UNIT-2

Machine Learning: Definition, Learning Methods: Supervised Learning, Unsupervised Learning, Semi-Supervised Learning, Reinforcement Learning.

Q) Define Machine Learning. Explain types of ML Algorithms.

Machine learning is the study of computer algorithms that allow computer programs to automatically improve through experience without being explicitly programmed. Simply machine learning algorithms learn by experience, similar to how humans. **--Arthur Samuel(1959)**

An algorithm can be thought of as a set of rules/instructions that a computer programmer specifies which a computer can process.

A computer program is said to learn
from experience E
with respect to some task T and
some performance measure P,
if its performance on T,
as measured by P,
improves with experience E. **-- Tom Mitchell(1997)**

Eg. So if you want your program to predict, for example, traffic patterns at a busy intersection (task T), you can run it through a machine learning algorithm with data about past traffic patterns (experience E) and, if it has successfully “learned”, it will then do better at predicting future traffic patterns (performance measure P).

Need: Machine learning enables analysis of massive quantities of data. While it generally delivers faster, more accurate results in order to identify profitable opportunities or dangerous risks, it may also require additional time and resources to train it properly. Combining machine learning with AI and cognitive technologies can make it even more effective in processing large volumes of information.

Machine learning algorithms are often categorized as supervised or unsupervised.

1. **Supervised** machine learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events.

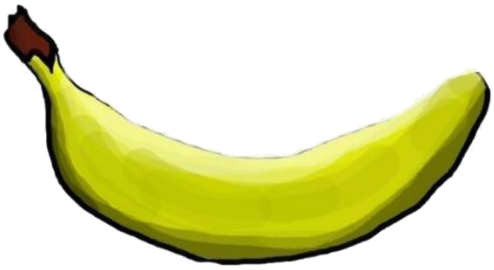
Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values.

The system is able to provide targets for any new input after sufficient training.
The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

Eg. suppose you are given an basket filled with different kinds of fruits.

Now the first step is to train the machine with all different fruits one by one like this:

If shape of object is rounded and depression at top having color Red then it will be labelled as –**Apple**.
If shape of object is long curving cylinder having color Green-Yellow then it will be labelled as –**Banana**.



Now suppose after training the data, you have given a new separate fruit say Banana from basket and asked to identify it.

Since machine has already learned the things from previous data and this time have to use it wisely. It will first classify the fruit with its shape and color, and would confirm the fruit name as BANANA and put it in Banana category. Thus machine learns the things from training data(basket containing fruits) and then apply the knowledge to test data(new fruit).

Supervised learning classified into two categories of algorithms:

Classification: A classification problem is when the output variable is a category, such as —Red or blue or disease and no disease.

Regression: A regression problem is when the output variable is a real value, such as price or weight or height.

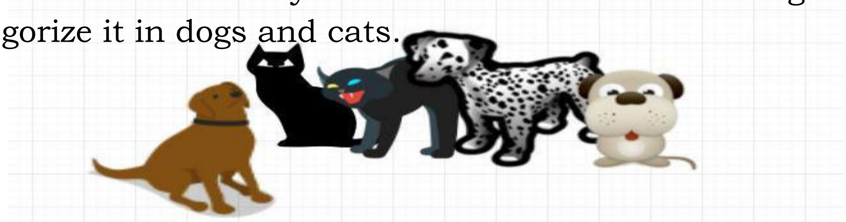
2. In contrast, **unsupervised** machine learning algorithms are used when the information used to train is neither classified nor labeled.

Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data.

The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

Eg.

suppose it is given an image having both dogs and cats which have not seen ever. Thus machine has no any idea about the features of dogs and cat so we can't categorize it in dogs and cats.



But it can categorize them according to their similarities, patterns and differences i.e., we can easily categorize the above picture into two parts. First may contain all pictures having **dogs** in it and second part may contain all pictures having **cats** in it based on some properties like color, size etc. Here you didn't learn anything before, means no labeled data.

3. **Semi-supervised** machine learning algorithms fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training – typically a small amount of labeled data and a large amount of unlabeled data.

The systems that use this method are able to considerably improve learning accuracy.

Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring unlabeled data generally doesn't require additional resources.

4. **Reinforcement learning** is a type of machine learning in which a computer learns to perform a task through repeated trial-and-error interactions with a dynamic environment. This learning approach enables the computer to make a series of decisions that maximize a reward metric for the task without human intervention and without being explicitly programmed to achieve the task.

Simple reward feedback is required for the agent to learn which action is best this is known as the reinforcement signal.

Eg.

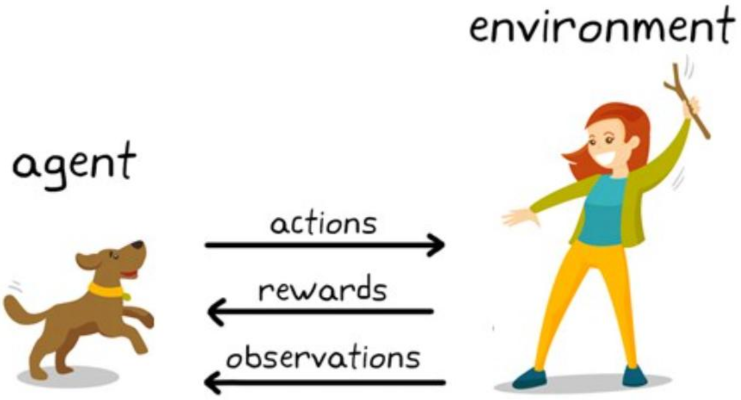


Fig. RL in dog training

The goal of reinforcement learning in this case is to train the dog (*agent)* to complete a task within an *environment*, which includes the surroundings of the dog as well as the trainer.

First, the trainer issues a command, which the dog observes (*observation*). The dog then responds by taking an *action*. If the action is close to the desired behavior, the trainer will likely provide a *reward*, such as a food treat; otherwise, no reward or a negative reward will be provided.

At the beginning of training, the dog will likely take more random actions like rolling over when the command given is "sit," as it is trying to associate specific observations with actions and rewards. This association, or mapping, between observations and actions is called *policy*.

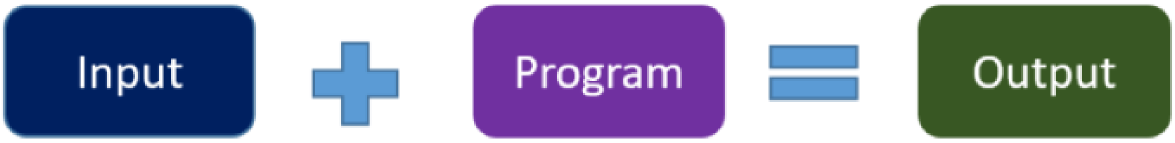
From the dog's perspective, the ideal case would be one in which it would respond correctly to every command, so that it gets as many treats as possible.

So, the whole meaning of reinforcement learning training is to "tune" the dog's policy so that it learns the desired behaviors that will maximize some reward. After training is complete, the dog should be able to observe the owner and take the appropriate action, for example, sitting when commanded to "sit" by using the internal policy it has developed.

Q) Differentiate ML Vs Classical/Traditional Algorithms.

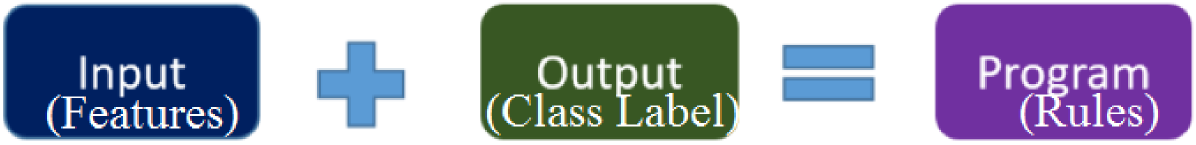
Traditional Programming

Traditional programming is a manual process—meaning a programmer creates the program.



Machine Learning

Unlike traditional programming, machine learning is an automated process. ML algorithm automatically formulates the rules from the data.



ML vs Classical Algorithms:

- ML algorithms do not depend on rules defined by human experts. Instead, they process data in raw form like text, emails, documents, social media content, images, voice and video.
- An ML system is truly a learning system if it is not programmed to perform a task, but is programmed to learn to perform the task
- Most ML models are uninterpretable, and for these reasons they are usually unsuitable when the purpose is to understand relationships. The mostly work well where one only needs predictions.
- One of the key differences is that classical approaches have a more rigorous mathematical approach while machine learning algorithms are more data-intensive

Q) Define a Data Object.

Data sets are made up of data objects. A **data object** represents an entity in a database table. Data objects are typically described by attributes. Data objects can also be referred to as *samples, examples, instances, data points*, or *objects*. If the data objects are stored in a database, they are *data tuples*. That is, the rows of a database correspond to the data objects, and the columns correspond to the attributes.

Q) Define an attribure. Explain types of attributes with an example.

An **attribute** is a data field, representing a characteristic or feature of a data object. It also called dimension, feature, variable. The type of an attribute is determined by the set of possible values—nominal, binary, ordinal, or numeric.

Nominal attributes: Each value represents some kind of category, code, or state, and so nominal attributes are also referred to as categorical. The values do not have any meaningful order.
Eg. marital status, occupation, ID numbers, zip codes, hair color.

Ordinal Attributes: An ordinal attribute is an attribute with possible values that have a meaningful order or ranking among them, but the magnitude between successive values is not known.
Eg. Grades, Qualification.

Binary Attributes: A binary attribute is a nominal attribute with only two categories or states: 0 or 1, where 0 typically means that the attribute is absent, and 1 means that it is present.
Binary attributes are referred to as Boolean if the two states correspond to true and false.

Symmetric binary: A binary attribute is symmetric if both of its states are equally valuable and carry the same weight; that is, there is no preference on which outcome should be coded as 0 or 1.
Eg. gender
Asymmetric binary: A binary attribute is asymmetric if the outcomes of the states are not equally important, such as the positive and negative outcomes of a medical test
Eg. test for fever, result of an examination.

Numeric Attributes: A numeric attribute is quantitative; that is, it is a measurable quantity, represented in integer or real values. Numeric attributes can be interval-scaled or ratio-scaled.

Interval-scaled attributes are measured on a scale of equal-size units. The values of interval-scaled attributes have order and can be positive, 0, or negative. It is not a Zero-Point.
E.g. temperature in C°or F°, calendar dates

A **ratio-scaled attribute** is a numeric attribute with an inherent zero-point. That is, if a measurement is ratio-scaled, we can speak of a value as being a multiple (or ratio) of another value.
Eg. temperature in Kelvin, length, counts, monetary quantities

Discrete versus Continuous Attributes

Discrete Attribute: A **discrete attribute** has a finite or countably infinite set of values, which may or may not be represented as integers. Binary attributes are a **special case** of discrete attributes.
E.g. zip codes, profession, or the set of words in a collection of documents.

Continuous Attribute: If an attribute is not discrete, it is **continuous**. It has real numbers as attribute values. Continuous attributes are typically represented as floating-point variables.
E.g. temperature, height, or weight of a person.

