

Machine Learning

Unit 1

HUMAN LEARNING

- In cognitive science, learning is typically referred to as the process of gaining information through observation.
- Why do we need to learn?

TYPES OF HUMAN LEARNING

- Thinking intuitively, human learning happens in one of the three ways –
 - (1) either somebody who is an expert in the subject directly teaches us,
 - (2) we build our own notion indirectly based on what we have learnt from the expert in the past,
 - (3) we do it ourselves, maybe after multiple attempts, some being unsuccessful.

WHAT IS MACHINE LEARNING?

- Tom M.Mitchell, Professor of Machine Learning Department, School of Computer Science, Carnegie Mellon University has defined machine learning as
‘ A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.’
- In the context of the learning to play checkers,
E represents the experience of playing the game,
T represents the task of playing checkers and
P is the performance measure indicated by the percentage of games won by the player.

MACHINE LEARNING

- Machine learning is programming computers to optimize a performance criterion using example data or past experience.
- There is no need to “learn” to calculate payroll
- Learning is used when:
 - Human expertise does not exist (navigating on Mars),
 - Humans are unable to explain their expertise (speech recognition)
 - Solution changes in time (routing on a computer network)
 - Solution needs to be adapted to particular cases (user biometrics)

A LITTLE HISTORY

- 1946: First computer called ENIAC to perform numerical computations
- 1950: Alan Turing proposes the Turing test. [Can machines think?](#)
- 1952: First game playing program for checkers by Arthur Samuel at IBM. Knowledge based systems such as ELIZA and MYCIN.
- 1957: Perceptron developed by Frank Roseblatt. Can be combined to form a neural network.
- Early 1990's: Statistical learning theory. Emphasize learning from data instead of rule-based inference.
- Current status: Used widely in industry, combination of various approaches but data-driven is prevalent.

1950	Alan Turing proposes “learning machine”
1952	Arthur Samuel developed first machine learning program that could play Checkers
1957	Frank Rosenblatt designed the first neural network program simulating human brain
1967	Nearest neighbour algorithm created – start of basic pattern recognition
1979	Stanford University students develop first self – driving cart that can navigate and avoid obstacles in a room
1982	Recurrent Neural Network developed
1989	- Reinforcement Learning conceptualized - Beginning of commercialization of Machine Learning
1995	Random Forest and Support Vector machine algorithms developed
1997	IBM's Deep Blue beats the world chess champion Gary Kasparov
TH	
2006	- First machine learning competition launched by Netflix - Geoffrey Hinton conceptualizes Deep Learning
2010	Kaggle, a website for machine learning competitions, launched
2011	IBM's Watson beats two human champions in Jeopardy
2016	Google's AlphaGo program beats unhandicapped professional human player

How do machines learn?

- The basic machine learning process can be divided into three parts.
 1. Data Input: Past data or information is utilized as a basis for future decision-making
 2. Abstraction: The input data is represented in a broader way through the underlying algorithm
 3. Generalization: The abstracted representation is generalized to form a framework for making decisions



Abstraction

- During the machine learning process, knowledge is fed in the form of input data.
- The data cannot be used in the original shape and form.
- It has to be in the form of summarized knowledge representation of the raw data, called as model.
- The model may be in any one of the following forms
 - Computational blocks like if/else rules
 - Mathematical equations
 - Specific data structures like trees or graphs
 - Logical groupings of similar observations

- The choice of the model used to solve a specific learning problem is a human task.
- The decision related to the choice of model is taken based on multiple aspects, some of which are :
 - The type of problem to be solved: Whether the problem is related to forecast or prediction, analysis of trend, understanding the different segments or groups of objects, etc.
 - Nature of the input data: How exhaustive the input data is, whether the data has no values for many fields, the data types, etc.
 - Domain of the problem: If the problem is in a business critical domain with a high rate of data input and need for immediate inference, e.g. fraud detection problem in banking domain.

Fitting a Model

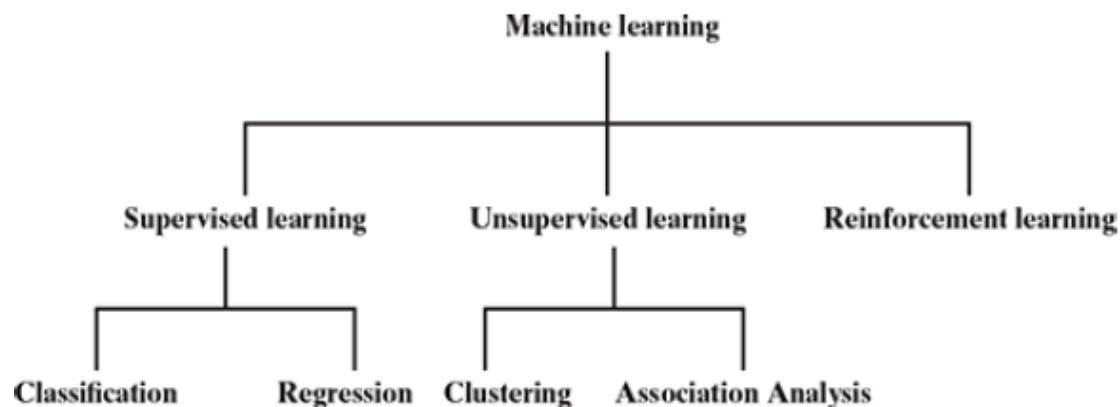
- After the model is chosen, the next task is to fit the model based on the input data.
- In a case where the model is represented by a mathematical equation, say ' $y = c_1 + c_2x$ ', based on the input data, we have to find out the values of c_1 and c_2 .
- So, fitting the model, in this case, means finding the values of the unknown coefficients or constants of the equation or the model.
- This process of fitting the model based on the input data is known as training.
- Also, the input data based on which the model is being finalized is known as training data.

Generalization

- Generalization is the ability of a trained model to accurately make predictions on new, unseen data.
- The purpose of generalization is to equip the model to understand the patterns and relationships within its training data and apply them to previously unseen examples from within the same distribution as the training set.

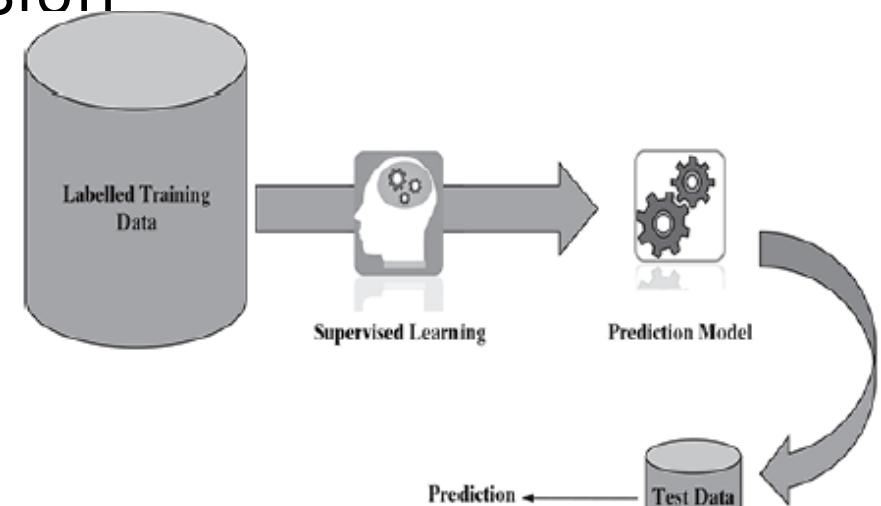
TYPES OF MACHINE LEARNING

- Machine learning can be classified into three broad categories:
 1. Supervised learning – Also called predictive learning. A machine predicts the class of unknown objects based on prior class-related information of similar objects.
 2. Unsupervised learning – Also called descriptive learning. A machine finds patterns in unknown objects by grouping similar objects together.
 3. Reinforcement learning – A machine learns to act on its own to achieve the given goals.



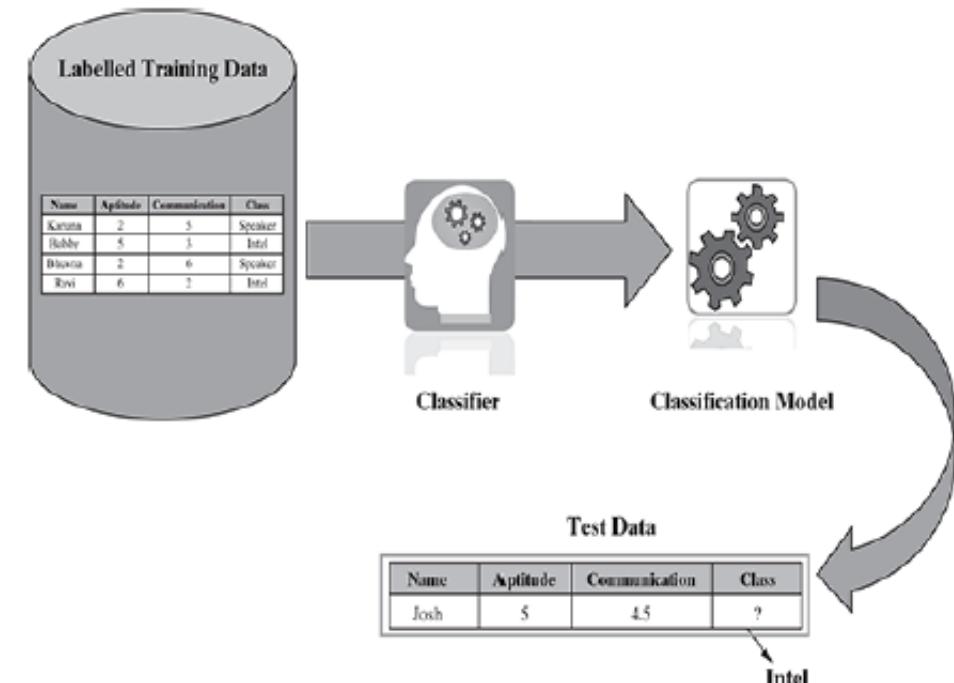
Supervised learning

- The major motivation of supervised learning is to learn from past information
 - Learning (training): Learn a model using the training data (Basic input)
 - Testing: Test the model using unseen test data to assess the model accuracy
- There are two areas of supervised learning, i.e. classification and regression



Types of Supervised Learning – Classification

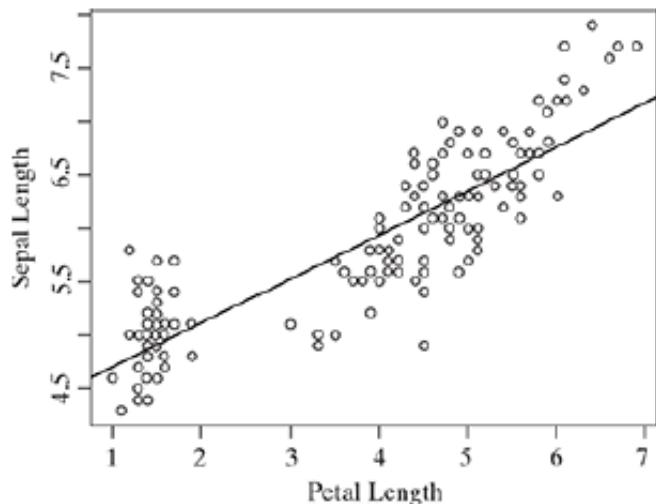
- When we are trying to predict a categorical or nominal variable, the problem is known as a **classification problem**.
- Machine Learning Algorithms for classification are Naïve Bayes, Decision tree, and k-Nearest Neighbour algorithms
- Some typical classification problems include:
 1. Image classification
 2. Prediction of disease
 3. Win–loss prediction of games
 4. Prediction of natural calamity like earthquake, flood, etc.
 5. Recognition of handwriting



Types of Supervised Learning – Regression

- In linear regression, the objective is to predict numerical features like real estate or stock price, temperature, marks in an examination, sales revenue, etc
- A typical linear regression model can be represented in the form – $y=a+bx$
where 'x' is the predictor variable and 'y' is the target variable.

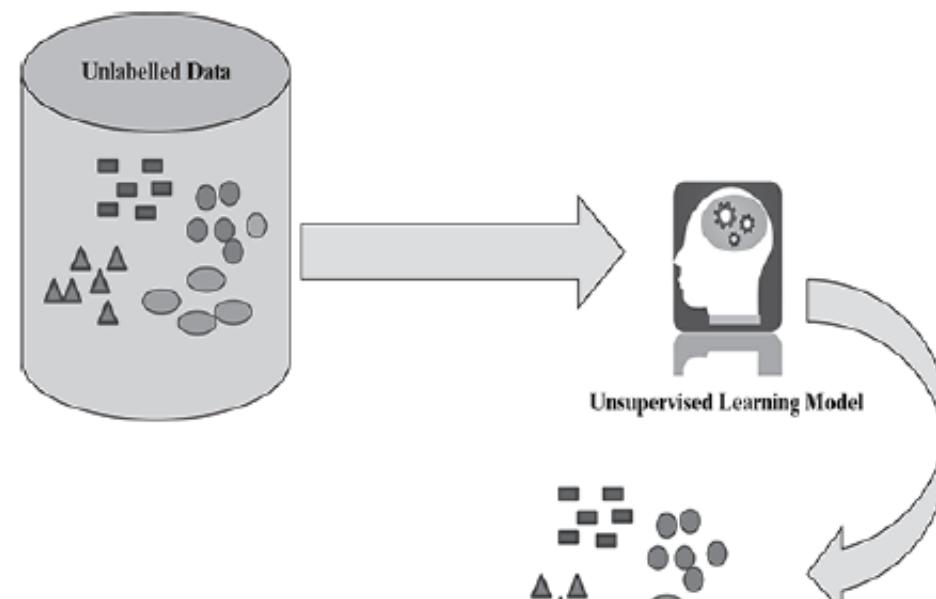
Typical applications of regression can be seen in



1. Demand forecasting in retail
2. Sales prediction for managers
3. Price prediction in real estate
4. Weather forecast
5. Skill demand forecast in job market

Unsupervised learning

- Unlike supervised learning, in unsupervised learning, there is no labelled training data to learn from and no prediction to be made.
- In unsupervised learning, the objective is to take a dataset as input and try to find natural groupings or **patterns** within the data elements or records.



Types of Unsupervised Learning

- Clustering is the main type of unsupervised learning. It intends to group or organize similar objects together.
- One more variant of unsupervised learning is **association analysis**. In association analysis, the association between data elements is identified

TransID	Items Bought
1	{Butter, Bread}
2	{Diaper, Bread, Milk, Beer}
3	{Milk, Chicken, Beer, Diaper}
4	{Bread, Diaper, Chicken, Egg}
5	{Diaper, Beer, Cookies, Ice Cream}
...	...

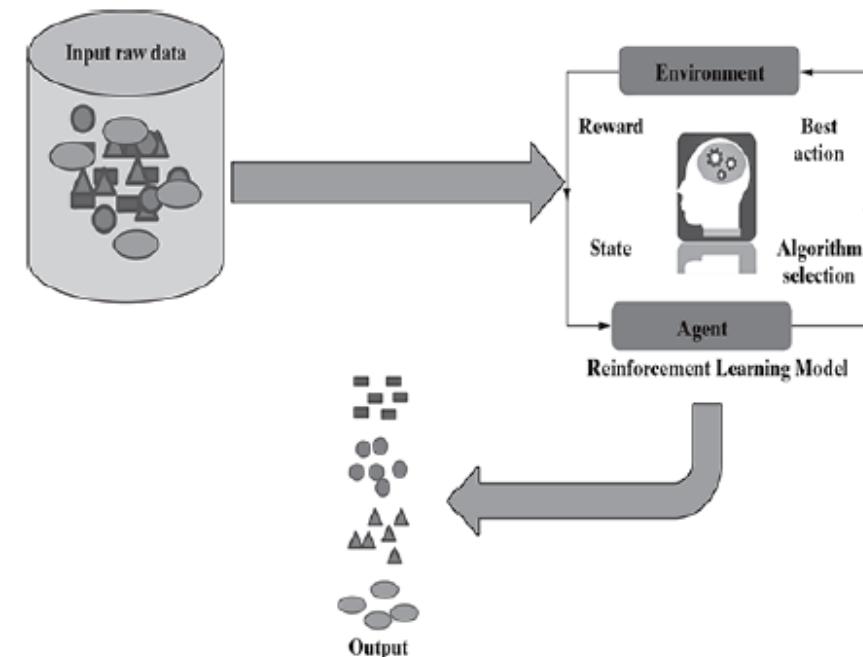
Market Basket transactions

Frequent itemsets \Rightarrow (Diaper, Beer)

Possible association: Diaper \Rightarrow Beer

Reinforcement learning

- Machines often learn to do tasks autonomously.
- It tries to improve its performance of doing the task.
- When a sub-task is accomplished successfully, a reward is given.
- When a sub-task is not executed correctly, obviously no reward is given.
- This continues till the machine is able to complete execution of the whole task.
- This process of learning is known as **reinforcement learning**



APPLICATIONS OF MACHINE LEARNING

- Web search
- Computational biology
- Banking and Finance
- E-commerce
- Space exploration
- Robotics
- Information extraction
- Social networks
- Debugging software
- Healthcare
- And many more

LANGUAGES/TOOLS IN MACHINE LEARNING

- Python is one of the most popular, open source programming language widely adopted by machine learning community
- Python has very strong libraries for advanced
 - mathematical functionalities (NumPy),
 - algorithms and mathematical tools (SciPy) and
 - numerical plotting (matplotlib).
- Built on these libraries, there is a machine learning library named **scikitlearn**, which has various classification, regression, and clustering algorithms embedded in it.

- R is a language for statistical computing and data analysis.
- R is a very simple programming language with a huge set of libraries available for different stages of machine learning.
- Some of the libraries standing out in terms of popularity are
 - plyr/dplyr (for data transformation),
 - caret ('Classification and Regression Training' for classification),
 - RJava (to facilitate integration with Java),
 - tm (for text mining),
 - ggplot2 (for data visualization).
- Other than the libraries, certain packages like Shiny and R Markdown have been developed around R to develop interactive web applications, documents and dashboards on R without much effort.

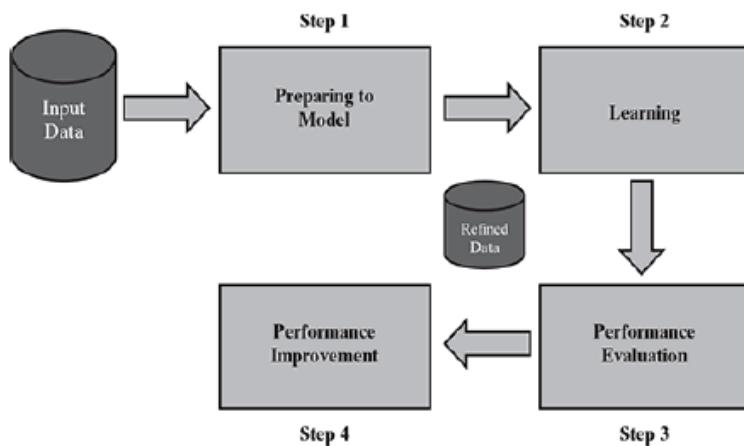
- MATLAB (matrix laboratory) is a licenced commercial software with a robust support for a wide range of numerical computing.
- MATLAB has a huge user base across industry and academia. MATLAB is developed by MathWorks, a company founded in 1984.
- Being proprietary software, MATLAB is developed much more professionally, tested rigorously, and has comprehensive documentation.
- MATLAB also provides extensive support of statistical functions and has a huge number of machine learning algorithms in-built.
- It also has the ability to scale up for large datasets by parallel processing on clusters and cloud.

- SAS (earlier known as ‘Statistical Analysis System’) is another licenced commercial software which provides strong support for machine learning functionalities.
- Developed in C by SAS Institute, SAS had its first release in the year 1976.
- SAS is a software suite comprising different components.
- The basic data management functionalities are embedded in the Base SAS component whereas the other components like SAS/INSIGHT, Enterprise Miner, SAS/STAT, etc. help in specialized functions related to data mining and statistical analysis.

MACHINE LEARNING ACTIVITIES

- Following are the typical **preparation activities** done once the input data comes into the machine learning system:
 - Understand the type of data in the given input data set.
 - Explore the data to understand the nature and quality.
 - Explore the relationships amongst the data elements, e.g. inter-feature relationship.
 - Find potential issues in data.
 - Do the necessary remediation, e.g. inpute missing data values, etc., if needed.
 - Apply pre-processing steps, as necessary.
 - Once the data is prepared for modelling, then the learning tasks start off. As a part of it, do the following activities:
 - The input data is first divided into parts – the training data and the test data (called holdout). This step is applicable for supervised learning only.
 - Consider different models or learning algorithms for selection.
 - Train the model based on the training data for supervised learning problem and apply to unknown data. Directly apply the chosen unsupervised model on the input data for unsupervised learning problem

MACHINE LEARNING ACTIVITIES



Step #	Step Name	Activities Involved
Step 1	Preparing to Model	<ul style="list-style-type: none">Understand the type of data in the given input data setExplore the data to understand data qualityExplore the relationships amongst the data elements, e.g., feature relationshipFind potential issues in dataRemediate data, if neededApply following pre-processing steps, as necessary:<ul style="list-style-type: none">✓ Dimensionality reduction✓ Feature subset selection
Step 2	Learning	<ul style="list-style-type: none">Data partitioning/holdoutModel selectionCross-validation
Step 3	Performance evaluation	<ul style="list-style-type: none">Examine the model performance, e.g. confusion matrix classificationVisualize performance trade-offs using ROC curves
Step 4	Performance improvement	<ul style="list-style-type: none">Tuning the modelEnsembling<ul style="list-style-type: none">BaggingBoosting

BASIC TYPES OF DATA

- A data set is a collection of related information or records.
- Each row of a data set is called a record.
- Each data set also has multiple attributes, each of which gives information on a specific characteristic
- Attributes can also be termed as feature, variable, dimension or field

- The attributes can be either discrete or continuous based on data values assigned to it

Student data set:

Roll Number	Name	Gender	Age
129/011	Mihir Karmarkar	M	14
129/012	Geeta Iyer	F	15
129/013	Chanda Bose	F	14
129/014	Sreenu Subramanian	M	14
129/015	Pallav Gupta	M	16
129/016	Gajanan Sharma	M	15

Student performance data set:

Roll Number	Maths	Science	Percentage
129/011	89	45	89.33%
129/012	89	47	90.67%
129/013	68	29	64.67%
129/014	83	38	80.67%
129/015	57	23	53.33%
129/016	78	35	75.33%

Data

- Data can broadly be divided into following two types:
 - 1. Qualitative data
 - 2. Quantitative data
- **Qualitative data** provides information about the quality of an object or information which cannot be measured. ('Good', 'Average', and 'Poor' and "name or roll number")
- Qualitative data is also called **categorical data**.
- Qualitative data can be further subdivided into two types
 - 1. Nominal data
 - 2. Ordinal data

Nominal Data

- **Nominal data** is one which has no numeric value, but a named value. It is used for assigning named values to attributes.
- Nominal values cannot be quantified.
- Examples of nominal data are
 - 1. Blood group: A, B, O, AB, etc.
 - 2. Nationality: Indian, American, British, etc.
 - 3. Gender: Male, Female, Other
- Mathematical operations/statistical functions cannot be performed on nominal data.
- A basic count is possible

Ordinal data

- **Ordinal data**, in addition to possessing the properties of nominal data, can also be naturally ordered
- Examples of ordinal data are
 - 1. Customer satisfaction: ‘Very Happy’, ‘Happy’, ‘Unhappy’, etc.
 - 2. Grades: A, B, C, etc.
 - 3. Hardness of Metal: ‘Very Hard’, ‘Hard’, ‘Soft’, etc.
- Since ordering is possible in case of ordinal data, median, and quartiles can be identified in addition. Mean can still not be calculated.

Quantitative data

- **Quantitative data** relates to information about the quantity of an object – hence it can be measured
- It can be measured using a scale of measurement.
- Quantitative data is also termed as numeric data.
- There are two types of quantitative data:
 - 1. Interval data
 - 2. Ratio data

Interval data

- **Interval data** is numeric data for which not only the order is known, but the exact difference between values is also known.
- For interval data, mathematical operations are possible.
- The central tendency can be measured by mean, median, or mode.
- Standard deviation can also be calculated

Ratio Data

- **Ratio data** represents numeric data for which exact value can be measured.
- Absolute zero is available for ratio data.
- Also, these variables can be added, subtracted, multiplied, or divided.
- The central tendency can be measured by mean, median, or mode and methods of dispersion such as standard deviation.
- Examples of ratio data include height, weight, age, salary, etc.

Type of attribute

- The attributes can be either discrete or continuous based on the number of values that can be assigned to it
- Discrete attributes can assume a finite or countably infinite number of values.
- Nominal attributes such as roll number, street number, pin code, etc. can have a finite number of values whereas numeric attributes such as count, rank of students, etc. can have countably infinite values.
- A special type of discrete attribute which can assume two values only is called binary attribute.
- Examples of binary attribute include male/ female, positive/negative, yes/no, etc
- Continuous attributes can assume any possible value which is a real number.
- Examples of continuous attribute include length, height, weight, price, etc.

EXPLORING STRUCTURE OF DATA

mpg	cylinder	displace- ment	horse- power	weight	accel- eration	model year	origin	car name
18	8	307	130	3504	12	70	1	Chevrolet chev- elle malibu
15	8	350	165	3693	11.5	70	1	Buick skylark 320
18	8	318	150	3436	11	70	1	Plymouth satellite
16	8	304	150	3433	12	70	1	Amc rebel sst
17	8	302	140	3449	10.5	70	1	Ford torino
15	8	429	198	4341	10	70	1	Ford galaxie 500
14	8	454	220	4354	9	70	1	Chevrolet impala
14	8	440	215	4312	8.5	70	1	Plymouth fury iii
14	8	455	225	4425	10	70	1	Pontiac catalina
15	8	390	190	3850	8.5	70	1	Amc ambassador dpl
15	8	383	170	3563	10	70	1	Dodge challenger se
14	8	340	160	3609	8	70	1	Plymouth 'cuda 340
15	8	400	150	3761	9.5	70	1	Chevrolet monte carlo
14	8	455	225	3086	10	70	1	Buick estate wagon (sw)
24	4	113	95	2372	15	70	3	Toyota corona mark ii
22	6	198	95	2933	15.5	70	1	Plymouth duster
18	6	199	97	2774	15.5	70	1	Amc hornet

- The data set is the Auto MPG data set available in the UCI repository
- the attributes such as ‘mpg’, ‘cylinders’, ‘displacement’, ‘horsepower’, ‘weight’, ‘acceleration’, ‘model year’, and ‘origin’ are all numeric.
- Out of these attributes, ‘cylinders’, ‘model year’, and ‘origin’ are discrete in nature
- The remaining of the numeric attributes, i.e. ‘mpg’, ‘displacement’, ‘horsepower’, ‘weight’, and ‘acceleration’ can assume any real value.
- ‘car name’ is of type categorical (nominal)

CENTRAL TENDENCY OF DATA

- In statistics, measures of central tendency help us understand the central point of a set of data.
- Mean, by definition, is a sum of all data values divided by the count of data elements.
- For example, mean of a set of observations – 21, 89, 34, 67, and 96 is calculated as
Mean = $(21+89+34+67+96)/5= 61.4$
- Median, is the value of the element appearing in the middle of an ordered list of data elements.
The median value of this set of data is 67.

Mean and Median

	mpg	cylinders	dis- place- ment	horse- power	weight	accel- eration	model year	origin
Median	23	4	148.5	7	2804	15.5	76	1
Mean	23.51	5.455	193.4	7	2970	15.57	76.01	1.573
Deviation	2.17	26.67%	23.22%		5.59%	0.45%	0.01%	36.43%
	Low	High	High		Low	Low	Low	High

Mean and Median

- Consider the data values of two attributes
 1. Attribute 1 values : 44, 46, 48, 45, and 47
 2. Attribute 2 values : 34, 46, 59, 39, and 52
- Both the set of values have a mean and median of 46.
- The first set of values is more concentrated or clustered around the mean/median value
- The second set of values is quite spread out or dispersed.
- To measure the extent of dispersion of a data, or to find out how much the different values of a data are spread out, the variance of the data is measured.

Variance

- The variance of a data is measured using the formula given by

$$\text{Variance } (x) = \frac{\sum_{i=1}^n x_i^2}{n} - \left(\frac{\sum_{i=1}^n x_i}{n} \right)^2$$

- where x is the variable whose variance is to be measured and n is the number of observations or values of variable x .
- Standard deviation of a data is measured as follows:
Standard deviation = $\sqrt{\text{Variance } (x)}$
- Larger value of variance or standard deviation indicates more dispersion in the data and vice versa

$$\begin{aligned}
 \text{Variance} &= \frac{\sum_{i=1}^n x_i^2}{n} - \left(\frac{\sum_{i=1}^n x_i}{n} \right)^2 \\
 &= \frac{44^2 + 46^2 + 48^2 + 45^2 + 47^2}{5} - \left(\frac{44 + 46 + 48 + 45 + 47}{5} \right)^2 \\
 &= \frac{1936 + 2116 + 2304 + 2025 + 2209}{5} - \left(\frac{230}{5} \right)^2 = \frac{10590}{5} - (46)^2 = 2
 \end{aligned}$$

For attribute 2,

$$\begin{aligned}
 \text{Variance} &= \frac{\sum_{i=1}^n x_i^2}{n} - \left(\frac{\sum_{i=1}^n x_i}{n} \right)^2 \\
 &= \frac{34^2 + 46^2 + 59^2 + 39^2 + 52^2}{5} - \left(\frac{34 + 46 + 59 + 39 + 52}{5} \right)^2
 \end{aligned}$$

- it is quite clear from the measure that attribute 1 values are quite concentrated around the mean while attribute 2 values are extremely spread out

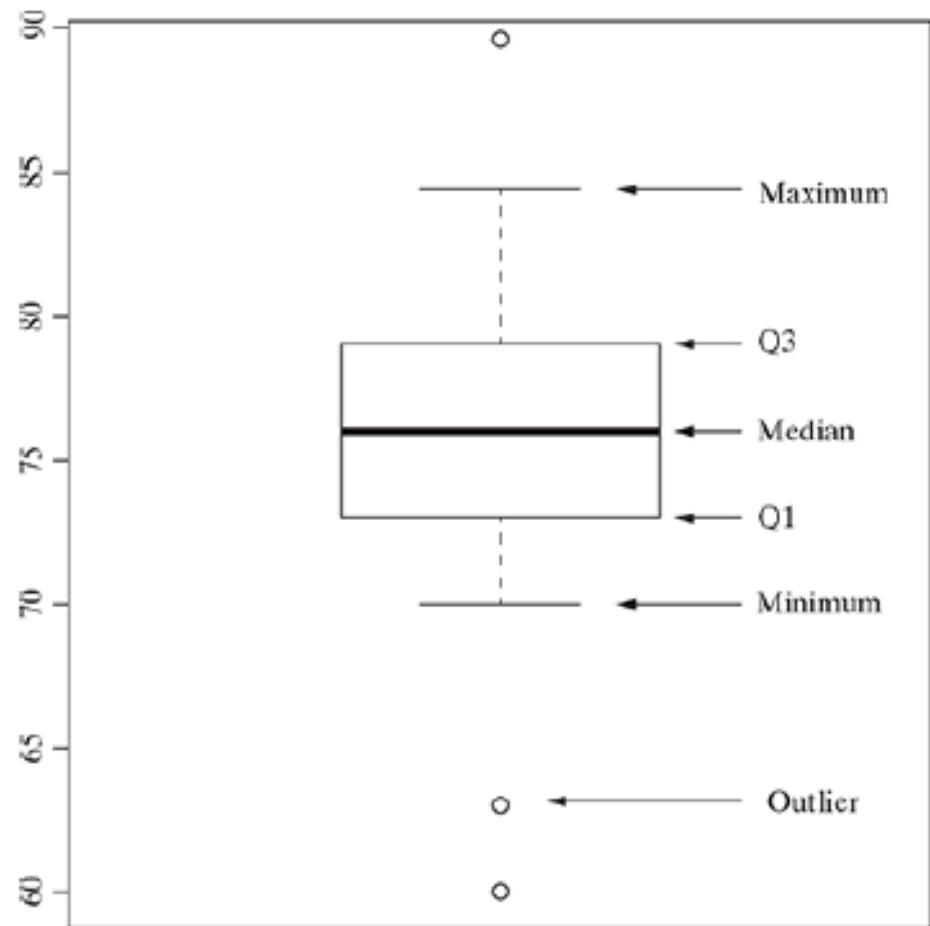
Measuring data value position

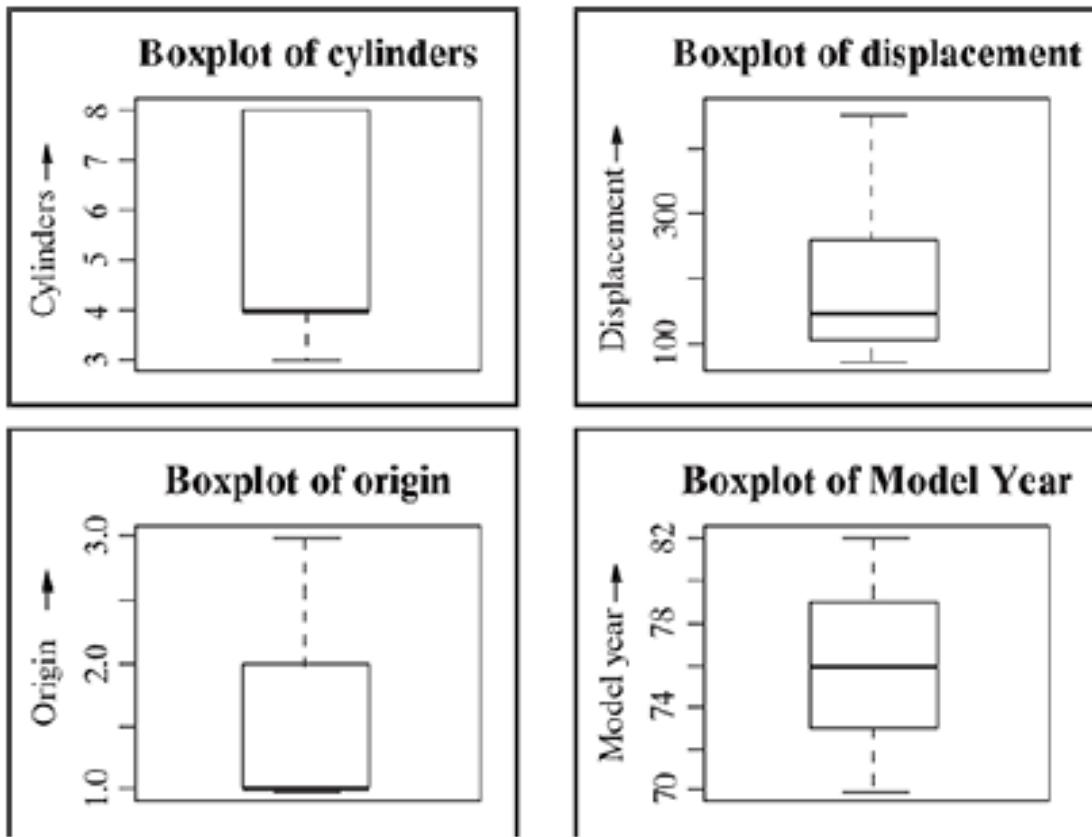
- When the data values of an attribute are arranged in an increasing order, that median gives the central data value, which divides the entire data set into two halves.
- Similarly, if the first half of the data is divided into two halves so that each half consists of one-quarter of the data set, then that median of the first half is known as first quartile or Q1 .
- In the same way, if the second half of the data is divided into two halves, then that median of the second half is known as third quartile or Q3 .
- The overall median is also known as second quartile or Q2 .
- So, any data set has five values –minimum, first quartile (Q1), median (Q2), third quartile (Q3), and maximum.

	cylinders	displacement	origin
Minimum	3	68	1
Q1	4	104.2	1
Median	4	148.5	1
Q3	8	262	2
Maximum	8	455	3

Plotting and exploring numerical data

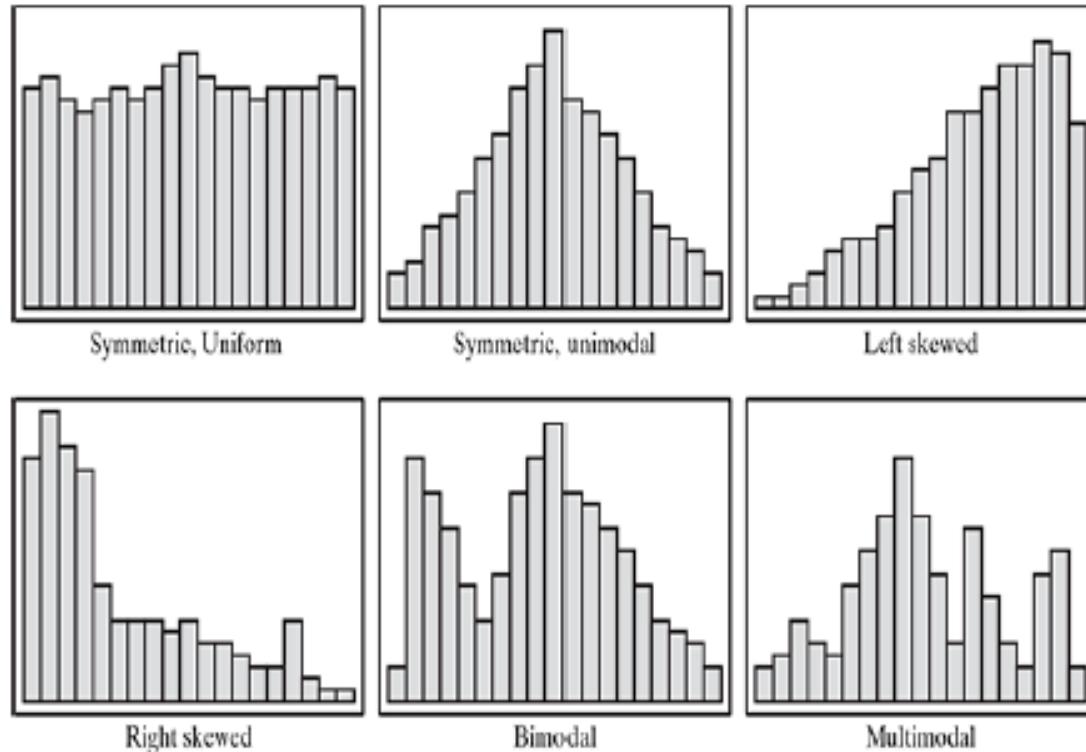
- The **box plot** (also called box and whisker plot) gives a standard visualization of the five-number summary statistics of a data, namely minimum, first quartile (Q1), median (Q2), third quartile (Q3), and maximum.

