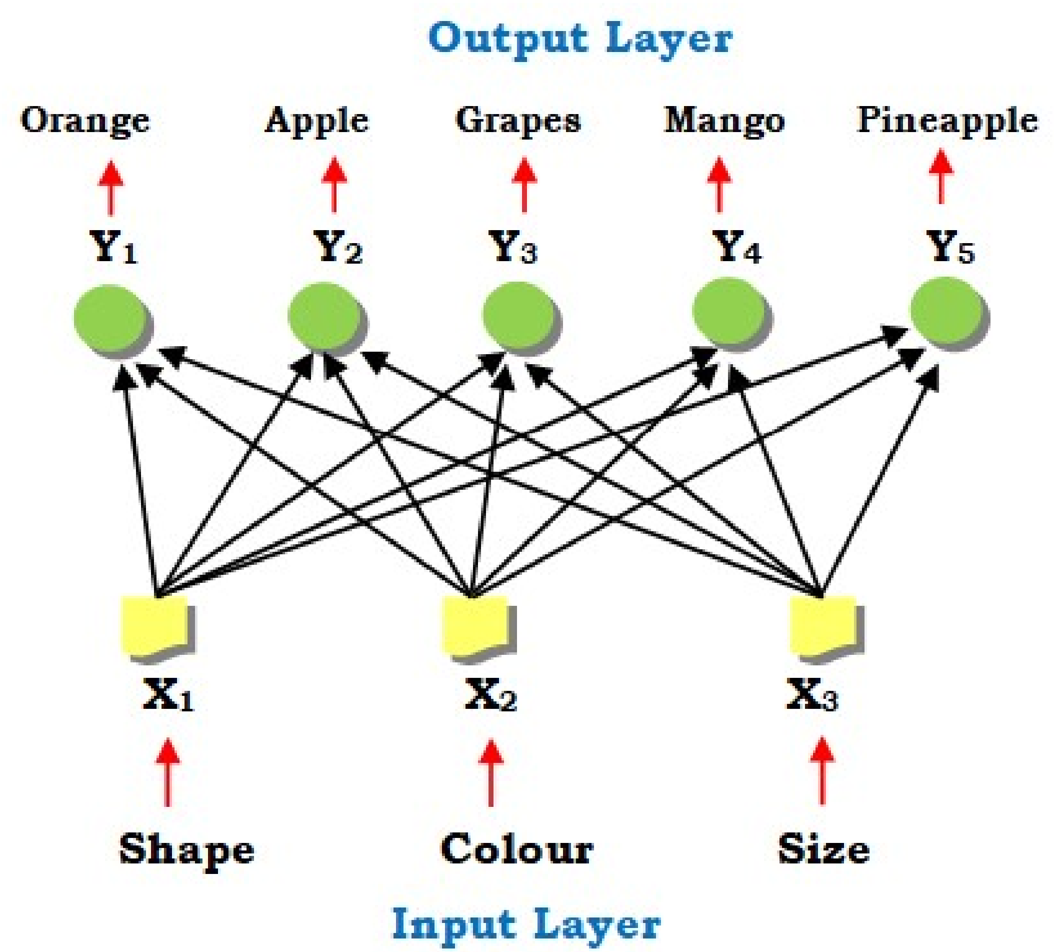
- **A self-organizing map (SOM) is a type of artificial neural network (ANN) that is trained using unsupervised learning.**
- The architecture of SOM is first proposed by the researcher from **Finland** named **Teuvo Kohonen** (1982). Besides **grouping similar data into clusters**, the SOM also acts as a **visualization and organization technique** that facilitates the **reduction of high dimensional data to a map**; hence called self-organizing map.
- That is, the SOM reduces the dimensionalities of data and highlights similarities within the data.
- Further, the features and characteristics of the input data are not mandatory to be known. That is why the self-organizing map is also referred to as a **self-organizing feature map**.
- In the typical SOM, the neurons are arranged in a flat grid. One can call it as **a two-dimensional array or map**. The SOM contains only an input layer and an output layer. There is **no hidden layer** typically in SOM.

Self Organizing Map

Table 2.7 Learning in self-organizing map

1.	Design a SOM network by arranging neurons in a flat grid/map and also design an input layer
2.	Initialize the weights of each node
3.	Chose one unit/vector of the training data set and feed it to the map
4.	Let every node determine whether its weight is similar to the weights of input vector or not. Take the help of the Euclidian distance function
5.	Consider the node whose weight is nearest to that of the input vector. The node is known as the winner or the 'Best Matching Unit (BMU)'
6.	Calculate the neighborhood of the winning node and provide a reward to the winning node in terms of weight correction
7.	Repeat step 3 onwards for a sufficient number of iterations

Self Organizing Map



Self Organizing Map

Let us consider the set of input vectors $V = [X_1, X_2, X_3, X_4]$ is given as follows.

