Module 3

Introduction to AI and Applications	Semester		I/II
Course Code	**1BAIA103/203**	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	3:0:0:0	SEE Marks	50
Total Hours of Pedagogy	40	Total Marks	100
Credits	3	Exam Hours	3
Examination type (SEE)		Theory	
Course outcome (Course Skill Set) At the end of the course, the student will be able to: CO1: Explain the concepts and types of artificial intelligence. CO2: Illustrate basic machine learning methods for regression, classification and clustering. CO3: Identify real-world applications across different disciplines. CO4: Make use of prompt engineering techniques to interact with generative AI tools. CO5: Outline recent trends in artificial intelligence and machine learning.			
Module-3			
Machine Learning: Techniques in AI, Machine Learning Model, Regression Analysis in Machine Learning, Classification Techniques, Clustering Techniques, Naïve Bayes Classification, Neural Network, Support Vector Machine (SVM).			

L1:

Artificial Intelligence Technologies

2.1 Techniques in AI

Artificial intelligence (AI) works by combining massive data with fast, iterative processing and intelligent algorithms, allowing the software to automatically learn to deduce patterns in data. Building an AI system is a process of reverse- engineering human traits and capabilities in a machine. AI is a broad field of study that includes many theories, methods and technologies. To understand how artificial intelligence actually works, we must look into its various sub domains. All these domains have one thing in common. They all process large amounts of data with fast and intelligent algorithms to allow the software to learn automatically from patterns or features in the data. The sub-domains of AI are given in Fig. 2.1.

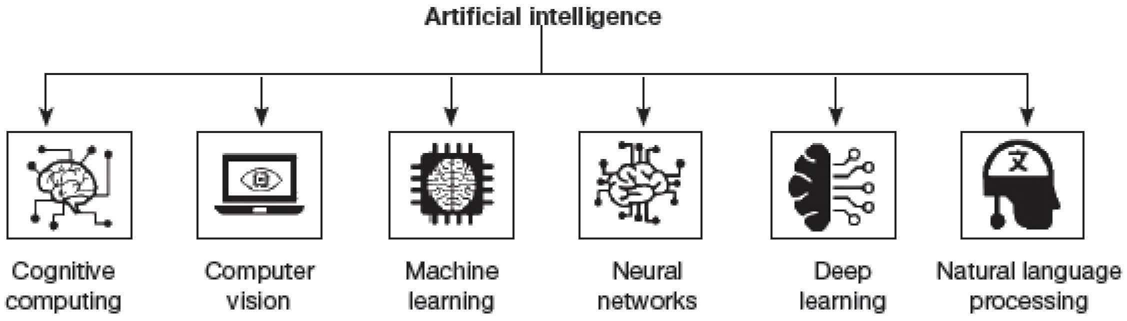


FIGURE 2.1 Sub-Domains of AI

1. **Neural Networks**: Neural networks work in the same way as human neural (brain) cells work. A series of nodes that capture the relationship between various underlying variables and processes the data. Every node (or neuron) process information by responding to external inputs and relaying information between each unit. The entire process of learning requires multiple passes at the data to derive meaning from undefined data (refer to Fig. 2.2).

   Note that even the most basic neural network consists of the following layers:

   An *input layer*, which is the layer from where data enters the network.

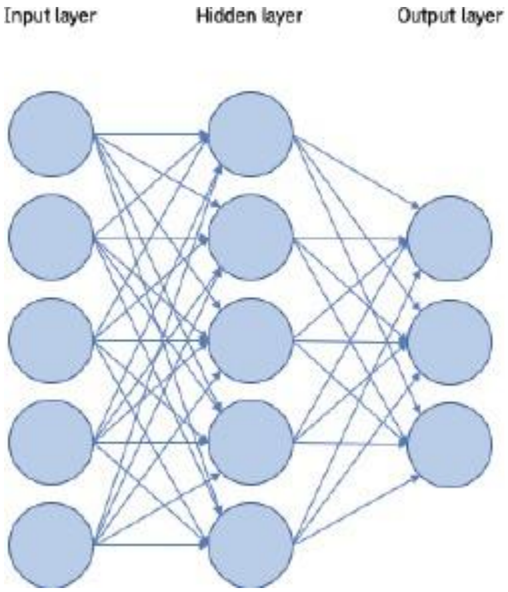


FIGURE 2.2 Basic neural network

At least one *hidden layer*, where machine learning algorithms process the inputs. Weights, biases and thresholds are applied to the received inputs and the results of processing are passed to the output layer.