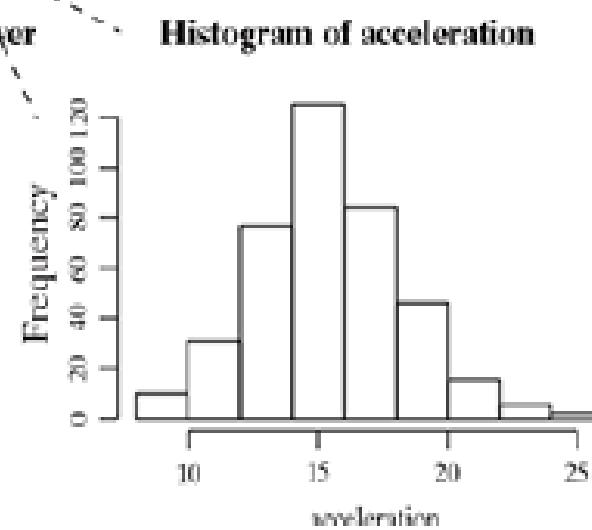
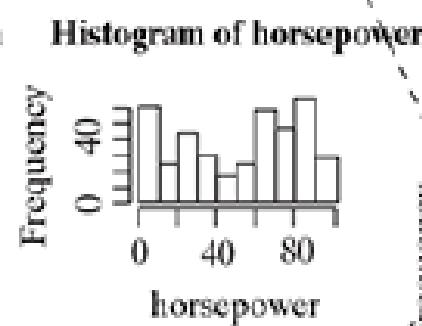
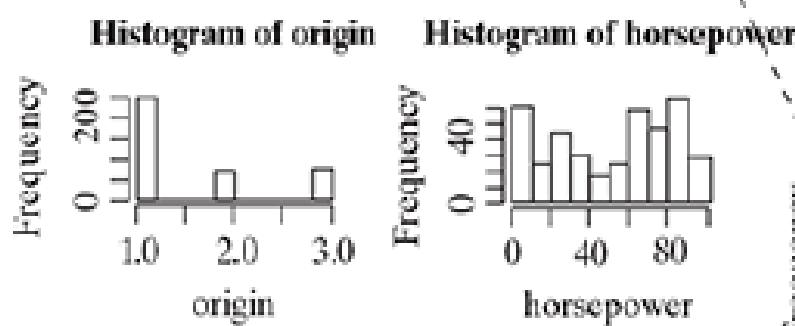
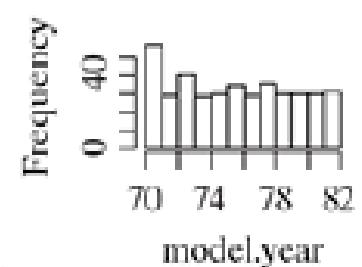
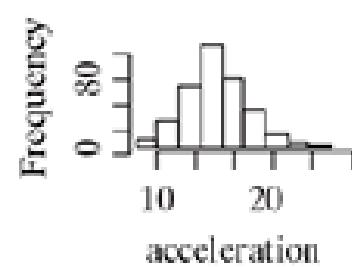
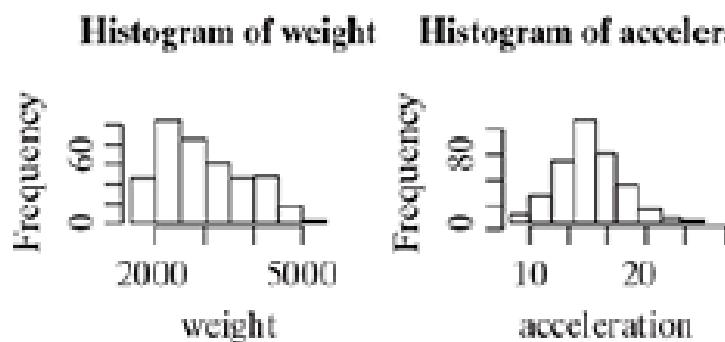
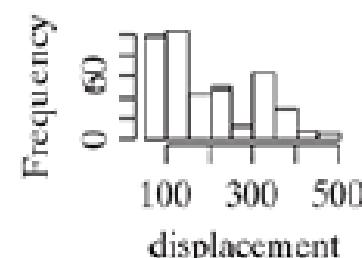
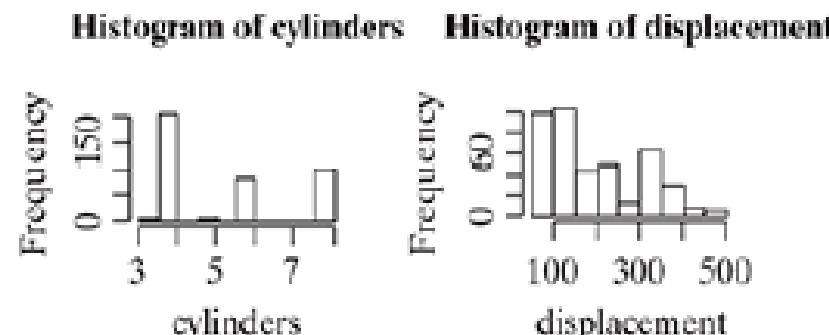
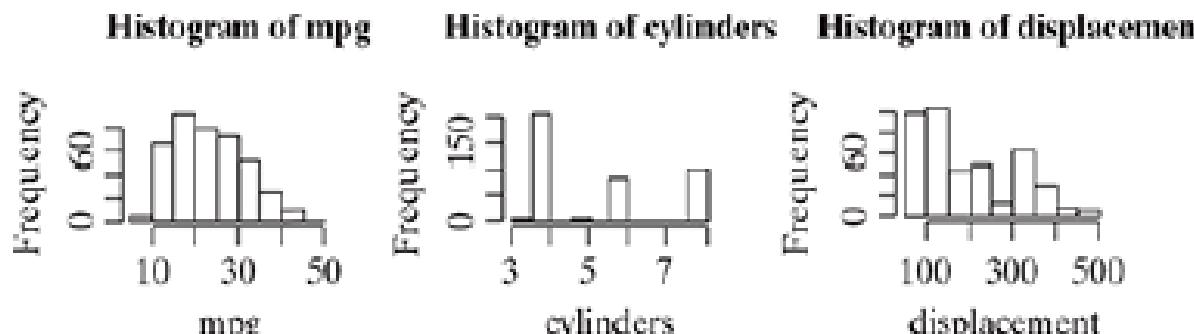




Histogram

- Histogram is another plot which helps in effective visualization of numeric attributes.
- It helps in understanding the distribution of a numeric data into series of intervals, also termed as ‘bins’.
- Types are





Exploring categorical data

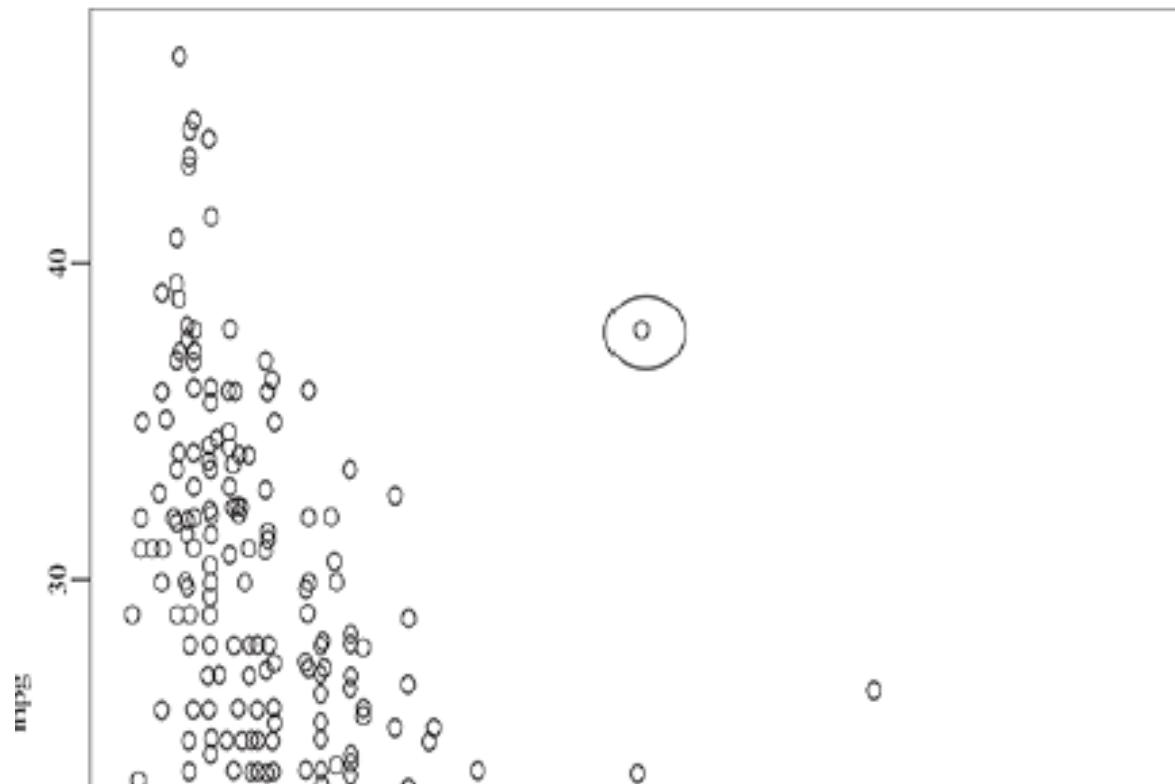
- attribute ‘car.name’ is categorical in nature.
- There are many unique names for the attribute ‘car name’ or how many unique values are there for ‘cylinders’ attribute.
- For attribute ‘car name’
 1. Chevrolet chevelle malibu
 2. Buick skylark 320
 3. Plymouth satellite
 4. Amc rebel sst
 5. Ford torino
 6. Ford galaxie 500
 7. Chevrolet impala
 8. Plymouth fury iii
 9. Pontiac catalina
 10. Amc ambassador dpl
- For attribute ‘cylinders’ 8 4 6 3 5

Attribute	amc	amc ambas-	amc	amc	amc	amc con-	amc	...
Value	ambas-	sador dpl	ambassa-	concord	concord	cord dl6	gremlin	
	sador		dor sst		d/l			
Count	1	1	1	1	2	2	4	...

Attribute	3	4	5	6	8
Value					
Count	4	204	3	84	103

Scatter plot

- A scatter plot helps in visualizing bivariate relationships, i.e. relationship between two variables.
- It is a two-dimensional plot in which points or dots are drawn on coordinates provided by values of the attributes.



Two-way cross-tabulations

- Two-way cross-tabulations (also called cross-tab or contingency table) are used to understand the relationship of two categorical attributes in a concise way.
- It has a matrix format that presents a summarized view of the bivariate frequency distribution.

Origin \ Model Year	70	71	72	73	74	75	76	77	78	79	80	81	82
1	22	20	18	29	15	20	22	18	22	23	7	13	20
2	5	4	5	7	6	6	8	4	6	4	9	4	2
3	2	4	5	4	6	4	4	6	8	2	13	12	9

Cylinders \ Origin	1	2	3
3	0	0	4
4	72	63	69
5	0	3	0
6	74	4	6
8	103	0	0

DATA QUALITY

- Success of machine learning depends largely on the quality of data. A data which has the right quality helps to achieve better prediction accuracy, in case of supervised learning.
- There can be two types of problems:
 - 1. Certain data elements without a value or data with a missing value.
 - 2. Data elements having value surprisingly different from the other elements, which we term as outliers.
- This is due to
 - Incorrect sample set selection
 - Errors in data collection

Data remediation

- The issues in data quality, need to be remediated, if the right amount of efficiency has to be achieved in the learning activity

Handling outliers

- **Remove outliers:** If the number of records which are outliers is not many, a simple approach may be to remove them.
- **Imputation:** One other way is to impute the value with mean or median or mode. The value of the most similar data element may also be used for imputation.
- **Capping:** For values that lie outside the $1.5 \times |IQR|$ limits, we can cap them by replacing those observations below the lower limit with the value of 5th percentile and those that lie above the upper limit, with the value of 95th percentile.
- (If there is a significant number of outliers, they should be treated separately in the statistical model.)
- In that case, the groups should be treated as two different groups, the model should be built for both groups and then the output can be combined.)

Handling missing values

- Eliminate records having a missing value of data elements
- Imputing missing values
 - Imputation is a method to assign a value to the data elements having missing values. Mean/mode/median is most frequently assigned value.
 - For quantitative attributes, all missing values are imputed with the mean, median, or mode of the remaining values under the same attribute.
 - For qualitative attributes, all missing values are imputed by the mode of all remaining values of the same attribute.
- Estimate missing values
 - If there are data points similar to the ones with missing attribute values, then the attribute values from those similar data points can be planted in place of the missing value.

DATA PRE-PROCESSING

- **Dimensionality reduction** refers to the techniques of reducing the dimensionality of a data set by creating new attributes by combining the original attributes.
- High-dimensional data sets need a high amount of computational space and time.
- Most of the machine learning algorithms perform better if the dimensionality of data set is reduced.
- Some popular dimensionality reduction techniques are Principal Component Analysis(PCA), Singular Value Decomposition(SVD), and feature selection.

- **Feature subset selection**, both for supervised as well as unsupervised learning, try to find out the optimal subset of the entire feature set
- This significantly reduces computational cost without any major impact on the learning accuracy.
- It may seem that a feature subset may lead to loss of useful information as certain features are going to be excluded from the final set of features used for learning.
- for elimination only features which are not relevant or redundant are selected.

Thankyou

Model selection and Feature Engineering

Unit 2

SELECTING A MODEL

- Multiple factors play a role when we try to select the model for solving a machine learning problem.
- The most important factors are
 - (i) the kind of problem we want to solve using machine learning and
 - (ii) the nature of the underlying data
- There is no one model that works best for every machine learning problem
- Machine learning algorithms are broadly of two types:
 - models for supervised learning, which primarily focus on solving predictive problems and
 - models for unsupervised learning, which solve descriptive problems

Predictive models

- Models for supervised learning or predictive models try to predict certain value using the values in an input data set.
- The learning model attempts to establish a relation between the target feature, i.e. the feature being predicted, and the predictor features.
- The predictive models have a clear focus on what they want to learn and how they want to learn.

Model for categorical value

- Predictive models, may need to predict the value of a category or class to which a data instance belongs to.
- some examples:
 - 1. Predicting win/loss in a cricket match
 - 2. Predicting whether a transaction is fraud
 - 3. Predicting whether a customer may move to another product
- The models which are used for prediction of target features of categorical value are known as classification models.
- The target feature is known as a class and the categories to which classes are divided into are called levels.
- Some of the popular classification models include *k-Nearest Neighbor (kNN)*, Naïve Bayes, and Decision Tree.

Model for numerical value

- Predictive models may also be used to predict numerical values of the target feature based on the predictor features.
- Below are some examples:
 - 1. Prediction of revenue growth in the succeeding year
 - 2. Prediction of rainfall amount in the coming monsoon
 - 3. Prediction of potential flu patients and demand for flu shots next winter
- The models which are used for prediction of the numerical value of the target feature of a data instance are known as regression models.
- Linear Regression and Logistic Regression models are popular regression models.

Descriptive models

- Models for unsupervised learning or descriptive models are used to describe a data set or gain insight from a data set.
- There is no target feature or single feature of interest in case of unsupervised learning.
- Based on the value of all features, interesting patterns or insights are derived about the data set.

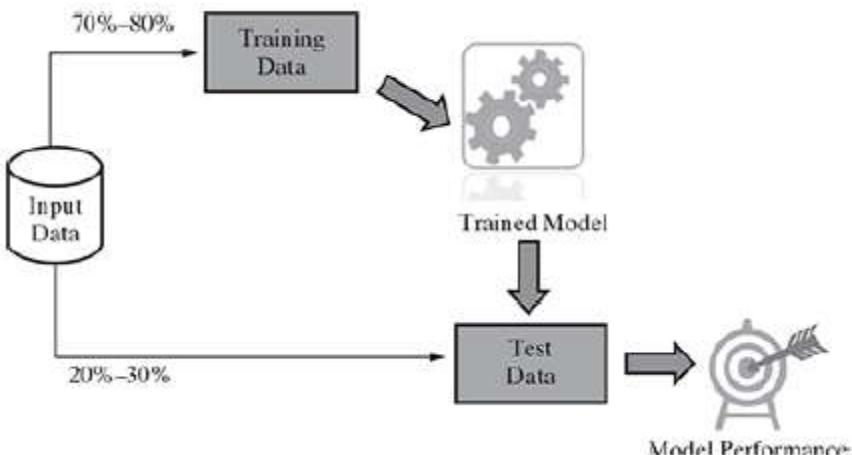
- Descriptive models which group together similar data instances, i.e. data instances having a similar value of the different features are called clustering models.
- Examples of clustering include
 - 1. Customer grouping or segmentation based on social, demographic, ethnic, etc. factors
 - 2. Grouping of music based on different aspects like genre (শৈলী), language, timeperiod, etc.
 - 3. Grouping of commodities in an inventory
- The most popular model for clustering is *k-Means*.

TRAINING A MODEL (FOR SUPERVISED LEARNING)

- There are 4 methods
 - Holdout method
 - *K-fold Cross-validation method*
 - Bootstrap sampling
 - Lazy vs. Eager learner

Holdout method

- In supervised learning, a model is trained using the labelled input data.
- The subset of the input data is used as the test data for evaluating the performance of a trained model.
- In general 70%–80% of the input data (which is obviously labelled) is used for model training.
- The remaining 20%–30% is used as test data for validation of the performance of the model.
- The division is done randomly.
- This method of partitioning the input data into two parts – training and test data and by holding back a part of the input data for validating the trained model is known as holdout method.



- In certain cases, the input data is partitioned into three portions – a training and a test data, and a validation data.
- The validation data is used in place of test data, for measuring the model performance.
- It is used in iterations and to refine the model in each iteration.
- The test data is used only for once, after the model is refined and finalized, to measure and report the final performance of the model as a reference for future learning efforts.

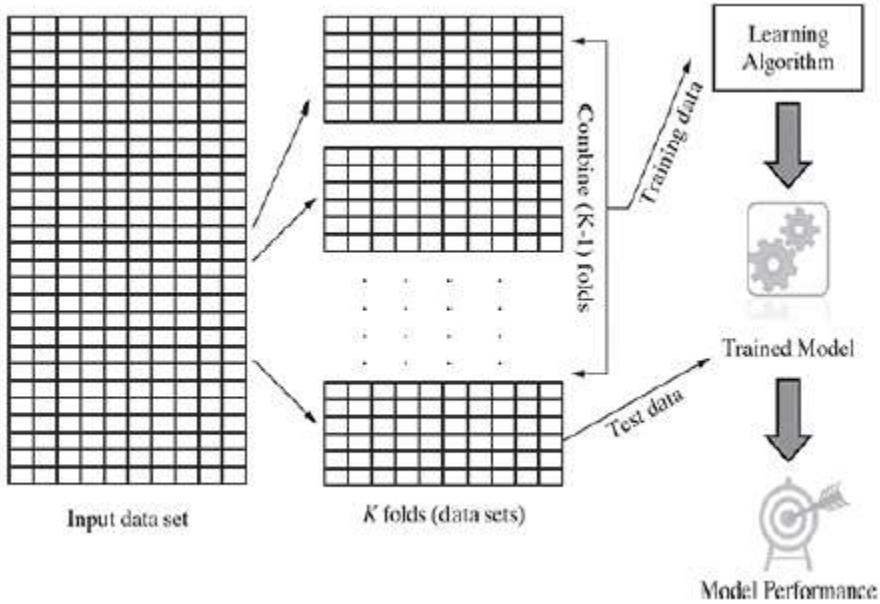
- The problem in this method is that the division of data of different classes into the training and test data may not be proportionate
- This problem can be addressed to some extent by applying stratified random sampling in place of sampling.
- In case of stratified random sampling, the whole data is broken into several homogenous groups or strata and a random sample is selected from each such stratum.
- This ensures that the generated random partitions have equal proportions of each class.
- The smaller data sets may have the challenge to divide the data of some of the classes proportionally amongst training and test data sets.

K-fold Cross-validation method

- A special variant of holdout method, called repeated holdout, is sometimes employed to ensure the randomness of the composed data sets.
- In repeated holdout, several random holdouts are used to measure the model performance.
- In the end, the average of all performances is taken.
- As multiple holdouts have been drawn, the training and test data are more likely to contain representative data from all classes and resemble the original input data closely.
- This process of repeated holdout is the basis of *k-fold cross validation* technique.
- In *k-fold cross-validation*, the data set is divided into *k*-completely distinct or non-overlapping random partitions called folds.

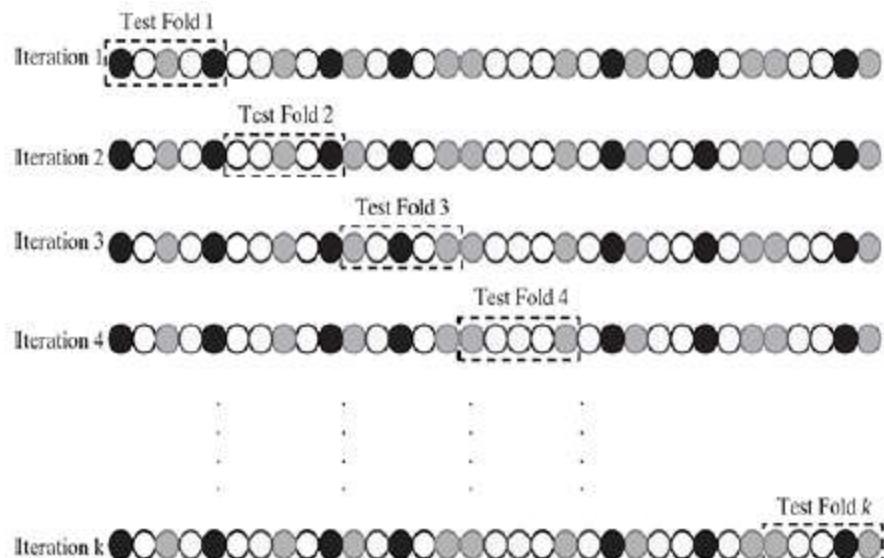
10-fold cross-validation (10-fold CV)

- The value of '*k*' in *k*-fold cross-validation can be set to any number. There are two approaches which are extremely popular:
 1. 10-fold cross-validation (10-fold CV)
 2. Leave-one-out cross-validation (LOOCV)
- 10-fold cross-validation is by far the most popular approach.
- In this approach, for each of the 10-folds, each comprising of approximately 10% of the data, one of the folds is used as the test data for validating model performance trained based on the remaining 9 folds (or 90% of the data).
- This is repeated 10 times, once for each of the 10 folds being used as the test data and the remaining folds as the training data.
- The average performance across all folds is being reported.



Overall approach for k fold cross validation

the records in a fold are drawn by using random sampling technique.



Detailed approach for fold selection

Leave-one-out cross-validation (LOOCV)

- LOOCV is an extreme case of *k-fold cross-validation using one record or data* instance at a time as a test data.
- This is done to maximize the count of data used to train the model.
- It is obvious that the number of iterations for which it has to be run is equal to the total number of data in the input data set.
- Hence, obviously, it is computationally very expensive and not used much in practice.

Feature

- A feature is an attribute of a data set that is used in a machine learning process.
- The features in a data set are also called its dimensions. So a data set having ' n ' features is called an *n-dimensional data set*.
- A famous machine learning data set, Iris, introduced by the British statistician and biologist Ronald Fisher, partly shown below. It is a five-dimensional data set.
- It has five attributes or features namely Sepal. Length, Sepal.Width, Petal.Length, Petal. Width and Species.
- the feature 'Species'- class variable
- the remaining features -predictor variables.

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
6.7	3.3	5.7	2.5	Virginica
4.9	3	1.4	0.2	Setosa
5.5	2.6	4.4	1.2	Versicolor
6.8	3.2	5.9	2.3	Virginica
5.5	2.5	4	1.3	Versicolor
5.1	3.5	1.4	0.2	Setosa
6.1	3	4.6	1.4	versicolor

Feature engineering

- Feature engineering is a critical preparatory process in machine learning.
- It is responsible for taking raw input data and converting that to well-aligned features which are ready to be used by the machine learning models.
- It has two major elements:
 1. feature transformation
 2. feature subset selection

Feature transformation

- Feature transformation transforms the data – structured or unstructured, into a new set of features which can represent the underlying problem which machine learning is trying to solve.
- There are two variants of feature transformation:
 1. feature construction
 2. feature extraction
- There are two distinct goals of feature transformation
 - Achieving best reconstruction of the original features in the data set
 - Achieving highest efficiency in the learning task

Feature construction

- Feature construction involves transforming a given set of input features to generate a new set of more powerful features.
- Example

apartment_length	apartment_breadth	apartment_price	apartment_length	apartment_breadth	apartment_area	apartment_price
80	59	23,60,000	80	59	4,720	23,60,000
54	45	12,15,000	54	45	2,430	12,15,000
78	56	21,84,000	78	56	4,368	21,84,000
63	63	19,84,000	63	63	3,969	19,84,500
83	74	30,71,000	83	74	6,142	30,71,000
92	86	39,56,000	92	86	7,912	39,56,000

- The three-dimensional data set is transformed to a four-dimensional data set, with the newly ‘discovered’ feature apartment area being added to the original data set.

- There are certain situations where feature construction is an essential activity before we can start with the machine learning task.
- These situations are
 - when features have categorical value and machine learning needs numeric value inputs
 - when features having numeric (continuous) values and need to be converted to ordinal values
 - when text-specific feature construction needs to be done

Encoding categorical (nominal) variables

- Example - data set on athletes
- The data set has features age, city of origin, parents athlete (i.e. indicate whether any one of the parents was an athlete) and chance of win.
- The feature chance of a win is a class variable while the others are predictor variables.
- Any machine learning algorithm requires numerical figures to learn from.
- In the dataset there are three features – City of origin, Parents athlete, and Chance of win, which are categorical in nature and cannot be used by any machine learning task.

Age (Years)	City of origin	Parents athlete	Chance of win
18	City A	Yes	Y
20	City B	No	Y
23	City B	Yes	Y
19	City A	No	N
18	City C	Yes	N
22	City B	Yes	Y

- feature construction can be used to create new dummy features which are usable by machine learning algorithms.

Age (Years)	origin_city_A	origin_city_B	origin_city_C	parents_athlete_Y	parents_athlete_N	win_chance_Y	win_chance_N
18	1	0	0	1	0	1	0
20	0	1	0	0	1	1	0
23	0	1	0	1	0	1	0
19	1	0	0	0	1	0	1
18	0	0	1	1	0	0	1
22	0	1	0	1	0	1	0

(b)

Age (Years)	origin_city_A	origin_city_B	origin_city_C	parents_athlete_Y	win_chance_Y
18	1	0	0	1	1
20	0	1	0	0	1
23	0	1	0	1	1
19	1	0	0	0	0
18	0	0	1	1	0
22	0	1	0	1	1

Encoding categorical (ordinal) variables

marks_science	marks_maths	Grade
78	75	B
56	62	C
87	90	A
91	95	A
45	42	D
62	57	B

marks_science	marks_maths	num_grade
78	75	2
56	62	3
87	90	1
91	95	1
45	42	4
62	57	2

Transforming numeric (continuous) features to categorical features

- Sometimes there is a need of transforming a continuous numerical variable into a categorical variable. eg. real estate price category
 - In that case, we can ‘bin’ the numerical data into multiple categories based on the data range.

apartment_area	apartment_price	apartment_area	apartment_grade
4,720	23,60,000	4,720	Medium
2,430	12,15,000	2,430	Low
4,368	21,84,000	4,368	Medium
3,969	19,84,500	3,969	Low
6,142	30,71,000	6,142	High
7,912	39,56,000	7,912	High

apartment_area	apartment_grade
4,720	2
2,430	1
4,368	2
3,969	1
6,142	3
7,912	3

Text-specific feature construction

- All machine learning models need numerical data as input.
- So the text data in the data sets need to be transformed into numerical features.
- Text data, or corpus which is the more popular keyword, is converted to a numerical representation following a process known as vectorization.
- In this process, word occurrences in all documents belonging to the corpus are consolidated in the form of bag-of-words.

- There are three major steps that are followed:
 1. tokenize
 2. count
 3. normalize
- In order to tokenize a corpus, the blank spaces and punctuations are used as delimiters to separate out the words, or tokens.
- Then the number of occurrences of each token is counted, for each document. Lastly, tokens are weighted with reducing importance when they occur in the majority of the documents.

- A matrix is then formed with each token representing a column and a specific document of the corpus representing each row.
 - Each cell contains the count of occurrence of the token in a specific document.
 - This matrix is known as a document-term matrix (also known as a term-document matrix).