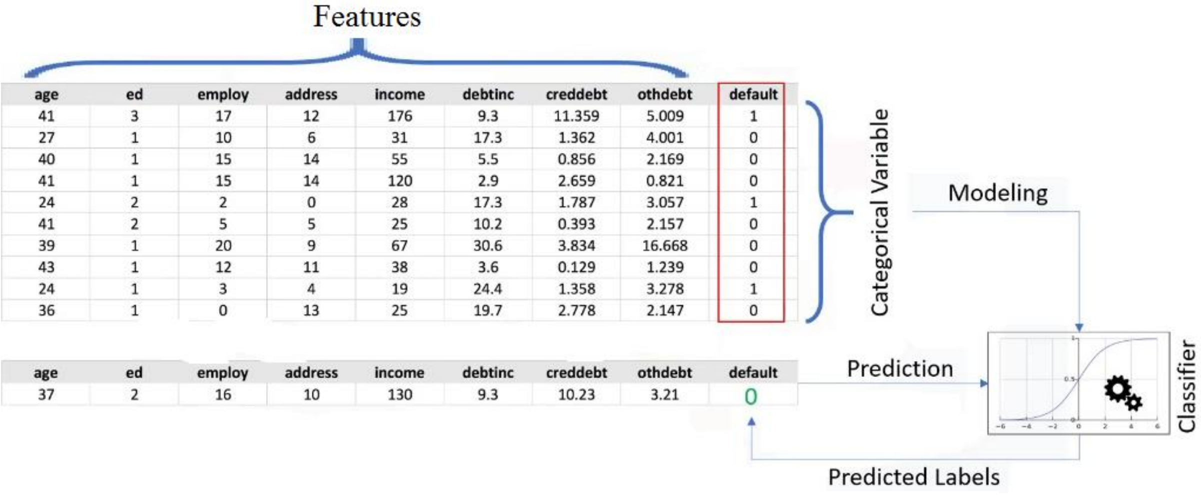
Q) Explain indetail about Supervised Learning approach.

Predictive or **Supervised** learning approach, the goal is to learn a mapping from inputs x to outputs y, given a labeled set of input-output pairs.

$$\mathbf{D = \{(x_i,\ y_i)\}_{i=1to\ N}.}$$

Here, D is called the training set, and N is the number of training examples.

Classification:
Classification is a type of supervised learning. It specifies the class to which data elements belong to and is best used when the output has finite and discrete values. It predicts a class for an input variable as well. Here the class label is categorical.



age	ed	employ	address	income	debtinc	creddebt	othdebt	default
41	3	17	12	176	9.3	11.359	5.009	1
27	1	10	6	31	17.3	1.362	4.001	0
40	1	15	14	55	5.5	0.856	2.169	0
41	1	15	14	120	2.9	2.659	0.821	0
24	2	2	0	28	17.3	1.787	3.057	1
41	2	5	5	25	10.2	0.393	2.157	0
39	1	20	9	67	30.6	3.834	16.668	0
43	1	12	11	38	3.6	0.129	1.239	0
24	1	3	4	19	24.4	1.358	3.278	1
36	1	0	13	25	19.7	2.778	2.147	0

age	ed	employ	address	income	debtinc	creddebt	othdebt	default
37	2	16	10	130	9.3	10.23	3.21	0

Fig. Illustration of Classification for Loan Sanction.

Each training input x_i is a D-dimensional vector of numbers, representing, say, the age, ed, employ, address, income, debtinc, creddebt and othdebt of a person. These are called features, attributes, dimensions, predictor variable, independent variables, columns or covariates. In general, however, x_i could be a complex structured object, such as an image, a sentence, an email message, a time series, a molecular shape, a graph, etc

Similarly the form of the output, dependent variable or response variable can in principle is anything, but most methods assume that y_i is a categorical or nominal variable from some finite set, $y_i \in \{1, . . . , C\}$, where C is no.of classes.

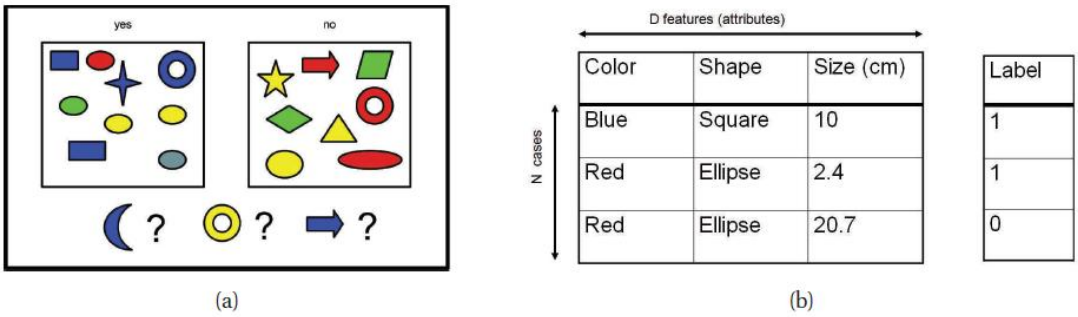
If C = 2, this is called binary classification (Eg. 0 or 1, yes/no)
if C > 2, this is called multiclass classification.(Eg. {0,1 or 2}, {high,low,medium})

When y_i is categorical, the problem is known as classification or pattern recognition, and when y_i is real-valued, the problem is known as regression.

We assume y = f(x) for some unknown function f, and the goal of learning is to estimate the function f given a labelled training set, and then to make predictions using ˆy = ˆ f(x). (We use the hat symbol to denote an estimate.)

Our main goal is to make predictions on new inputs, meaning ones that we have not seen before called generalization. Since predicting the response on the training set is easy as we can just look up the answer.

Need for probabilistic predictions:



Color	Shape	Size (cm)	Label
Blue	Square	10	1
Red	Ellipse	2.4	1
Red	Ellipse	20.7	0

Figure Left: Some labeled training examples of colored shapes, along with 3 unlabeled test cases. Right: Representing the training data as an $N \times D$ design matrix. Row i represents the feature vector $\mathbf{x}_i$. The last column is the label, $y_i \in \{0, 1\}$.

In above fig., the yellow circle is harder to classify, since some yellow things are labeled y = 1 and some are labeled y = 0, and some circles are labeled y = 1 and some y = 0.

Consequently it is not clear what the right label should be in the case of the yellow circle. Similarly, the correct label for the blue arrow is unclear.

To handle ambiguous cases, such as the yellow circle above, it is desirable to return a probability.

Given a probabilistic output, we can always compute our “best guess” as to the “true label” using

$$\hat{y} = \hat{f}(\mathbf{x}) = \operatorname*{argmax}_{c=1}^{C} p(y = c|\mathbf{x}, \mathcal{D})$$

This corresponds to the most probable class label, and is called the mode of the distribution p(y|x,D); it is also known as a MAP estimate (MAP stands for maximum a posteriori).

Real-world applications of classification:
Classification is probably the most widely used form of machine learning, and has been used to solve many interesting and often difficult real-world problems.

1. In **document classification**, the goal is to classify a document, such as a web page or email message, into one of C classes, that is, to compute p(y = c|x,D), where x is some representation of the text.
   A special case of this is **email spam filtering**, where the classes are spam y = 1 or ham y = 0.
2. **Image classification**:
   i. Classifying flowers: The goal is to learn to distinguish three different kinds of iris flower, called setosa, versicolor and virginica.

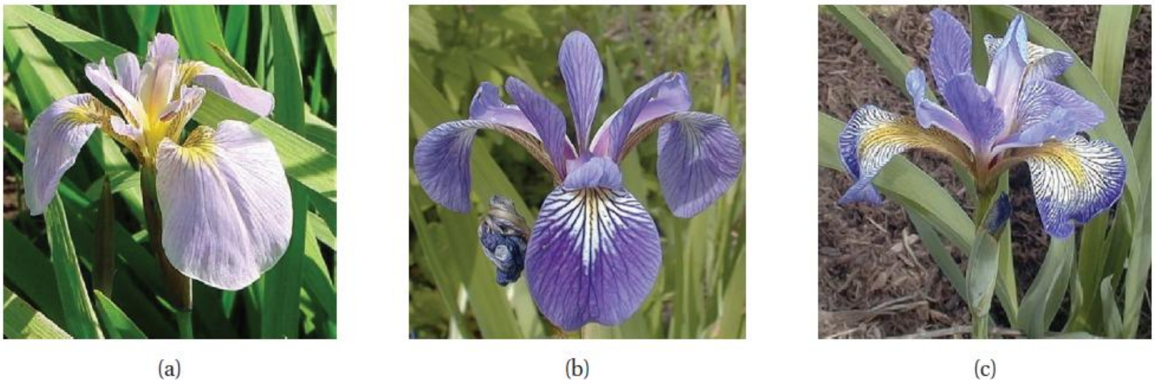


Figure 1.3 Three types of iris flowers: setosa, versicolor and virginica. Source: http://www.statlab.uni-heidelberg.de/data/iris/ . Used with kind permission of Dennis Kramb and SIGNA.

   ii. Hand writing recognition

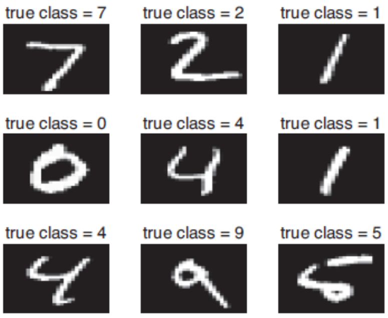


Figure First 9 test MNIST gray-scale images.

MNIST, which stands for “Modified National Institute of Standards, This dataset contains 60,000 training images and 10,000 test images of the digits 0 to 9, as written by various people. The images are size 28×28 and have grayscale values in the range 0 : 255.

3. **Object detection and recognition:** A harder problem is to find objects within an image; this is called object detection or object localization. A special case of this is face detection.

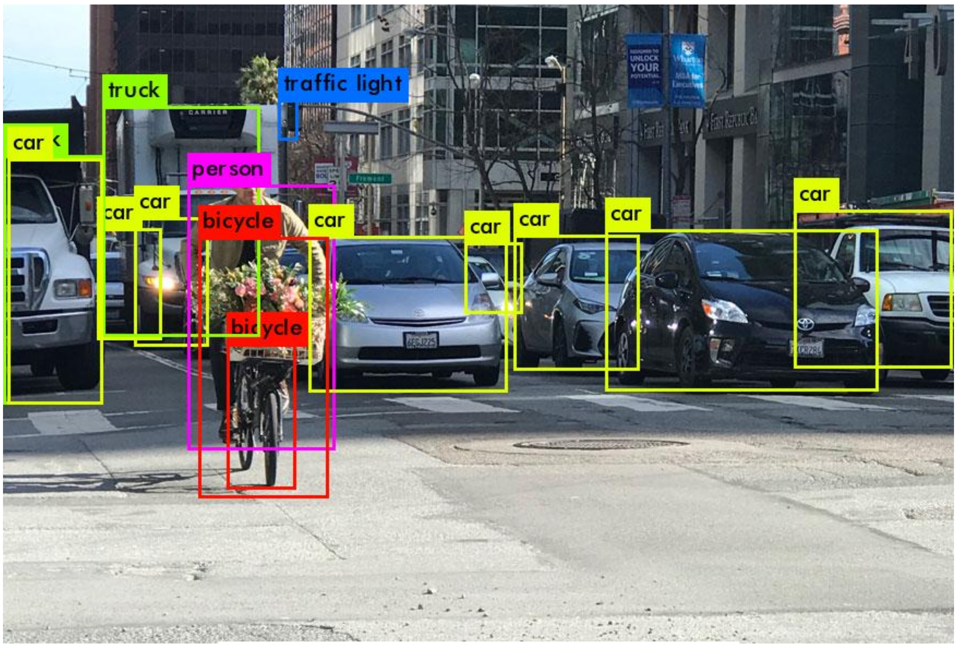


Fig. Example for Object Detection

4. In banking, predicting loan sanction.
5. Speech Identification

Regression:

Regression is just like classification except the response variable is continuous.