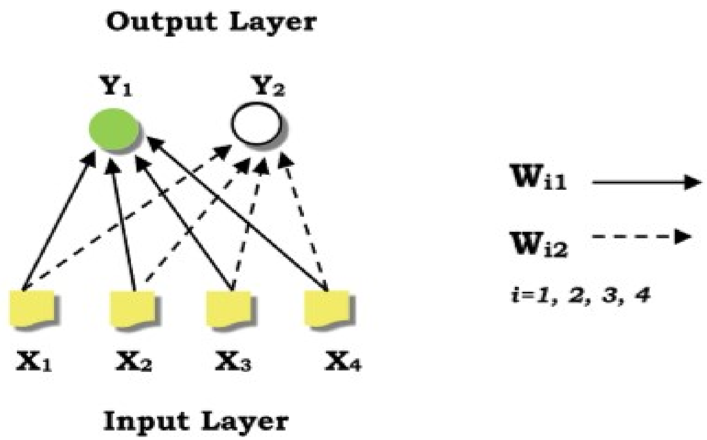
$$V = \{[1, 0, 1, 0], [0, 1, 0, 1], [1, 0, 0, 0], [0, 0, 0, 1]\}$$

Our objective is to categorize these inputs into two clusters with a given learning rate $(\alpha) = 0.5$. The solution steps are as follows.

1. As the length of each input vector is 4 and the number of clusters required is 2, the architecture of the Kohonen Self Organizing Map (SOM) is given as shown in Fig. 4.23.
2. Let us initialize random weights between 0 and 1 to all connections. The weight matrix is given as follows.

$$\begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \\ W_{31} & W_{32} \\ W_{41} & W_{42} \end{bmatrix} = \begin{bmatrix} 0.5 & 0.7 \\ 0.4 & 0.3 \\ 0.3 & 0.6 \\ 0.8 & 0.1 \end{bmatrix}$$

Fig. 4.23 Architecture of the SOM



3. For the first input vector $V_1 = [1, 0, 1, 0]$, let us find out the distance of the vector to the first (Y_1) as well as second (Y_2) output neurons. The distance is defined as:

$$D(Y_1) = \sum_{i=1}^{4} (W_{i1} - X_i)^2 \text{and } D(Y_2) = \sum_{i=1}^{4} (W_{i2} - X_i)^2$$

$$\begin{aligned} D(Y_1) &= (0.5 - 1)^{\wedge 2} + (0.4 - 0)^2 + (0.3 - 1)^2 + (0.8 - 0)^2 \\ &= 0.25 + 0.16 + 0.49 + 0.64 \\ &= 1.54 \end{aligned}$$

$$\begin{aligned} D(Y_2) &= (0.7 - 1)^2 + (0.3 - 0)^2 + (0.6 - 1)^2 + (0.1 - 0)^2 \\ &= 0.09 + 0.09 + 0.16 + 0.01 \\ &= 0.35 \end{aligned}$$

The winner is Node 2 (Y_2) of the output layer as $D(Y_2) < D(Y_1)$.

4. As the Y_2 is the winner, 2nd column ($j = 2$) weights are needed to be changed using the following formula.

$$W_{i2}(\text{New}) = W_{i2}(\text{Old}) + a(X_i - W_{i2});\ \text{where } \alpha = 0.5$$

The update is as follows.

$$\begin{aligned} W_{12}(\text{New}) &= 0.7 + 0.5(1 - 0.7) = 0.85 \\ W_{22}(\text{New}) &= 0.3 + 0.5(0 - 0.3) = 0.15 \\ W_{32}(\text{New}) &= 0.6 + 0.5(1 - 0.6) = 0.80 \\ W_{42}(\text{New}) &= 0.1 + 0.5(0 - 0.1) = 0.05 \end{aligned}$$

Hence, the new weight matrix is:

$$\begin{bmatrix} 0.50 & 0.85 \\ 0.40 & 0.15 \\ 0.30 & 0.80 \\ 0.80 & 0.05 \end{bmatrix}$$

Self Organizing Map

5. For the second input vector $V_2 = [0, 1, 0, 1]$, let us find out the distance of the vector to the first (Y_1) as well as second (Y_2) output neurons. The distance is defined as:

$$D(Y_1) = \sum_{i=1}^{4} (W_{41} - X_4)^2 \text{and } D(Y_2) = \sum_{i=1}^{4} (W_{i2} - X_6)^2$$

$$\begin{aligned} D(Y_1) &= (0.5 - 0)^2 + (0.4 - 1)^2 + (0.3 - 0)^2 + (0.8 - 1)^2 \\ &= 0.25 + 0.36 + 0.09 + 0.04 \\ &= 0.74 \end{aligned}$$

$$\begin{aligned} D(Y_2) &= (0.85 - 0)^2 + (0.15 - 1)^2 + (0.8 - 0)^2 + (0.05 - 1)^2 \\ &= 0.7225 + 0.7225 + 0.64 + 0.9025 \\ &= 2.9875 \end{aligned}$$

The winner is Node 1 (Y_1) of the output layer as $D(Y_1) < D(Y_2)$.