Feature extraction

- In feature extraction, new features are created from a combination of original features.
- Some of the commonly used operators for combining the original features include
 1. For Boolean features: Conjunctions, Disjunctions, Negation, etc.
 2. For nominal features: Cartesian product, M of N, etc.
 3. For numerical features: Min, Max, Addition, Subtraction, Multiplication, Division, Average, Equivalence, Inequality, etc.
- Say, we have a data set with a feature set $F_i(F_1, F_2, \dots, F_n)$. After feature extraction using a mapping function $f(F_1, F_2, \dots, F_n)$ say, we will have a set of features such that
 $F'_i = f(F_i)$ and $m < n$. For example,

Feat_A	Feat_B	Feat_C	Feat_D		Feat₁	Feat₂
34	34.5	23	233		41.25	185.80
44	45.56	11	3.44		54.20	53.12
78	22.59	21	4.5		43.73	35.79
22	65.22	11	322.3		65.30	264.10
22	33.8	355	45.2		37.02	238.42
11	122.32	63	23.2		113.39	167.74



$$\text{Feat}_1 = 0.3 \times \text{Feat}_A + 0.9 \times \text{Feat}_C$$

$$\text{Feat}_2 = \text{Feat}_A + 0.5 \times \text{Feat}_B + 0.6 \times \text{Feat}_C$$

$$F'_1 = k_1 F_1 + k_2 F_2$$

Feature Extraction Algorithms

The most popular feature extraction algorithms used in machine learning are

- *Principal Component Analysis*
- *Singular value decomposition*
- *Linear Discriminant Analysis*

Principal Component Analysis

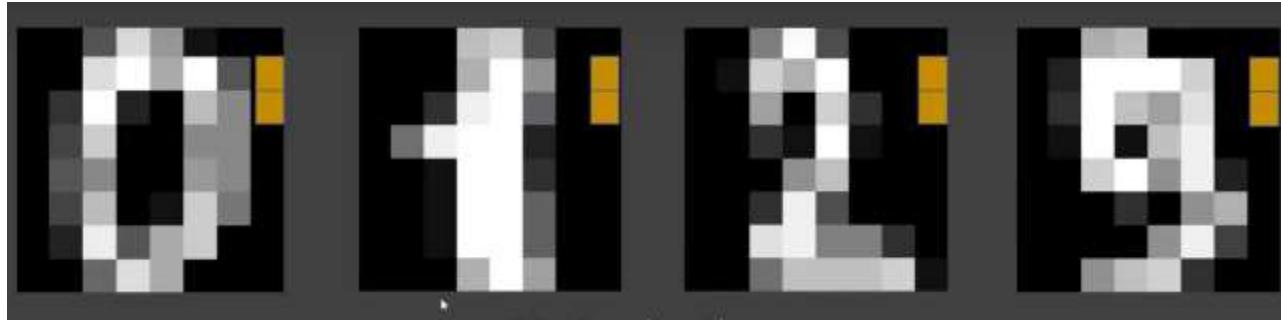
- The features should be less in number as well as the similarity between them should be very less.
- This is the main guiding philosophy of principal component analysis (PCA) technique of feature extraction.

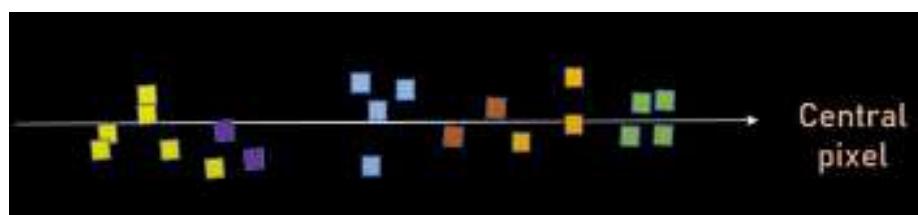
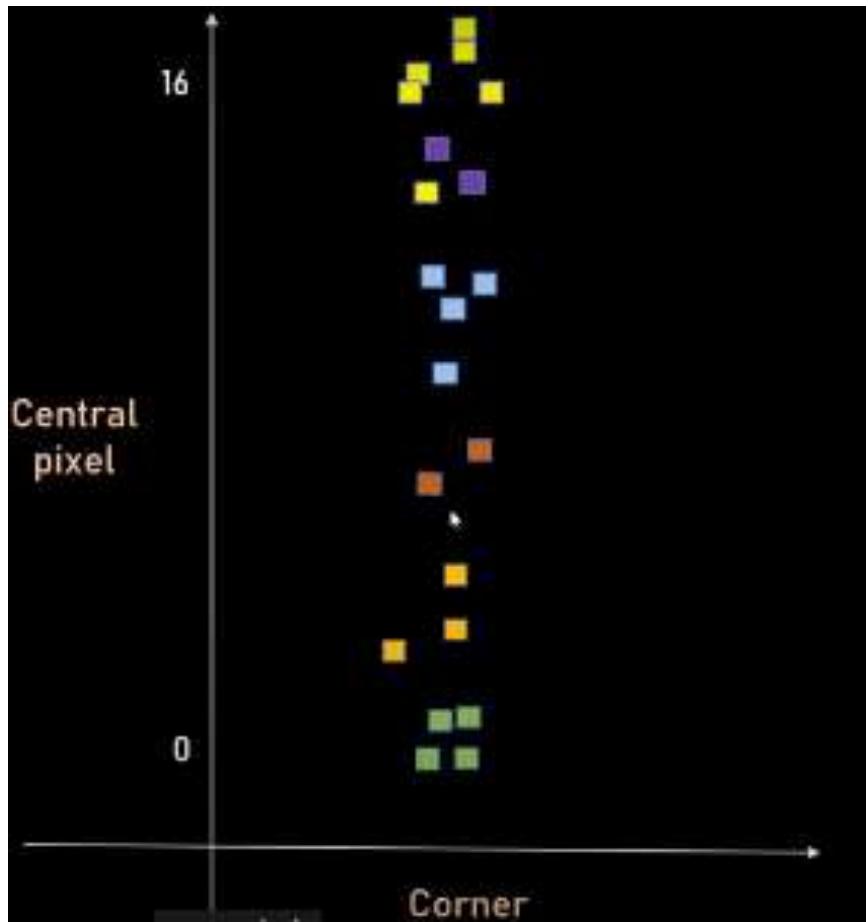
town	area	bathroom	plot	trees nearby	price
monroe	2600	2	8500	2	550000
monroe	3000	3	9200	2	565000
monroe	3200	3	8750	2	610000
monroe	3600	4	10200	2	680000
monroe	4000	4	15000	2	725000
west windsor	2600	2	7000	2	585000
west windsor	2800	3	9000	2	615000
west windsor	3300	4	10000	1	650000
west windsor	3600	4	10500	1	710000

Examples

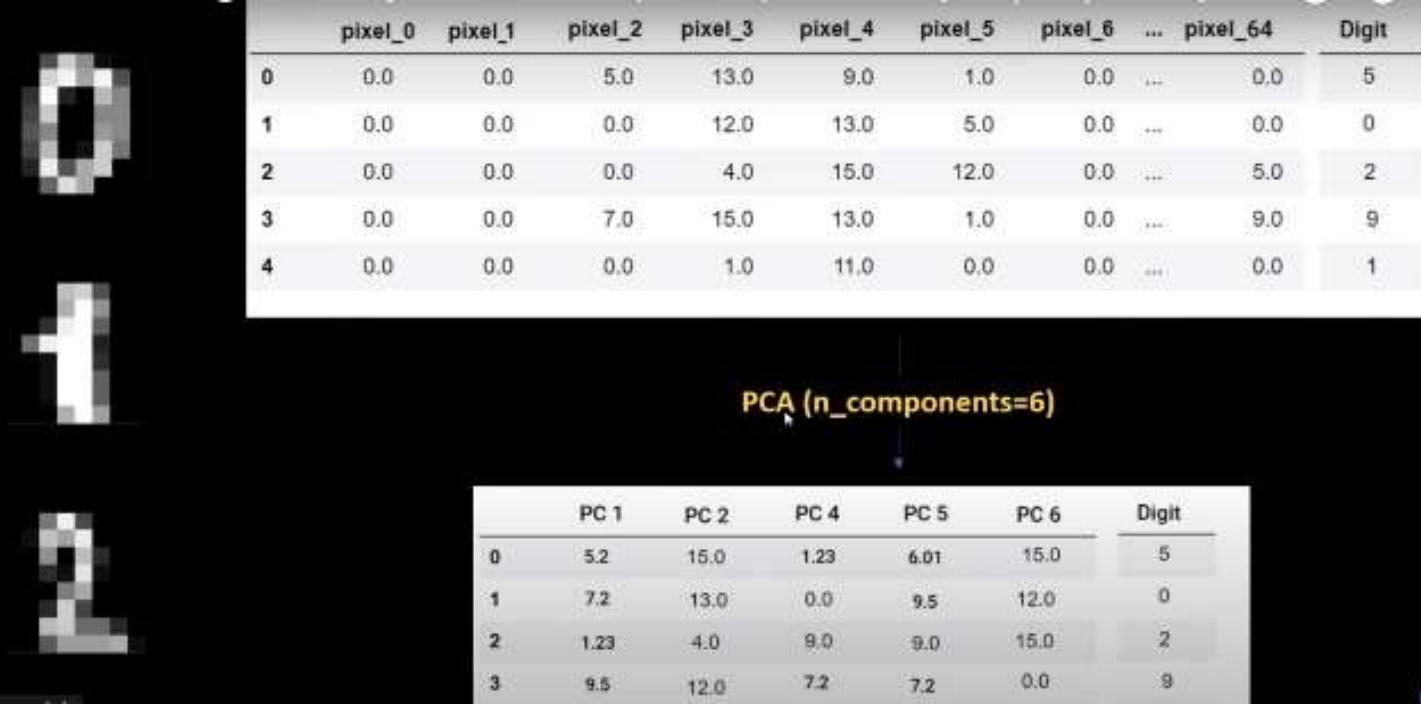


0	0	11	16	9	0	0	0
0	0	13	11	12	0	0	0
0	0	5	0	2	7	0	0
0	0	3	0	4	5	0	0
0	0	0	6	13	4	0	0
0	0	3	3	16	7	0	0
0	0	8	1	3	10	0	0
0	0	7	8	8	8	11	1





- In PCA, a new set of features are extracted from the original features
- The new set are quite dissimilar in nature
- So an n -dimensional feature space gets transformed to an m -dimensional feature space, where the dimensions are orthogonal to each other, i.e. completely independent of each other.



The figure illustrates a dataset of handwritten digits. On the left, three small grayscale images of digits are shown: a 5, a 0, and a 2. To the right of each image is a table showing the pixel values and the digit's classification.

Table 1: Raw Pixel Data

	pixel_0	pixel_1	pixel_2	pixel_3	pixel_4	pixel_5	pixel_6	...	pixel_64	Digit
0	0.0	0.0	5.0	13.0	9.0	1.0	0.0	...	0.0	5
1	0.0	0.0	0.0	12.0	13.0	5.0	0.0	...	0.0	0
2	0.0	0.0	0.0	4.0	15.0	12.0	0.0	...	5.0	2
3	0.0	0.0	7.0	15.0	13.0	1.0	0.0	...	9.0	9
4	0.0	0.0	0.0	1.0	11.0	0.0	0.0	...	0.0	1

PCA (n_components=6)

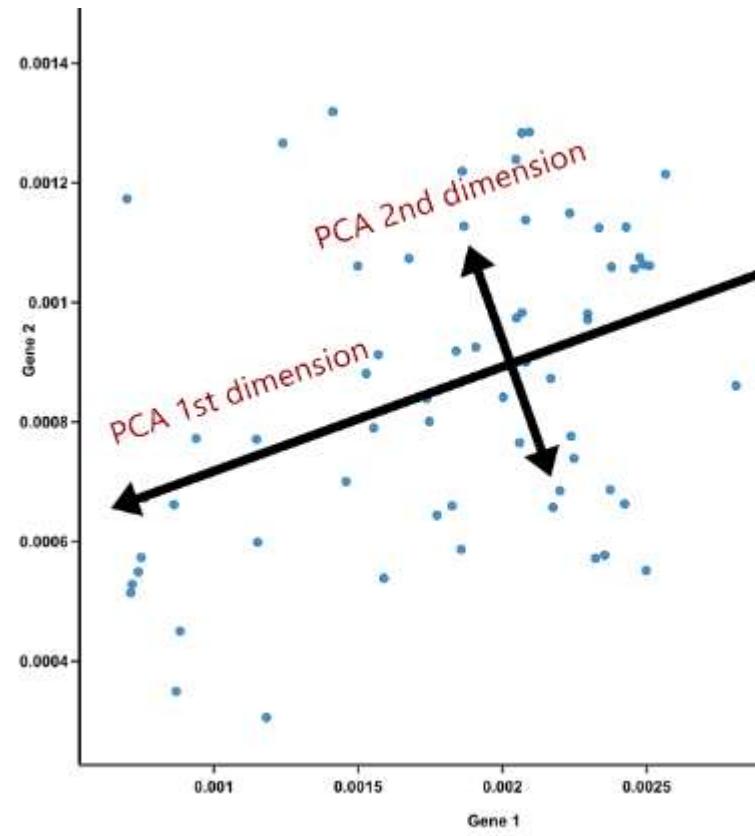
	PC 1	PC 2	PC 4	PC 5	PC 6	Digit
0	5.2	15.0	1.23	6.01	15.0	5
1	7.2	13.0	0.0	9.5	12.0	0
2	1.23	4.0	9.0	9.0	15.0	2
3	9.5	12.0	7.2	7.2	0.0	9

Principal components

- A set of feature vectors which may have similarity with each other is transformed to a set of principal components which are completely unrelated.
- Principal components are new variables that are constructed as linear combinations or mixtures of the initial variables.
- The principal components capture the variability of the original feature space.
- Also, the number of principal component derived, much like the basis vectors, is much smaller than the original set of features.

- The objective of PCA is to make the transformation in such a way that
 1. The new features are distinct, i.e. the covariance between the new features, i.e. the principal components is 0.
 2. The principal components are generated in order of the variability in the data that it captures. Hence, the first principal component should capture the maximum variability, the second principal component should capture the next highest variability etc.
 3. The sum of variance of the new features or the principal components should be equal to the sum of variance of the original features.

- A principal component is a normalized linear combination of the original features in a data set.
- The first principal component(PC1) will always be in the **direction of maximum variation** and then the other PC's follow.
- We need to note that all the PC's will be **perpendicular** to each other.
- The main intention behind this is that no information present in PC1 will be present in PC2 when they are perpendicular to each other.



- PCA works based on a process called eigenvalue decomposition of a covariance matrix of a data set. Below are the steps to be followed:
 1. First, calculate the covariance matrix of a data set.
 2. Then, calculate the eigenvalues of the covariance matrix.
 3. The eigenvector having highest eigenvalue represents the direction in which there is the highest variance. So this will help in identifying the first principal component.
 4. The eigenvector having the next highest eigenvalue represents the direction in which data has the highest remaining variance and also orthogonal to the first direction. So this helps in identifying the second principal component.
 5. Like this, identify the top ' k ' eigenvectors having top ' k ' eigenvalues so as to get the ' k ' principal components.

Step by step

- **STEP 1: STANDARDIZATION**
- The aim of this step is to standardize the range of the continuous initial variables so that each one of them contributes equally to the analysis.
- Mathematically, this can be done by subtracting the mean and dividing by the standard deviation for each value of each variable.

$$z = \frac{\text{value} - \text{mean}}{\text{standard deviation}}$$

- Once the standardization is done, all the variables will be transformed to the same scale.

- **STEP 2: COVARIANCE MATRIX COMPUTATION**
- The aim of this step is to understand how the variables of the input data set are varying from the mean with respect to each other, or in other words, to see if there is any relationship between them.
- The covariance matrix is a $p \times p$ symmetric matrix (where p is the number of dimensions) that has as entries the covariances associated with all possible pairs of the initial variables. For example, for a 3-dimensional data set with 3 variables x , y , and z , the covariance matrix is a 3×3 matrix of this form:

$$\text{Cov}(x,y) = \sum ((x\text{-mean})(y\text{-mean}))/(\text{n}-1)$$

$$\begin{bmatrix} \text{Cov}(x,x) & \text{Cov}(x,y) & \text{Cov}(x,z) \\ \text{Cov}(y,x) & \text{Cov}(y,y) & \text{Cov}(y,z) \\ \text{Cov}(z,x) & \text{Cov}(z,y) & \text{Cov}(z,z) \end{bmatrix}$$

FEATURE SUBSET SELECTION

- Feature subset selection is intended to derive a subset of features from the full feature set.
- No new feature is generated.
- The objective of feature selection is three-fold:
 - 1. Having faster and more cost-effective (i.e. less need for computational resources) learning model
 - 2. Improving the efficiency of the learning model
 - 3. Having a better understanding of the underlying model that generated the data
- Feature selection intends to remove all features which are irrelevant and take a representative subset of the features which are potentially redundant.
- This leads to a meaningful feature subset in context of a specific learning task.

Feature relevance

- In supervised learning , each of the predictor variables, is expected to contribute information to decide the value of the class label.
- If a variable is not contributing any information, it is said to be irrelevant.
- If the information contribution for prediction is very little, the variable is said to be weakly relevant.
- Remaining variables, which make a significant contribution to the prediction task are said to be strongly relevant variables.

- In unsupervised learning, there is no training data set or labelled data.
- Grouping of similar data instances are done and similarity of data instances are evaluated based on the value of different variables.
- Certain variables do not contribute any useful information for deciding the similarity or dissimilarity of data instances.
- Those variables make no significant information contribution in the grouping process. They are marked as irrelevant variables.
- Example- student data

Feature redundancy

- A feature may contribute information which is similar to the information contributed by one or more other features.
- A situation when one feature is similar to another feature, the feature is said to be potentially redundant in the context of the learning problem.
- All such features (Except few) having potential redundancy are candidates for rejection in the final feature subset.
- Example : Age, height, weight

Objective of feature selection

- The main objective of feature selection is
 - to remove all features which are irrelevant
 - and take a representative subset of the features which are potentially redundant.
- This leads to a meaningful feature subset in context of a specific learning task.

Measures of Feature redundancy

- There are multiple measures of similarity of information contribution,
 1. Correlation-based measures
 2. Distance-based measures, and

Correlation-based similarity measure

- Correlation is a measure of linear dependency between two random variables.
- For two random feature variables F_1 and F_2 , *Pearson correlation coefficient is defined as:*

$$\alpha = \frac{cov(F_1, F_2)}{\sqrt{var(F_1).var(F_2)}}$$

$$cov(F_1, F_2) = \sum (F_{1i} - \bar{F}_1).(F_{2i} - \bar{F}_2)$$

$$var(F_1) = \sum (F_{1i} - \bar{F}_1)^2, \text{ where } \bar{F}_1 = \frac{1}{n} \cdot \sum F_{1i}$$

$$var(F_2) = \sum (F_{2i} - \bar{F}_2)^2, \text{ where } \bar{F}_2 = \frac{1}{n} \cdot \sum F_{2i}$$

- Correlation values range between +1 and -1.
- A correlation of 1 (+ / -) indicates perfect correlation,
- A correlation of 0, indicate no linear relationship.

Distance-based similarity measure

- The most common distance measure is the **Euclidean distance**, which, between two features F_1 and F_2 are calculated as:

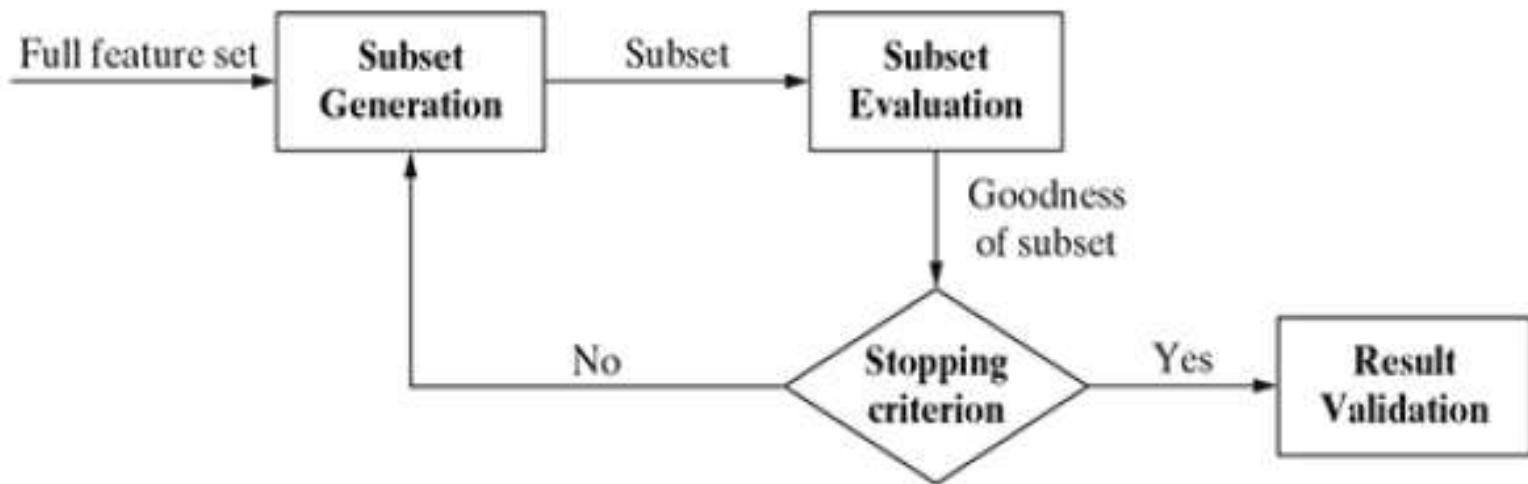
$$d(F_1, F_2) = \sqrt{\sum_{i=1}^n (F_{1i} - F_{2i})^2}$$

- Example :

Aptitude (F_1)	Communication (F_2)	$(F_1 - F_2)$	$(F_1 - F_2)^2$
2	6	-4	16
3	5.5	-2.5	6.25
6	4	2	4
7	2.5	4.5	20.25
8	3	5	25
6	5.5	0.5	0.25
6	7	-1	1
7	6	1	1
8	6	2	4
9	7	2	4

Overall feature selection process

- A typical feature selection process consists of four steps:
 1. generation of possible subsets
 2. subset evaluation
 3. stop searching based on some stopping criterion
 4. validation of the result



Feature selection process

- **Subset generation**, which is the first step of any feature selection algorithm
- It is a search procedure which ideally should produce all possible candidate subsets.
- for an *n-dimensional* data set, 2^n subsets can be generated.
- So, as the value of ‘*n*’ becomes high, *finding an optimal subset from all* the 2^n candidate subsets becomes difficult
- So, to solve this problem, different approximate search strategies are employed to find candidate subsets for evaluation.

Feature selection process- Conti...

- Different approximate search strategies are
 - the search may start with an empty set and keep adding features. This search strategy is termed as a sequential forward selection.
 - On the other hand, a search may start with a full set and successively remove features. This strategy is termed as sequential backward elimination.
 - In certain cases, search start with both ends and add and remove features simultaneously. This strategy is termed as a bi-directional selection.
- Each candidate subset is then evaluated and compared with the previous best performing subset based on certain **evaluation criterion**.
- If the new subset performs better, it replaces the previous one.

Feature selection process- Conti...

- This cycle of subset generation and evaluation continues till a pre-defined **stopping criterion is fulfilled.**
- Some commonly used stopping criteria are
 1. the search completes
 2. some given bound (e.g. a specified number of iterations) is reached
 3. subsequent addition (or deletion) of the feature is not producing a better subset
 4. a sufficiently good subset (e.g. a subset having better classification accuracy than the existing benchmark) is selected

Feature selection process- Conti...

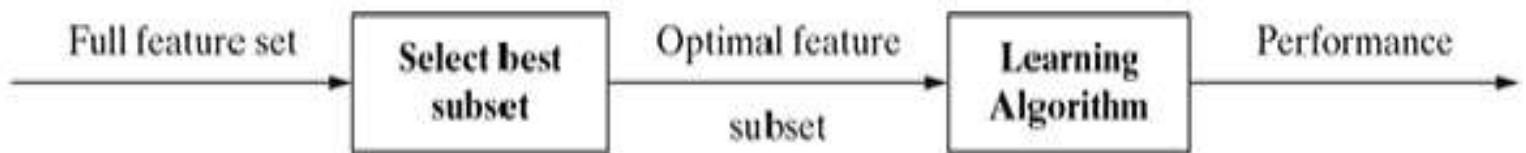
- Then the selected best subset is **validated** either against prior benchmarks or by experiments using real-life or synthetic but authentic data sets.
- In case of supervised learning, the accuracy of the learning model may be the performance parameter considered for validation.
- The accuracy of the model using the subset derived is compared against the model accuracy of the subset derived using some other benchmark algorithm.
- In case of unsupervised, the cluster quality may be the parameter for validation.

Feature selection approaches

- There are four types of approach for feature selection:
 1. Filter approach
 2. Wrapper approach
 3. Hybrid approach
 4. Embedded approach

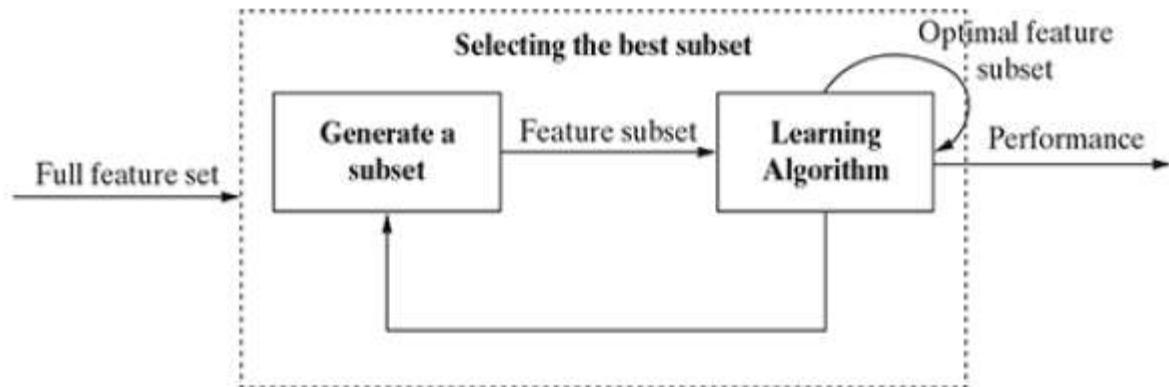
filter approach

- In the filter approach , the feature subset is selected based on statistical measures done to assess the merits of the features from the data perspective.
- No learning algorithm is employed to evaluate the goodness of the feature selected.
- Some of the common statistical tests conducted on features as a part of filter approach are – Pearson’s correlation, information gain, Fisher score, analysis of variance (ANOVA), Chi-Square, etc.



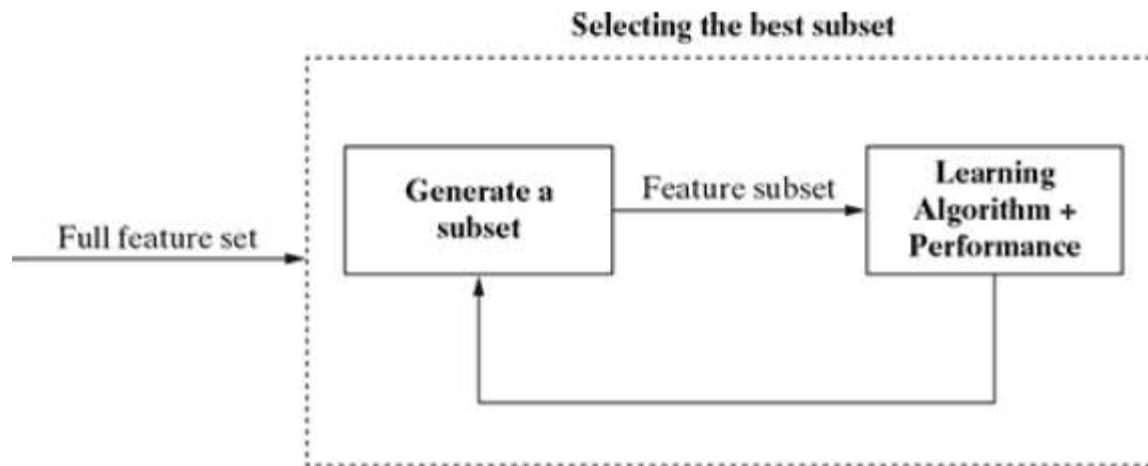
wrapper approach

- In this approach, identification of best feature subset is done using the induction algorithm as a black box.
- The feature selection algorithm searches for a good feature subset using the induction algorithm itself as a part of the evaluation function.
- Since for every candidate subset, the learning model is trained and the result is evaluated by running the learning algorithm, wrapper approach is computationally very expensive.
- the performance is generally superior compared to filter approach.



Embedded approach

- Embedded approach is quite similar to wrapper approach as it also uses inductive algorithm to evaluate the generated feature subsets.
- The difference is it performs feature selection and classification simultaneously.



Thankyou

Supervised Learning- Classification

Unit 3

Supervised Learning

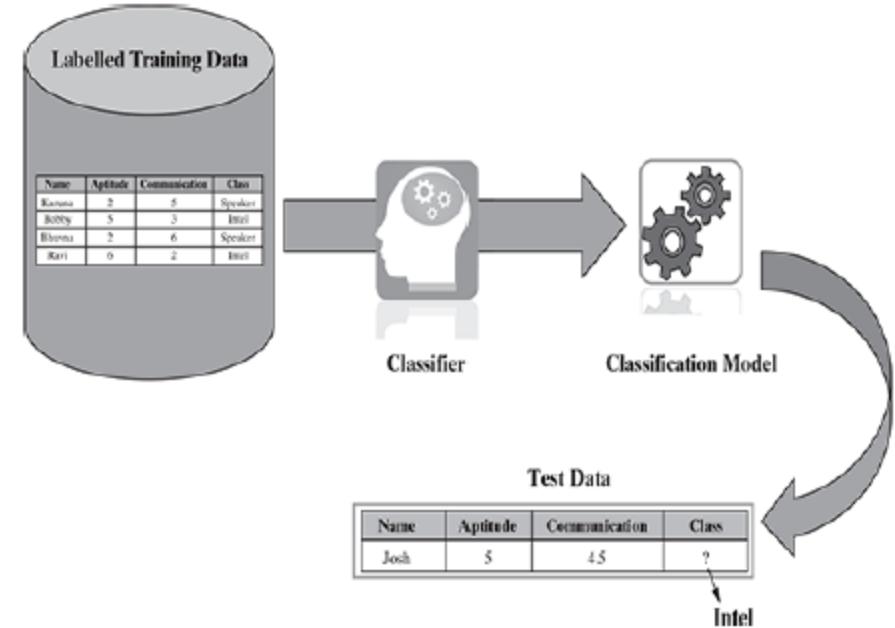
- In supervised learning, the labelled training data provide the basis for learning.
- Here the supervisor is the training data.
- According to the definition of machine learning, this labelled training data is the experience or prior knowledge or belief with known value of class field or '**label**'.
- Examples -
 - Patients in Hospital
 - Prediction of results of a game based on the past analysis of results
 - Predicting whether a tumour is malignant or benign on the basis of the analysis of data
 - Price prediction in domains such as real estate, stocks, etc.

CLASSIFICATION MODEL

- When we are trying to predict a categorical or nominal variable, the problem is known as a classification problem.
- A classification problem is one where the output variable is a category such as ‘red’ or ‘blue’ or ‘malignant tumour’ or ‘benign tumour’, etc.
- In classification, the whole problem centres around assigning a label or category or class to a test data on the basis of the label or category or class information that is imparted by the training data.
- Because the target objective is to assign a class label, we call this type of problem as a classification problem.

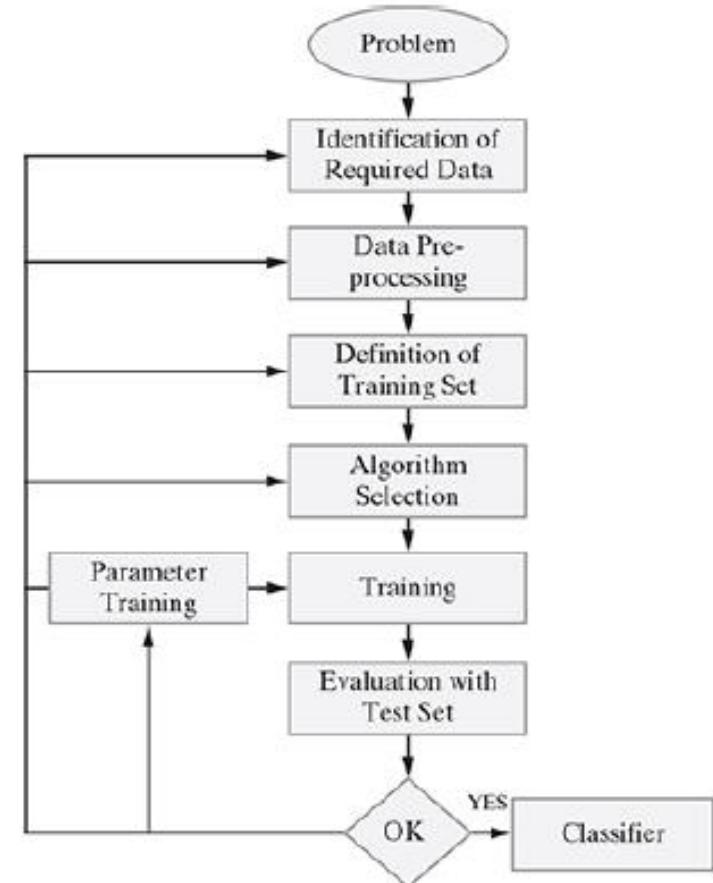
Summery

- Classification is a type of supervised learning where a target feature, which is of categorical type, is predicted for test data on the basis of the information imparted by the training data.
- The target categorical feature is known as **class** .
- Some typical classification problems include the following:
 - Image classification
 - Disease prediction
 - Win-loss prediction of games
 - Prediction of natural calamity such as earthquake, flood, etc.
 - Handwriting recognition



CLASSIFICATION LEARNING STEPS

- Identify the problem
- The required data (related to the problem, which is already stored in the system) is evaluated and pre-processed based on the algorithm.
- Algorithm selection is a critical point in supervised learning.
- The result after iterative training rounds is a classifier for the problem in hand



CLASSIFICATION LEARNING STEPS

Problem Identification:

- Identifying the problem is the first step in the supervised learning model.
- The problem needs to be a well-formed problem, i.e. a problem with well-defined goals and benefit, which has a long-term impact.