Regression analysis consists of a set of machine learning methods that allow us to predict a continuous outcome variable (y) based on the value of one or multiple predictor variables (x).

In **linear regression**, the data are modeled to fit a straight line.
y = wx+b, variable, y (called a response variable), can be modeled as a linear function of another random variable, x (called a predictor variable) where the variance of y is assumed to be constant.

In the context of data mining, x and y are numeric database attributes. The coefficients, w and b (called regression coefficients), specify the slope of the line and the y-intercept, respectively.
Eg.

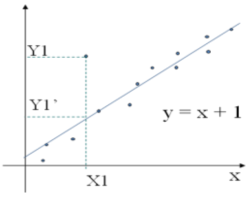
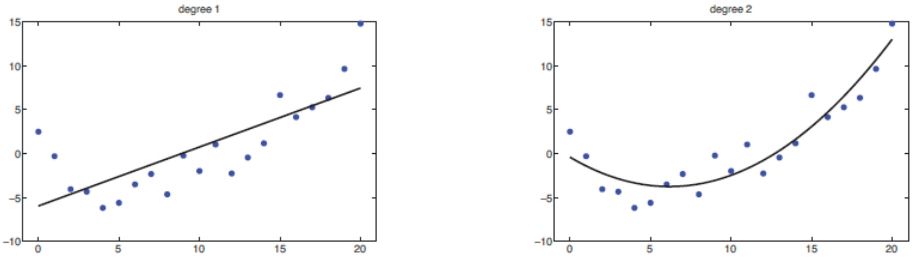


Fig. Linear Regression

In non-**linear regression**, the data can't be modelled to fit a straight line i.e. modelled to fit a curve as shown in below figure.



(a)

(b)

Figure (a) Linear regression on some 1d data. (b) Same data with polynomial regression (degree 2).

For example the equation can have the form: X^{Theta}

Here are some examples of real-world regression problems.

- Predict tomorrow's stock market price given current market conditions and other possible side information.
- Predict the age of a viewer watching a given video on YouTube.
- Predict the location in 3d space of a robot arm end effector, given control signals (torques) sent to its various motors.
- Predict the amount of prostate specific antigen (PSA) in the body as a function of a number of different clinical measurements.
- Predict the temperature at any location inside a building using weather data, time, door sensors, etc.

Q) Differentiate Simple linear regression and Multi linear regression.

Linear regression models the relationship between a dependent variable and one or more explanatory variables using a linear function can be formulated as below:

$$\mathbf{y = b_o + b_1 * x_1}$$

Eg. Prediction of height based on age.

Feature(X): age

Class(y): height

Regression equation will be: height = w*age + b

If two or more explanatory variables have a linear relationship with the dependent variable, the regression is called a multiple linear regression, which can be formulated as below:

$$\mathbf{y = b_o + b_1 * x_1 + b_2 * x_2 \quad .. + b_n * x_n}$$

where,

y is the response variable.

a, b_1, $b_2 \ldots b_n$ are the coefficients.
x_1, x_2, $\ldots x_n$ are the predictor variables.

Eg. Prediction of height based on age and gender.
Features(X):age, gender
class(y): height
Regression equation will be: height = w1*age + w2*gender + b

Q) Explain about KNN classification

Eager learners, when given a set of training tuples, will construct a generalization (i.e., classification) model before receiving new (e.g., test) tuples to classify.

Eg. decision tree induction, Bayesian classification, rule-based classification, classification by backpropagation, support vector machines, and classification based on association rule mining.

A **lazy learner** simply stores the given set of training tuples and only when it sees the test tuple does it perform generalization to classify the tuple based on its similarity to the stored training tuples.

Eg. KNN

k-Nearest-Neighbor Classifiers

Algorithm:

- Initialize k value.
- Calculate the distance between test data and each row of training data. Here we will use Euclidean distance as our distance metric since it's the most popular method. The other metrics that can be used are manhatten, cosine, etc.
- Sort the calculated distances in ascending order based on distance values
- Get top k rows from the sorted array
- Get the most frequent class of these rows
- Return the predicted class

Eg.

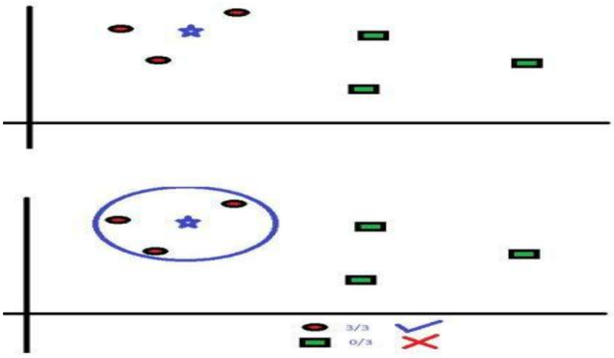


Fig. KNN Classifier

In the above fig. the new data point star belongs to circle class as 3 nearest neighbours are circles.

Advantages:

- Quick calculation time
- Simple algorithm – to interpret
- useful for regression and classification
- No assumptions about data – no need to make additional assumptions, tune several parameters, or build a model. This makes it crucial in nonlinear data case.

Disadvantages:

- Accuracy depends on the quality of the data
- With large data, the prediction stage might be slow
- Require high memory – need to store all of the training data

Q) Briefly explain about logistic regression.

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes.

Types of Logistic Regression

Generally, logistic regression means binary logistic regression having binary target variables, but there can be two more categories of target variables that can be predicted by it. Based on the number of categories, Logistic regression can be divided into following types:

- **Binary Logistic Regression Model:** The simplest form of logistic regression is binary or binomial logistic regression in which the target or dependent variable can have only 2 possible types either 1 or 0.

  Eg. Spam detection is a binary classification problem where we are given an email and we need to classify whether or not it is spam. If the email is spam, we label it 1; if it is not spam, we label it 0. In order to apply Logistic Regression to the spam detection problem, the following features of the email are extracted:

  Sender of the email
  Number of typos in the email
  Occurrence of words/phrases like “offer”, “prize”, “free gift”, etc.

  The resulting feature vector is then used to train a Logistic classifier which emits a score in the range 0 to 1. If the score is more than 0.5, we label the email as spam. Otherwise, we don’t label it as spam.

- **Multinomial Logistic Regression Model**: Another useful form of logistic regression is multinomial logistic regression in which the target or dependent variable can have 3 or more possible unordered types i.e. the types having no quantitative significance.

Eg. Entering high school students make program choices among general program, vocational program and academic program. Their choice might be modeled using their writing score and their social economic status.

Q) Differentiate Key Differences between Linear and Logistic Regression.

1. The Linear regression models data using continuous numeric value. As against, logistic regression models the data in the binary values.
2. Linear regression requires establishing the linear relationship among dependent and independent variable whereas it is not necessary for logistic regression.
3. In the linear regression, the independent variable can be correlated with each other. In contrast the logistic regression, the variable must not be correlated with each other.

Q) Explain indetail about Unsupervised learning.

In unsupervised learning, we are just given output data, without any inputs. The goal is to discover "interesting structure" in the data; this is sometimes called knowledge discovery.

Unlike supervised learning, we are not told what the desired output is for each input. Instead, we will formalize our task as one of density estimation, that is, we want to build models of the form $p(x_i|\theta)$.

Difference between Supervised and Unsupervised learning:

1. Supervised learning is conditional density estimation, whereas unsupervised learning is unconditional density estimation.
2. x_i is a vector of features, so we need to create multivariate probability models. By contrast, in supervised learning, y_i is usually just a single variable that we are trying to predict.

Unsupervised learning is arguably more typical of human and animal learning. It is also more widely applicable than supervised learning, since it does not require a human expert to manually label the data. Labeled data is not only expensive to acquire, but it also contains relatively little information, certainly not enough to reliably estimate the parameters of complex models.

Here are some real world applications of clustering:

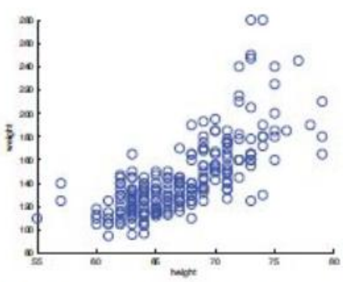
- In astronomy, the autoclass system (Cheeseman et al. 1988) discovered a new type of star, based on clustering astrophysical measurements.
- In e-commerce, it is common to cluster users into groups, based on their purchasing or web-surfing behavior, and then to send customized targeted advertising to each group.
- In biology, it is common to cluster flow-cytometry data into groups, to discover different sub-populations of cells.

• In banking, it is used to find credit card fraud detection by using user transaction behavior.

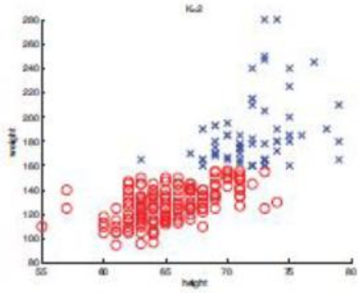
Canonical examples of unsupervised learning:

1. **Discovering clusters**

consider the problem of clustering data into groups.



The height and weight of some people.



A possible clustering using *K = 2 clusters.*

The above figure, plots some 2d data, representing the height and weight of a group of 210 people. It seems that there might be various clusters, or subgroups, although it is not clear how many. Let K denote the number of clusters. Our **first goal** is to estimate the distribution over the number of clusters, p(K|D); this tells us if there are subpopulations within the data. For simplicity, we often approximate the distribution p(K|D) by its mode,

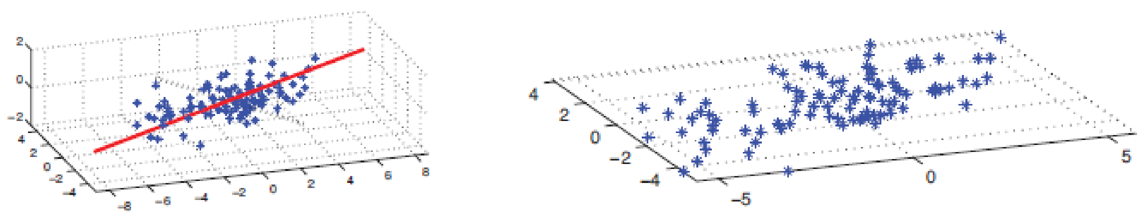
$$\mathbf{K^* = arg\ max_K\ p(K|D).}$$

In the supervised case, we were told that there are two classes (male and female), but in the unsupervised case, we are free to choose as many or few clusters as we like. Picking a model of the "right" complexity is called model selection.

Our **second goal** is to estimate which cluster each point belongs to. Let z_i(z_i is an example of a hidden or latent variable, since it is never observed in the training set.) ∈ {1, . . . , K} represent the cluster to which data point i is assigned.