K Means Algorithm

- Consider the following data sets about the review of various products. It can be likes on the social media platform such as Facebook or Twitter, it can be likes/ranks for a movie on online streaming video channels such as Youtube or Netflix, or ranking of products available on online shopping sites such as Amazon or Flipkart. The data about the ranks/likes per product are available in a set-D as follows.
- **Consider** D = {2, 4, 5, 6, 8, 9, 13, 16, 18, 27, 29, 31}.
- The objective is to divide the set-D in various clusters using the K-means algorithm.

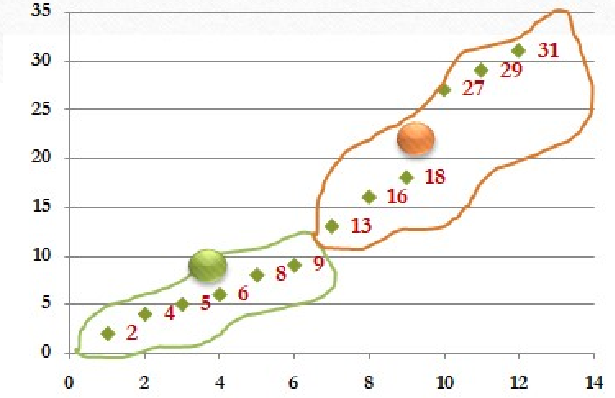
K Means Algorithm

- Acquire the number of clusters from the user and initialize K value with it.
- Determine the required number of clusters, say K, and find the K number of random mean values. That is, if the data are to be divided into two clusters, the K is assigned value 2. In this case, 2 random means need to be taken, say M_1 and M_2. If the number of clusters is N, the K value is N, and the N number of random means has to be considered, which are given as $M_1, M_2, \ldots, M_N$.
- Initialize N number of clusters namely $C_1, C_2, \ldots, C_N$ as empty sets.
- Find out the nearest value from the data set and put it into a cluster where the distance between the data item and mean is the lowest one. If there are two clusters (K=2), the picked data items have to be compared with both the randomly selected means and assigned to an appropriate cluster. That is, if the absolute value of (item–M_1) is less than the absolute value of (item–M_2), the item has to be placed into the cluster 1 called C_1. If there are N clusters, the item has to be compared with all the N means before assigning it into an appropriate cluster.

K Means Algorithm

- Once all the data items are assigned into proper clusters after the first iteration, one can visualize N clusters with separated data items. From the **N derived clusters, find out new N means and replace** the original (randomly selected) means with these means.
- Since the initial means are taken randomly to categorize the data items into clusters, the clusterization might not be perfect. Hence, the procedure has to be repeated with new means.
- Repeat the procedure until you cannot see improvement in the means.

K Means Algorithm



- Let us apply these steps to the data set D given above.
- D = {2, 4, 5, 6, 8, 9, 13, 16, 18, 27, 29, 31}.
- Let K=2 i.e. two clusters are needed. Hence, two means are required.
- Let M_1=6 and M_2=18.
- Here, D is the given data set, **Diff M_1 is the absolute difference between the M_1 (First Mean) and the data item**, Diff M_2 is the absolute difference between the M_2(Second Mean) and the data item, and C is the suitable(nearer) cluster index (Here1 or 2) where the item belongs to.

D	2	4	5	6	8	9	13	16	18	27	29	31
Diff M_1	4	2	1	0	2	3	7	10	12	21	23	25
Diff M_2	16	14	13	12	10	9	5	2	0	9	11	13
C	1	1	1	1	1	1	2	2	2	2	2	2