Identification of Required Data:

- On the basis of the problem identified , the required data set that precisely represents the identified problem needs to be identified/evaluated.

CLASSIFICATION LEARNING STEPS

Data Pre-processing:

- This is related to the cleaning/transforming the data set.
- This step ensures that all the unnecessary/irrelevant data elements are removed.
- Data pre-processing refers to the transformations applied to the identified data before feeding the same into the algorithm.
- As the data is gathered from different sources, it is usually collected in a raw format and is not ready for immediate analysis.
- This step ensures that the data is ready to be fed into the machine learning algorithm.

CLASSIFICATION LEARNING STEPS

Definition of Training Data Set:

- Before starting the analysis, the user should decide what kind of data set is to be used as a training set.
- A set of ‘input meta-objects’ and corresponding ‘output meta-objects’ are also gathered.
- The training set needs to be actively representative of the real-world use of the given scenario.
- Thus, a set of data input (X) and *corresponding* outputs (Y) is gathered either from human experts or experiments.

CLASSIFICATION LEARNING STEPS

Algorithm Selection:

- This involves determining the structure of the learning function and the corresponding learning algorithm.
- This is the most critical step of supervised learning model.
- On the basis of various parameters, the best algorithm for a given problem is chosen.

Training:

- The learning algorithm identified in the previous step is run on the gathered training set for further fine tuning.
- Some supervised learning algorithms require the user to determine specific control parameters (which are given as inputs to the algorithm).
- These parameters may also be adjusted by optimizing performance on a subset (called as validation set) of the training set.

CLASSIFICATION LEARNING STEPS

Evaluation with the Test Data Set:

- Test data is run on the algorithm, and its performance is measured here.
- If a suitable result is not obtained, further training of parameters may be required.

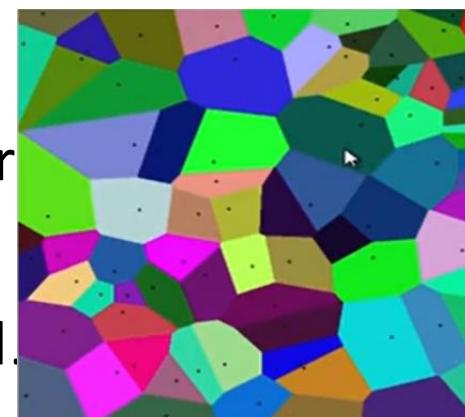
COMMON CLASSIFICATION ALGORITHMS

- Following are the most common classification algorithms,
 1. *k-Nearest Neighbor (kNN)*
 2. Decision tree
 3. Random forest
 4. Support Vector Machine (SVM)
 5. Naïve Bayes classifier

K-NEAREST NEIGHBOR (KNN)

k -Nearest Neighbour (kNN)

- The *kNN algorithm* is a simple but extremely powerful classification algorithm. It is called Lazy algorithm.
- The name of the algorithm originates from the underlying philosophy of *kNN* – *i.e. people having* similar background or mindset tend to stay close to each other.
- In the same way, as a part of the *kNN algorithm*, the unknown and unlabelled data which comes for a prediction problem is judged on the basis of the training data set elements which are similar to the unknown element.
- It is called Lazy algorithm. No training is required.
- Can be used for Classification and regression



Voronoi Diagram

Example

- Consider a very simple Student data set
- It consists of 15 students studying in a class. Each of the students has been assigned a score on a scale of 10 on two performance parameters – ‘Aptitude’ and ‘Communication’.
- Also, a class value is assigned to each student based on the following criteria:
 - 1. Students having good communication skills as well as a good level of aptitude have been classified as ‘Leader’
 - 2. Students having good communication skills but not so good level of aptitude have been classified as ‘Speaker’
 - 3. Students having not so good communication skill but a good level of aptitude have been classified as ‘Intel’

Name	Aptitude	Communication	Class
Karuna	2	5	Speaker
Bhuvna	2	6	Speaker
Gaurav	7	6	Leader
Parul	7	2.5	Intel
Dinesh	8	6	Leader
Jani	4	7	Speaker
Bobby	5	3	Intel
Parimal	3	5.5	Speaker
Govind	8	3	Intel
Susant	6	5.5	Leader
Gouri	6	4	Intel
Bharat	6	7	Leader
Ravi	6	2	Intel
Pradeep	9	7	Leader
Josh	5	4.5	Intel

kNN

- First 14 students data are taken as training data
- Last student's data is taken as test data
- In the *kNN algorithm*, the class label of the test data elements is decided by the class label of the training data elements which are neighbouring, i.e. similar in nature.

But there are two challenges:

1. What is the basis of this similarity or when can we say that two data elements are similar?
2. How many similar elements should be considered for deciding the class label of each test data element?

Name	Aptitude	Communication	Class
Karuna	2	5	Speaker
Bhuvna	2	6	Speaker
Gaurav	7	6	Leader
Parul	7	2.5	Intel
Dinesh	8	6	Leader
Jani	4	7	Speaker
Bobby	5	3	Intel
Parimal	3	5.5	Speaker
Govind	8	3	Intel
Susant	6	5.5	Leader
Gouri	6	4	Intel
Bharat	6	7	Leader
Ravi	6	2	Intel
Pradeep	9	7	Leader
Josh	5	4.5	Intel

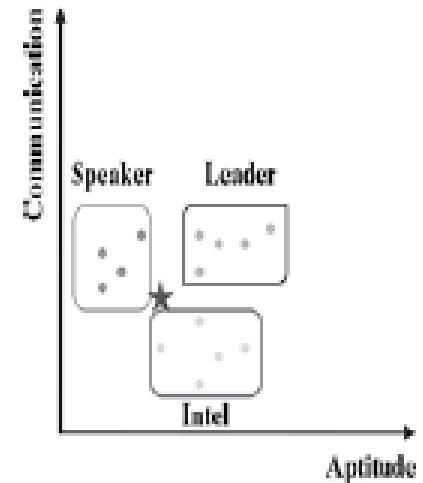
1. What is the basis of this similarity or when can we say that two data elements are similar?

- To measure similarity between two data elements kNN uses Euclidean distance
- Considering a very simple data set having two features (say f_1 and f_2)
- *Euclidean distance between two data elements d_1 and d_2 can be measured by*

$$\text{Euclidean distance} = \sqrt{(f_{11} - f_{12})^2 + (f_{21} - f_{22})^2}$$

- where f_{11} = value of feature f_1 for data element d_1
- f_{12} = value of feature f_1 for data element d_2
- f_{21} = value of feature f_2 for data element d_1
- f_{22} = value of feature f_2 for data element d_2

- The training data points can be represented as dots in a two-dimensional feature space.
- The test data point for student Josh is represented as an asterisk in the same space.
- To find out the closest or nearest neighbours of the test data point, Euclidean distance of the different dots need to be calculated from the asterisk.
- Then, the class value of the closest neighbours helps in assigning the class value of the test data element.



How many similar elements should be considered for deciding the class label of each test data element?

- In the *kNN algorithm*, the value of 'k' indicates the number of neighbours that need to be considered.
- It is a user-defined parameter given as an input to the algorithm.

Name	Aptitude	Communication	Class	Distance	k = 1	k = 2	k = 3
Karuna	2	5	Speaker	3.041			
Bhuvna	2	6	Speaker	3.354			
Parimal	3	5.5	Speaker	2.236			
Jani	4	7	Speaker	2.693			
Bobby	5	3	Intel	1.500			1.500
Ravi	6	2	Intel	2.693			
Gouri	6	4	Intel	1.118	1.118	1.118	1.118
Parul	7	2.5	Intel	2.828			
Govind	8	3	Intel	3.354			
Susant	6	5.5	Leader	1.414			
Bharat	6	7	Leader	2.693			
Gaurav	7	6	Leader	2.500			
Dinesh	8	6	Leader	3.354			
Pradeep	9	7	Leader	4.717			
Josh	5	4.5	???				

K value

- It is often a tricky decision to decide the value of k . *The reasons are as follows:*
 - If the value of k is *very large* (*in the extreme case equal to the total number* of records in the training data), the class label of the majority class of the training data set will be assigned to the test data regardless of the class labels of the neighbours nearest to the test data.
 - If the value of k is *very small* (*in the extreme case equal to 1*), *the class value* of a noisy data or outlier in the training data set which is the nearest neighbour to the test data will be assigned to the test data.
- The best k value is *somewhere between these two extremes*.

K value

- Few strategies are adopted by machine learning practitioners to arrive at a value for k .
- One common practice is to set k *equal to the square root of the number of training records*.
- An alternative approach is to test several *k values on a variety of test data sets* and choose the one that delivers the best performance.
- Another interesting approach is to choose a larger value of k , *but apply a weighted voting process* in which the vote of close neighbours is considered more influential than the vote of distant neighbours.

kNN algorithm

- **Input:** Training data set, test data set (or data points), value of '*k*' (*i.e.* number of nearest neighbours to be considered)

Steps:

Do for all test data points

Calculate the distance (usually Euclidean distance) of the test data point from the different training data points.

Find the closest '*k*' *training data points*, *i.e.* *training data points whose* distances are least from the test data point.

If *k* = 1

Then assign class label of the training data point to the test data point

Else

 Whichever class label is predominantly present in the training data points, assign that class label to the test data point

End do

Strengths and Weaknesses

- *Strengths of the kNN algorithm*
 - Extremely simple algorithm – easy to understand
 - Very effective in certain situations, e.g. for recommender system design
 - Very fast or almost no time required for the training phase
- *Weaknesses of the kNN algorithm*
 - Does not learn anything in the real sense. Classification is done completely on the basis of the training data. So, it has a heavy reliance on the training data. If the training data does not represent the problem domain comprehensively, the algorithm fails to make an effective classification.
 - Because there is no model trained in real sense and the classification is done completely on the basis of the training data, the classification process is very slow.
 - Also, a large amount of computational space is required to load the training data for classification.

Application of the kNN algorithm

- One of the most popular areas in machine learning where the *kNN algorithm is widely adopted is recommender systems*
- Document classification
- Video recommendations

1 Classification example

Document Classification Task

Classes - Political Documents and Academic Documents

	x1	x2	class	Eucli Dist
d1	7	7	c1	4
d2	7	4	c2	5
d3	3	4	c1	3
d4	1	4	c1	3.605551
d5	3	7	?	k=3

$$\text{Euclidian Distance} = \sqrt{(x_{1i}-3)^2 + (x_{2i}-7)^2}$$

Take 3 nearest values. We get

d3 c1
d4 c1
d1 c1

So, d5 is assigned to c1

2 Regression Example

sr. no	height	age	weight	Eucl dist
1	6	40	60	3.006659
2	6.11	26	65	11.00437
3	5.9	30	56	7.000714
4	5.8	32	58	5
5	5.3	33	75	4.031129
6	5.6	34	78	3.006659
7	5.5	35	80	2.022375
8	5.8	37	?	

Take nearest 3 values.

They are = 2.02, 3.00, 3.00

Their corresponding weights are = 80

78

60

Average = **72.66667**

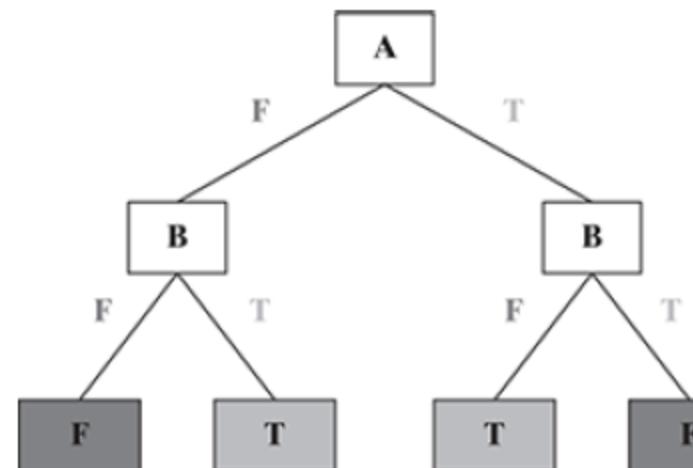
for a person having height = 5.8, age = 37 the weight is 72.66

DECISION TREE

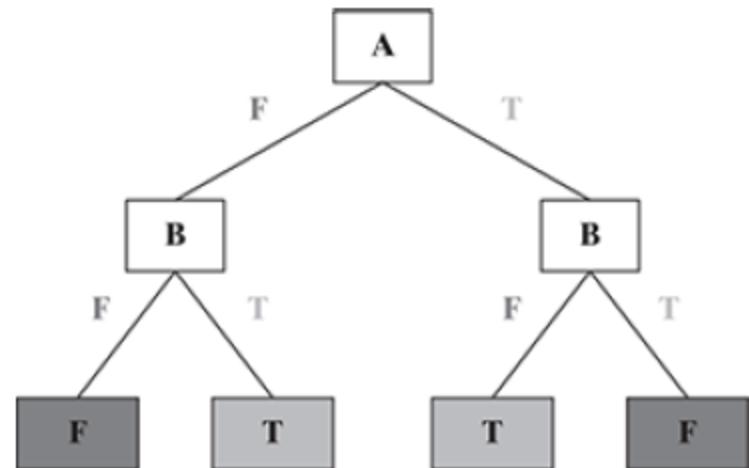
Decision tree

- Decision tree learning is one of the most widely adopted algorithms for classification.
- It builds a model in the form of a tree structure.
- Its grouping exactness is focused with different strategies, and it is exceptionally productive.
- A decision tree is used for multi-dimensional analysis with multiple classes.
- It is characterized by fast execution time and ease in the interpretation of the rules.
- The goal of decision tree learning is to create a model (based on the past data called past vector) that predicts the value of the output variable based on the input variables in the feature vector

- Each node (or decision node) of a decision tree corresponds to one of the feature vector.
- From every node, there are edges to children, wherein there is an edge for each of the possible values (or range of values) of the feature associated with the node.
- The tree terminates at different leaf nodes (or terminal nodes) where each leaf node represents a possible value for the output variable.
- The output variable is determined by following a path that starts at the root and is guided by the values of the input variables.

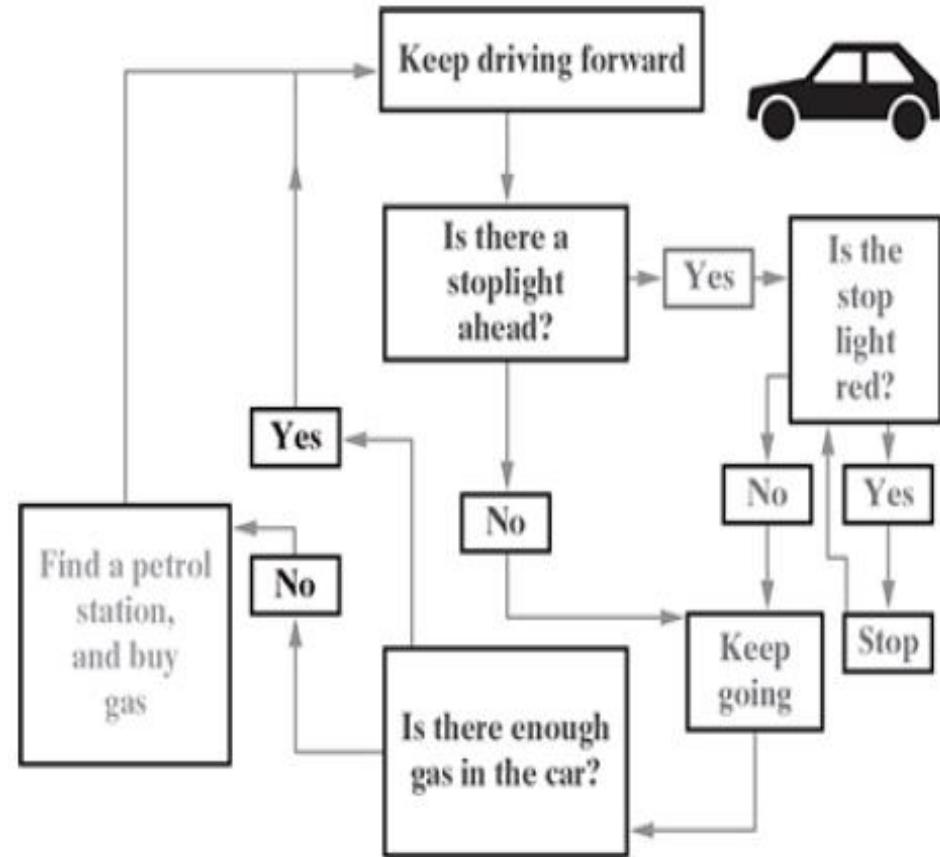


- A decision tree consists of three types of nodes:
 - Root Node
 - Branch Node
 - Leaf Node
- Each internal node (represented by boxes) tests an attribute (represented as ‘A’/‘B’ within the boxes).
- Each branch corresponds to an attribute value (T/F) in the above case.
- Each leaf node assigns a classification.

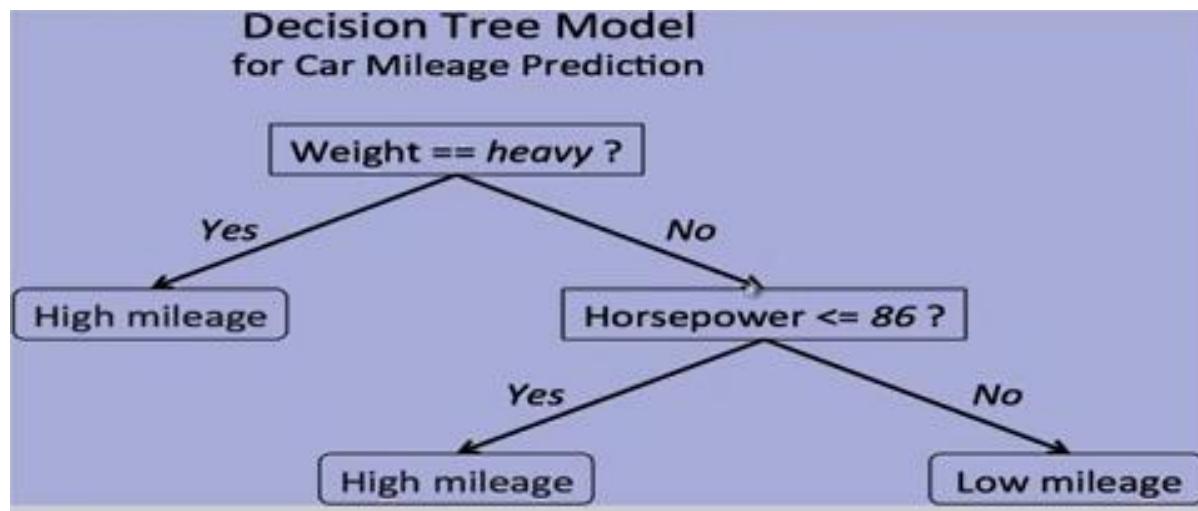


Example

- An example decision tree for a car driving – the decision to be taken is whether to ‘Keep Going’ or to ‘Stop’, which depends on various situations.
- If the signal is RED in colour, then the car should be stopped.
- If there is not enough gas (petrol) in the car, the car should be stopped at the next available gas station.



Example



Building a decision tree

- Decision trees are built corresponding to the training data following an approach called recursive partitioning.
- The approach splits the data into multiple subsets on the basis of the feature values.
- It starts from the root node, which is nothing but the entire data set.
- It first selects the feature which predicts the target class in the strongest way.
- The decision tree splits the data set into multiple partitions, with data in each partition having a distinct value for the feature based on which the partitioning has happened.
- This is the first set of branches.

- Likewise, the algorithm continues splitting the nodes on the basis of the feature which helps in the best partition.
- This continues till a stopping criterion is reached.
- The usual stopping criteria are –
 1. All or most of the examples at a particular node have the same class
 2. All features have been used up in the partitioning
 3. The tree has grown to a pre-defined threshold limit

Example

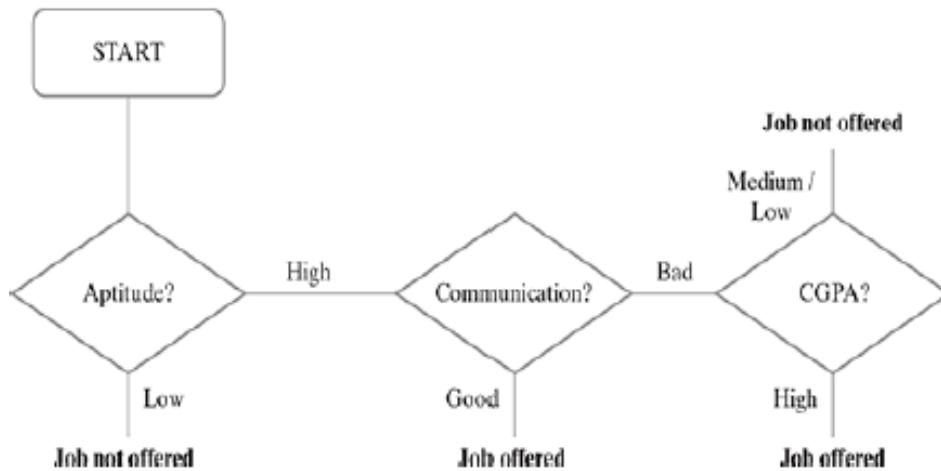
- Global Technology Solutions (GTS), a leading provider of IT solutions, is coming to College of Engineering and Management (CEM) for hiring B.Tech. students.
- Last year during campus recruitment, they had shortlisted 18 students for the final interview.
- Being a company of international repute, they follow a stringent interview process to select only the best of the students.
- The information related to the interview evaluation results of shortlisted students (hiding the names) on the basis of different evaluation parameters is available for reference.
- Chandra, a student of CEM, wants to find out if he may be offered a job in GTS. His CGPA is quite high.
- His self-evaluation on the other parameters is as follows:
Communication – Bad; Aptitude – High; Programming skills – Bad

CGPA	Communication	Aptitude	Programming Skill	Job offered
High	Good	High	Good	Yes
Medium	Good	High	Good	Yes
Low	Bad	Low	Good	No
Low	Good	Low	Bad	No
High	Good	High	Bad	Yes
High	Good	High	Good	Yes
Medium	Bad	Low	Bad	No
Medium	Bad	Low	Good	No
High	Bad	High	Good	Yes
Medium	Good	High	Good	Yes
Low	Bad	High	Bad	No
Low	Bad	High	Bad	No
Medium	Good	High	Bad	Yes
Low	Good	Low	Good	No
High	Bad	Low	Bad	No
Medium	Bad	High	Good	No
High	Bad	Low	Bad	No
Medium	Good	High	Bad	Yes

- Let us try to solve this problem, i.e. predicting whether Chandra will get a job offer, by using the decision tree model.
- First, we need to draw the decision tree corresponding to the training data given.
- According to the table, job offer condition (i.e. the outcome) is FALSE for all the cases where **Aptitude = Low**, irrespective of other conditions.
- So, the feature Aptitude can be taken up as the first node of the decision tree.
- For **Aptitude = High**, job offer condition is TRUE for all the cases where **Communication = Good**.
- For cases where **Communication = Bad**, job offer condition is TRUE for all the cases where **CGPA = High**.

decision tree

- the complete decision tree diagram for the data given in table is



- By using the decision tree we need to predict whether Chandra might get a job offer for the given parameter values: **CGPA = High, Communication = Bad, Aptitude = High, Programming skills = Bad.**

Searching a decision tree

- There are multiple ways to search through the trained decision tree for a solution to the given prediction problem.
 - *Exhaustive search*
 - *Branch and bound search*

Exhaustive search

1. Place the item in the first group (class). Recursively examine solutions with the item in the first group (class).
 2. Place the item in the second group (class). Recursively examine solutions with the item in the second group (class).
 3. Repeat the above steps until the solution is reached.
- Exhaustive search travels through the decision tree exhaustively, but it will take much time when the decision tree is big with multiple leaves and multiple attribute values.

Branch and bound search

- Branch and bound uses an existing best solution to sidestep searching of the entire decision tree in full.
- When the algorithm starts, the best solution is well defined to have the worst possible value; thus, any solution it finds out is an improvement.
- This makes the algorithm initially run down to the left-most branch of the tree, even though that is unlikely to produce a realistic result.
- In the partitioning problem, that solution corresponds to putting every item in one group, and it is an unacceptable solution.
- A programme can speed up the process by using a fast heuristic to find an initial solution.
- This can be used as an input for branch and bound.
- If the heuristic is right, the savings can be substantial.

- The biggest challenge of a decision tree algorithm is to find out which feature to split upon.
- The main driver for identifying the feature is that the data should be split in such a way that the partitions created by the split should contain examples belonging to a single class.
- If that happens, the partitions are considered to be **pure**.
- Entropy is a measure of impurity of an attribute or feature.
- The information gain is calculated on the basis of the decrease in entropy (S) after a data set is split according to a particular attribute (A).
- Constructing a decision tree is all about finding an attribute that returns the highest information gain (i.e. the most homogeneous branches).

Entropy of a decision tree

- Let us say S is the sample set of training examples.
- Then, Entropy (S) measuring the impurity of S is defined as

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

- where c is the number of different class labels and p refers to the proportion of values falling into the *i-th class label*.
- For example, with respect to the training data, we have two values for the target class ‘Job Offered?’ – Yes and No. The value of p for class value ‘Yes’ is 0.44 (i.e. 8/18) and that for class value ‘No’ is 0.56 (i.e. 10/18).
- So, we can calculate the entropy as

$$\text{Entropy}(S) = -0.44 \log_2(0.44) - 0.56 \log_2(0.56) = 0.99.$$

Information gain of a decision tree

- The information gain is created on the basis of the decrease in entropy (S) *after a data set is split according to a particular attribute (A)*.
- *Constructing a decision tree is all about finding* an attribute that returns the highest information gain (i.e. the most homogeneous branches).
- If the information gain is 0, it means that there is no reduction in entropy due to split of the data set according to that particular feature.
- On the other hand, the maximum amount of information gain which may happen is the entropy of the data set before the split.

- Information gain for a particular feature A is calculated by the difference in entropy before a split (or S) *with the* entropy after the split (S).

$$\text{Information Gain } (S, A) = \text{Entropy } (S_{bs}) - \text{Entropy } (S_{as})$$

- For calculating the entropy after split, entropy for all partitions needs to be considered.
- Then, the weighted summation of the entropy for each partition can be taken as the total entropy after split.
- For performing weighted summation, the proportion of examples falling into each partition is used as weight.

$$\text{Entropy}(S_{as}) = \sum_{i=1}^n w_i \text{Entropy}(p_i)$$

- Consider the training data set
- We will find the value of entropy at the beginning before any split happens and then again after the split happens.
- We will compare the values for all the cases –
 1. when the feature ‘CGPA’ is used for the split
 2. when the feature ‘Communication’ is used for the split
 3. when the feature ‘Aptitude’ is used for the split
 4. when the feature ‘Programming Skills’ is used for the split
- As calculated, entropy of the data set before split (i.e. Entropy (S_0)) = 0.99, and entropy of the data set after split (i.e. Entropy (S_1)) is
 - 0.69 when the feature ‘CGPA’ is used for split
 - 0.63 when the feature ‘Communication’ is used for split
 - 0.52 when the feature ‘Aptitude’ is used for split
 - 0.95 when the feature ‘Programming skill’ is used for split

CGPA	Communication	Aptitude	Programming Skill	Job offered
High	Good	High	Good	Yes
Medium	Good	High	Good	Yes
Low	Bad	Low	Good	No
Low	Good	Low	Bad	No
High	Good	High	Bad	Yes
High	Good	High	Good	Yes
Medium	Bad	Low	Bad	No
Medium	Bad	Low	Good	No
High	Bad	High	Good	Yes
Medium	Good	High	Good	Yes
Low	Bad	High	Bad	No
Low	Bad	High	Bad	No
Medium	Good	High	Bad	Yes
Low	Good	Low	Good	No
High	Bad	Low	Bad	No
Medium	Bad	High	Good	No
High	Bad	Low	Bad	No
Medium	Good	High	Bad	Yes

(a) Original data set:

	Yes	No	Total
Count	8	10	18
pi	0.44	0.56	
-pi*log(pi)	0.52	0.47	0.99

Total Entropy = 0.99

(b) Splitted data set (based on the feature 'CGPA'):

CGPA = High			CGPA = Medium			CGPA = Low					
	Yes	No	Total		Yes	No	Total		Yes	No	Total
Count	4	2	6	Count	4	3	7	Count	0	5	5
pi	0.67	0.33		pi	0.57	0.43		pi	0.00	1.00	
-pi*log(pi)	0.39	0.53	0.92	-pi*log(pi)	0.46	0.52	0.99	-pi*log(pi)	0.00	0.00	0.00

Total Entropy = 0.69

Information Gain = 0.30

(c) Splitted data set (based on the feature 'Communication'):

Communication = Good

	Yes	No	Total
Count	7	2	9
pi	0.78	0.22	
-pi*log(pi)	0.28	0.48	0.76

Total Entropy = 0.63

Communication = Bad

	Yes	No	Total
Count	1	8	9
pi	0.11	0.89	
-pi*log(pi)	0.35	0.15	0.50

Information Gain = 0.36

(d) Splitted data set (based on the feature 'Aptitude'):

Aptitude = High

	Yes	No	Total
Count	8	3	11
pi	0.73	0.27	
-pi*log(pi)	0.33	0.51	0.85

Total Entropy = 0.52

Aptitude = Low

	Yes	No	Total
Count	0	7	7
pi	0.00	1.00	
-pi*log(pi)	0.00	0.00	0.00

Information Gain = 0.47

(e) Splitted data set (based on the feature 'Programming Skill'):

Programming Skill = Good

	Yes	No	Total
Count	5	4	9
pi	0.56	0.44	
-pi*log(pi)	0.47	0.52	0.99

Total Entropy = 0.95

Programming Skill = Bad

	Yes	No	Total
Count	3	6	9
pi	0.33	0.67	
-pi*log(pi)	0.53	0.39	0.92

Information Gain = 0.04

Calculations

$$\text{Information Gain } (S, A) = \text{Entropy } (S_{\text{bs}}) - \text{Entropy } (S_{\text{as}})$$

For CGPA

$$\begin{aligned}\text{Entropy}(S_{\text{as}}) &= \sum_{i=1}^3 w_i * \text{Entropy}(p_i) \\ &= \sum_{\text{High, low, medium}} w_{\text{high}} \text{Entropy}_{\text{high}} * w_{\text{low}} \text{Entropy}_{\text{low}} * \\ &\quad w_{\text{medium}} \text{Entropy}_{\text{medium}} \\ &= (6/18)*0.92 + (7/18)*0.99 + (5/18)*0 = 0.306 + 0.385 = 0.691\end{aligned}$$

$$\begin{aligned}\text{Information Gain}(S, \text{CGPA}) &= \text{Entropy}(S_{\text{original dataset}}) - \text{Entropy}(S_{\text{as}}) \\ &= 0.99 - 0.691 = 0.30\end{aligned}$$

For Communication

$$\begin{aligned}\text{Entropy}(S_{\text{as}}) &= \sum_{i=1}^2 w_i * \text{Entropy}(p_i) \\ &= \sum_{\text{good, bad}} w_{\text{good}} \text{Entropy}_{\text{good}} * w_{\text{bad}} \text{Entropy}_{\text{bad}} \\ &= (9/18)*0.76 + (9/18)*0.50 = 0.38 + 0.25 = 0.63\end{aligned}$$

$$\begin{aligned}\text{Information Gain}(S, \text{CGPA}) &= \text{Entropy}(S_{\text{original dataset}}) - \text{Entropy}(S_{\text{as}}) \\ &= 0.99 - 0.63 = 0.36\end{aligned}$$

- The information gain from the feature ‘CGPA’ = $0.99 - 0.69 = 0.3$,
- The information gain from the feature ‘Communication’ = $0.99 - 0.63 = 0.36$. Likewise, the information gain for ‘Aptitude’ and ‘Programming skills’ is 0.47 and 0.04, respectively.
- It is quite evident that among all the features, ‘Aptitude’ results in the best information gain when adopted for the split.
- So, at the first level, a split will be applied according to the value of ‘Aptitude’
- For **Aptitude = Low, entropy is 0**, which indicates that always the result will be the same irrespective of the values of the other features.
- Hence, the branch towards Aptitude = Low will not continue any further.

- As a part of level 2, we will thus have only one branch to navigate in this case – the one for **Aptitude = High**.
- The entropy value is as follows:
 - 0.85 before the split
 - 0.33 when the feature ‘CGPA’ is used for split
 - 0.30 when the feature ‘Communication’ is used for split
 - 0.80 when the feature ‘Programming skill’ is used for split

Aptitude = High

CGPA	Communication	Programming Skill	Job offered?
High	Good	Good	Yes
Medium	Good	Good	Yes
High	Good	Bad	Yes
High	Good	Good	Yes
High	Bad	Good	Yes
Medium	Good	Good	Yes
Low	Bad	Bad	No
Low	Bad	Bad	No
Medium	Good	Bad	Yes
Medium	Bad	Good	No
Medium	Good	Bad	Yes

(a) Level 2 starting set:

	Yes	No	Total
Count	8	3	11
pi	0.73	0.27	
-pi*log(pi)	0.33	0.51	0.85

Total Entropy = 0.85

(b) Splitted data set (based on the feature 'CGPA'):

CGPA = High			CGPA = Medium			CGPA = Low					
	Yes	No	Total		Yes	No	Total		Yes	No	Total
Count	4	0	4	Count	4	1	5	Count	0	2	2
pi	1.00	0.00		pi	0.80	0.20		pi	0.00	1.00	
-pi*log(pi)	0.00	0.00	0.00	-pi*log(pi)	0.26	0.46	0.72	-pi*log(pi)	0.00	0.00	0.00

Total Entropy = 0.33

Information Gain = 0.52

(c) Splitted data set (based on the feature 'Communication'):

Communication = Good

	Yes	No	Total
Count	7	0	7
pi	1.00	0.00	

Total Entropy = 0.30

Communication = Bad

	Yes	No	Total
Count	1	3	4
pi	0.25	0.75	

Information Gain = 0.55

(d) Splitted data set (based on the feature 'Programming Skill'):

Programming Skill = Good

	Yes	No	Total
Count	5	1	6
pi	0.83	0.17	

Total Entropy = 0.80

Programming Skill = Bad

	Yes	No	Total
Count	3	2	5
pi	0.60	0.40	

Information Gain = 0.05

- The information gain after split with the features CGPA, Communication and Programming Skill is 0.52, 0.55 and 0.05, respectively.
- So, the feature Communication should be used for this split as it results in the highest information gain.
- At the second level, a split will be applied on the basis of the value of ‘Communication’.
- For **Communication = Good**, entropy is 0, which indicates that always the result will be the same irrespective of the values of the other features.
- Hence, the branch towards Communication = Good will not continue any further.