An *output layer*, is the layer that gives the final result to be displayed.

2. **Machine learning**: Machine learning (ML) is a branch of computer science that analyses data and identifies patterns to teach a machine to deduce results and make decisions without any human intervention. ML algorithms learn from experiences rather than instructions. They automatically learn and improve by learning from their output. For this, humans do not have to write instructions for them to produce the desired output. They learn by analysing data sets and comparing the final output. In case of any error, they repeat the learning process until the accuracy of the outputs improve.

   This automation not only saves human time and effort but also make better decisions. In chapter 1, we have already seen that technologies like Machine Learning, Natural Language Processing, Deep Learning are all sub domains of Artificial Intelligence (refer Fig. 2.3)

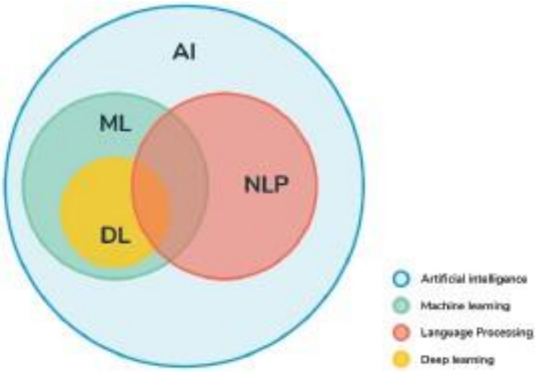


FIGURE 2.3 Relationship between AI, ML, DL and NLP.

3. **Deep learning**: Deep learning is an ML technique that teaches a machine to process inputs through layers to more accurately classify, infer and predict an outcome. DL creates huge neural networks with several layers of processing units to take full advantage of advances in computing power and improved training techniques. This helps the algorithm to learn complex patterns in large amounts of data. Some applications of DL include image and speech recognition.

   Deep learning models are based on *deep neural networks* (refer Fig. 2.4), that is, neural networks with multiple hidden layers. In such a network, each hidden layer further processes the temporary outputs received from the previous layer.

This movement of computations through the hidden layers to the output layer is known as *forward propagation*.

Once the final result is produced by the output layer, its accuracy is calculated. In case of unsatisfactory results, errors are identified, weights assigned to each node are updated, and pushed back to the previous layers to refine or train the model. This process of moving backward to update weights of all nodes is known as ***backward propagation***.

Deep learning models can work with labeled as well as unlabelled data. This means that deep learning supports both supervised and *unsupervised learning.*

4. **Natural language processing**: NLP is a science in which a machine is made to read, understand, interpret and respond to a human language. This is specifically done to make machine capable of communicating with a human.

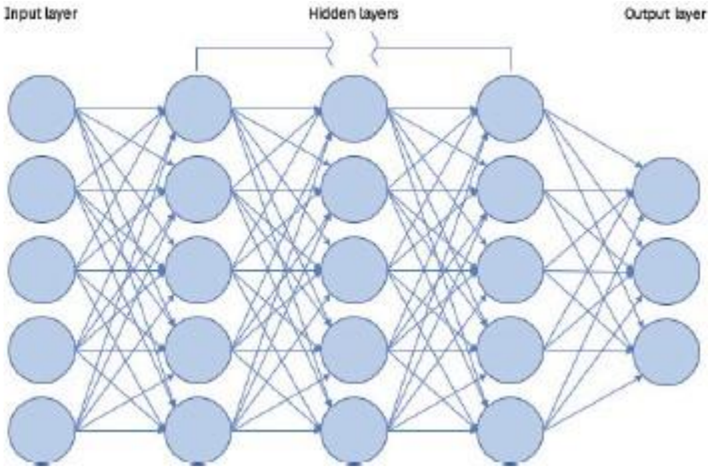


FIGURE 2.4 Deep neural network

5. **Computer vision**: Computer vision is a branch of AI that tries to understand an image by breaking it down into several parts and then studying each part of the image. This helps machine to classify and learn from a set of images so that it can make better decisions based on previous observations. For example, when a machine can process, analyse and understand images, they can capture images or videos in real time and interpret their surroundings.