2. **Discovering latent factors**

When dealing with high dimensional data, it is often useful to reduce the dimensionality by projecting the data to a lower dimensional subspace which captures the "essence" of the data. This is called **dimensionality reduction.**



(a)

(b)

Figure (a) A set of points that live on a 2d linear subspace embedded in 3d. The solid red line is the first principal component direction. The dotted black line is the second PC direction. (b) 2D representation of the data.

Such low dimensional representations often result in better predictive accuracy, because they focus on the “essence” of the object, filtering out inessential features.
Also, low dimensional representations are useful for enabling fast nearest neighbor searches and two dimensional projections are very useful for visualizing high dimensional data.

The most common approach to dimensionality reduction is called principal components analysis (PCA).

PCA in particular, has been applied in many different areas. Some examples include the following:

- In biology, it is common to use PCA to interpret gene microarray data, to account for the fact that each measurement is usually the result of many genes which are correlated in their behavior by the fact that they belong to different biological pathways.
- In natural language processing, it is common to use a variant of PCA called latent semantic analysis for document retrieval.
- In signal processing (e.g., of acoustic or neural signals), it is common to use ICA (which is a variant of PCA) to separate signals into their different sources
- In computer graphics, it is common to project motion capture data to a low dimensional space, and use it to create animations.

3. Discovering graph structure

Sometimes we measure a set of correlated variables, and we would like to discover which ones are most correlated with which others. This can be represented by a graph G, in which nodes represent variables, and edges represent direct dependence between variables. We can then learn this graph structure from data, i.e., we compute ˆG = argmax p(G|D).

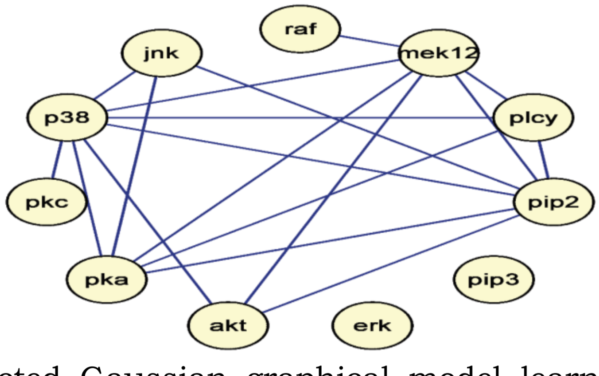


Fig. sparse undirected Gaussian graphical model learned using graphical lasso applied to some flow cytometry data which measures the phosphorylation status of 11 proteins.

As with unsupervised learning in general, there are two main applications for learning sparse graphs: to discover new knowledge, and to get better joint probability density estimators.

4. Matrix Completion

Sometimes we have missing data, that is, variables whose values are unknown.
For example, we might have conducted a survey, and some people might not have answered certain questions. Or we might have various sensors, some of which fail.
The corresponding design matrix will then have "holes" in it; these missing entries are often represented by NaN, which stands for "not a number". The goal of imputation is to infer plausible values for the missing entries also called matrix completion.

Applications of Matrix completion:

1. Image inpainting:

The goal is to "fill in" holes (e.g., due to scratches or occlusions) in an image with realistic texture, where we denoise the image, as well as impute the pixels hidden behind the occlusion. This can be tackled by building a joint probability model of the pixels, given a set of clean images, and then inferring the unknown variables (pixels) given the known variables(pixels).

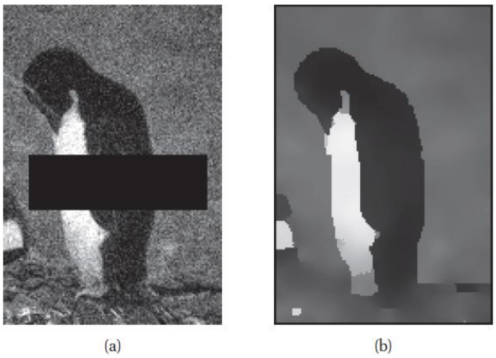
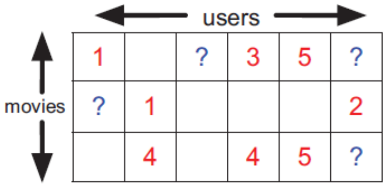


Fig. (a) A noisy image with an occluder. (b) An estimate of the underlying pixel intensities, based on a pairwise MRF model.

2. Collaborative filtering:

Collaborative filtering is the predictive process behind recommendation engines. Recommendation engines analyze information about users with similar tastes to assess the probability that a target individual will enjoy something, such as a video, a book or a product. Collaborative filtering is also known as social filtering.

Eg. Predicting which movies people will want to watch based on how they, and other people, have rated movies which they have already seen.


users
movies

1		?	3	5	?
?	1				2
	4		4	5	?

Fig. movie-rating data. Training data is in red, test data is denoted by ?, empty cells are unknown.

3. Market basket analysis

A set of items is referred to as an **itemset.** The occurrence or frequency of an itemset in the number of transactions that contain the itemset is called count of the itemset.

If the count of the itemset satisfy a minimum support threshold then the itemset is said to be a **frequent itemset**.

Frequent itemset mining leads to the discovery of associations and correlations among items in large transactional or relational data sets.

A typical example of frequent itemset mining is **market basket analysis**. This process analyzes customer buying habits by finding associations between the different items that customers place in their "shopping baskets".

Eg. 1. Bread => Jam and 2. Bread, Milk => Sugar

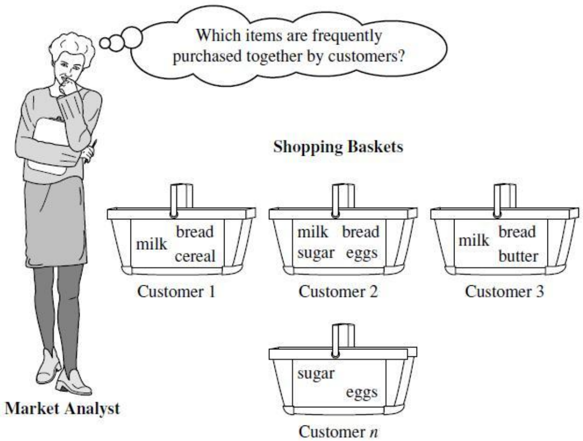


Fig. Market basket analysis.

The discovery of these associations can help retailers develop marketing strategies by gaining insight into which items are frequently purchased together by customers. For instance, if customers are buying milk, how likely are they to also buy bread.

Q) Explain about Partitioning cluster algorithms with an example.

Given a set of *n* objects, a partitioning method constructs *k* partitions of the data, where each partition represents a cluster and *k* <= *n*.

The basic partitioning methods typically adopt *exclusive cluster separation*. That is, each object must belong to exactly one group.

Given *k*, the *k-means* algorithm consists of four steps:

1. Select initial centroids at random.

2. Assign each object to the cluster with the nearest centroid.

3. Compute each centroid as the mean of the objects assigned to it.

4. Repeat previous 2 steps until no change.

Euclidean distance:

$$d(i,j)=\sqrt{(|x_{i_1}-x_{j_1}|^2+|x_{i_2}-x_{j_2}|^2+\ldots+|x_{i_p}-x_{j_p}|^2)}$$

Updating centroid point:

$$CP(x_1,x_2,\ldots,x_k)=\left(\frac{\sum_{i=1}^{k} x1st_i}{k},\frac{\sum_{i=1}^{k} x2nd_i}{k},\ldots,\frac{\sum_{i=1}^{k} xnth_i}{k}\right)$$

Algorithm: The k-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

Input:

k: the number of clusters,

D: a data set containing i objects.

Output: A set of k clusters.

Method:

(1) arbitrarily choose k objects from D as the initial cluster centers;

(2) repeat

(3) (re)assign each object to the cluster to which the object is the most similar,

based on the mean value of the objects in the cluster;

(4) update the cluster means, that is, calculate the mean value of the objects for

each cluster;

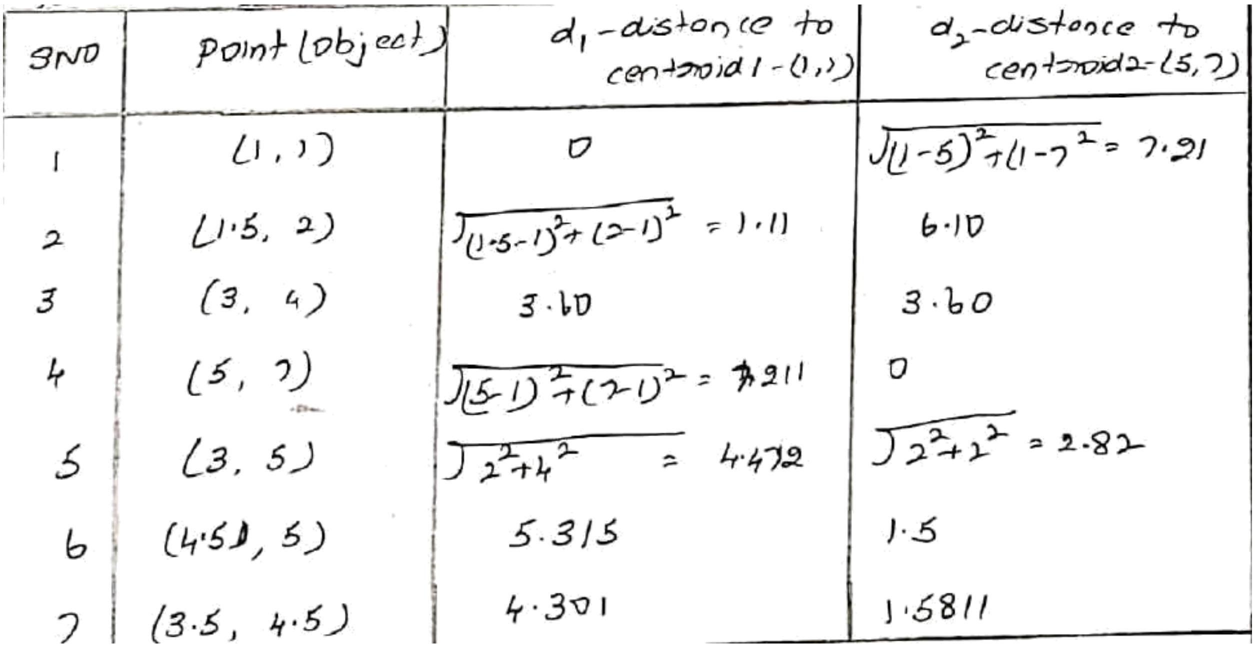
(5) until no change;

Eg. Consider the following 2-dimensional data objects:

Sl.No.	A	B
1	1	1
2	1.5	2
3	3	4
4	5	7
5	3.5	5
6	4.5	5
7	3.5	4.5

Let us start with random k=2 and the centroids be c1= (1,1) and c2= (5,7)

Now we find the Euclidean distance from all the given points to the centroids taken and is as shown below:



SNO	point (object)	d_1-distance to centroid 1-(1,1)	d_2-distance to centroid2-(5,7)
1	(1, 1)	0	$\sqrt{(1-5)^2+(1-7^2}$ = 7.21
2	(1.5, 2)	$\sqrt{(1.5-1)^2+(2-1)^2}$ = 1.11	6.10
3	(3, 4)	3.60	3.60
4	(5, 7)	$\sqrt{(5-1)^2+(7-1)^2}$ = 7.211	0
5	(3, 5)	$\sqrt{2^2+4^2}$ = 4.472	$\sqrt{2^2+2^2}$ = 2.82
6	(4.5, 5)	5.315	1.5
7	(3.5, 4.5)	4.301	1.5811