Aptitude = High & Communication = Bad

CGPA	Programming Skill	Job offered?
High	Good	Yes
Low	Bad	No
Low	Bad	No
Medium	Good	No

(a) Level 2 starting set:

	Yes	No	Total
Count	1	3	4
pi	0.25	0.75	
-pi*log(pi)	0.50	0.31	0.81

Total Entropy = 0.81

(b) Splitted data set (based on the feature 'CGPA'):

CGPA = High			CGPA = Medium			CGPA = Low					
	Yes	No	Total		Yes	No	Total		Yes	No	Total
Count	1	0	1	Count	0	1	1	Count	0	2	2
pi	1.00	0.00		pi	0.00	1.00		pi	0.00	1.00	
-pi*log(pi)	0.00	0.00	0.00	-pi*log(pi)	0.00	0.00	0.00	-pi*log(pi)	0.00	0.00	0.00

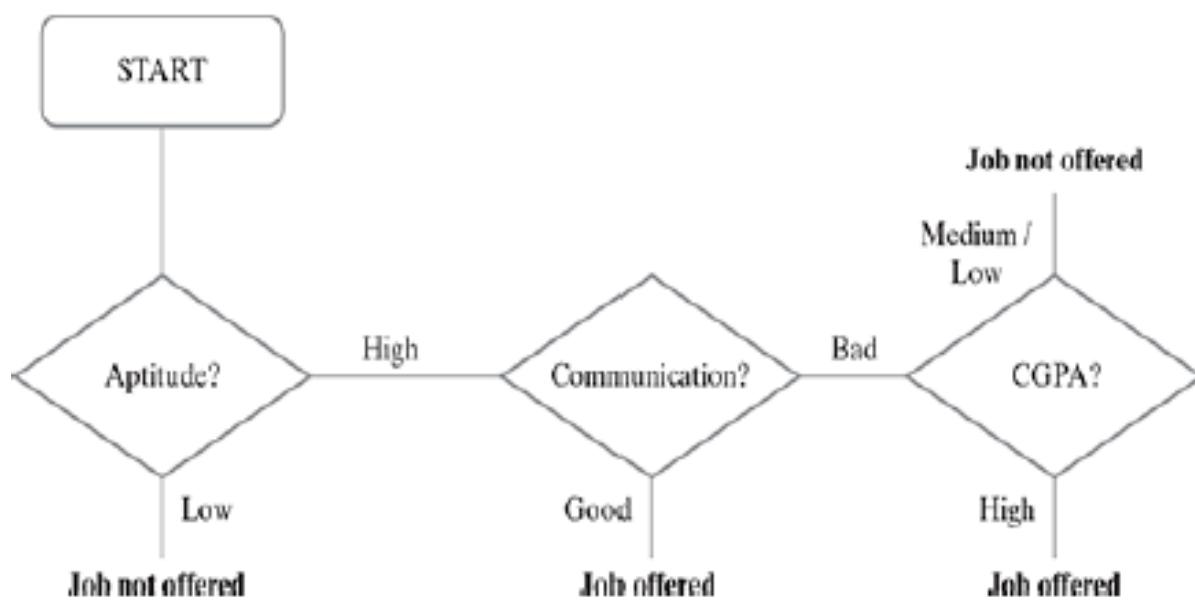
Total Entropy = 0.00 Information Gain = 0.81

(c) Splitted data set (based on the feature 'Programming Skill'):

Programming Skill = Good			Programming Skill = Bad				
	Yes	No	Total		Yes	No	Total
Count	1	1	2	Count	0	2	2
pi	0.50	0.50		pi	0.00	1.00	
-pi*log(pi)	0.50	0.50	1.00	-pi*log(pi)	0.00	0.00	0.00

Total Entropy = 0.50 Information Gain = 0.31

- As a part of level 3, we will thus have only one branch to navigate in this case – the one for **Communication = Bad**.
- As can be seen, the entropy value is as follows:
 - 0.81 before the split
 - 0 when the feature ‘CGPA’ is used for split
 - 0.50 when the feature ‘Programming Skill’ is used for split
- The information gain after split with the feature CGPA is 0.81, which is the maximum possible information gain (as the entropy before split was 0.81).
- So, a split will be applied on the basis of the value of ‘CGPA’.
- Because the maximum information gain is already achieved, the tree will not continue any further.



Algorithm for decision tree

- **Input:** Training data set, test data set (or data points)
- **Steps:**

Do for all attributes

Calculate the entropy E_i of the attribute F_i

if $E_i < E_{min}$

 then $E_{min} = E_i$ and $F_{min} = F_i$

end if

End do

Split the data set into subsets using the attribute F_{min}

Draw a decision tree node containing the attribute F_{min} and split the data set into subsets

Repeat the above steps until the full tree is drawn covering all the attributes of the original table.

Avoiding overfitting in decision tree – pruning

- Unless a stopping criterion is applied, the decision tree algorithm may keep growing indefinitely – splitting for every feature and dividing into smaller partitions till the point that the data is perfectly classified.
- This results in overfitting problem.
- To prevent a decision tree getting overfitted to the training data, pruning of the decision tree is essential.
- Pruning a decision tree reduces the size of the tree such that the model is more generalized and can classify unknown and unlabelled data in a better way.

- There are two approaches of pruning:
 - Pre-pruning:
 - Stop growing the tree before it reaches perfection/it reaches a certain number of decision nodes or decisions. .
 - It also has a chance to ignore important information contributed by a feature which was skipped, thereby resulting in miss out of certain patterns in the data.
 - Post-pruning:
 - Allow the tree to grow entirely and then post-prune some of the branches from it, by using certain pruning criterion, e.g. error rates at the nodes, the size of the tree is reduced
 - This is a more effective approach in terms of classification accuracy as it considers all minute information available from the training data.
 - The computational cost is obviously more than that of pre-pruning.

Strengths of decision tree

- It produces very simple understandable rules.
- For smaller trees, not much mathematical and computational knowledge is required to understand this model.
- Works well for most of the problems.
- It can handle both numerical and categorical variables.
- Can work well both with small and large training data sets.
- Decision trees provide a definite clue of which features are more useful for classification

Weaknesses of decision tree

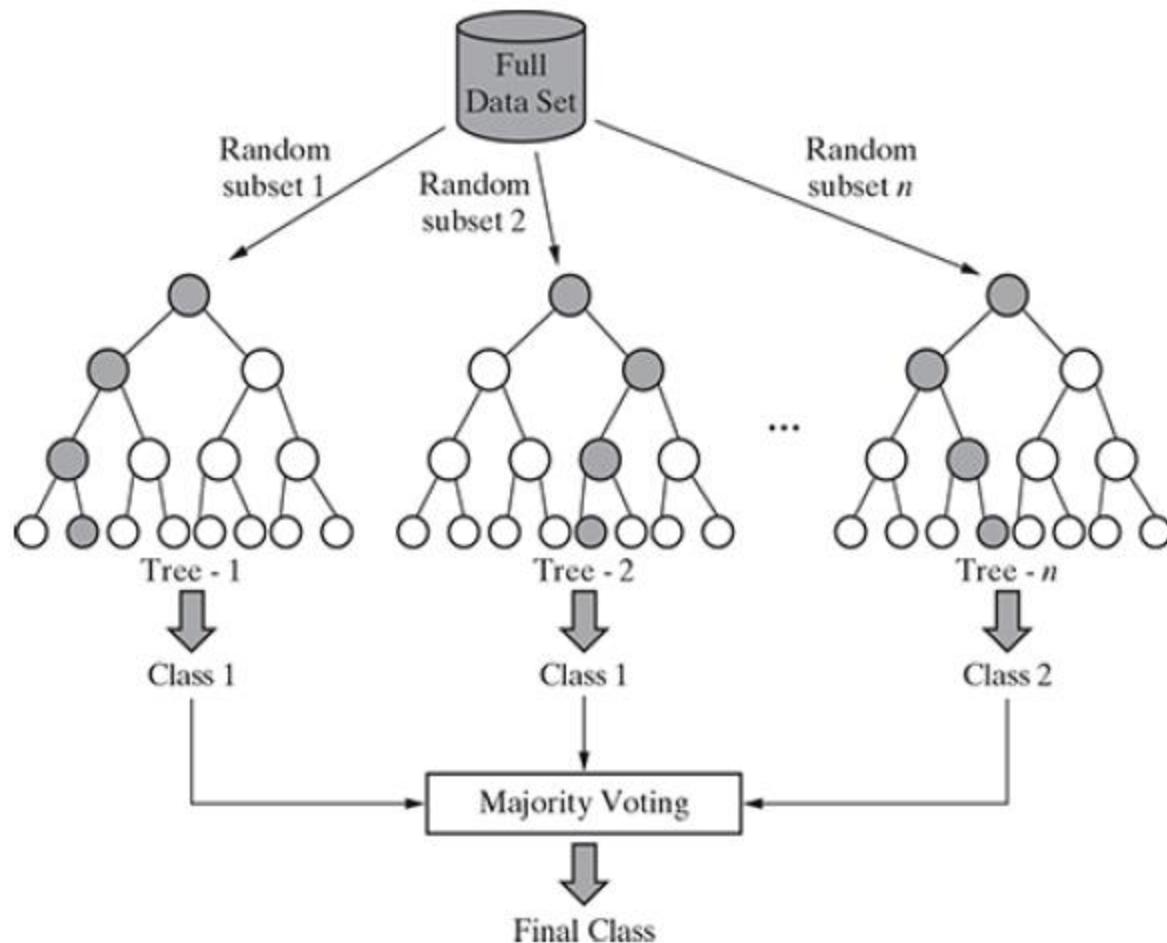
- Decision tree models are often biased towards features having more number of possible values, i.e. levels.
- This model gets overfitted or underfitted quite easily.
- Decision trees are prone to errors in classification problems with many classes and relatively small number of training examples.
- A decision tree can be computationally expensive to train.
- Large trees are complex to understand.

RANDOM FOREST MODEL

Random forest model

- Random forest is an ensemble classifier, i.e. a combining classifier that uses and combines many decision tree classifiers.
- Ensembling is usually done using the concept of bagging with different feature sets.
- The reason for using large number of trees in random forest is to train the trees enough such that contribution from each feature comes in a number of models.
- After the random forest is generated by combining the trees, majority vote is applied to combine the output of the different trees.
- The result from the ensemble model is usually better than that from the individual decision tree models.

Random forest model



Random forest algorithm

1. If there are N variables or features in the input data set, select a subset of ' m ' ($m < N$) features at random out of the N features. Also, the observations or data instances should be picked randomly.
2. Use the best split principle on these ' m ' features to calculate the number of nodes ' d '.
3. Keep splitting the nodes to child nodes till the tree is grown to the maximum possible extent.
4. Select a different subset of the training data 'with replacement' to train another decision tree following steps (1) to (3). Repeat this to build and train ' n ' decision trees.
5. Final class assignment is done on the basis of the majority votes from the ' n ' trees.

Strengths of random forest

- It runs efficiently on large and expansive data sets.
- It has a robust method for estimating missing data and maintains precision when a large proportion of the data is absent.
- It has powerful techniques for balancing errors in a class population of unbalanced data sets.
- It gives estimates (or assessments) about which features are the most important ones in the overall classification.
- It generates an internal unbiased estimate (gauge) of the generalization error as the forest generation progresses.
- Generated forests can be saved for future use on other data.
- Lastly, the random forest algorithm can be used to solve both classification and regression problems.

Weaknesses of random forest

- This model, because it combines a number of decision tree models, is not as easy to understand as a decision tree model.
- It is computationally much more expensive than a simple model like decision tree.

Application of random forest

- Random forest is a very powerful classifier which combines the versatility of many decision tree models into a single model.
- Because of the superior results, this ensemble model is gaining wide adoption and popularity amongst the machine learning practitioners to solve a wide range of classification problems.

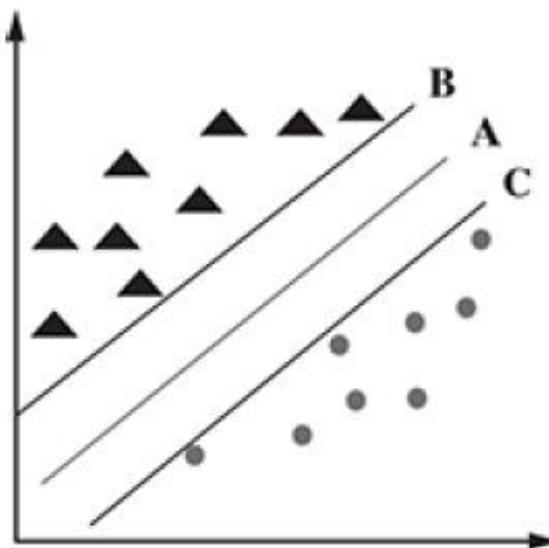
SUPPORT VECTOR MACHINES

Support vector machines

- SVM is a model, which can do linear classification as well as regression.
- SVM is based on the concept of a surface, called a hyperplane, which draws a boundary between data instances plotted in the multi-dimensional feature space.
- The output prediction of an SVM is one of two conceivable classes which are already defined in the training data.
- The SVM algorithm builds an N-dimensional hyperplane model that assigns future instances into one of the two possible output classes.

Support Vectors

- Support vectors are the data points (representing classes), the critical component in a data set, which are near the identified set of lines (hyperplane).
- If support vectors are removed, they will alter the position of the dividing hyperplane.



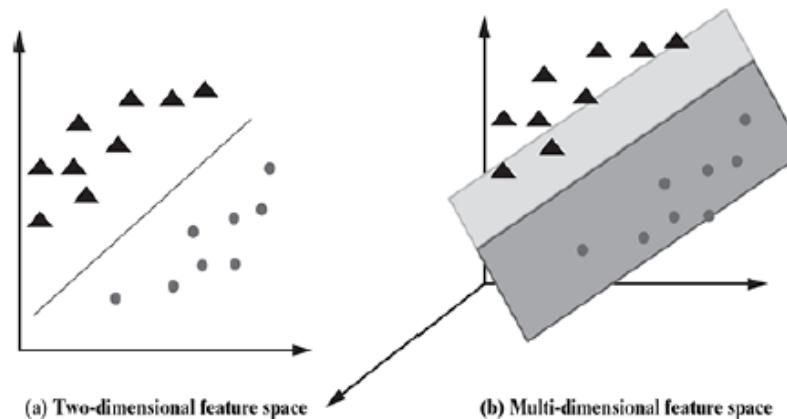
Hyperplane and Margin

- For an N-dimensional feature space, hyperplane is a flat subspace of dimension (N-1) that separates and classifies a set of data.
- A two-dimensional feature space has two features and a class variable
- For a two-dimensional feature space hyperplane will be a one-dimensional subspace or a straight line.
- A three-dimensional feature space has three features and a class variable
- For a three-dimensional feature space the hyperplane is a two-dimensional subspace or a simple plane.
- It is difficult to visualize a feature space greater than three dimensions, much like for a subspace or hyperplane having more than three dimensions.

- Mathematically, in a two-dimensional space, hyperplane can be defined by the equation:
 $c + c_1X_1 + c_2X_2 = 0$, *an equation of a straight line.*
- Extending this concept to an *N-dimensional space*, hyperplane can be defined by the equation:
 $c + c_1X_1 + c_2X_2 + \dots + c_nX_n = 0$
- The further (or more distance) from the hyperplane the data points lie, the more confident we can be about correct categorization.
- So, when a new testing data point/data set is added, the side of the hyperplane it lands on will decide the class that we assign to it.
- The distance between hyperplane and data points is known as **margin**.

Classification using hyperplanes

- In SVM, a model is built to discriminate the data instances belonging to different classes.
- when mapped in a two-dimensional space, the data instances belonging to different classes fall in different sides of a straight line drawn in the two-dimensional space
- In a multidimensional feature space, the straight line dividing data instances belonging to different classes transforms to a hyperplane



- An SVM model is a representation of the input instances as points in the feature space, which are mapped so that an apparent gap between them divides the instances of the separate classes.
- The goal of the SVM analysis is to find a plane, or rather a hyperplane, which separates the instances on the basis of their classes
- In the overall training process, the SVM algorithm analyses input data and identifies a surface in the multi-dimensional feature space called the hyperplane.
- There may be many possible hyperplanes, and one of the challenges with the SVM model is to find the optimal hyperplane.

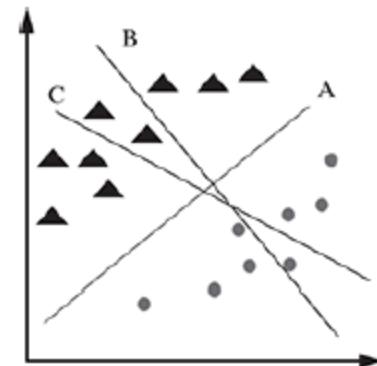
- Training data sets which have a substantial large group size will function well with SVM.
- Generalization error in SVM is the measure of how accurately and precisely this SVM model can predict values for previously unseen data (new data).
- A hard margin in terms of SVM means that an SVM model is inflexible in classification and tries to work exceptionally fit in the training set, thereby causing overfitting.

Identifying the correct hyperplane in SVM

- There may be multiple options for hyperplanes dividing the data instances belonging to the different classes.
- We need to identify which one will result in the best classification.

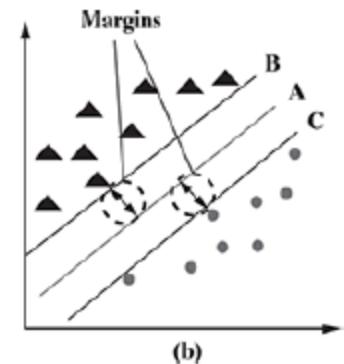
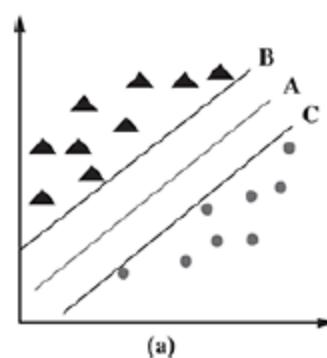
Scenario 1

- In this scenario, we have three hyperplanes: A, B, and C.
- We need to identify the correct hyperplane which better segregates the two classes represented by the triangles and circles.
- Hyperplane ‘A’ has performed this task quite well.



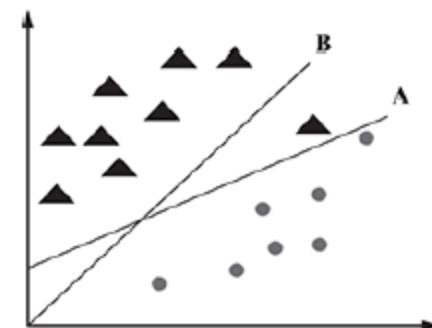
Scenario 2

- We have three hyperplanes: A, B, and C.
- We have to identify the correct hyperplane which classifies the triangles and circles in the best possible way.
- Maximizing the distances between the nearest data points of both the classes and hyperplane will help us decide the correct hyperplane. This distance is called as **margin**.
- The margin for hyperplane A is high as compared to those for both B and C.
- Hence, hyperplane A is the correct hyperplane.



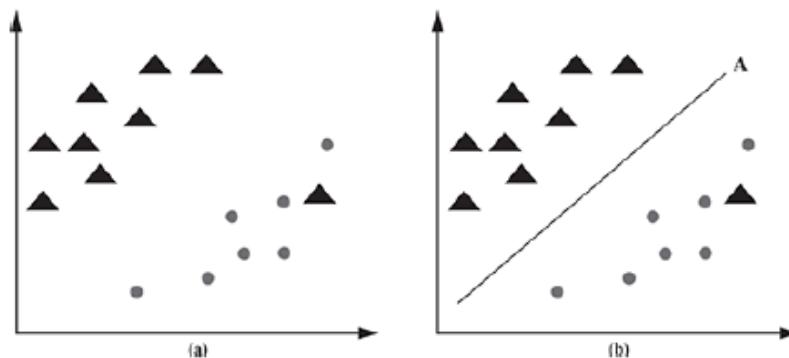
Scenario 3

- Use the rules as discussed in the previous section to identify the correct hyperplane
- B has a higher margin (distance from the class) than A.
- SVM selects the hyperplane which classifies the classes accurately before maximizing the margin.
- Here, hyperplane B has a classification error, and A has classified all data instances correctly.
- Therefore, A is the correct hyperplane.



Scenario 4

- It is not possible to distinctly segregate the two classes by using a straight line, as one data instance belonging to one of the classes (triangle) lies in the territory of the other class (circle) as an outlier.
- SVM has a feature to ignore outliers and find the hyperplane that has the maximum margin



Summery

1. The hyperplane should segregate the data instances belonging to the two classes in the best possible way.
2. It should maximize the distances between the nearest data points of both the classes, i.e. maximize the margin.
3. If there is a need to prioritize between higher margin and lesser misclassification, the hyperplane should try to reduce misclassifications.

Strengths of SVM

- SVM can be used for both classification and regression.
- It is robust, i.e. not much impacted by data with noise or outliers.
- The prediction results using this model are very promising.

Weaknesses of SVM

- SVM is applicable only for binary classification, i.e. when there are only two classes in the problem domain.
- The SVM model is very complex – almost like a black box when it deals with a high-dimensional data set. Hence, it is very difficult and close to impossible to understand the model in such cases.
- It is slow for a large dataset, i.e. a data set with either a large number of features or a large number of instances.
- It is quite memory-intensive.

Application of SVM

- SVM is most effective when it is used for binary classification, i.e. for solving a machine learning problem with two classes.
- One common problem on which SVM can be applied is in the field of bioinformatics – more specifically, in detecting cancer and other genetic disorders.
- It can also be used in detecting the image of a face by binary classification of images into face and non-face components.

- Naïve Bayes is a simple technique for building classifiers
- It builds the models that assign class labels to problem instances.
- The basic idea of Bayes rule is that the outcome of a hypothesis can be predicted on the basis of some evidence (E) that can be observed.
- From Bayes rule, we observed that
 1. A prior probability of hypothesis h or $P(h)$: This is the probability of an event or hypothesis before the evidence is observed.
 2. A posterior probability of h or $P(h | D)$: This is the probability of an event after the evidence is observed within the population D .

Thankyou

Supervised Learning- Regression

Unit 4

Introduction to Regression Analysis

- Regression analysis is used to:
 - Predict the value of a dependent variable based on the value of at least one independent variable
 - Explain the impact of changes in an independent variable on the dependent variable
- Dependent variable: the variable we wish to predict or explain
- Independent variable: the variable used to predict or explain the dependent variable

COMMON REGRESSION ALGORITHMS

- The most common regression algorithms are
 - Simple linear regression
 - Multiple linear regression
 - Polynomial regression
 - Multivariate adaptive regression splines
 - Logistic regression
 - Maximum likelihood estimation (least squares)

Simple Linear Regression Model

A

- Only **one** independent variable, X
- Relationship between X and Y is described by a linear function
- Changes in Y are assumed to be related to changes in X
- Example – Real Estate

Simple Linear Regression Model

A

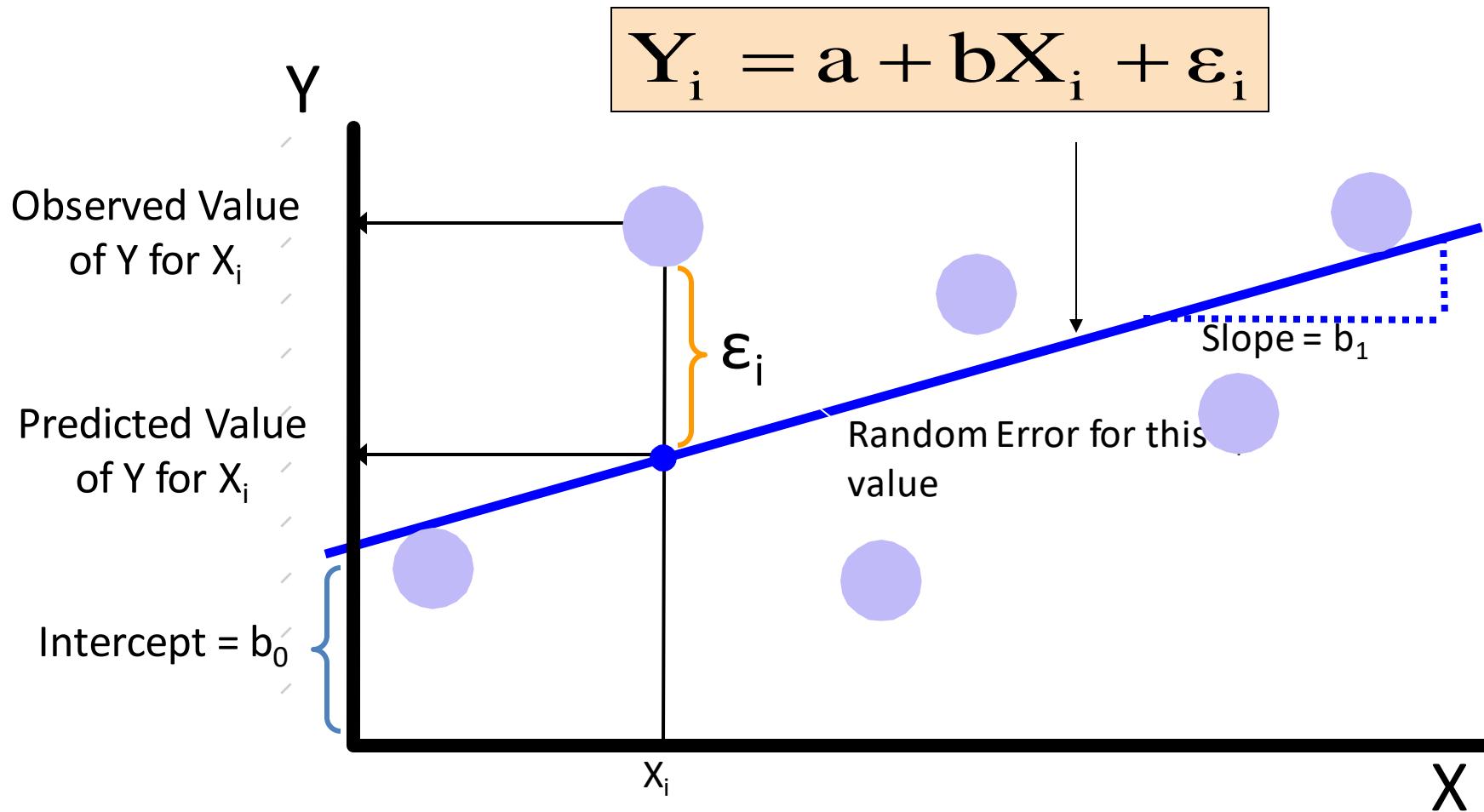
$$Y_i = a + bX_i + \varepsilon_i$$

Population Y intercept Population Slope Coefficient Independent Variable Random Error term

Linear component Random Error component

The diagram illustrates the components of the Simple Linear Regression Model equation. The equation is shown in a light orange box:
$$Y_i = a + bX_i + \varepsilon_i$$
. Above the box, four labels with red arrows point to their respective terms: "Population Y intercept" points to a , "Population Slope Coefficient" points to bX_i , "Independent Variable" points to X_i , and "Random Error term" points to ε_i . Below the box, a purple brace groups $a + bX_i$ as the "Linear component", and another purple brace groups ε_i as the "Random Error component".

Simple Linear Regression Model



Simple Linear Regression Equation (Prediction Line)

A

The simple linear regression equation provides an estimate of the population regression line

Estimated (or predicted) Y value for observation i

Estimate of the regression intercept

Estimate of the regression slope

Value of X for observation i

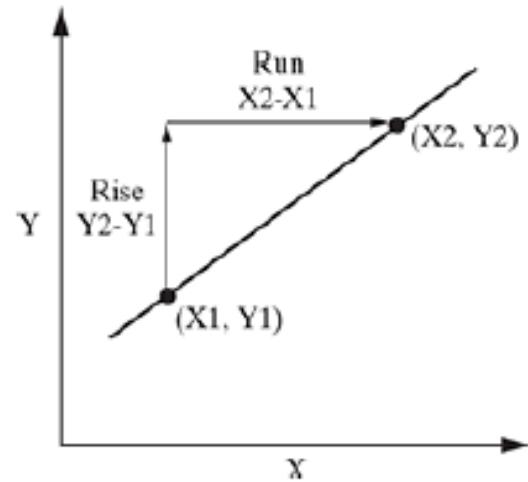
$$Y_i = a + bX_i + \varepsilon_i$$

Slope of the simple linear regression model

- Slope of a straight line represents how much the line in a graph changes in the vertical direction (*Y-axis*) over a change in the horizontal direction (*X-axis*)
Slope = Change in Y/Change in X
- Rise is the change in *Y-axis* ($Y - Y_1$) and Run is the change in *X-axis* ($X - X_1$). So,
- *Slope is represented as given below:*

$$\text{Slope} = \frac{\text{Rise}}{\text{Run}} = \frac{Y_2 - Y_1}{X_2 - X_1}$$

- *Find the slope for the line passing through (-1,-5) and (3,4)*



Types of slopes in LR

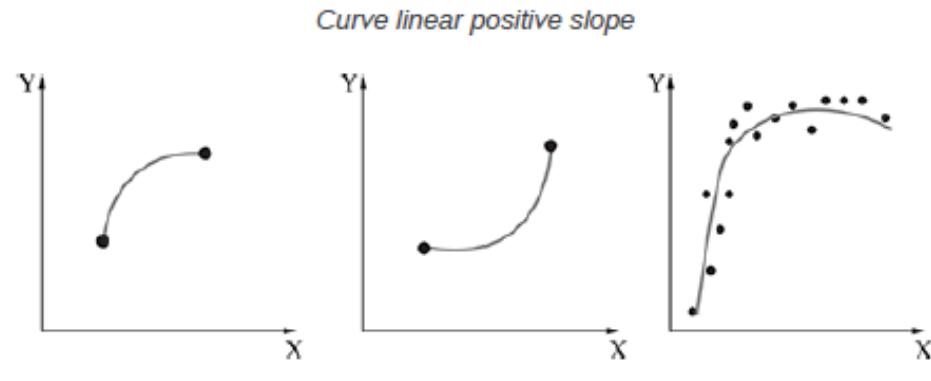
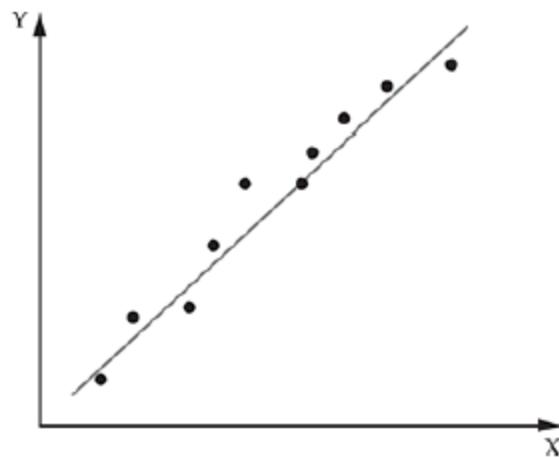
- Types of slopes
 - positive slope and negative slope
- Different types of regression lines based on the type of slope include
 - Linear positive slope
 - Curve linear positive slope
 - Linear negative slope
 - Curve linear negative slope

Linear positive slope

- A positive slope always moves upward on a graph from left to right

$$\text{Slope} = \text{Rise/Run} = (Y - Y_1) / (X - X_1) = \Delta(Y) / \Delta(X)$$

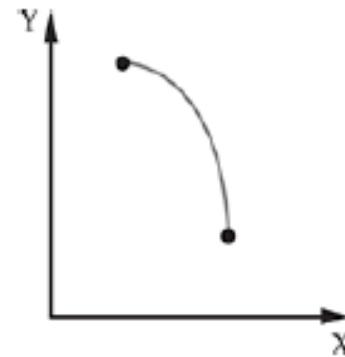
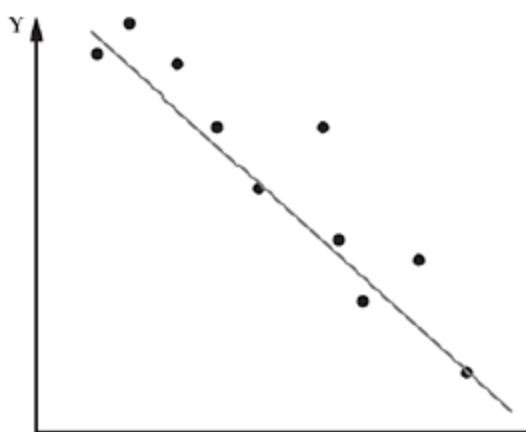
- Scenario 1 for positive slope: *Delta (Y) is positive and Delta (X) is positive*
- Scenario 2 for positive slope: *Delta (Y) is negative and Delta (X) is negative*



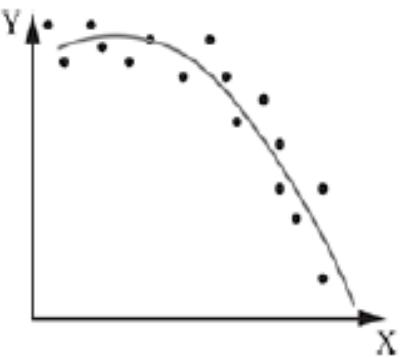
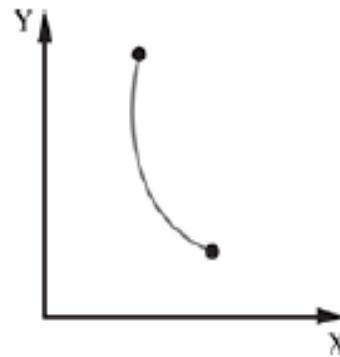
- A negative slope always moves downward on a graph from left to right.

$$\text{Slope} = \text{Rise/Run} = (Y - Y) / (X - X) = \Delta(Y) / \Delta(X)$$

- Scenario 1 for negative slope: *Delta (Y) is positive and Delta (X) is negative*
- Scenario 2 for negative slope: *Delta (Y) is negative and Delta (X) is positive*

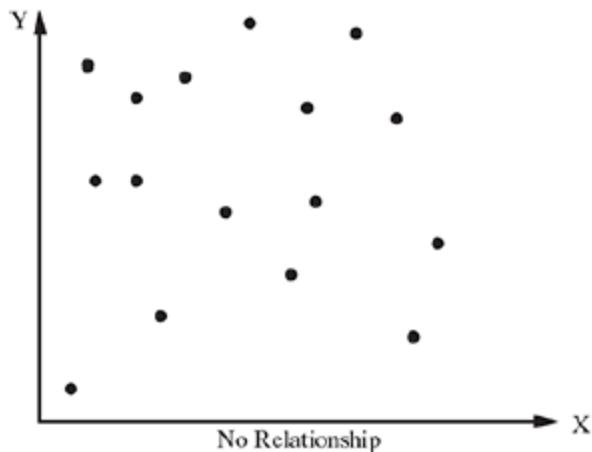


Curve linear negative slope



No relationship graph

- Scatter graph which indicates ‘no relationship’ curve as it is very difficult to conclude whether the relationship between X and Y is positive or negative



Error in simple regression

- The regression equation model in machine learning uses the slope–intercept format in algorithms.
- X and Y values are provided to the machine, and it identifies the values of a (intercept) and b (slope) by relating the values of X and Y.
- Identifying the exact match of values for a and b is not always possible.
- There will be some error value (ε) associated with it.
- This error is called marginal or residual error.