   Computer vision techniques are used today for facial recognition that helps in surveillance and security systems, autonomous vehicles, retail stores for tracking inventory and customers, in medicine for diagnosing diseases, in financial Institutions to prevent fraud, and so on.

6. **Cognitive Computing**: Cognitive computing algorithms try to mimic human brain by analysing text, speech, images or objects in the same way to give the desired output. **It** is basically a subfield of AI that is used to provide a natural, human-like interaction with machines. The ultimate goal of using cognitive computing is to make the machine speak coherently in response to a human.

Apart from the above-mentioned techniques, some additional technologies that enable and support AI include the following:

1. **Graphical processing units** that provide heavy computing power required for iterative processing and training neural networks.
2. **Internet of Things to** generate massive amounts of data from connected devices. Usually, this data remains unanalysed. Automating models with AI helps to analyse this data and extract useful information from it. Advanced algorithms **can be used** to analyse data faster at multiple levels to identify and predict rare events, understanding complex systems and optimizing unique scenarios.

Review Questions:

1.What is ML?

2.What is DL?

3.What is NLP?

L2:

2.2 Machine Learning Model

We can better understand the role of machine learning techniques through a very simple definition given by professor Mitchell –

'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.' Based on this definition, the Machine Learning Model can be given as shown in Fig. 2.5. Here,

Task(T), is the real-world problem to be solved. For example, predicting sales of a product, classifying an email as spam or not a spam, etc. Technically, examples of ML based tasks are classification, regression, clustering, recognition, etc.

Experience (E) is the knowledge gained from data provided to the algorithm or model. Once data is provided, the model runs iteratively to learn some inherent patterns. Therefore, like humans, machines now learn from experience by analysing situation, relationships etc. Supervised, unsupervised and reinforcement learning are some ways to learn or gain experience. Experience acquired through ML model or algorithm is used to solve task T.

Performance (P) is a measure that indicates how well a particular ML algorithm has performed the given task T using experience E. Performance is analysed based on well-defined metrics including accuracy, F1, confusion matrix, precision, recall, sensitivity, etc.

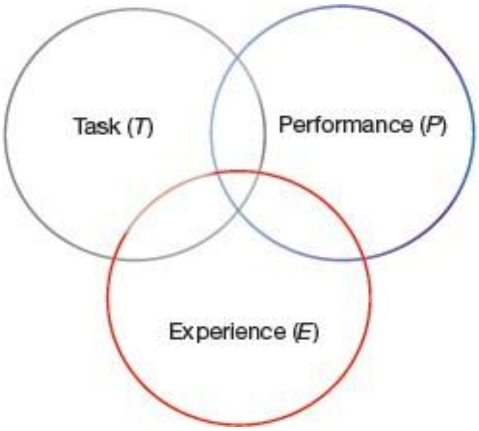


FIGURE 2.5 ML Model is a combination of Task, Performance and Experience

2.2.1 Types of Machine Learning Algorithms

Machine learning algorithms can be categorized as supervised or unsupervised.

2.2.1.1 Supervised Machine Learning Algorithms

As the name indicates, such algorithms have a supervisor as teacher. These algorithms apply learning from past data (or experiences) to new data using labelled examples.

Technically, supervised learning algorithms learn an association between input data and output. For example, if we have input variable(s) X and output variable (Y), then the mapping function from input to output can be given as, Y=f(x). This mapping function can be used to predict the output value for any new input after learning from the existing data.

For example, if you have a basket filled with different varieties of fruits then the first step is to train the machine to identify a fruit. Now, our machine can easily identify an apple and a banana. Supervised learning algorithms can be further classified into two categories:

Supervised learning should be used when output of data in the training set is known.