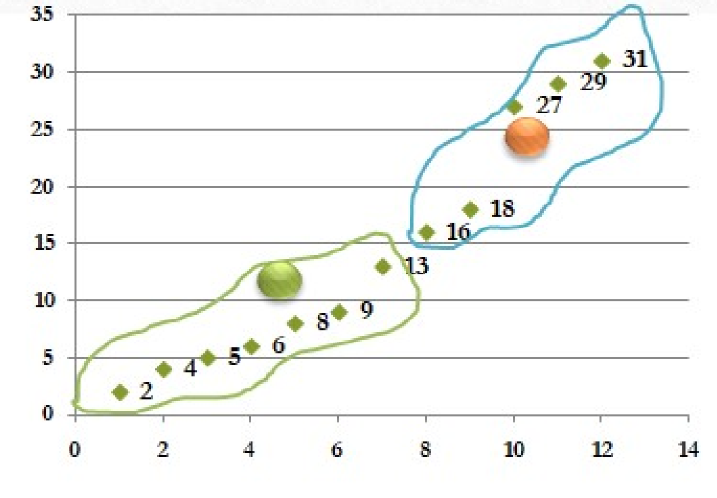
K Means Algorithm

D	2	4	5	6	8	9	13	16	18	27	29	31
Diff M_1	4	2	1	0	2	3	7	10	12	21	23	25
Diff M_2	16	14	13	12	10	9	5	2	0	9	11	13
C	1	1	1	1	1	1	2	2	2	2	2	2

- From the above calculation, it can be observed that the first 6 items (since data are sorted) fall into the first cluster (C_1). The remaining 6 items fall into the second cluster (C_2).
- Let us calculate new means from the data items from both the clusters C_1 and C_2.
- The **new means are 5.67 and 22.33.** For ease of calculation, we may take the **rounded value as 6 and 22**. With these new means, the assignment of the data items into both clusters is as follows.

K Means Algorithm

D	2	4	5	6	8	9	13	16	18	27	29	31
Diff M_1	4	2	1	0	2	3	7	10	12	21	23	25
Diff M_2	20	18	17	16	14	13	9	6	4	5	7	9
C	1	1	1	1	1	1	1	2	2	2	2	2



- From the above calculation, it can be seen that the first 7 items fall into the first cluster (C_1). The remaining 5 items fall into the second cluster (C_2).
- The new means are 6.71 and 26.8. The rounded means are 7 and 27. Reassigning the data items into clusters with modified means yields the following result.

D	2	4	5	6	8	9	13	16	18	27	29	31
Diff M_1	5	3	2	1	1	2	6	9	11	20	22	24
Diff M_2	25	23	22	21	19	18	14	11	9	0	2	4
C	1	1	1	1	1	1	1	1	2	2	2	2

- It can be seen that gradually, the number of elements in the cluster 1 (C_1) is increasing from 6 to 8.
- From the above calculation, new means evolved are 7.87 and 26.25. Considering the rounded values **of means as 8 and 26 following** result is achieved.

D	2	4	5	6	8	9	13	16	18	27	29	31
Diff M_1	6	4	3	2	0	1	5	8	10	19	21	23
Diff M_2	24	22	21	20	18	17	13	10	8	1	3	5
C	1	1	1	1	1	1	1	1	2	2	2	2

- As per the calculation given above, no improvement is seen in the cluster assignment as well as in the means. The new means are 7.87 and 26.25, which are the same as the previous ones. Hence, the final clusters are as follows.
- $C_1 = \{2, 4, 5, 6, 8, 9, 13, 16\}$; and
- $C_2 = \{18, 27, 29, 31\}$.

Applications of Un-Supervised Learning

- Clustering : Also, e-commerce websites like Amazon use clustering algorithms to implement a user-specific recommendation system.
- Feature identification and dimensionality reduction
- Visualization
- Recognization
- Anomaly detection – to discover important data points in your dataset which is useful for finding fraudulent transactions.

Introduction to Hybrid Learning

- A Hybrid system is an intelligent system which is framed **by combining at least two intelligent technologies** like Fuzzy Logic, Neural networks, Genetic algorithm, reinforcement Learning, etc.
- The combination of different techniques in one computational model make these systems possess an **extended range of capabilities**.
- These systems are capable of reasoning and learning in an uncertain and imprecise environment. These systems can provide human-like expertise like domain knowledge, adaptation in noisy environment etc.
- **Types of Hybrid Systems:**
  - Neuro Fuzzy Hybrid systems
  - Neuro Genetic Hybrid systems
  - Fuzzy Genetic Hybrid systems
  - Neuro Fuzzy Geentic Hybrid System

References

- *Shutterstocks.com, Microsoft.com,*
- *Amazinganimations.com*
- *illustrationof.com, Clipartof.com*
- *towardsdatascience.com*
- *Javapoint.com, tutorialspoint.com,*
- *Medium.com, section.io,*
- *Dotnetlovers.com,*
- *Machinelearningmastery.com*
- *Sajja, P.S. "Illustrated computational intelligence: Examples and applications", Springer International Publishing, Singapore (Nov'20)*