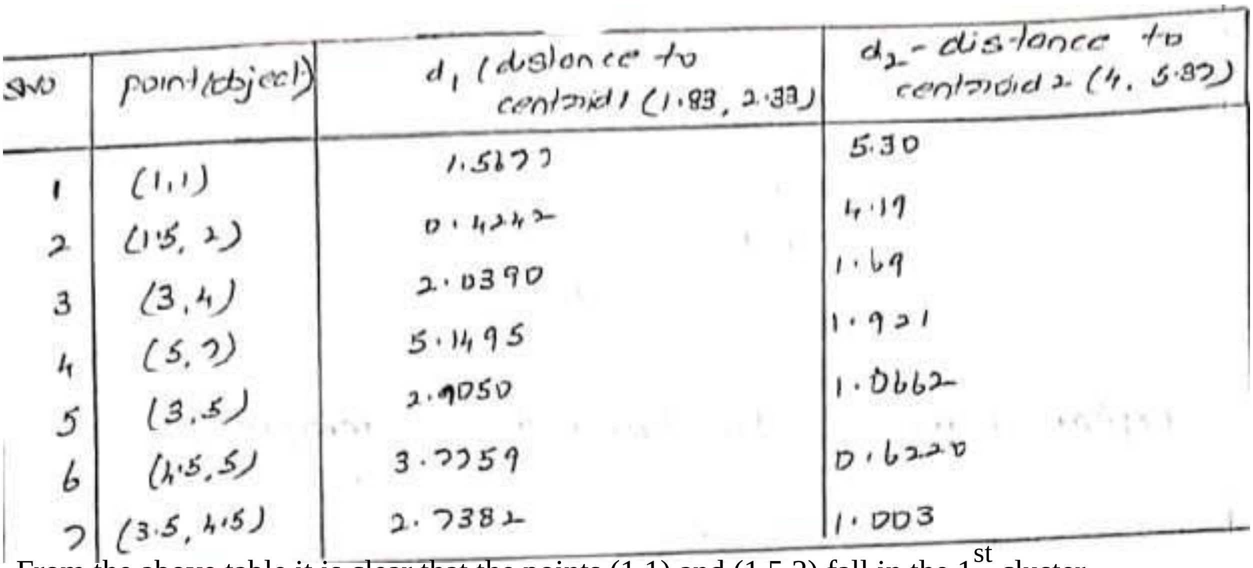
From the above table it is clear that the points (1,1), (1.5,2) and (3,4) falls in the 1st cluster and remaining points fall in the 2nd cluster.

So we find the new centroids as:

C1 = [(1+1.5+3)/3, (1+2+4)/3] = (1.83, 2.33)

C2 = [(5+3+4.5+3.5)/4, (7+5+5+4.5)/4] = (4, 5.37)

The distances from all the given points to the updated centroids are as follows:



SNO	point(object)	d_1 (distance to centroid1 (1.83, 2.33)	d_2 - distance to centroid2 (4, 5.37)
1	(1,1)	1.5177	5.30
2	(1.5, 2)	0.4242	4.19
3	(3,4)	2.0390	1.69
4	(5,7)	5.1495	1.921
5	(3,5)	2.9050	1.0662
6	(4.5,5)	3.7759	0.6220
7	(3.5, 4.5)	2.7382	1.003

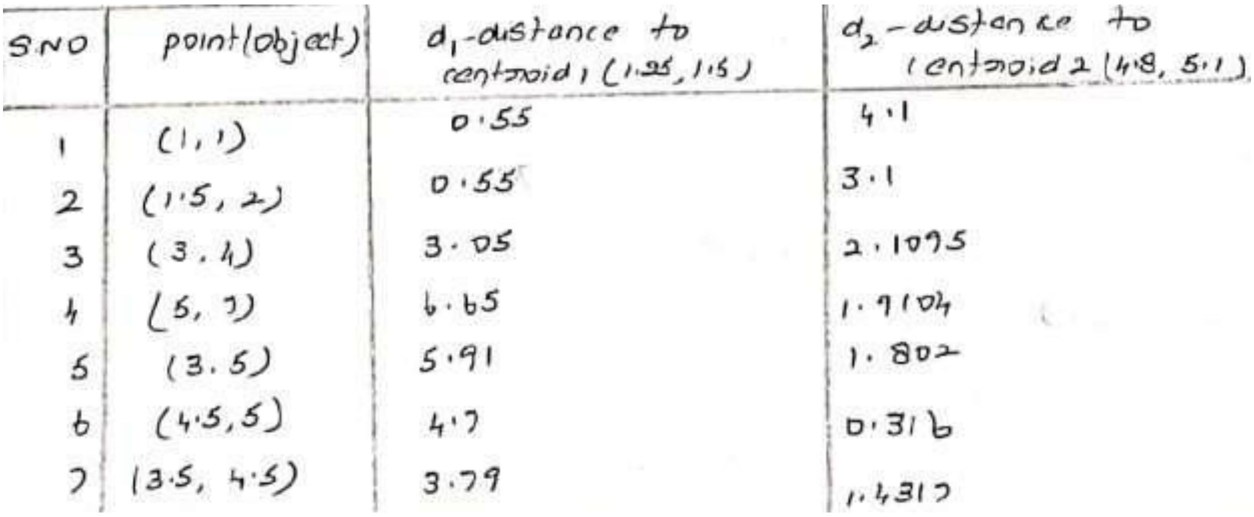
From the above table it is clear that the points (1,1) and (1.5,2) fall in the 1st cluster and the remaining into the 2nd cluster.

Since we get different points into the cluster compared to the previous step we continue the process and update the centroids as follows:

C1 = [(1+1.5)/2, (1+2)/2] = (1.25, 1.5)

C2 = [(3+5+3+4.5+3.5)/5, (4+7+5+5+4.5)/5] = (4.8, 5.1)

Now again we calculate the distance between all the given points and the updated C1 and C2 as follows.



S.NO	point(object)	d_1-distance to centroid1 (1.25, 1.5)	d_2-distance to centroid2 (4.8, 5.1)
1	(1,1)	0.55	4.1
2	(1.5, 2)	0.55	3.1
3	(3.4)	3.05	2.1075
4	(5, 7)	6.65	1.9104
5	(3.5)	5.91	1.802
6	(4.5,5)	4.7	0.316
7	(3.5, 4.5)	3.79	1.4317

From the above table it is evident that the points in the 1st cluster and 2nd cluster are same as the previous step.

So we stop the iterations and the final cluster of points are cluster1 = {(1,1) , (1.5,2) } and cluster2 = {(3,4),(5,7),(3,5),(4.5,5),(3.5,4.5)}

Strengths

–Relatively efficient: O(tkn), where n is # objects, k is # clusters, and t is # iterations. Normally, k, t << n.

–Often terminates at a local optimum. The global optimum may be found using techniques such as simulated annealing and genetic algorithms

Weaknesses

–Applicable only when mean is defined.

–Need to specify k, the number of clusters, in advance

–Trouble with noisy data and outliers

–Not suitable to discover clusters with non-convex shapes.

Q) Explain about Parametric vs non-parametric models in Machine Learning.

A learning model that summarizes data with a set of parameters of fixed size (independent of the number of training examples) is called a **parametric model.** No matter how much data you throw at a parametric model, it won't change its mind about how many parameters it needs.

The algorithms involve two steps:

1. Select a form for the function.
2. Learn the coefficients for the function from the training data.

An easy to understand functional form for the mapping function is a line, as is used in linear regression:

$$b0 + b1*x1 + b2*x2 = 0$$

Where b0, b1 and b2 are the coefficients of the line that control the intercept and slope, and x1 and x2 are two input variables.

Some more examples of popular nonparametric machine learning algorithms are:

- Naive Bayes
- Logistic Regression

Benefits of Parametric Machine Learning Algorithms:

- Simpler: These methods are easier to understand and interpret results.
- Speed: Parametric models are very fast to learn from data.

- Less Data: They do not require as much training data and can work well even if the fit to the data is not perfect.

Limitations of Parametric Machine Learning Algorithms:

- Constrained: By choosing a functional form these methods are highly constrained to the specified form.
- Limited Complexity: The methods are more suited to simpler problems.
- Poor Fit: In practice the methods are unlikely to match the underlying mapping function.

Nonparametric methods seek to best fit the training data in constructing the mapping function, while maintaining some ability to generalize to unseen data. As such, they are able to fit a large number of functional forms.

An easy to understand nonparametric model is the k-nearest neighbors algorithm that makes predictions based on the k most similar training patterns for a new data instance. The method does not assume anything about the form of the mapping function other than patterns that are close likely to have a similar output variable.

Some popular nonparametric machine learning algorithms are:

- k-Nearest Neighbors
- Decision Trees like CART and C4.5
- Support Vector Machines

Benefits of Nonparametric Machine Learning Algorithms:

- Flexibility: Capable of fitting a large number of functional forms.
- Power: No assumptions (or weak assumptions) about the underlying function.
- Performance: Can result in higher performance models for prediction.

Limitations of Nonparametric Machine Learning Algorithms:

- More data: Require a lot more training data to estimate the mapping function.
- Slower: A lot slower to train as they often have far more parameters to train.
- Overfitting: More of a risk to overfit the training data and it is harder to explain why specific predictions are made.

Q) Briefly explain Semi supervised learning.

Semi-supervised machine learning is a combination of supervised and unsupervised machine learning methods.

With more common supervised machine learning methods, you train a machine learning algorithm on a "labeled" dataset in which each record includes the outcome information.

This allows the algorithm to deduce patterns and identify relationships between your target variable and the rest of the dataset based on information it already has.

In contrast, unsupervised machine learning algorithms learn from a dataset without the outcome variable.

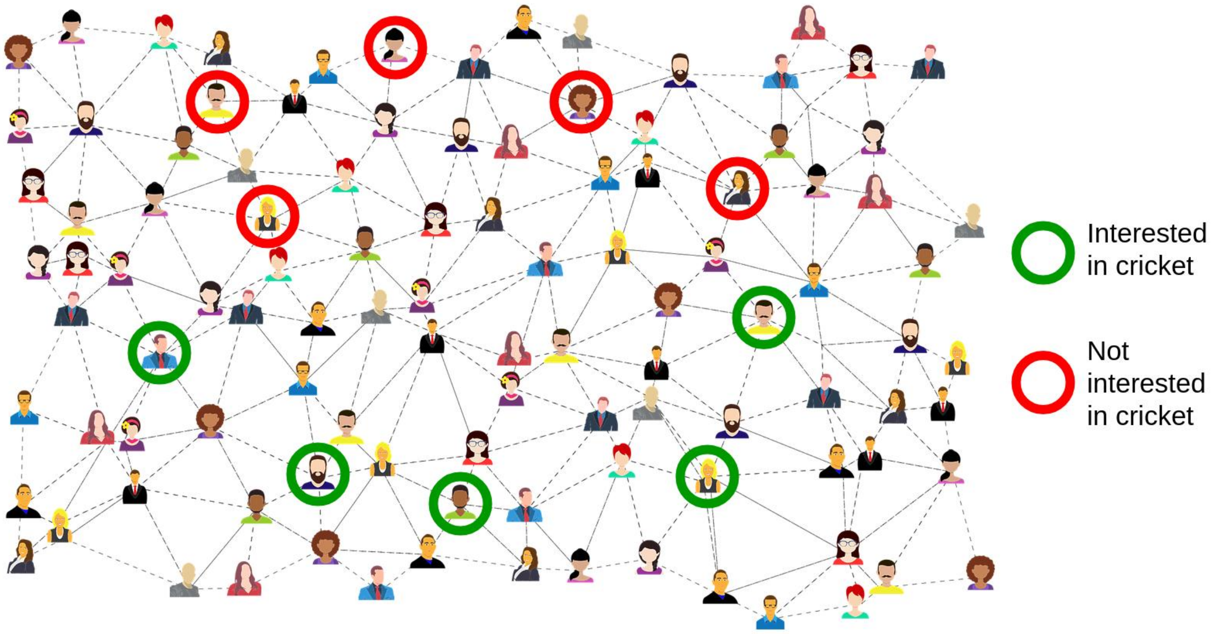
In semi-supervised learning, an algorithm learns from a dataset that includes both labeled and unlabeled data, usually mostly unlabeled.