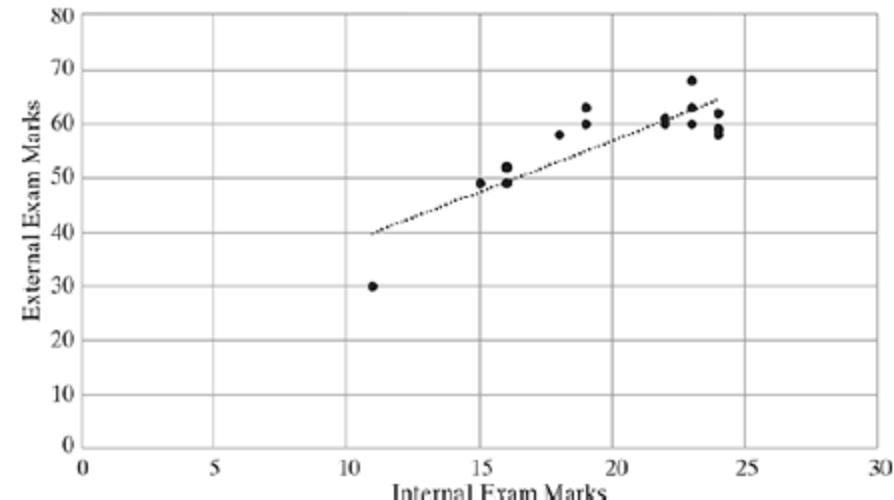
$$Y = (a + bX) + \varepsilon$$

Example of simple regression

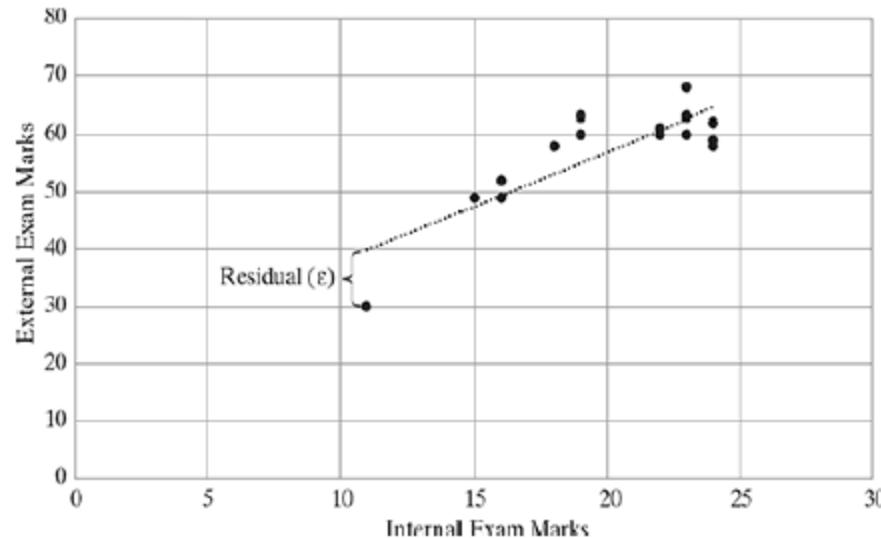
- A college professor believes that if the grade for internal examination is high in a class, the grade for external examination will also be high.
- A random sample of 15 students in that class was selected, and the data is given as

Internal Exam	15	23	18	23	24	22	22	19	19	16	24	11	24	16	23
External Exam	49	63	58	60	58	61	60	63	60	52	62	30	59	49	68

A scatter plot was drawn to explore the relationship between the independent variable (internal marks) mapped to *X-axis* and dependent variable (external marks) mapped to *Y-axis*



- The line does not predict the data exactly
- Some predictions are lower than expected, while some others are higher than expected.
- Residual is the distance between the predicted point (on the regression line) and the actual point



Calculating the values of a and b

- In simple linear regression, the line is drawn using the regression formula.

$$Y = (a + bX) + \varepsilon$$

- If we know the values of '*a*' and '*b*', *then it is easy to predict the value of *Y* for any given *X* by using the formula.*
- How to calculate the values of '*a*' and '*b*' *for a given set of *X* and *Y* values?*
- A straight line is drawn as close as possible over the points on the scatter plot.
- Ordinary Least Squares (OLS) is the technique used to estimate a line that will minimize the error(ε), which is the difference between the predicted and the actual values of *Y*.

The Least Squares Method

- Summing the errors of each prediction or, more appropriately, the Sum of the Squares of the Errors (SSE)

(i.e. $\sum_i \varepsilon_i^2$).

- The SSE is least when b takes the value

$$b = \frac{\sum_i (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_i (X_i - \bar{X})^2} = \frac{\text{cov}(X, Y)}{\text{var}(X)}$$

- The corresponding value of ' a ' *calculated using the value of ' b ' is* $a = \bar{Y} - b\bar{X}$

Ordinary Least Squares *algorithm*

- Step 1: Calculate the mean of X and Y
 - Step 2: Calculate the errors of X and Y
 - Step 3: Get the product
 - Step 4: Get the summation of the products
 - Step 5: Square the difference of X
 - Step 6: Get the sum of the squared difference
 - Step 7: Divide output of step 4 by output of step 6 to calculate ' b '
 - Step 8: Calculate ' a ' using the value of ' b '

		Step 2		Step 3		Step 5	
X	Y	X- mean (X)	Y- Mean (Y)	$(X_i - \bar{X})(Y_i - \bar{Y})$	$(X_i - \bar{X})^2$	$(Y_i - \bar{Y})^2$	
15	49	-4.93	-7.8	38.454	24.3049	100.0000	
23	63	3.07	6.2	19.034	9.4289	36.0000	
18	58	-1.93	1.2	-2.316	3.7289	1.4400	
23	60	3.07	3.2	9.824	9.4289	4.0000	
24	58	4.07	1.2	4.884	16.5649	1.4400	
22	61	2.07	4.2	8.694	4.2889	16.0000	
22	60	2.07	3.2	6.624	4.2889	10.2400	
19	63	-0.93	6.2	-5.766	0.8649	36.0000	
19	60	-0.93	3.2	-2.976	0.8649	10.2400	
16	52	-3.93	-4.8	18.864	15.4449	23.0400	
24	62	4.07	5.2	21.164	16.5649	25.6000	
11	30	-8.93	-26.8	230.324	79.7449	688.6400	
24	59	4.07	2.2	8.954	16.5649	4.8400	
16	49	-3.93	-7.8	30.654	15.4449	60.8400	
23	68	3.07	11.2	34.384	9.4289	125.4400	
19.9	56.8			$\sum(X_i - \bar{X})(Y_i - \bar{Y})$	429.8		226.9335

Example

Linear
regression

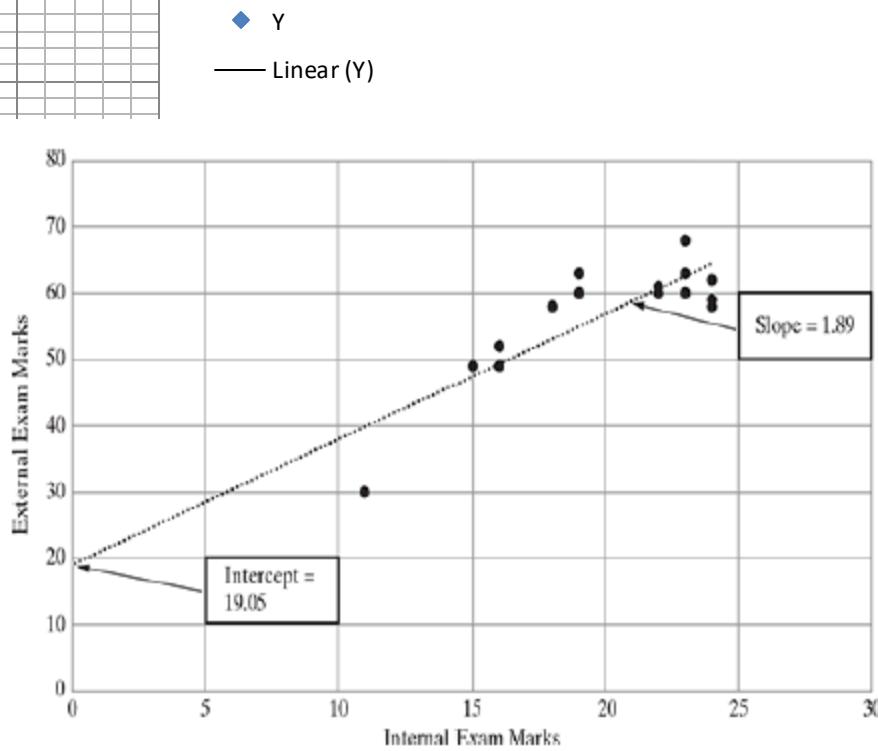
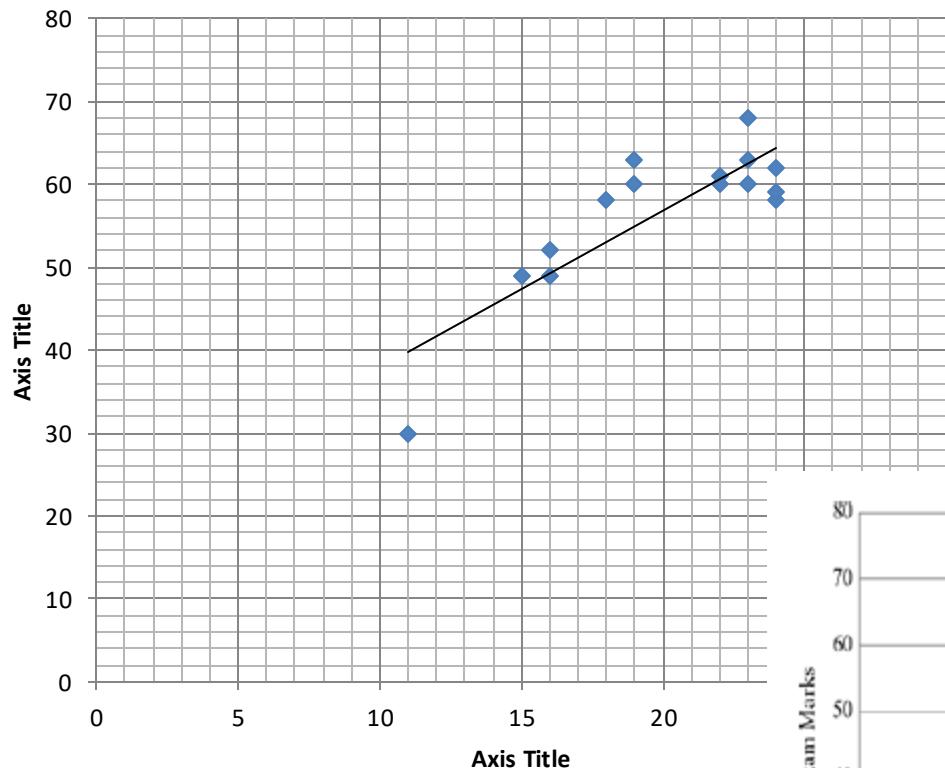
X	Y	$X_i - \bar{X}$	$Y_i - \bar{Y}$	$(X_i - \bar{X})(Y_i - \bar{Y})$	$(X_i - \bar{X})^2$
15	49	-4.93	-7.8	38.48	24.34
23	63	3.07	6.2	19.01	9.40
18	58	-1.93	1.2	-2.32	3.74
23	60	3.07	3.2	9.81	9.40
24	58	4.07	1.2	4.88	16.54
22	61	2.07	4.2	8.68	4.27
22	60	2.07	3.2	6.61	4.27
19	63	-0.93	6.2	-5.79	0.87
19	60	-0.93	3.2	-2.99	0.87
16	52	-3.93	-4.8	18.88	15.47
24	62	4.07	5.2	21.15	16.54
11	30	-8.93	-26.8	239.41	79.80
24	59	4.07	2.2	8.95	16.54
16	49	-3.93	-7.8	30.68	15.47
23	68	3.07	11.2	34.35	9.40
sum	299	852		429.80	226.93
mean(X')	19.93	56.8			

$$b = ((\sum (X_i - \bar{X})(Y_i - \bar{Y})) / (\sum (X_i - \bar{X})^2)) \quad (429 / 226.93) = 1.89$$

$$a = (\bar{Y} - b\bar{X}) = 19.05$$

Regression line
 $Y = 19.05 + 1.89X$

Scatter Graph



Solve the problem

x y

245	1400
312	1600
279	1700
308	1875
199	1100
219	1550
405	2350
324	2450
319	1425
255	1700

Find the values of a and b

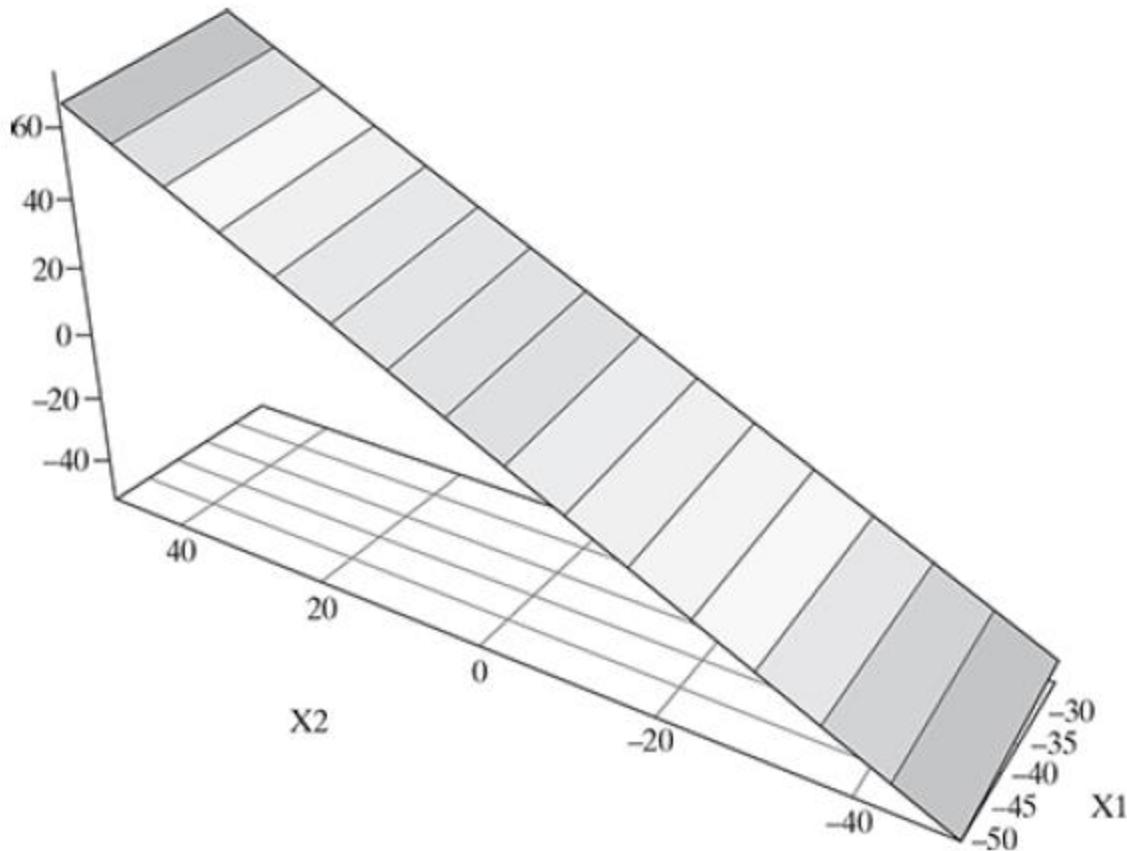
Multiple Linear Regression

- In a multiple regression model, two or more independent variables, i.e. predictors are involved in the model
- The following expression describes the equation involving the relationship with two predictor variables, namely X_1 and X_2 .

$$\hat{Y} = a + b_1 X_1 + b_2 X_2$$

- The model describes a plane in the three-dimensional space of \hat{Y} , X_1 , and X_2 . *Parameter 'a' is the intercept of this plane.*
- Parameters ' b_1 ' and ' b_2 ' are referred to as *partial regression coefficients*.
- Parameter b_1 represents the change in the mean response corresponding to a unit change in X_1 when X_2 is held constant.
- Parameter b_2 represents the change in the mean response corresponding to a unit change in X_2 when X_1 is held constant.

Multiple regression plane



- Multiple regression for estimating equation when there are '*n*' *predictor variables* is as follows:

$$\hat{Y} = a + b_1 X_1 + b_2 X_2 + b_3 X_3 + \dots + b_n X_n$$

- While finding the best fit line, we can fit either a polynomial or curvilinear regression.

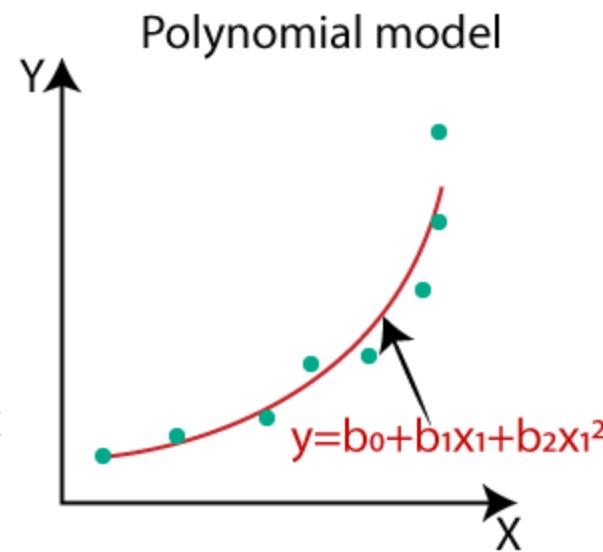
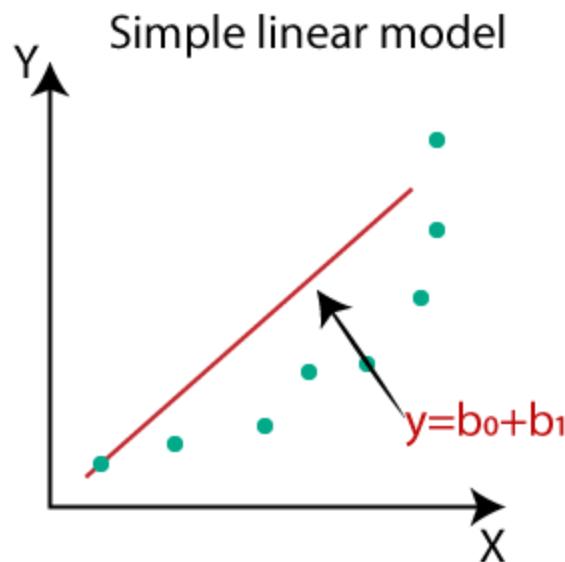
Polynomial Regression Model

- Polynomial regression model is the extension of the simple linear model by adding extra predictors obtained by raising (squaring) each of the original predictors to a power.
- For example, if there are three variables, X , X^2 , and X^3 are used as predictors. This approach provides a simple way to yield a non-linear fit to data.

$$f(x) = c_0 + c_1 \cdot X^1 + c_2 \cdot X^2 + c_3 \cdot X^3$$

- In the above equation, c_0 , c_1 , c_2 and c_3 are the coefficients.

Need for Polynomial Regression



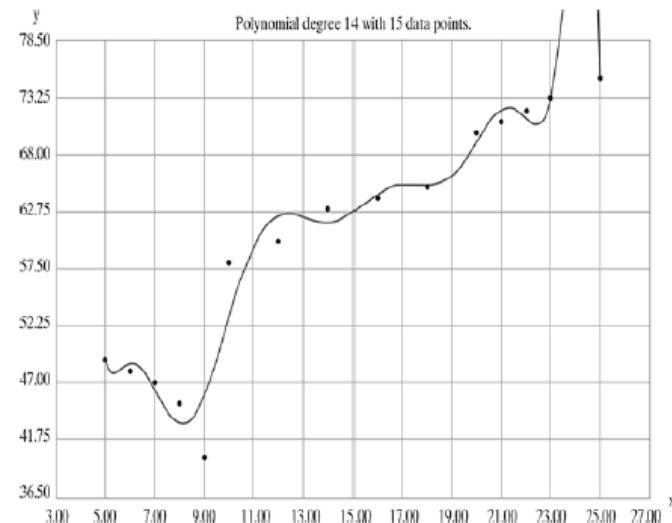
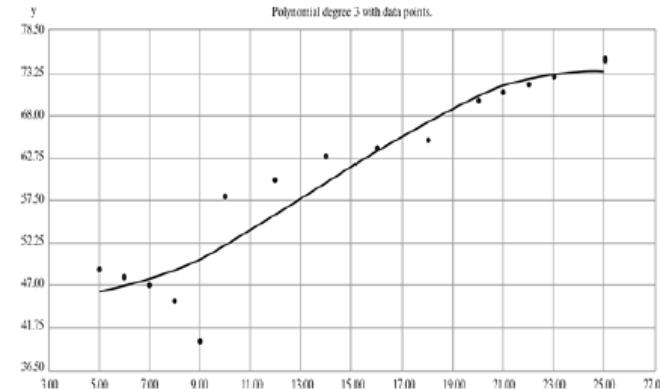
- Used for nonlinear data set
- There is a large range of different functions that can be used for fitting.

Disadvantages of polynomial regression

- Even a single outlier in the data plot can seriously mess up the results.
- PR models are prone to overfitting. If enough parameters are used, you can fit anything.
- As a consequence, PR models might not generalize well outside of the data used

- Let us use the data set of (X, Y) for degree 3 polynomial.
- The regression line is slightly curved for polynomial degree 3 with the 15 data points.
- The regression line will curve further if we increase the polynomial degree
- At the extreme value as, the regression line will be overfitting into all the original values of X .

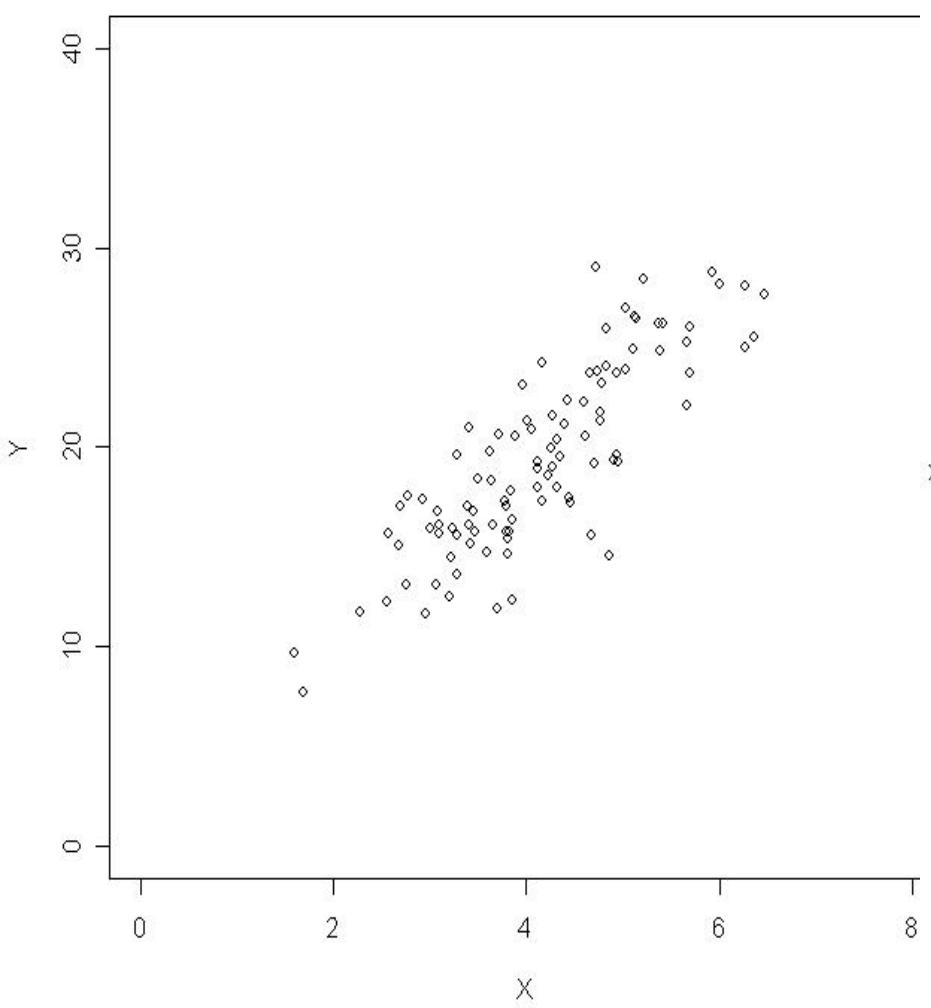
Internal Exam (X)	15	23	18	23	24	22	22	19	19	16	24	11	24	16	23
External Exam (Y)	49	63	58	60	58	61	60	63	60	52	62	30	59	49	68



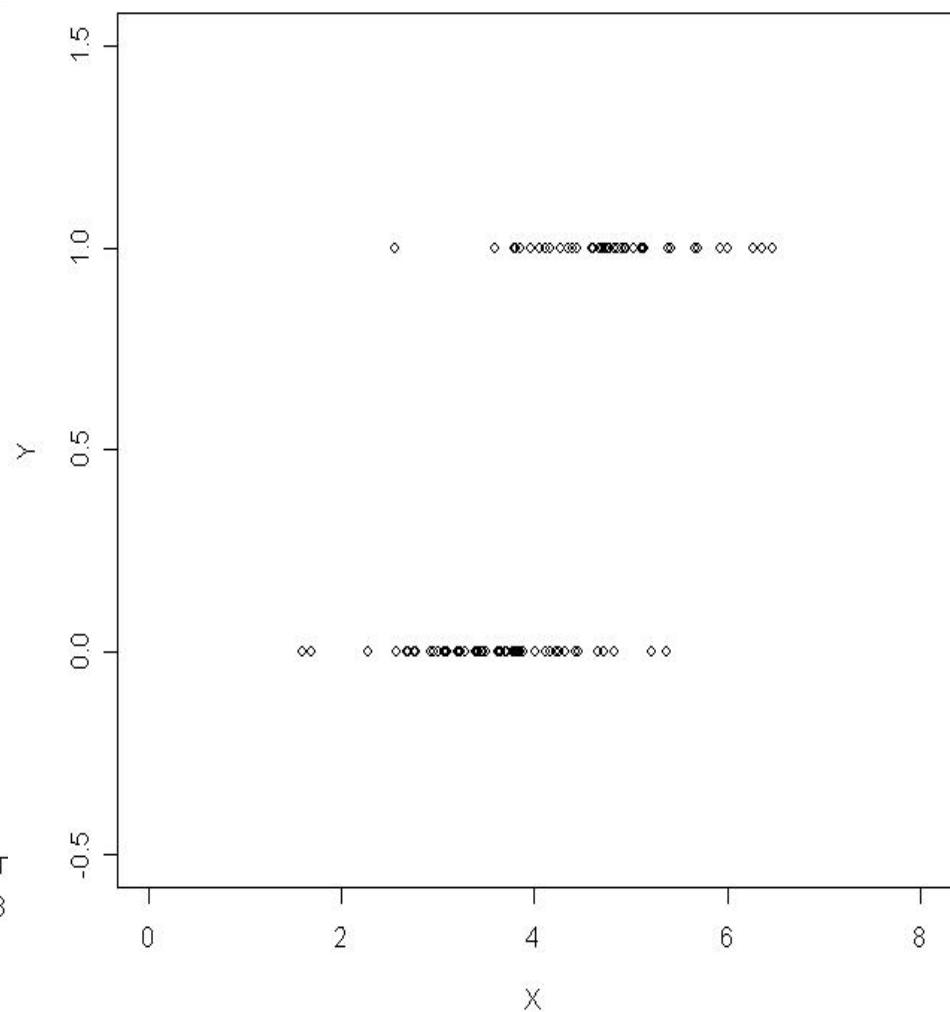
Logistic Regression

- Logistic regression is both classification and regression technique depending on the scenario used.
- Logistic regression is a type of regression analysis used for predicting the outcome of a categorical dependent variable.
- In logistic regression, dependent variable (Y) is binary (0,1) and independent variables (X) are continuous in nature.
- The goal of logistic regression is to predict the likelihood that Y is equal to 1 (probability that $Y = 1$ rather than 0) given certain values of X .
- If X and Y have a strong positive linear relationship, the probability that a person will have a score of $Y = 1$ will increase as values of X increase

Y vs. X: Data Appropriate for Least Squares Regression



Y vs. X: Data Appropriate for Logistic Regression (DO NOT use least-squares regression)



- Logistic regression uses logistic function which always takes the value between 0 and 1.
- The logistic formulae are stated in terms of the probability that $Y = 1$, *which is referred to as P*.
- The probability that Y is 0 is $(1 - P)$.

$$\ln\left(\frac{P}{1 - P}\right) = a + bX$$

$$\ln(p/1 - p) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \epsilon$$

- The ‘In’ symbol refers to a natural logarithm and $a + bX$ is the regression line equation.

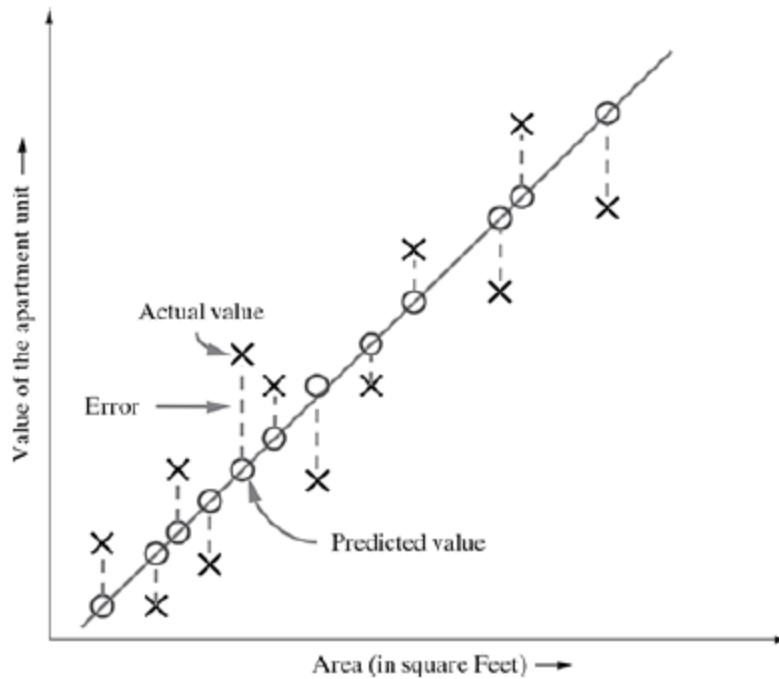
- Probability (P) can also be computed from the regression equation.
- The expected probability that $Y = 1$ for a given value of X is given by

$$P = \frac{\exp(a + bX)}{1 + \exp(a + bX)} = \frac{e^{a+bX}}{1 + e^{a+bX}}$$

- ‘exp’ is the exponent function, which is sometimes also written as e.

Evaluating performance of Regression model

- A well-fitted regression model churns out predicted values close to actual values.
- A regression model which ensures that the difference between predicted and actual values is low can be considered as a good model.



Thankyou

Unsupervised Learning

Unit 5

Unsupervised learning

- Unsupervised learning is a machine learning concept where the unlabelled and unclassified information is analysed to discover hidden knowledge.
- The algorithms work on the data without any prior training, but they are constructed in such a way that they can identify patterns, groupings, sorting order, and numerous other interesting knowledge from the set of data.

Clustering

- Clustering refers to a broad set of techniques for finding subgroups, or clusters, in a data set on the basis of the characteristics of the objects within that data set.
- The objects within the group are similar (or related to each other) but are different from (or unrelated to) the objects from the other groups.
- The effectiveness of clustering depends on how similar or related the objects within a group are or how different or unrelated the objects in different groups are from each other.
- It is often domain specific to define what is meant by two objects to be similar or dissimilar.