Classification Algorithm

A classification algorithm classifies data into a particular group. Classification techniques predict discrete categories. The output will be based on what the model has learned in training phase.

For example, a fruit as either an apple or a banana. In real world applications, classification can be used in medical imaging, speech recognition, hand-writing recognition, credit scoring, predict if an incoming email is authentic or spam, or whether a tumour is cancerous or benign.

Classification algorithms are best used if data can be tagged, categorized, or separated into specific groups or classes.

Regression

A regression algorithm predicts a real value. The output value is based on what the model has learned in its training phase. In contrast to classification algorithms, regression predict continuous values. For example, the cost of a product, the value of a stock, changes in temperature or fluctuations in power demand.

Thus, in supervised machine learning algorithm, data input and desired output, along with a feedback about the accuracy of predictions during algorithm training are

provided. Data scientists can select variables or features that can be used by the model to analyse data and make predictions. For example, supervised learning can be used by a company to identify customers who are likely to churn. It can also be used by insurance companies to predict the likelihood of occurrence of a mishap and determine the total insurance value.
In supervised learning, clear instructions are given specifying what needs to be learnt and how it needs to be learnt.

2.2.1.2 Unsupervised Learning

Unsupervised learning trains the machine using information that is neither classified nor labelled. In this case, the machine learning algorithm works on that information without any guidance. The unsorted information is grouped based on similarities, patterns and differences without any prior training of data. For example, if we give an image of mango and an orange, then initially, the machine has no idea about how a mango looks and how the orange looks.

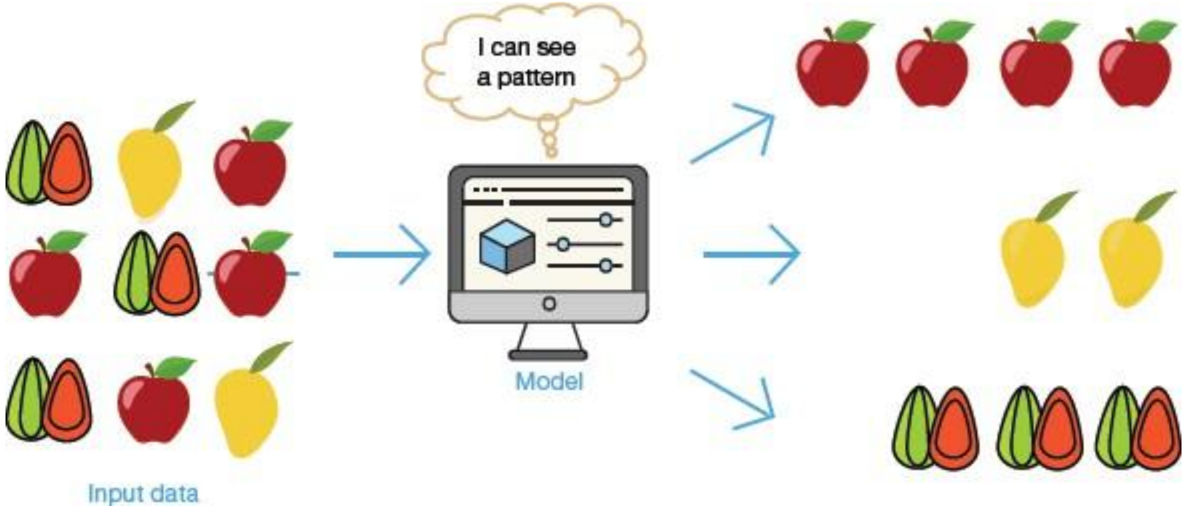


FIGURE 2.6 Grouping fruits by identifying patterns

Unsupervised machine learning algorithm learns through observation and finding structures in the data. That is, the model automatically finds patterns and relationships in the dataset by creating clusters in it. For example, if given a dataset of pictures of both mangoes and oranges, the algorithm can make two clusters- one containing only pictures of oranges and the other of mangoes (refer Fig. 2.6). What an unsupervised machine learning cannot do is specifying labels to the clusters. That is, it can only segregate the pictures but cannot tell that this is a real-world orange and that is a mango.