Need: When you don't have enough labeled data to produce an accurate model and you don't have the ability or resources to get more data, you can use semi-supervised techniques to increase the size of your training data.

Label Propagation Algorithm (LPA) is an iterative algorithm where we assign labels to unlabelled points by propagating labels through the dataset.

Eg.

Assume that we have a network of people as given below with two label classes "*interested in cricket*" and "*not interested in cricket*". So the question is, can we predict whether the remaining people are interested in cricket or not?



For LPA to work in this case, we have to make an assumption; **an edge connecting two nodes carry a notion of similarity**. i.e. if two people are connected together, that means that it is highly likely that these two people share the same interests.

Consider the sample graph given in below, where we have 2 label classes (red and green) and 4 nodes coloured (2 for each class). We want to predict the label of node 4.

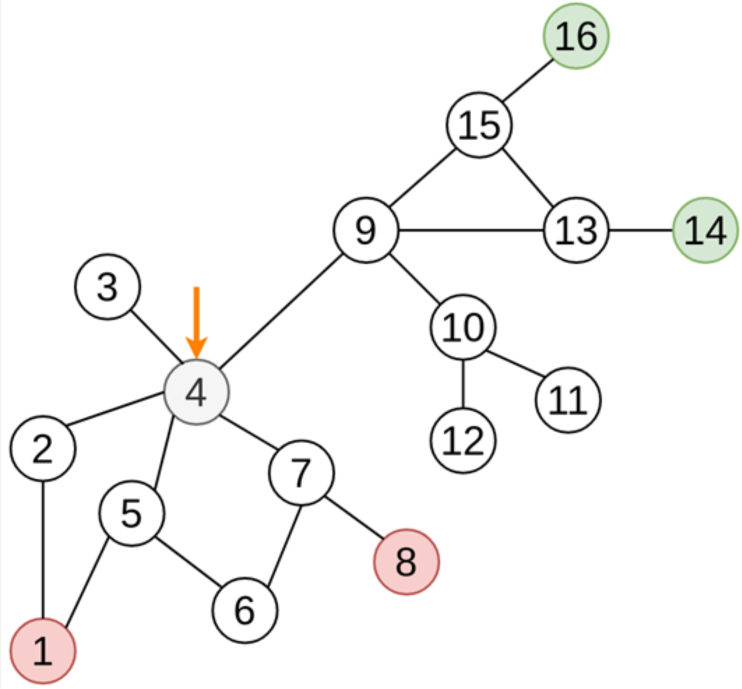


Fig . Sample graph

We can walk randomly in the graph, starting from node 4 until we meet any labelled node. When we hit a labelled node, we stop the walk. Hence, these labelled nodes are known as **absorbing states**. Let's consider all the possible walks from node 4. Out of all the possible walk, the following walks will end in a green node.

1. 4 → 9 → 15 → 16
2. 4 → 9 → 13 → 14
3. 4 → 9 → 13 → 15 → 16
4. 4 → 9 → 15 → 13 → 14

The following walks will end in a red node.

1. 4 → 7 → 8
2. 4 → 7 → 6 → 5 → 1
3. 4 → 5 → 1
4. 4 → 5 → 6 → 7 → 8
5. 4 → 2 → 1

Based on all the possible random walks starting from node 4, we can see that the majority of the walks end in a red node. So, we can colour node 4 in red. This is the basic intuition behind LPA.

Q) What is Reinforcement Learning? Explain about agent in RL.

Reinforcement learning (**RL**) is an area of machine learning concerned with how software agent have to take actions in an environment in order to maximize the notion of cumulative reward.

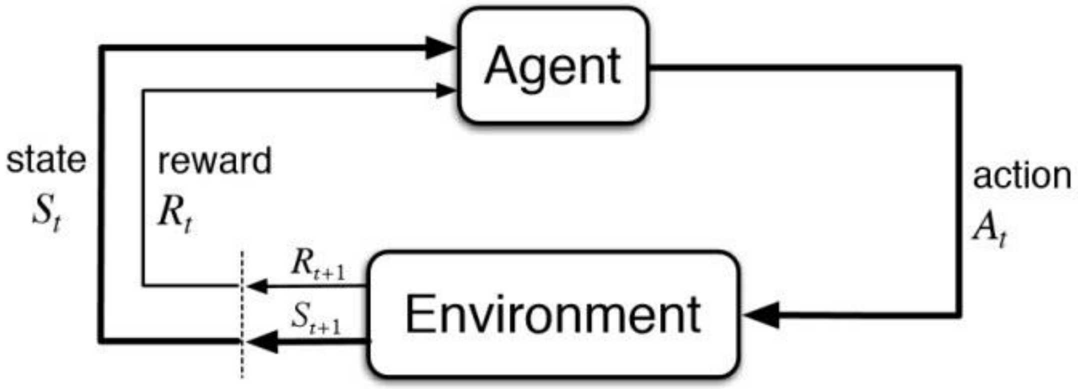


Fig. Illustrating of RL

Agent

Agents are the software programs that make intelligent decisions and they are basically learners in RL.

Agents take action by interacting with the environment and they receive rewards based on their actions.

Eg. Mario navigating in a video game.

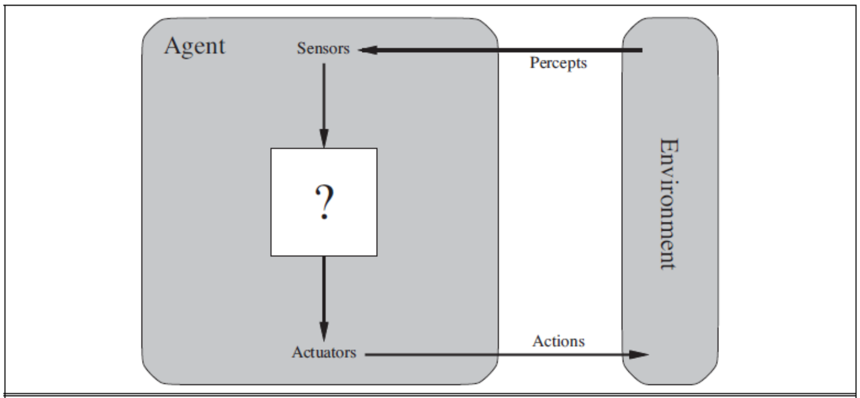


Figure 2.1 Agents interact with environments through sensors and actuators.

Eg. 1: A human agent has eyes, ears, and other organs for sensors and hands, legs, vocal tract, and so on for actuators.

Eg. 2: A robotic agent might have cameras and infrared range finders for sensors and various motors for actuators.

Eg. 3: A software agent receives keystrokes, file contents, and network packets as sensory inputs and acts on the environment by displaying on the screen, writing files, and sending network packets.

Eg. 4: Vaccum cleaner that cleans blocks A & B. A reward of +1 will be given if it sucks dust and a reward of 0 will be given if in same block and -1 if moves from one block to another.

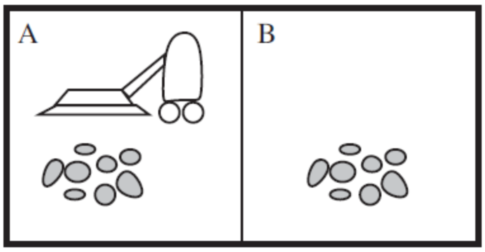
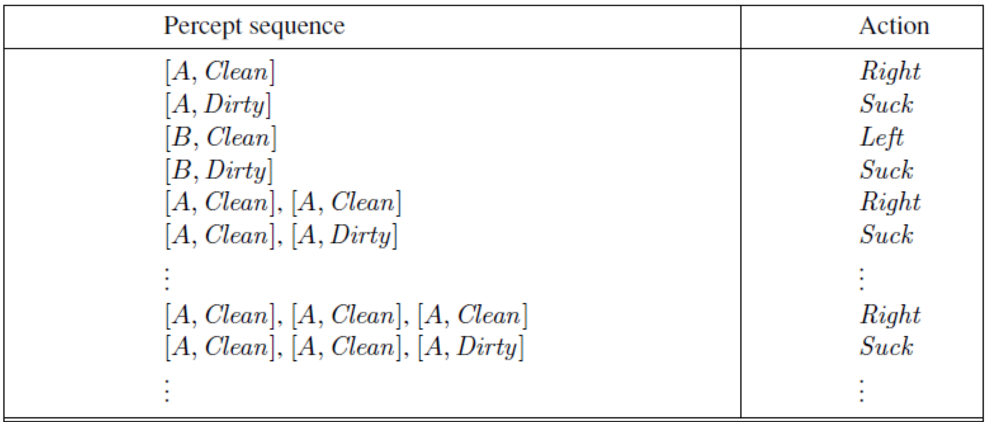


Figure 2.2 A vacuum-cleaner world with just two locations.



Percept sequence	Action
[*A*, *Clean*]	*Right*
[*A*, *Dirty*]	*Suck*
[*B*, *Clean*]	*Left*
[*B*, *Dirty*]	*Suck*
[*A*, *Clean*], [*A*, *Clean*]	*Right*
[*A*, *Clean*], [*A*, *Dirty*]	*Suck*
⋮	⋮
[*A*, *Clean*], [*A*, *Clean*], [*A*, *Clean*]	*Right*
[*A*, *Clean*], [*A*, *Clean*], [*A*, *Dirty*]	*Suck*
⋮	⋮

Figure 2.3 Partial tabulation of a simple agent function for the vacuum-cleaner world shown in Figure 2.2.

The elements of RL are:

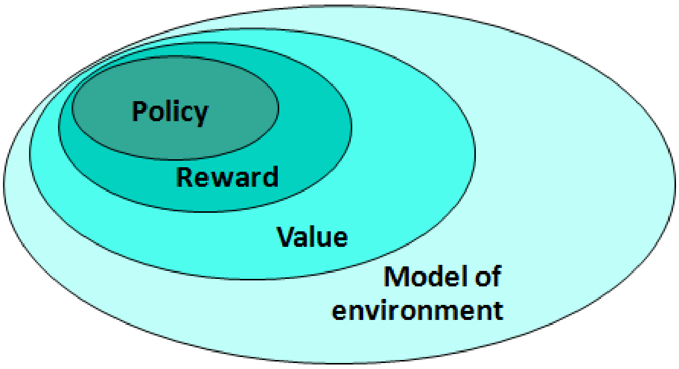


Fig. Elements of RL

Policy function

A policy defines the agent's behavior in an environment. The way in which the agent decides which action to perform depends on the policy.
A policy is often denoted by the symbol **π**. A policy can be in the form of a lookup table or a complex search process.