- There are many different fields where cluster analysis is used effectively, such as
 - Text data mining: this includes tasks such as text categorization, text clustering, document summarization, concept extraction, sentiment analysis, and entity relation modelling
 - Customer segmentation: creating clusters of customers on the basis of parameters such as demographics, financial conditions, buying habits, etc., which can be used by retailers and advertisers to promote their products in the correct segment
 - Anomaly checking: checking of anomalous behaviours such as fraudulent bank transaction, unauthorized computer intrusion, suspicious movements on a radar scanner, etc.
 - Data mining: simplify the data mining task by grouping a large number of features from an extremely large data set to make the analysis manageable

- Clustering is defined as an unsupervised machine learning task that automatically divides the data into clusters or groups of similar items.
- The analysis achieves this without prior knowledge of the types of groups required and thus can provide an insight into the natural groupings within the data set.
- The primary guideline of clustering task is that the data inside a cluster should be very similar to each other but very different from those outside the cluster.

- The definition of similarity might vary across applications, but the basic idea is always the same, that is, to create the group such that related elements are placed together.
- Using this principle, whenever a large set of diverse and varied data is presented for analysis, clustering enables to represent the data in a smaller number of groups.
- It helps to reduce the complexity and provides insight into patterns of relationships to generate meaningful and actionable structures within the data.
- The effectiveness of clustering is measured by the homogeneity within a group as well as the difference between distinct groups.

- Through clustering, the objects are labeled with class labels.
- Clustering is different from the classification and numeric prediction of supervised learning
- In each of these cases, the goal was to create a model that relates features to an outcome or to other features and the model identifies patterns within the data.
- In contrast, clustering creates new data.
- Unlabelled objects are given a cluster label which is inferred entirely from the relationship of attributes within the data.

Different types of clustering techniques

Method	Characteristics
Partitioning methods	<ul style="list-style-type: none">• Uses mean or medoid (etc.) to represent cluster centre• Adopts distance-based approach to refine clusters• Finds mutually exclusive clusters of spherical or nearly spherical shape• Effective for data sets of small to medium size
Hierarchical methods	<ul style="list-style-type: none">• Creates hierarchical or tree-like structure through decomposition or merger• Uses distance between the nearest or furthest points in neighbouring clusters as a guideline for refinement• Erroneous merges or splits cannot be corrected at subsequent levels
Density-based methods	<ul style="list-style-type: none">• Useful for identifying arbitrarily shaped clusters• Guiding principle of cluster creation is the identification of dense regions of objects in space which are separated by low-density regions• May filter out outliers

Partitioning methods

- Two of the most important algorithms for partitioning-based clustering are *k-means* and *k-medoid*.
- In the *k-means* algorithm, the centroid of the prototype is identified for clustering, which is normally the mean of a group of points.
- The *k-medoid algorithm identifies the medoid* which is the most representative point for a group of points.
- In most cases, the centroid does not correspond to an actual data point, whereas medoid is always an actual data point.

K-means - A centroid-based technique

- This is one of the oldest and most popularly used algorithm for clustering.
- The principle of the *k-means algorithm* is to assign each of the '*n*' data points to one of the *K* clusters where '*K*' is a user defined parameter as the number of clusters desired.
- The objective is to maximize the homogeneity within the clusters and also to maximize the differences between the clusters.
- The homogeneity and differences are measured in terms of the distance between the objects or points in the data set.

The simple algorithm of *K-means*

Step 1: Select K points in the data space and mark them as initial centroids
loop

Step 2: Assign each point in the data space to the nearest centroid to form K clusters

Step 3: Measure the distance of each point in the cluster from the centroid

Step 4: Calculate the Sum of Squared Error (SSE) to measure the quality of clusters

Step 5: Identify the new centroid of each cluster on the basis of distance between points

Step 6: Repeat Steps 2 to 5 to refine until centroids do not change
end loop

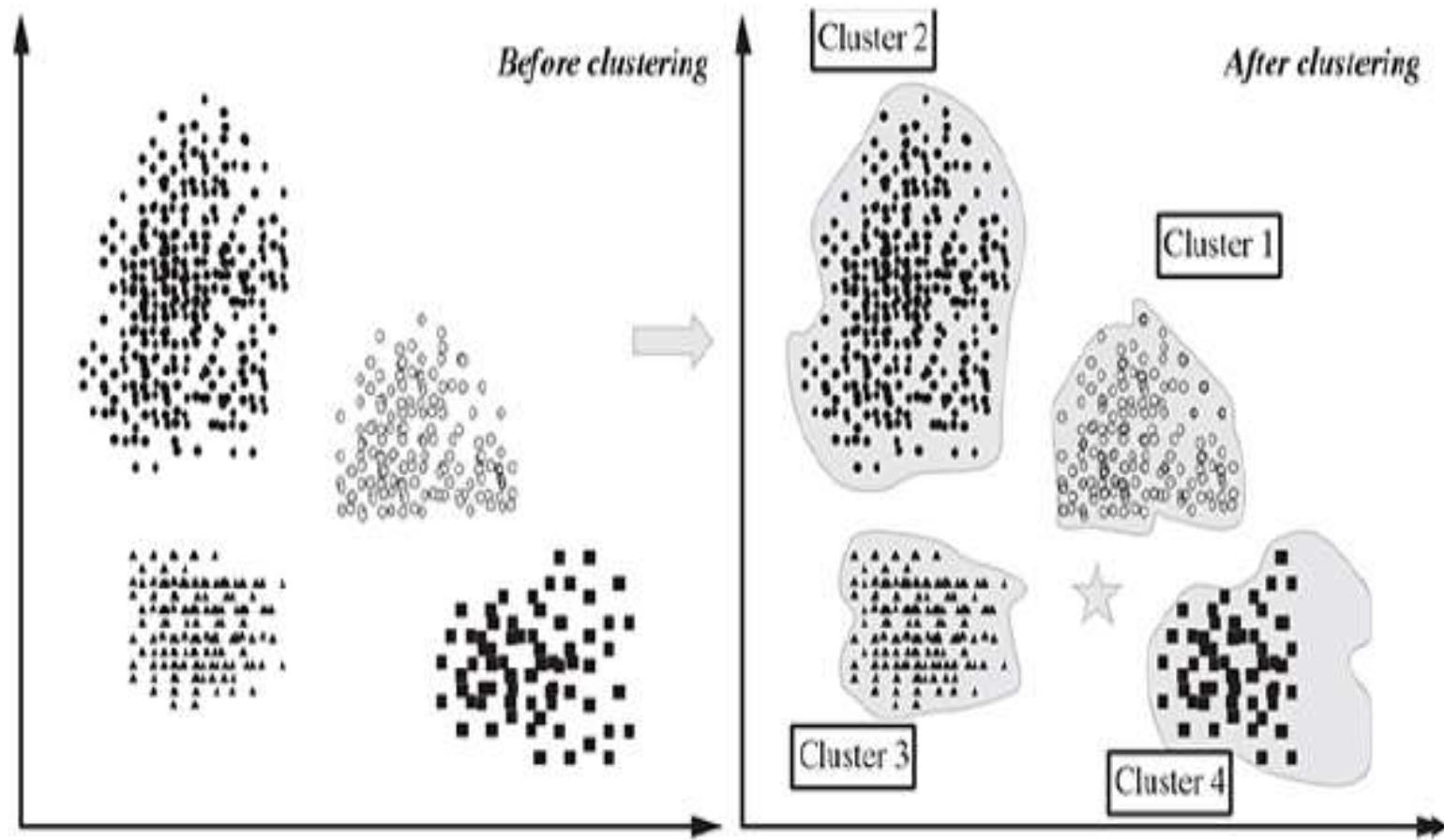
Strengths and Weaknesses of K-means

Strengths

- The principle used for identifying the clusters is very simple and involves very less complexity of statistical terms
 - The algorithm is very flexible and thus can be adjusted for most scenarios and complexities
 - The performance and efficiency are very high and comparable to those of any sophisticated algorithm in term of dividing the data into useful clusters
-

Weaknesses

- The algorithm involves an element of random chance and thus may not find the optimal set of cluster in some cases
- The starting point of guessing the number natural clusters within the data requires some experience of the user, so that the final outcome is efficient

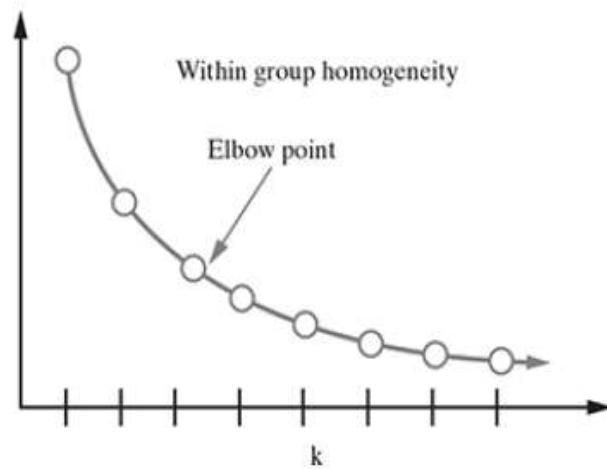
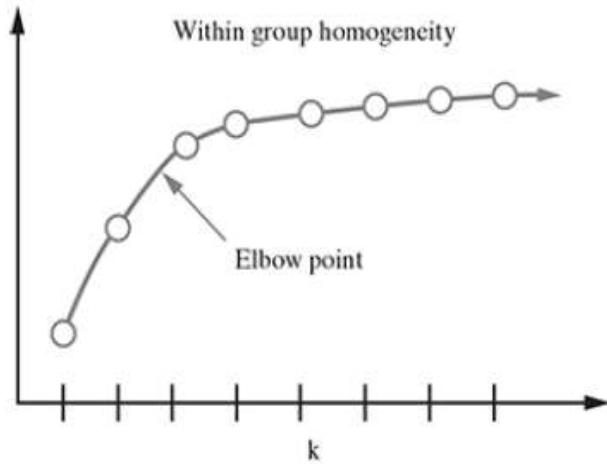


Choosing appropriate number of clusters

- One of the most important success factors in arriving at correct clustering is to start with the correct number of cluster assumptions.
- Different numbers of starting cluster lead to completely different types of data split.
- It will always help if we have some prior knowledge about the number of clusters and we start our *k-means algorithm with that prior knowledge*.
- There are several statistical methods to arrive at the suitable number of clusters.

Elbow method

- This method tries to measure the homogeneity or heterogeneity within the cluster for various values of ‘ K ’ and helps in arriving at the optimal ‘ K ’.
- We can see the homogeneity will increase or heterogeneity will decrease with increasing ‘ K ’ as the number of data points inside each cluster reduces with this increase.
- But these iterations take significant computation effort, and after a certain point, the increase in homogeneity benefit is no longer in accordance with the investment required to achieve it.
- This point is known as the elbow point, and the ‘ K ’ value at this point produces the optimal clustering performance.



Choosing the initial centroids

- One common practice is to choose random points in the data space on the basis of the number of cluster requirement and refine the points as we move into the iterations.
- This often leads to higher squared error in the final clustering, thus resulting in sub-optimal clustering solution.
- The assumption for selecting random centroids is that multiple subsequent runs will minimize the sum of squared error(SSE) and identify the optimal clusters.
- But this does not work
- Another effective approach is to employ the hierarchical clustering technique on sample points from the data set and then arrive at sample K clusters.

- The centroids of these initial K clusters are used as the initial centroids.
- This approach is practical when the data set has small number of points and K is relatively small compared to the data points.

Recomputing cluster centroids

- In the k -means algorithm the iterative step is used to recalculate the centroids of the data set after each iteration.
- The proximities of the data points from each other within a cluster is measured to minimize the distances.
- The distance of the data point from its nearest centroid can also be calculated to minimize the distances to arrive at the refined centroid.
- The Euclidean distance between two data points is measured as

$$\text{dist}(x, y) = \sqrt{\sum_1^n (x_i - y_i)^2}$$

- It calculates the Euclidean distance between the centroid c_i of the cluster C_i and the data points x in the cluster.

- By using this function, the distance between the example data and its nearest centroid and the objective is calculated to minimize this distance.
- The measure of quality of clustering uses the SSE technique. The formula used is as follows

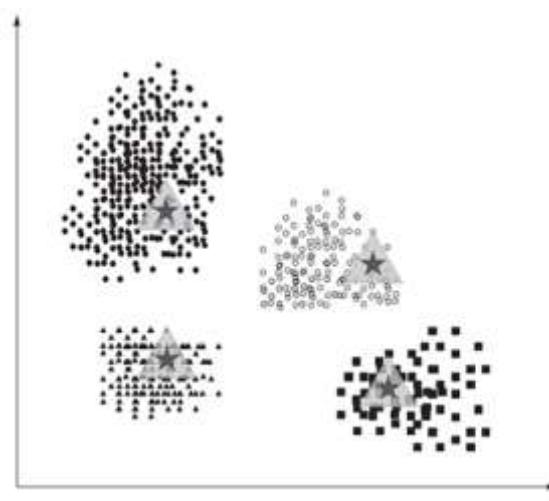
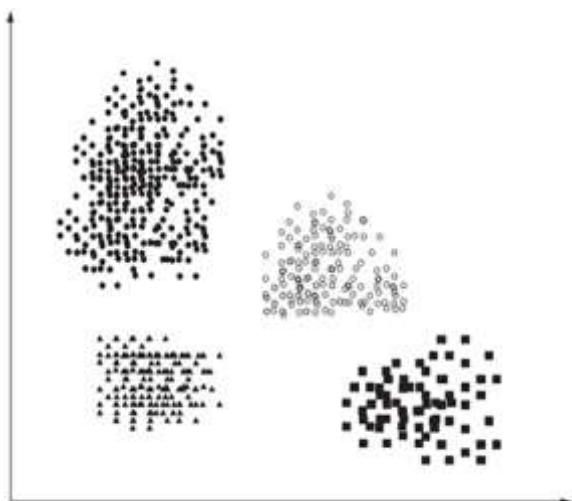
$$\text{SSE} = \sum_{i=1}^k \sum_{x \in C_i} \text{dist}(c_i, x)^2$$

- The summation of distances over all the ‘K’ clusters gives the total sum of squared error.
- The lower the SSE for a clustering solution, the better is the representative position of the centroid.

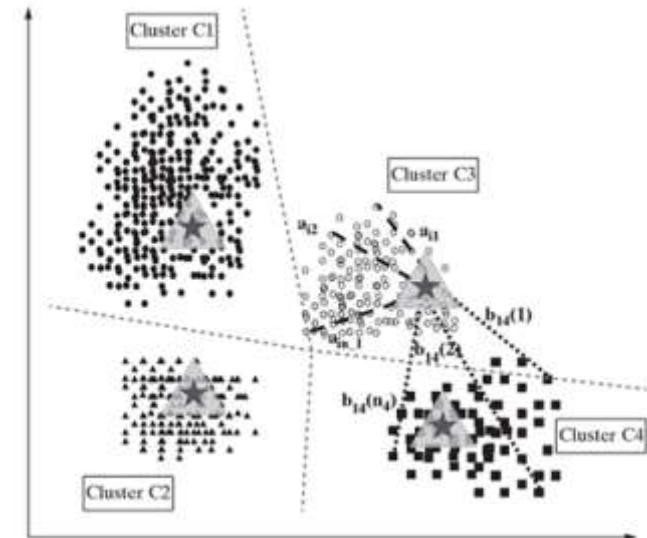
- In our clustering algorithm, the recomputation of the centroid involves calculating the SSE of each new centroid and arriving at the optimal centroid identification.
- After the centroids are repositioned, the data points nearest to the centroids are assigned to form the refined clusters.
- One limitation of the squared error method is that in the case of presence of outliers in the data set, the squared error can distort the mean value of the clusters.

Example

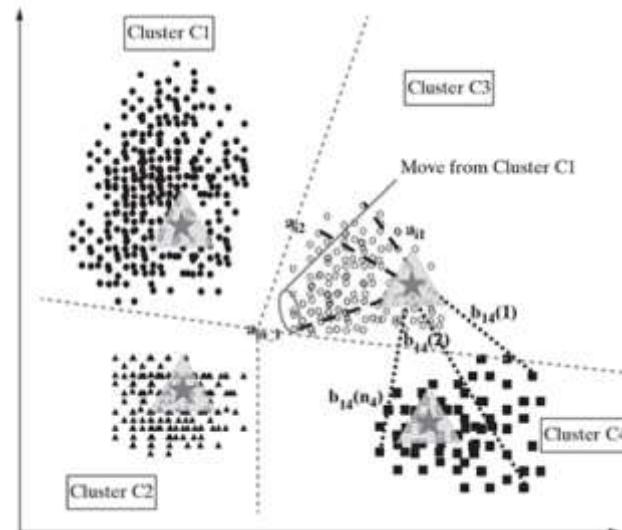
- Consider the data set (figure 1)
- Assume the number of cluster requirement, $K = 4$.
- We will randomly select four cluster centroids as indicated in the figure 2



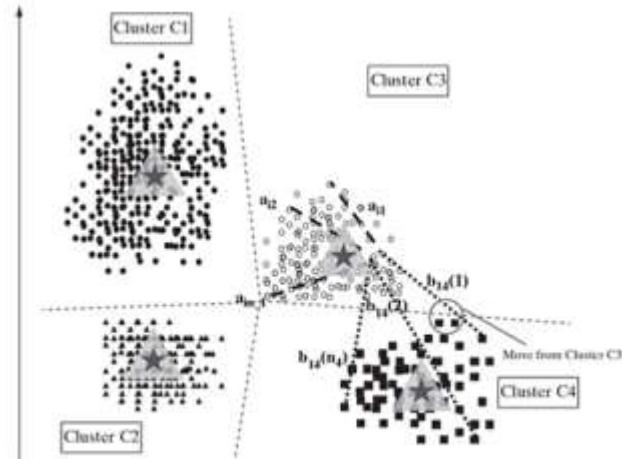
- On the basis of the proximity of the data points in this data set to the centroids, we partition the data set into four segments as represented by dashed lines
- This diagram is called **Voronoi diagram** which creates the boundaries of the clusters.
- We got the initial four clusters, namely C_1 , C_2 , C_3 , and C_4 , created by the dashed lines from the vertex of the clusters, which is the point with the maximal distance from the centre of the clusters.
- It is now easy to understand the areas covered by each cluster and the data points within each cluster.



- The next step is to calculate the SSE of this clustering and update the position of the centroids.
- Our aim is to minimize the homogeneity within the clusters and maximize the heterogeneity among the different clusters.
- We can also find out that the cluster boundaries are refined on the basis of the new centroids and the identification of the nearest centroids for the data points and reassigning them to the new centroids.



- The k -means algorithm continues with the update of the centroid according to the new cluster and reassignment of the points, until no more data points are changed due to the centroid shift.
- At this point, the algorithm stops.



Problem 1

We will apply k-means on the following 1 dimensional data set for K=2.

Data set {2, 4, 10, 12, 3, 20, 30, 11, 25}

Iteration 1

M1, M2 are the two randomly selected centroids/means where

M1= 4, M2=11

and the initial clusters are

C1= {4}, C2= {11}

Calculate the Euclidean distance as

$$D=[x,a]=\sqrt{(x-a)^2}$$

D1 is the distance from M1

D2 is the distance from M2

Datapoint	D1	D2	Cluster
2	2	9	C1
4	0	7	C1
10	6	1	C2
12	8	1	C2
3	1	8	C1
20	16	9	C2
30	26	19	C2
11	7	0	C2
25	21	14	C2

2 datapoints are added to cluster C1 and other datapoints added to cluster C2

Therefore

C1= {2, 4, 3}

C2= {10, 12, 20, 30, 11, 25}

Iteration 2

Calculate new mean of datapoints in C1 and C2.

Therefore

$$M1 = (2+3+4)/3 = 3$$

$$M2 = (10+12+20+30+11+25)/6 = 18$$

Calculating distance and updating clusters based on table

New Clusters

$$C1 = \{2, 3, 4, 10\}$$

$$C2 = \{12, 20, 30, 11, 25\}$$

Datapoint	D1	D2	Cluster
2	1	16	C1
4	1	14	C1
3	0	15	C1
10	7	8	C1
12	9	6	C2
20	17	2	C2
30	27	12	C2
11	8	7	C2
25	22	7	C2

Iteration 3

Calculate new mean of datapoints in C1 and C2.

Therefore

$$M1 = (2+3+4+10)/4 = 4.75$$

$$M2 = (12+20+30+11+25)/5 = 19.6$$

Calculating distance and updating clusters based on table

Datapoint	D1	D2	Cluster
2	2.75	17.6	C1
4	0.75	15.6	C1
3	1.75	16.6	C1
10	5.25	9.6	C1
12	7.25	7.6	C1
20	15.25	0.4	C2
30	25.25	10.4	C2
11	6.25	8.6	C1
25	20.25	5.4	C2

New Clusters

$$C1 = \{2, 3, 4, 10, 12, 11\}$$

$$C2 = \{20, 30, 25\}$$

Iteration 4

Calculate new mean of datapoints in C1 and C2.

Therefore

$$M1 = (2+3+4+10+12+11)/6 = 7$$

$$M2 = (20+30+25)/3 = 25$$

Calculating distance and updating clusters based on table

Datapoint	D1	D2	Cluster
2	5	23	C1
4	3	21	C1
3	4	22	C1
10	3	15	C1
12	5	13	C1
11	4	14	C1
20	13	5	C2
30	23	5	C2
25	18	0	C2

New Clusters

$$C1 = \{2, 3, 4, 10, 12, 11\}$$

$$C2 = \{20, 30, 25\}$$

The data points in the cluster C1 and C2 in iteration 3 are same as the data points of the cluster C1 and C2 of iteration 4.

It means that none of the data points has moved to other cluster.

So this becomes the stopping condition for our algorithm.

Solve

Cluster the following eight points (with (x, y) representing locations) into three clusters:

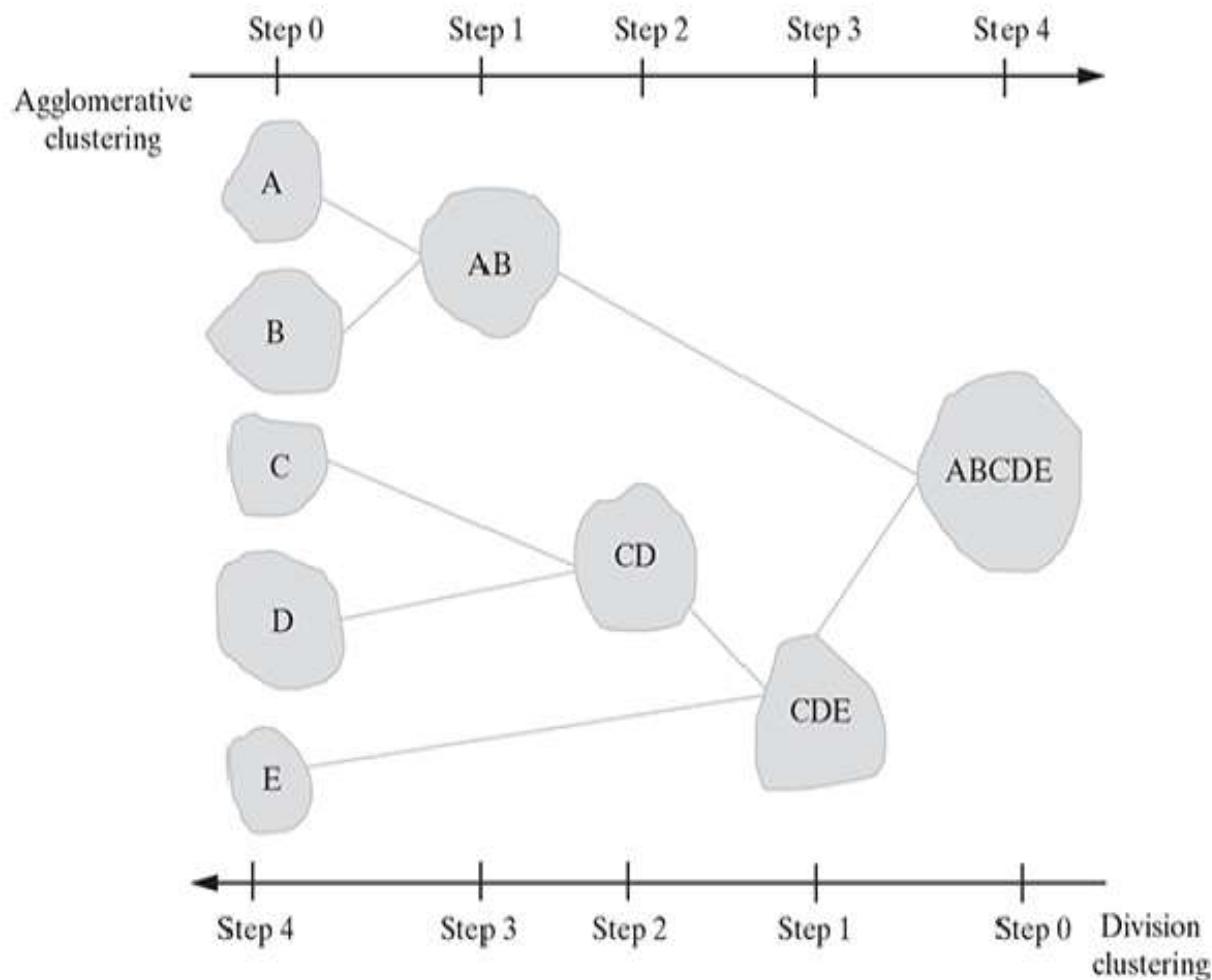
A1(2, 10), A2(2, 5), A3(8, 4), A4(5, 8), A5(7, 5), A6(6, 4), A7(1, 2), A8(4, 9)

Initial cluster centers are: A1(2, 10), A4(5, 8) and A7(1, 2).

Use K-Means Algorithm to find the three cluster centers after the second iteration.

Hierarchical clustering

- The hierarchical clustering methods are used to group the data into hierarchy or tree-like structure.
- There are two main hierarchical clustering methods:
 - agglomerative clustering and
 - divisive clustering.
- Agglomerative clustering is a bottom-up technique which starts with individual objects as clusters and then iteratively merges them to form larger clusters.
- The divisive method starts with one cluster with all given objects and then splits it iteratively to form smaller clusters



Agglomerative hierarchical clustering method

- It uses the bottom-up strategy.
- It starts with each object forming its own cluster and then iteratively merges the clusters according to their similarity to form larger clusters.
- It terminates either
 - when a certain clustering condition imposed by the user is achieved or
 - all the clusters merge into a single cluster.

Divisive hierarchical clustering method

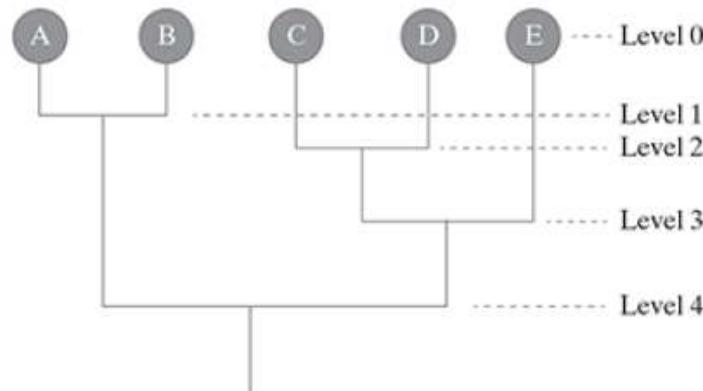
- The divisive hierarchical clustering method uses a top-down strategy.
- The starting point is the largest cluster with all the objects in it, and then, it is split recursively to form smaller and smaller clusters, thus forming the hierarchy.
- The end of iterations is achieved
 - when the objects in the final clusters are sufficiently homogeneous to each other or
 - the final clusters contain only one object or
 - the user-defined clustering condition is achieved.

Drawbacks

- In both these cases, it is important to select the split and merger points carefully, as the subsequent splits or mergers will use the result of the previous ones
- There is no option to perform any object swapping between the clusters or rectify the decisions made in previous steps, which may result in poor clustering quality at the end.

Dendrogram

- A dendrogram is a commonly used tree structure
- representation of step-by-step creation of hierarchical clustering.
- It shows how the clusters are **merged** iteratively or **split** iteratively to arrive at the optimal clustering solution.



Distances Measurement

- One of the core measures of proximities between clusters is the distance between them.
- There are four standard methods to measure the distance between clusters:

(Let C_i and C_j be the two clusters with n_i and n_j respectively.

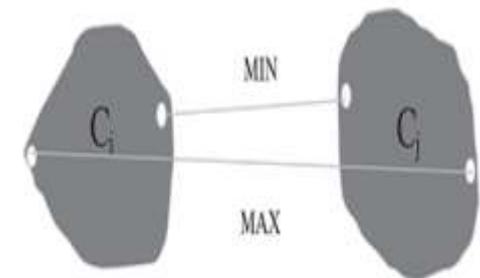
p_i and p_j represents the points in clusters C_i and C_j respectively. We will denote the mean of cluster C_i as m_i)

$$\text{Minimum distance } D_{\min}(C_i, C_j) = \min_{p_i \in C_i, p_j \in C_j} \{ |p_i - p_j| \}$$

$$\text{Maximum distance } D_{\max}(C_i, C_j) = \max_{p_i \in C_i, p_j \in C_j} \{ |p_i - p_j| \}$$

$$\text{Mean distance } D_{\text{mean}}(C_i, C_j) = \{ |m_i - m_j| \}$$

$$\text{Average distance } D_{\text{avg}}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{p_i \in C_i, p_j \in C_j} |p_i - p_j|$$



- Often the distance measure is used to decide when to terminate the clustering algorithm.
 - In an agglomerative clustering, the merging iterations may be stopped once the MIN distance between two neighbouring clusters becomes less than the user-defined threshold.
 - When an algorithm uses the minimum distance D_{min} to measure the distance between the clusters, then it is referred to as nearest neighbour clustering algorithm, and if the decision to stop the algorithm is based on a user-defined limit on D_{min} , then it is called single linkage algorithm.
 - when an algorithm uses the maximum distance D_{max} to measure the distance between the clusters, then it is referred to as furthest neighbour clustering algorithm, and if the decision to stop the algorithm is based on a userdefined limit on D_{max} then it is called complete linkage algorithm.

- As minimum and maximum measures provide two extreme options to measure distance between the clusters, they are prone to the outliers and noisy data.
- Instead, the use of mean and average distance helps in avoiding such problem and provides more consistent results.

Density-based methods(DBSCAN)

- In the partitioning and hierarchical clustering methods, the resulting clusters are spherical or nearly spherical in nature.
- In the case of the other shaped clusters such as S-shaped or uneven shaped clusters, these two types of method do not provide accurate results.
- The density-based clustering approach provides a solution to identify clusters of arbitrary shapes.
- The principle is based on identifying the dense area and sparse area within the data set and then run the clustering algorithm.
- DBSCAN is one of the popular density-based algorithm which creates clusters by using connected regions with high density.

Neural Network

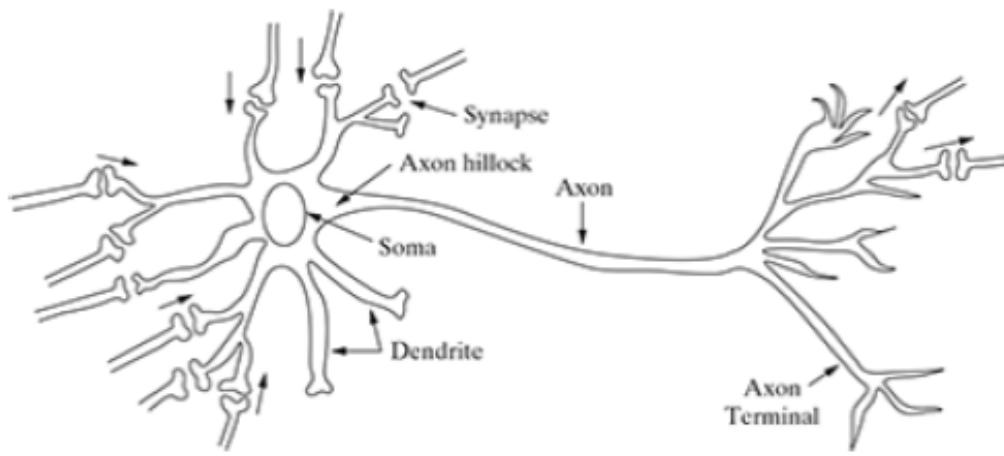
Unit 6

UNDERSTANDING THE BIOLOGICAL NEURON

- The human nervous system has two main parts –
 - the central nervous system (CNS) consisting of the brain and spinal cord
 - the peripheral nervous system consisting of nerves and ganglia outside the brain and spinal cord.
- The CNS integrates all information, in the form of signals, from the different parts of the body.
- The peripheral nervous system, on the other hand, connects the CNS with the limbs and organs.
- Neurons are basic structural units of the CNS.
- A neuron is able to receive, process, and transmit information in the form of chemical and electrical signals

Neuron

- The structure of a neuron has three main parts to carry out its primary functionality of receiving and transmitting information:
 1. **Dendrites** – to receive signals from neighbouring neurons.
 2. **Soma** – main body of the neuron which accumulates the signals coming from the different dendrites. It ‘fires’ when a sufficient amount of signal is accumulated.
 3. **Axon** – last part of the neuron which receives signal from soma, once the neuron ‘fires’, and passes it on to the neighbouring neurons through the axon terminals (to the adjacent dendrite of the neighbouring neurons).