Eg. If you want to reach your office from home; there will be different routes to reach your office, and some routes are shortcuts, while some routes are long. These routes are called policies because they represent the way in which we choose to perform an action to reach our goal.

Value function

- A value function denotes how good it is for an agent to be in a particular state. It is dependent on the policy and is often denoted by v(s). It is equal to the total expected reward received by the agent starting from the initial state.
- There can be several value functions. The optimal value function is the one that has the highest value for all the states compared to other value functions.
- Similarly, an optimal policy is the one that has the optimal value function.

Model

Model is the agent's representation of an environment. The learning can be of two types—

In **model-based learning**, the agent exploits previously learned information to accomplish a task. Eg. Epsilon-Greedy approach, random selection approach.

whereas in **model-free learning**, the agent simply relies on a trial-and-error experience for performing the right action. Eg. Q-learning or policy gradient.

Eg. If you want to reach your office from home faster. In model-based learning, you simply use a previously learned experience (map) to reach the office faster, whereas in model-free learning you will not use a previous experience and will try all different routes and choose the faster one.

Reward Function Engineering determines the rewards for actions.

Q) Explain the Nature of Environment in RL.

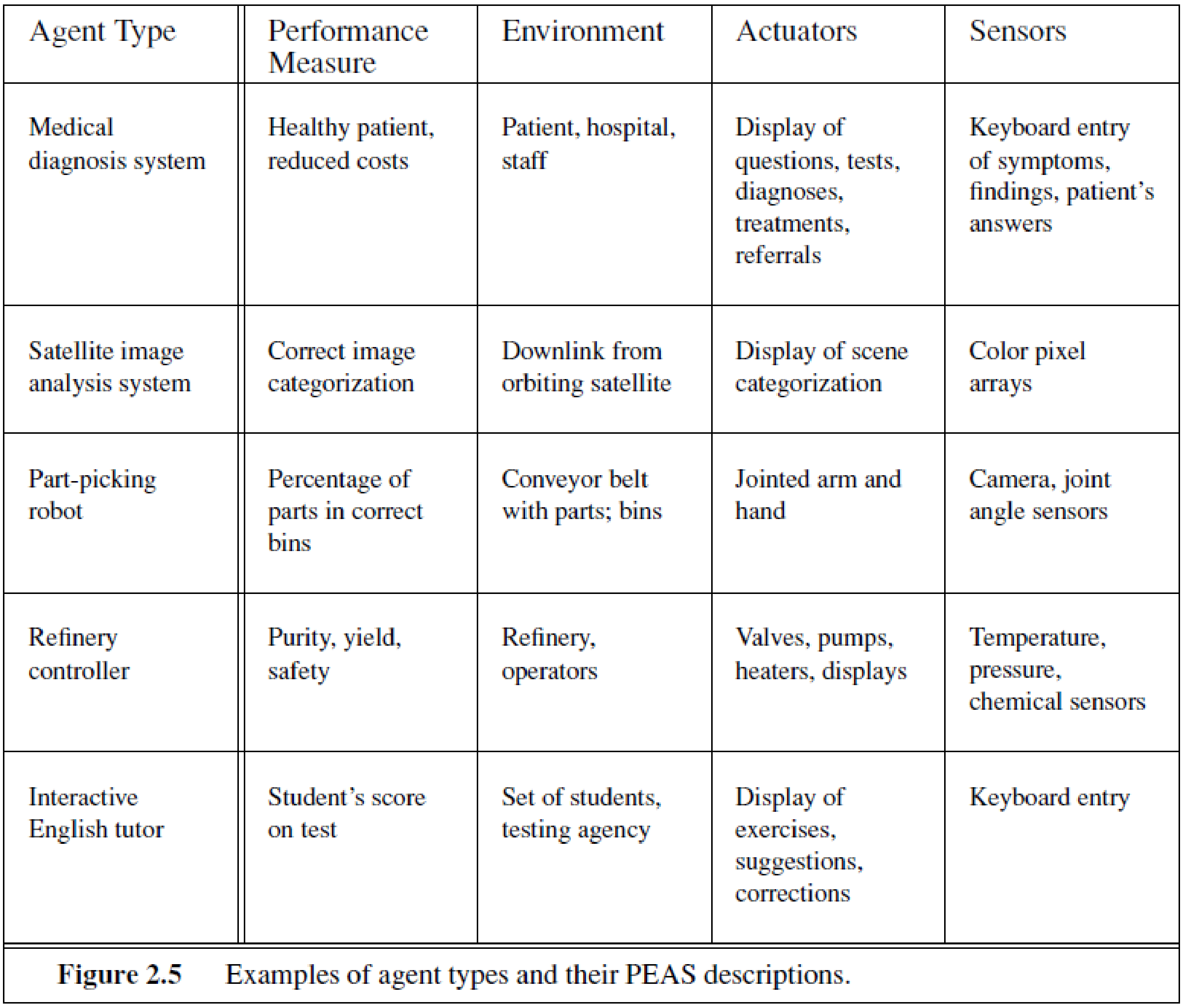
Task environments, which are essentially the "problems" to which rational agents are the "solutions." we had to specify the performance measure, the environment, and the agent's actuators and sensors. We group all these under the heading of the **task environment**.

For the acronymically minded, we call this as the **PEAS** (**Performance, Environment, Actuators, Sensors**) description. In designing an agent, the first step must always be to specify the task environment as fully as possible.

Eg.: an automated taxi driver.

The full driving task is extremely open-ended. There is no limit to the novel combinations of circumstances that can arise.

Type Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits	Roads, other traffic, pedestrians, customers	Steering, accelerator, brake, signal, horn, display	Cameras, sonar, speedometer, GPS, odometer, accelerometer, engine sensors, keyboard.



Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments, referrals	Keyboard entry of symptoms, findings, patient's answers
Satellite image analysis system	Correct image categorization	Downlink from orbiting satellite	Display of scene categorization	Color pixel arrays
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts; bins	Jointed arm and hand	Camera, joint angle sensors
Refinery controller	Purity, yield, safety	Refinery, operators	Valves, pumps, heaters, displays	Temperature, pressure, chemical sensors
Interactive English tutor	Student's score on test	Set of students, testing agency	Display of exercises, suggestions, corrections	Keyboard entry

Figure 2.5 Examples of agent types and their PEAS descriptions.

Q) Explain about different types of Enviornment in RL.

Fully observable vs. partially Fully observable:

If an agent's sensors give it access to the complete state of the environment at each point in time, then we says that the task environment is fully observable.

An environment might be partially observable because of noisy and inaccurate sensors or because parts of the state are simply missing from the sensor data.

Eg. a vacuum agent with only a local dirt sensor cannot tell whether there is dirt in other squares.

Single agent Vs. Multi agent:

An environment in which only single agent exists is called a Single Agent Environment.

Eg. Crossword puzzle.

An environment in which more than one single agent exists is called a Multi-agent Environment.

Eg. Chess is a competitive multiagent environment because both the agents try to maximize their performance (win). In the taxi-driving environment, on the other hand, avoiding collisions maximizes the performance measure of all agents, so it is a partially cooperative multiagent environment.

Deterministic Vs. Stochastic:
If the next state of the environment is completely determined by the current state and the action executed by the agent, then we say the environment is deterministic; otherwise, it is stochastic. If the environment is partially observable, however, then it could appear to be stochastic.
Eg. The vacuum world as we described it is deterministic, but variations can include stochastic elements such as randomly appearing dirt and an unreliable suction mechanism.
Taxi driving is clearly stochastic in this sense, because one can never predict the behavior of traffic exactly. We say an environment is uncertain if it is not fully observable or not deterministic.

Episodic Vs. Sequential:
In an episodic task environment, the agent's experience is divided into atomic episodes. In each episode the agent receives a percept and then performs a single action. Crucially, the next episode does not depend on the actions taken in previous episodes. Many classification tasks are episodic. Episodic environments are much simpler than sequential environments because the agent does not need to think ahead.
Eg. an agent that has to spot defective parts on an assembly line bases each decision on the current part, regardless of previous decisions; moreover, the current decision doesn't affect whether the next part is defective.

In sequential environments the current decision could affect all future decisions. Eg. Chess and taxi driving.

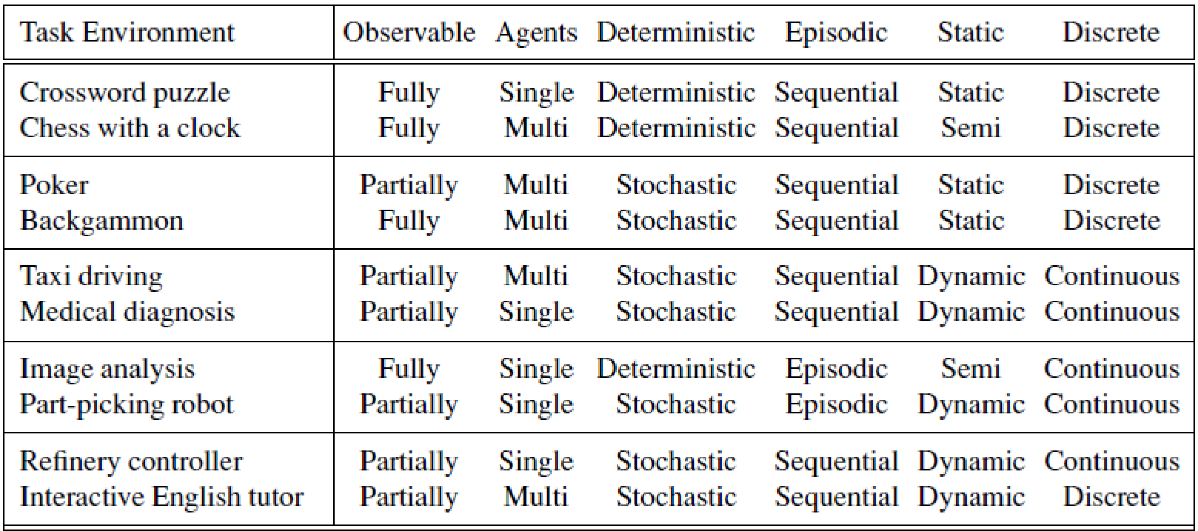
Static vs. Dynamic:
If the environment can change while an agent is deliberating, then we say the environment is dynamic for that agent; otherwise, it is static.
Eg. Taxi Driving is dynamic
Static environments are easy to deal with because the agent need not keep looking at the world while it is deciding on an action, nor need it worry about the passage of time.
Eg. Crossword puzzle is static

If the environment itself does not change with the passage of time but the agent's performance score does, then we say the environment is semidynamic.
Eg. Chess, when played with a clock, is semidynamic.

Discrete vs. Continuous:
The discrete/continuous distinction applies to the state of the environment, to the way time is handled, and to the percepts and actions of the agent.
Eg. Input from digital cameras is discrete; Taxi driving is a continuous-state

Known vs. Unknown:

This distinction refers not to the environment itself but to the agent's state of knowledge about the environment. In a known environment the outcomes (or outcome probabilities if the environment is stochastic) for all actions are given. Obviously, if the environment is unknown, the agent will have to learn how it works in order to make good decisions.