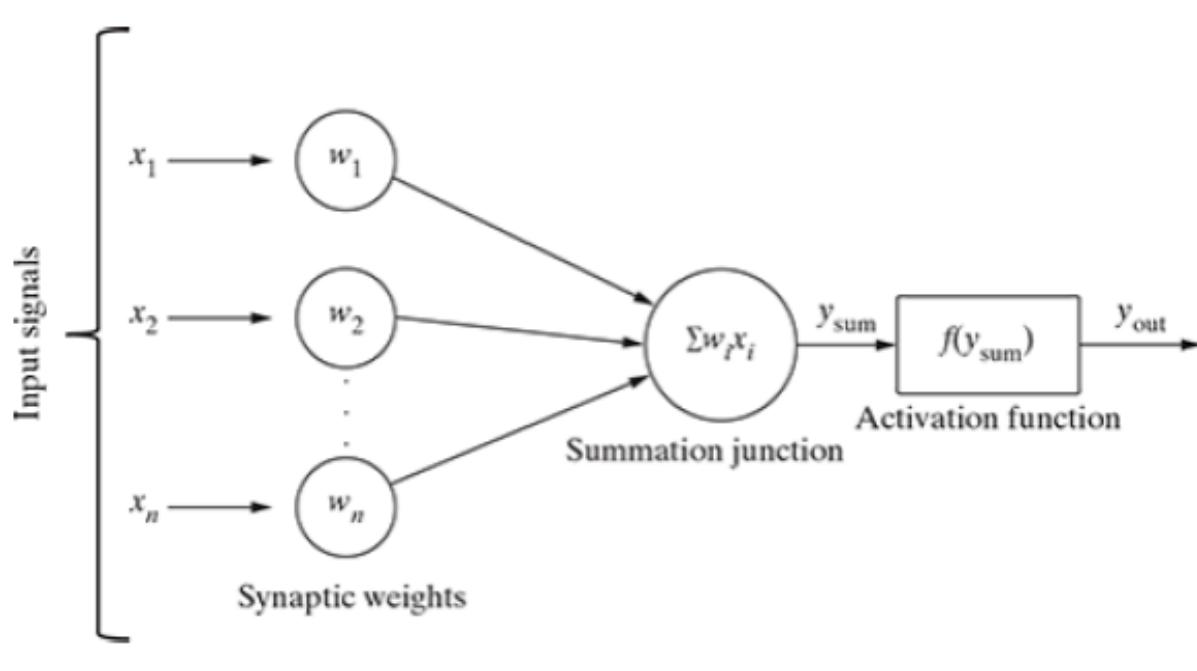
- There is a very small gap between the axon terminal of one neuron and the adjacent dendrite of the neighbouring neuron. This small gap is known as **synapse**.
- The signals transmitted through synapse may be excitatory or inhibitory.

Human Brain

- The adult human brain, which forms the main part of the central nervous system, is approximately 1.3 kg in weight and 1200 cm^3 in volume.
- It is estimated to contain about 100 billion (i.e. 10^{11}) neurons and 10 times more glial or glue cells.
- Glial cells act as support cells for the neurons.
- It is believed that neurons represent about 10% of all cells in the brain. On an average, each neuron is connected to 10^5 of other neurons, which means that altogether there are 10^{16} connections.
- The axon, a human neuron, is 10–12 μm in diameter.
- Each synapse spans a gap of about a millionth of an inch wide.

THE ARTIFICIAL NEURON

- Each neuron has three major components:
 1. A set of ' i ' **synapses** having weight w_i . A signal x_i forms the input to the i -th synapse having weight w_i . The value of weight w_i may be positive or negative.
A positive weight has an excitatory effect, while a negative weight has an inhibitory effect on the output of the summation junction, y_{sum} .
 2. A **summation junction** for the input signals is weighted by the respective synaptic weight. Because it is a linear combiner or adder of the weighted input signals, the output of the summation junction, y_{sum} , can be expressed as :
$$y_{sum} = b + \sum_{i=1}^n w_i x_i$$
 3. A threshold activation function (or simply activation function, also called squashing function) results in an output signal only when an input signal exceeding a specific threshold value comes as an input. It is similar in behaviour to the biological neuron which transmits the signal only when the total input signal meets the firing threshold.



Output of the activation function, y_{out} , can be expressed as

$$y_{\text{out}} = f(y_{\text{sum}})$$

TYPES OF ACTIVATION FUNCTIONS

- Identity function
- Threshold function
- ReLU(Rectified linear unit)
- Sigmoid function
- Hyperbolic Tangent function

Identity function

- Identity function is used as an activation function for the input layer. It is a linear function having the form

$$y_{\text{out}} = f(x) = x, \text{ for all } x$$

- As obvious, the output remains the same as the input.

Threshold/step function

- Step/threshold function is a commonly used activation function.
- The **step function** gives 1 as output if the input is either 0 or positive.
- If the input is negative, the step function gives 0 as output.
- Expressing mathematically,

$$y_{\text{out}} = f(y_{\text{sum}}) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

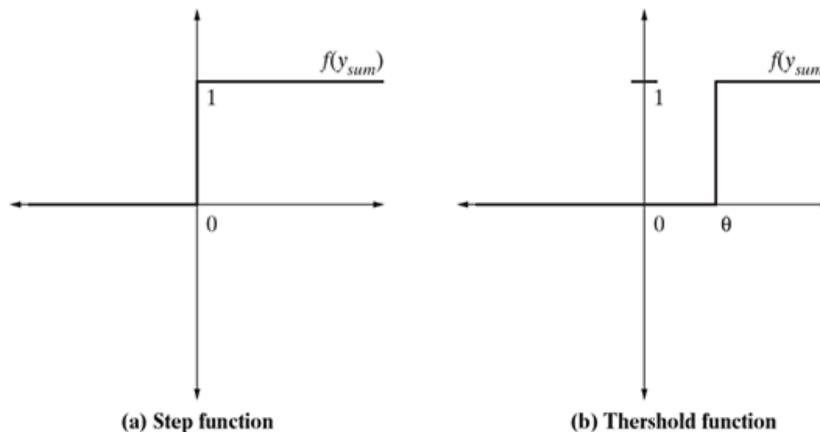


FIG. 10.3 Step and threshold functions

- The **threshold function** is almost like the step function, with the only difference being the fact that θ is used as a threshold value instead of 0.
- Expressing mathematically,

$$y_{\text{out}} = f(y_{\text{sum}}) = \begin{cases} 1, & x \geq \theta \\ 0, & x < \theta \end{cases}$$

ReLU (Rectified Linear Unit) function

- ReLU is the most popularly used activation function in the areas of convolutional neural networks and deep learning.
- It is of the form

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

- This means that $f(x)$ is zero when x is less than zero and $f(x)$ is equal to x when x is above or equal to zero.
- The curve for a ReLU activation function.
- This function is differentiable, except at a single point $x = 0$.
- In that sense, the derivative of a ReLU is actually a subderivative.

Sigmoid function

- Sigmoid function is by far the most commonly used activation function in neural networks.
- The need for sigmoid function stems from the fact that many learning algorithms require the activation function to be differentiable and hence continuous.
- Step function is not suitable in those situations as it is not continuous.
- There are two types of sigmoid function:
 1. Binary sigmoid function
 2. Bipolar sigmoid function

Binary sigmoid function

- A binary sigmoid function is of the form

$$y_{\text{out}} = f(x) = \frac{1}{1 + e^{-kx}}$$

- where k = steepness or slope parameter of the sigmoid function.
- By varying the value of k , sigmoid functions with different slopes can be obtained. It has range of $(0, 1)$.
- The slope at origin is $k/4$.
- As the value of k becomes very large, the sigmoid function becomes a threshold function.

Bipolar sigmoid function

- A bipolar sigmoid function is of the form.

$$y_{\text{out}} = f(x) = \frac{1 - e^{-kx}}{1 + e^{-kx}}$$

- The range of values of sigmoid functions can be varied depending on the application.
- However, the range of $(-1, +1)$ is most commonly adopted.

Hyperbolic tangent function

- Hyperbolic tangent function is another continuous activation function, which is bipolar in nature.
- It is a widely adopted activation function for a special type of neural network known as backpropagation network
- The hyperbolic tangent function is of the form
- This function is similar to the bipolar sigmoid function.

EARLY IMPLEMENTATIONS OF ANN

McCulloch–Pitts model of neuron

- The McCulloch–Pitts neural model was the earliest ANN model
- It has only two types of inputs – excitatory and inhibitory.
 - The excitatory inputs have weights of positive magnitude and
 - the inhibitory weights have weights of negative magnitude.
- The inputs of the McCulloch–Pitts neuron could be either 0 or 1.
- It has a threshold function as activation function.
- So, the output signal y_{out} is 1 if the input y_{sum} is greater than or equal to a given threshold value, else 0.
- Simple McCulloch–Pitts neurons can be used to design logical operations. For that purpose, the connection weights need to be correctly decided along with the threshold function

Example

- John carries an umbrella if it is sunny or if it is raining.
- There are four given situations. We need to decide when John will carry the umbrella.
- The situations are as follows:
 - Situation 1 – It is not raining nor is it sunny.
 - Situation 2 – It is not raining, but it is sunny.
 - Situation 3 – It is raining, and it is not sunny.
 - Situation 4 – Wow, it is so strange! It is raining as well as it is sunny.
- To analyse the situations using the McCulloch–Pitts neural model, we can consider the input signals as follows:
 - $x_1 \rightarrow$ Is it raining?
 - $x_2 \rightarrow$ Is it sunny?
- So, the value of both x_1 and x_2 can be either 0 or 1. We can use the value of both weights x_1 and x_2 as 1 and a threshold value of the activation function as 1.

- So, the neural model will look as
- Formally, we can say
- The truth table built with respect to the problem is depicted
- From the truth table, we can conclude that in the situations where the value of y_{out} is 1, John needs to carry an umbrella.
- Hence, he will need to carry an umbrella in situations 2, 3, and 4.

Rosenblatt's perceptron

- Rosenblatt's perceptron is built around the McCulloch–Pitts neural model.
- The perceptron, receives a set of input x_1, x_2, \dots, x_n .
- The linear combiner or the adder node computes the linear combination of the inputs applied to the synapses with synaptic weights being w_1, w_2, \dots, w_n .
- Then, the hard limiter checks whether the resulting sum is positive or negative.
- If the input of the hard limiter node is positive, the output is +1, and if the input is negative, the output is -1.
Mathematically, the hard limiter input is

- Perceptron includes an adjustable value or bias as an additional weight w . This additional weight w is attached to a dummy input x , which is always assigned a value of 1.
- This consideration modifies the above equation to
- The output is decided by the expression
- The objective of perceptron is to classify a set of inputs into two classes, c_1 and c_2 .
- This can be done using a very simple decision rule – assign the inputs $x_0, x_1, x_2, x_3, \dots, x_n$ to c_1 if the output of the perceptron, i.e. y_{out} , is +1 and c_2 if y_{out} is -1.
- So, for an n-dimensional signal space, i.e. a space for 'n' input signals $x_0, x_1, x_2, x_3, \dots, x_n$, the simplest form of perceptron will have two decision regions, resembling two classes, separated by a hyperplane defined by

- Therefore, for two input signals denoted by variables x_1 and x_2 , the decision boundary is a straight line of the form
- So, for a perceptron having the values of synaptic weights w_0 , w_1 , and w_2 as -2 , $\frac{1}{2}$, and $\frac{1}{4}$, respectively, the linear decision boundary will be of the form
- So, any point (x_1, x_2) which lies above the decision boundary, will be assigned to class c_1 and the points which lie below the boundary are assigned to class c_2 .

Example

- Let us examine if this perceptron is able to classify a set of points given below:
 $p_1 = (5, 2)$ and $p_2 = (-1, 12)$ belonging to c_1
 $p_3 = (3, -5)$ and $p_4 = (-2, -1)$ belonging to c_2
- We can see that on the basis of activation function output, only points p_1 and p_2 generate an output of 1.
- Hence, they are assigned to class c_1 as expected.
- On the other hand, p_3 and p_4 points having activation function output as negative generate an output of 0.
- Hence, they are assigned to class c_2 , again as expected.
- Class assignment through perceptron

- The same classification is obtained by mapping the points in the input space, as shown
- Thus, we can see that for a data set with linearly separable classes, perceptrons can always be employed to solve classification problems using decision lines (for two dimensional space), decision planes (for three-dimensional space), or decision hyperplanes (for n -dimensional space).

- Appropriate values of the synaptic weights $w_1, w_2, w_3, \dots, w_n$ can be obtained by training a perceptron.
- However, one assumption for perceptron to work properly is that the two classes should be linearly separable (as depicted in Figure 10.12a), i.e. the classes should be sufficiently separated from each other.
- Otherwise, if the classes are non-linearly separable (as depicted in Figure 10.12b), then the classification problem cannot be solved by perceptron.

Neural Network

Unit 6

ARCHITECTURES OF NEURAL NETWORK

- ANN is a computational system consisting of a large number of interconnected units called artificial neurons.
- The connection between artificial neurons can transmit signal from one neuron to another.
- There are multiple possibilities for connecting the neurons based on which architecture we are going to adopt for a specific solution.
- Some of the choices are listed below:
 - There may be just two layers of neuron in the network – the input and output layer.
 - Other than the input and output layers, there may be one or more intermediate ‘hidden’ layers of neuron.
 - The neurons may be connected with one or more of the neurons in the next layer.
 - The neurons may be connected with all neurons in the next layer.
 - There may be single or multiple output signals. If there are multiple output signals, they might be connected with each other.
 - The output from one layer may become input to neurons in the same or preceding layer

Single-layer feed forward network

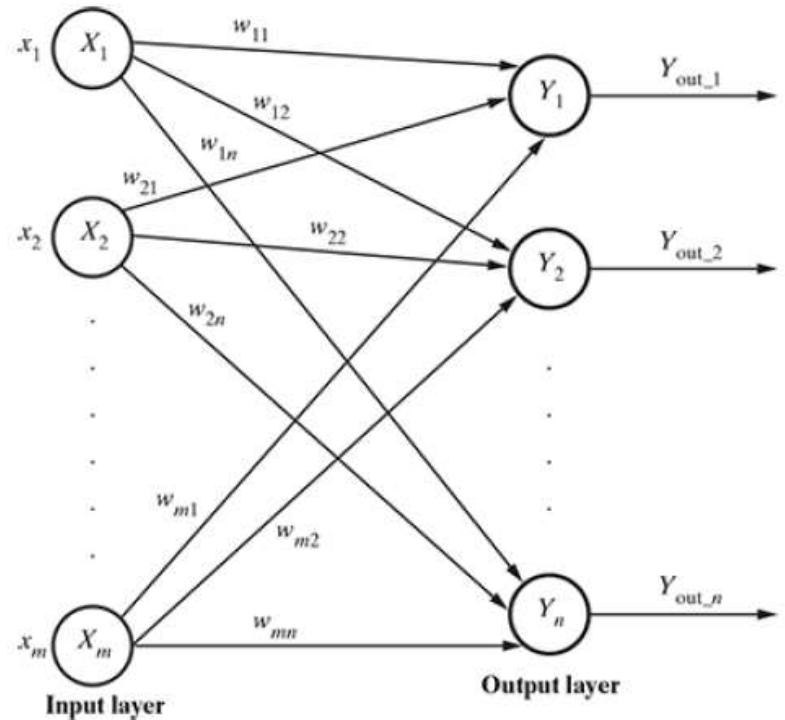
- Single-layer feed forward is the simplest and most basic architecture of ANNs.
- It consists of only two layers-the input layer and the output layer.
- The input layer consists of a set of ' m ' input neurons X_1, X_2, \dots, X_m connected to each of the ' n ' output neurons Y_1, Y_2, \dots, Y_n . The connections carry weights $w_{11}, w_{12}, \dots, w_{mn}$.
- The input layer of neurons does not conduct any processing – they pass the input signals to the output neurons.
- The computations are performed only by the neurons in the output layer.
- The network is known as single layer in spite of having two layers of neurons.
- As the signals always flow from the input layer to the output layer, this network is known as feed forward.

- The net signal input to the output neurons is given by

$$y_{\text{in},k} + x_1 w_{1k} + x_2 w_{2k} + \dots + x_m w_{mk} = \sum_{i=1}^m x_i w_{ik},$$

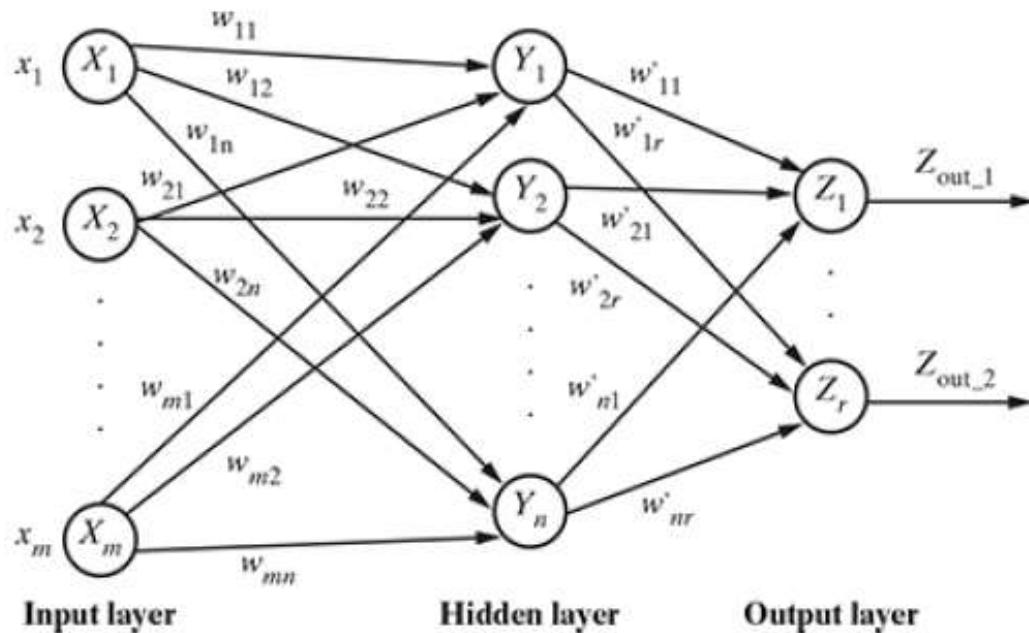
for the k -th output neuron.

- The signal output from each output neuron will depend on the activation function used.



Multi-layer feed forward ANNs

- There are one or more intermediate layers of neurons between the input and the output layers.
- Each of the layers may have varying number of neurons.



- The net signal input to the neuron in the hidden layer is given by

$$y_{in_k} = x_1 w_{1k} + x_2 w_{2k} + \dots + x_m w_{mk} = \sum_{i=1}^m x_i w_{ik},$$

for the k -th hidden layer neuron.

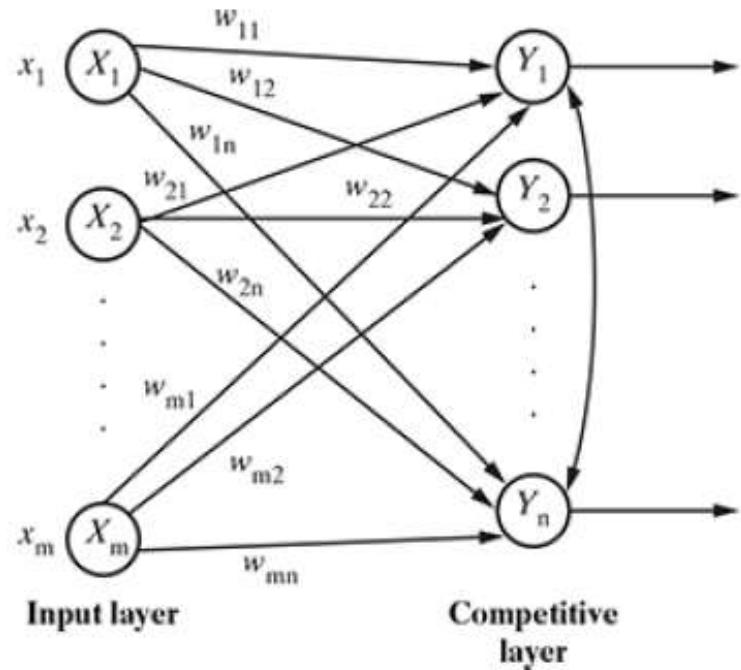
- The net signal input to the neuron in the output layer is given by

$$z_{in_k} = y_{out_1} w'_{1k} + y_{out_2} w'_{2k} + \dots + y_{out_n} w'_{nk} = \sum_{i=1}^n y_{out_i} w'_{ik}$$

for the k -th output layer neuron.

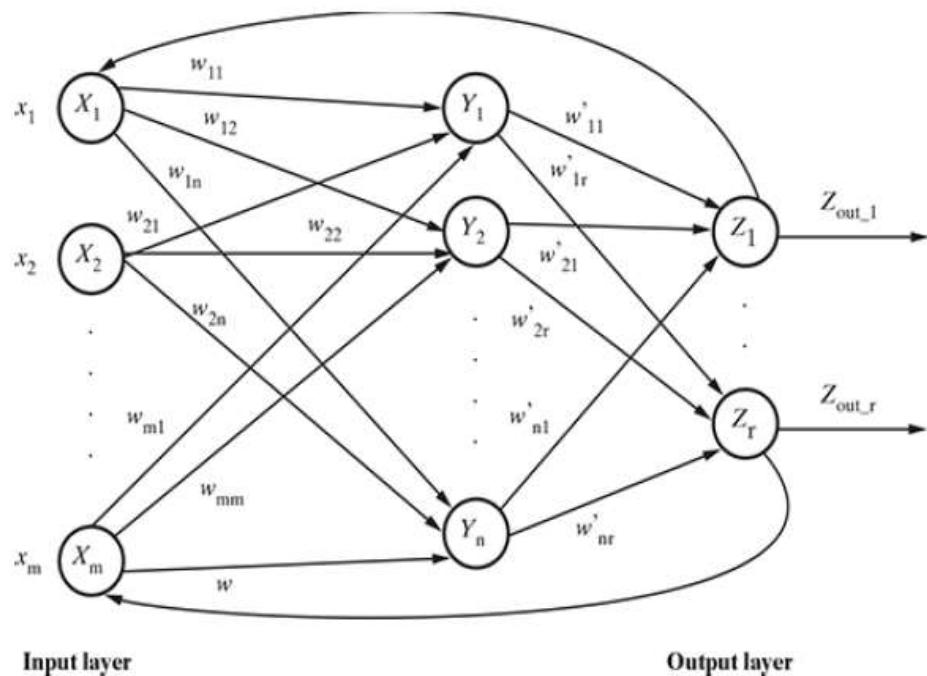
Competitive network

- The competitive network is almost the same in structure as the single-layer feed forward network.
- The only difference is that the output neurons are connected with each other (either partially or fully).
- For a given input, the output neurons compete amongst themselves to represent the input.
- It represents a form of unsupervised learning algorithm in ANN that is suitable to find clusters in a data set.



Recurrent network

- In feed forward networks, signals always flow from the input layer towards the output layer (through the hidden layers in the case of multi-layer feed forward networks), i.e. in one direction.
- In recurrent neural networks, there is a feedback loop, from the neurons in the output layer to the input layer neurons.



LEARNING PROCESS IN ANN

- Learning process using ANN is a combination of multiple aspects
- There are four major aspects which need to be decided before actual learning process starts in ANN:
 1. The number of layers in the network
 2. The direction of signal flow
 3. The number of nodes in each layer
 4. The value of weights attached with each interconnection between neurons

Number of layers

- A neural network may have a single layer or multi-layer.
- In the case of a single layer, a set of neurons in the input layer receives signal, i.e. a single feature per neuron, from the data set.
- The value of the feature is transformed by the activation function of the input neuron.
- The signals processed by the neurons in the input layer are then forwarded to the neurons in the output layer.
- The neurons in the output layer use their own activation function to generate the final prediction.
- More complex networks may be designed with multiple hidden layers between the input layer and the output layer.
- Most of the multi-layer networks are fully connected.

Direction of signal flow

- In feed forward networks, signal is always fed in one direction, i.e. from the input layer towards the output layer through the hidden layers, if there is any.
- The networks like recurrent network allow signals to travel from the output layer to the input layer.

Number of nodes in layers

- In the case of a multi-layer network, the number of nodes in each layer can be varied.
- The number of nodes or neurons in the input layer is equal to the number of features of the input data set.
- The number of output nodes will depend on possible outcomes, e.g. number of classes in the case of supervised learning.
- The number of nodes in each of the hidden layers is to be chosen by the user.
- A larger number of nodes in the hidden layer help in improving the performance.
- Too many nodes may result in overfitting as well as an increased computational expense.

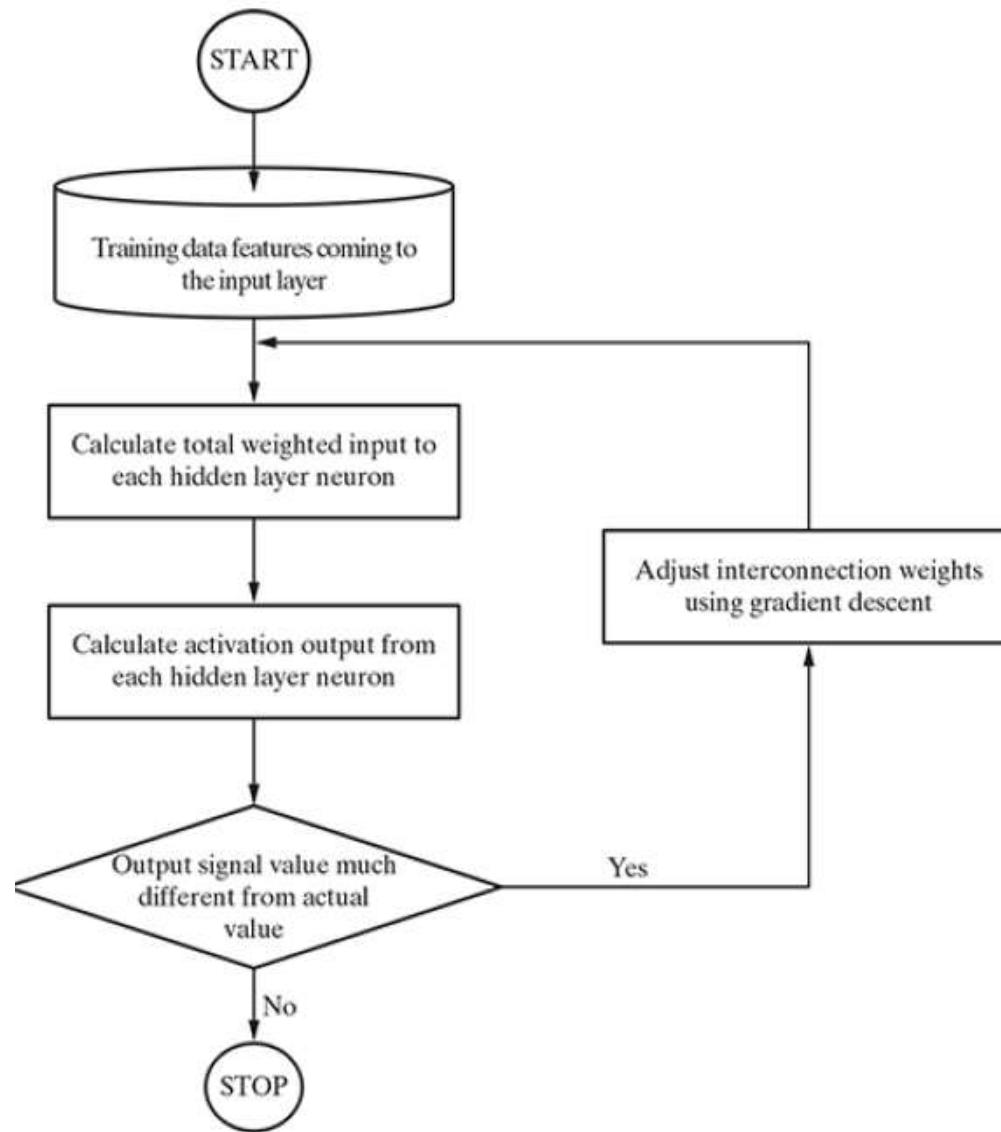
Weight of interconnection between neurons

- We can start with a set of values for the synaptic weights and keep doing changes to those values in multiple iterations.
- In the case of supervised learning, the objective to be pursued is to reduce the number of misclassifications.
- Ideally, the iterations for making changes in weight values should be continued till there is no misclassification.
- In practice, such a stopping criterion may not be possible to achieve.
- Practical stopping criteria may be the rate of misclassification less than a specific threshold value, say 1%, or the maximum number of iterations reaches a threshold, say 25, etc.
- There may be other practical challenges to deal with, such as the rate of misclassification is not reducing progressively.
- This may become a bigger problem when the number of interconnections and hence the number of weights keeps increasing.

BACK PROPAGATION

- Multi-layer feed forward network is a commonly adopted architecture.
- It has been observed that a neural network with even one hidden layer can be used to reasonably approximate any continuous function.
- The learning method adopted to train a multi-layer feed forward network is termed as backpropagation

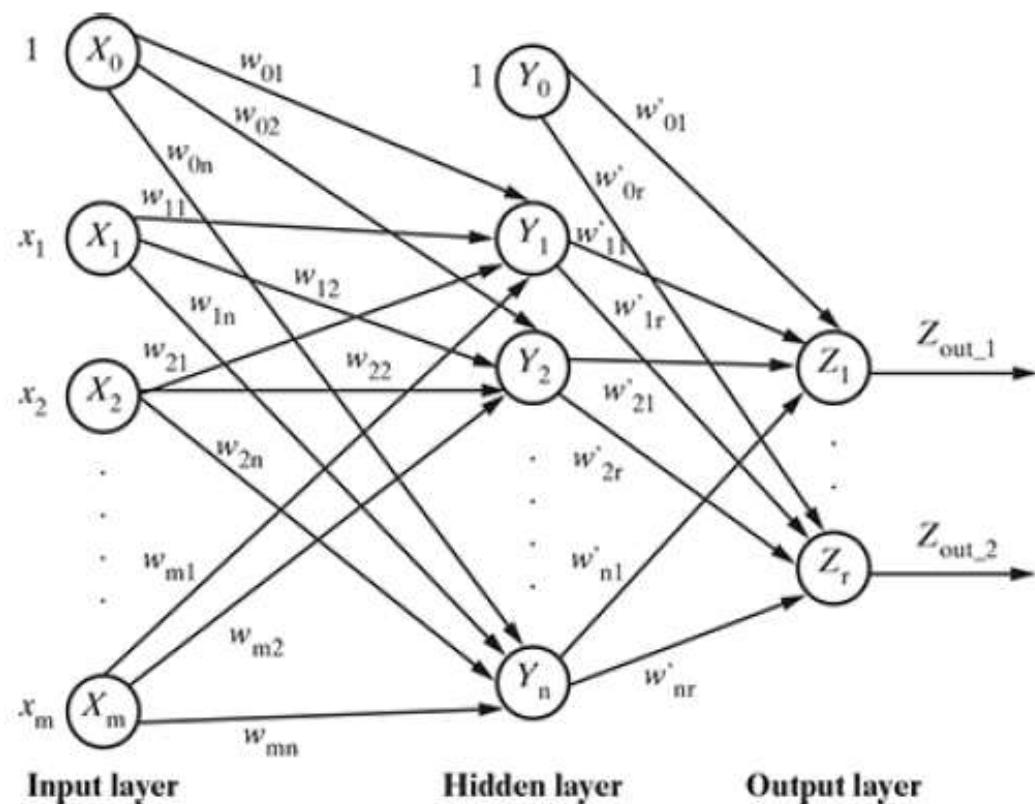
- The backpropagation algorithm is applicable for multi-layer feed forward networks.
- It is a supervised learning algorithm which continues adjusting the weights of the connected neurons with an objective to reduce the deviation of the output signal from the target output.
- This algorithm consists of multiple iterations, also known as **epochs**.
- Each epoch consists of two phases –
 - A **forward phase** in which the signals flow from the neurons in the input layer to the neurons in the output layer through the hidden layers. The weights of the interconnections and activation functions are used during the flow. In the output layer, the output signals are generated.
 - A **backward phase** in which the output signal is compared with the expected value. The computed errors are propagated backwards from the output to the preceding layers. The errors propagated back are used to adjust the interconnection weights between the layers.



Backpropagation algorithm

- One main part of the algorithm is adjusting the interconnection weights.
- A technique termed as gradient descent is used to adjust the interconnection weights between neurons of different layers.
- The algorithm calculates the partial derivative of the activation function by each interconnection weight to identify the ‘gradient’ or extent of change of the weight required to minimize the cost function.
- The multi-layer neural networks have multiple hidden layers.
- During the learning phase, the interconnection weights are adjusted on the basis of the errors generated by the network, i.e. difference in the output signal of the network and the expected value.

- These errors generated at the output layer are propagated back to the preceding layers.
- Because of the backward propagation of errors which happens during the learning phase, these networks are also called back-propagation networks or simply backpropagation nets.
- X_0 is the bias input to the hidden layer and Y_0 is the bias input to the output layer.



- The net signal input to the hidden layer neurons is given by

$$y_{in_k} = x_0w_{0k} + x_1w_{1k} + x_2w_{2k} + \dots + x_mw_{mk} = w_{0k} + \sum_{i=1}^m x_iw_{ik},$$

for the k-th neuron in the hidden layer.

- If f_y is the activation function of the hidden layer, then

$$y_{out_k} = f_y(y_{in_k})$$

- The net signal input to the output layer neurons is given by

$$z_{in_k} = y_{0}w'_{0k} + y_{out_1}w'_{1k} + y_{out_2}w'_{2k} + \dots + y_{out_n}w'_{nk} = w'_{0k} + \sum_{i=1}^n y_{out_i}w'_{ik}$$

for the k-th neuron in the output layer.

- If f_z is the activation function of the hidden layer, then

$$z_{out_k} = f_z(z_{in_k})$$

- If t_k is the target output of the k -th output neuron, then the cost function defined as the squared error of the output layer is given by

$$\begin{aligned} E &= \frac{1}{2} \sum_{k=1}^n (t_k - z_{\text{out},k})^2 \\ &= \frac{1}{2} \sum_{k=1}^n (t_k - f_z(z_{\text{in},k}))^2 \end{aligned}$$

DEEP LEARNING

- When the number of hidden layers in multi-layer neural networks is higher than three, it is termed as deep neural network.
- Deep learning is a more contemporary branding of deep neural networks.

Thankyou