Task Environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with a clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete
Backgammon	Fully	Multi	Stochastic	Sequential	Static	Discrete
Taxi driving	Partially	Multi	Stochastic	Sequential	Dynamic	Continuous
Medical diagnosis	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Image analysis	Fully	Single	Deterministic	Episodic	Semi	Continuous
Part-picking robot	Partially	Single	Stochastic	Episodic	Dynamic	Continuous
Refinery controller	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Interactive English tutor	Partially	Multi	Stochastic	Sequential	Dynamic	Discrete

Figure 2.6 Examples of task environments and their characteristics.

Q) Differentiate Supervised, Unsupervised and Reinforcement Learnings.

Criteria	**Supervised ML**	**Unsupervised ML**	**Reinforcement ML**
Definition	Learns by using labelled data	Trained using unlabelled data without any guidance.	Works on interacting with the environment
Type of data	Labelled data	Unlabelled data	No - predefined data
Type of problems	Regression and classification	Association and Clustering	Exploitation or Exploration
Supervision	Extra supervision	No supervision	No supervision
Algorithms	Linear Regression, Logistic Regression, SVM, KNN etc.	K - Means, Hierarchical clustering, Apriori, etc.	Q - Learning, SARSA, etc.
Aim	Calculate outcomes	Discover underlying patterns	Learn a series of action

Application	Risk Evaluation, Forecast Sales	Recommendation System, Anomaly Detection	Self Driving Cars, Gaming, Healthcare

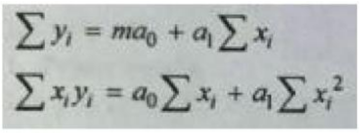
Q) Fit a straight line to the following data

x	0	1	2	3	4
y	1	1.8	3.3	4.5	6.3

Sol:

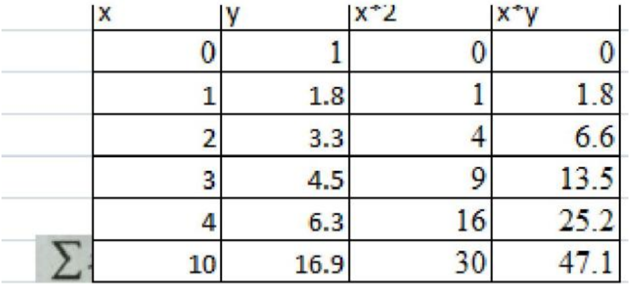
Let the required linear equation be

y=a1x+a0. We know,



$$\sum y_i = ma_0 + a_1 \sum x_i$$
$$\sum x_i y_i = a_0 \sum x_i + a_1 \sum x_i^2$$

Here m = 5



	x	y	x^2	x*y
	0	1	0	0
	1	1.8	1	1.8
	2	3.3	4	6.6
	3	4.5	9	13.5
	4	6.3	16	25.2
$\sum$	10	16.9	30	47.1

There fore, we get equations as

16.9=5a0+10a1

47.1=10a0+30a1

By solving above two equations we get a0=0.72 a1=1.33
Therefore, required equation is y = 1.33x + 0.72.

Q) Fit a curve of the type $y=ae^{bx}$ to the following data

x	0	1	2	3
y	1.05	2.1	3.85	8.3

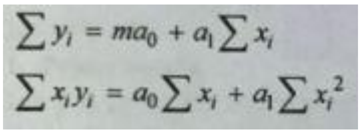
The given relation is $y=ae^{bx}$

Taking logarithms on both sides we obtain log y = log a + bx----(1)

Let log y = Y and x = X, log a = a0 and b = a1

The eq(1) takes the form Y = a0 + a1X, which is a straight line.

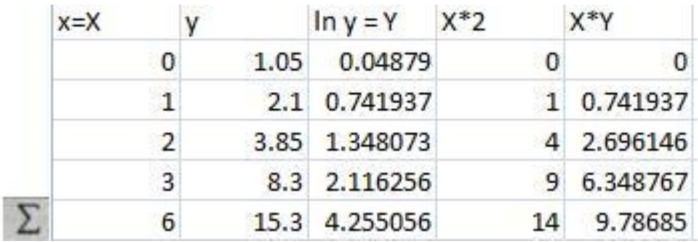
We know,



$$\sum y_i = ma_0 + a_1 \sum x_i$$

$$\sum x_i y_i = a_0 \sum x_i + a_1 \sum x_i^2$$

And m=4



	x=X	y	In y = Y	X*2	X*Y
	0	1.05	0.04879	0	0
	1	2.1	0.741937	1	0.741937
	2	3.85	1.348073	4	2.696146
	3	8.3	2.116256	9	6.348767
Σ	6	15.3	4.255056	14	9.78685

4.255 = 4a0 + 6a1   and 9.786 = 6a0 + 14a1

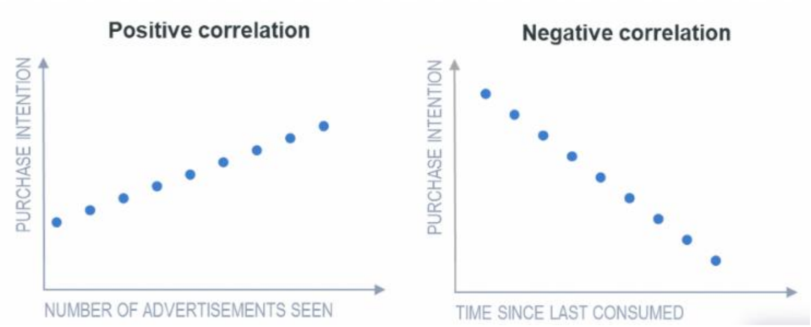
By solving above equations we get, a0 = 0.0416, a1 = 0.6811

Q) Define Correlation.

Correlation is a term that is a measure of the strength of a linear relationship between two quantitative variables (e.g., height, weight).

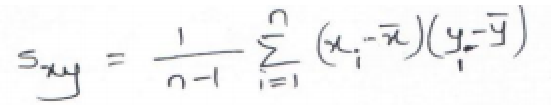
Positive correlation is a relationship between two variables in which both variables move in the same direction and viceversa is Negative correlation.

eg. postivie correlation: the more you exercise, the more calories you will burn.



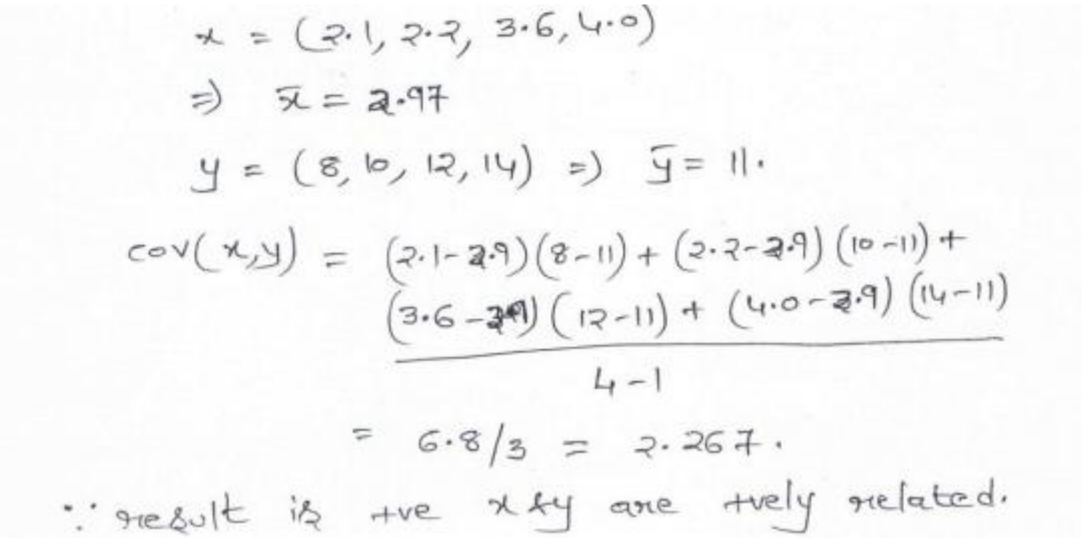
Q) Define Covariance.

The covariance of two variables x and y in a dataset measures how they are linearly related. A positive covariance would indicate a positive linear relationship between the variables and negative covariance indicates opposite.



$$s_{xy} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

Eg. Find the covariance between x and y where x = (2.1, 2.2, 3.6, 4.0) and y = (8, 10, 12, 14).



$$x = (2.1, 2.2, 3.6, 4.0)$$

$$\Rightarrow \bar{x} = 2.97$$

$$y = (8, 10, 12, 14) \Rightarrow \bar{y} = 11.$$

$$\text{cov}(x,y) = \frac{(2.1-2.9)(8-11) + (2.2-2.9)(10-11) + (3.6-2.9)(12-11) + (4.0-2.9)(14-11)}{4-1}$$

$$= 6.8/3 = 2.267.$$

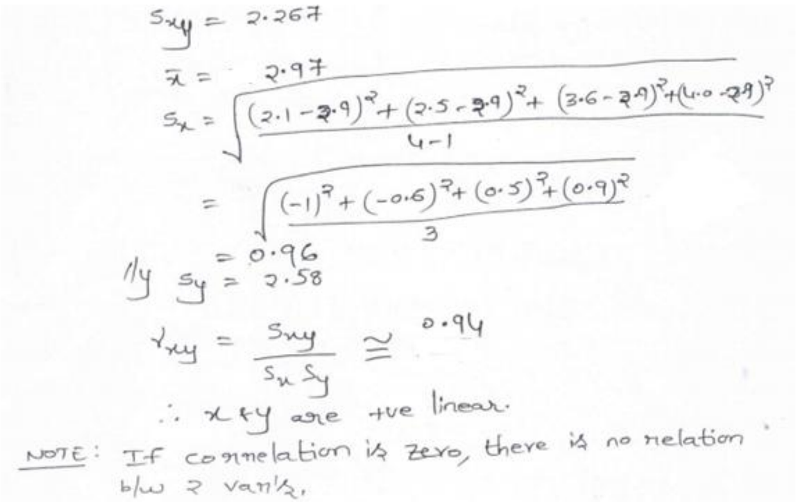
$\because$ result is +ve x & y are +vely related.

Q) Explain in brief about correlation coefficient.

The correlation coefficient of two variables in a dataset equals to their covariance divided by product of their individual standard deviations.

The range of correlation coefficient is -1 to +1.

Eg. Find the correlation coefficient for above x & y.



$$s_{xy} = 2.267$$

$$\bar{x} = 2.97$$

$$s_x = \sqrt{\frac{(2.1-2.9)^2 + (2.5-2.9)^2 + (3.6-2.9)^2 + (4.0-2.9)^2}{4-1}}$$

$$= \sqrt{\frac{(-1)^2 + (-0.6)^2 + (0.5)^2 + (0.9)^2}{3}}$$

$$= 0.96$$

$$\text{llly } s_y = 2.58$$

$$r_{xy} = \frac{s_{xy}}{s_x s_y} \cong 0.94$$

$\therefore$ x & y are +ve linear.

<u>NOTE</u>: If correlation is zero, there is no relation b/w 2 vari's.