Unsupervised learning can be categorized into following sets of algorithms:

1. ***Clustering:*** In clustering, the aim is to discover inherent groupings in the data or discover hidden patterns (refer Fig. 2.7). It is one of the most useful unsupervised machine learning techniques as it finds similarity as well as relationship in the underlying data. For example, a company may like to group its customers by their purchasing behaviour. A cell phone company can use clustering to optimally decide the locations where they can build cell phone towers (these clusters can depict the number of people relying on their towers). Other applications include gene sequence analysis, market research, and object recognition.
2. ***Association analysis:*** In association mining (or analysis), the aim is to discover rules that describe large portions of data. For example, a company can use association analysis to conclude that customer who buys X also buys product Y.
3. ***Dimensionality reduction:*** It is used to reduce the number of feature variables for the data set. It is done by selecting a set of principal or representative features. Dimensionality reduction is very important technique especially when the data set has a large number (millions).
4. ***Outlier detection or anomaly detection:*** This technique is used to find out the occurrences of rare events or observations that generally do not occur. Application of learned knowledge in anomaly detection techniques helps to differentiate between anomalous or a normal data point. Generally, clustering (more specifically, KNN) is used to detect anomalies based in data.

Thus, unsupervised machine learning algorithms (also called neural networks) are used when the information used to train is neither classified nor labelled. These algorithms use an iterative approach called deep learning to review data and arrive at conclusions. Unsupervised learning algorithms are used for more complex tasks than supervised learning systems. For example, these algorithms are used in image recognition, speech-to-text and natural language processing applications, predict the probability of presence of a particular disease. A retailer can use unsupervised learning technique to find out products that are frequently bought by customers tends to buy more frequently.

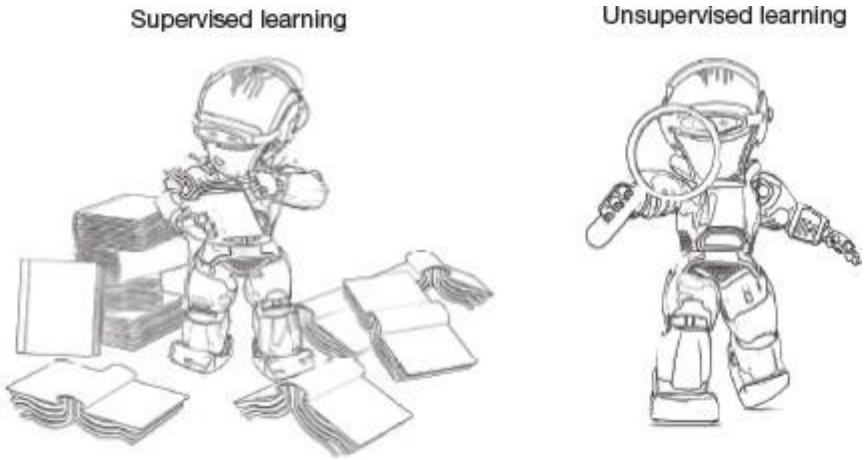


FIGURE 2.7 In supervised learning, we have some clue about what exactly we are finding but unsupervised learning explores data to find just any hidden pattern

2.2.1.3 Semi-Supervised Learning

Semi-supervised machine learning algorithms fall somewhere between supervised and unsupervised algorithms since they use both labelled and unlabelled data for training. These algorithms improve accuracy.

Semi-supervised learning algorithms use small amount of pre- labelled data and a large number of unlabelled data for training. Semi-supervised learning techniques can be applied using any of the two approaches given below.

First, to build the supervised model based on small amount of labelled data followed by building an unsupervised model by applying the same to the large amounts of unlabeled data. Experience gained by generating more labelled data is used to train the model. This process is repeated multiple times. In the second approach, unsupervised methods are used to cluster similar data samples, annotate these groups and then use this information to train the model.

Application of the discussed techniques can be summarized in the flowchart given in Fig. 2.8 below.

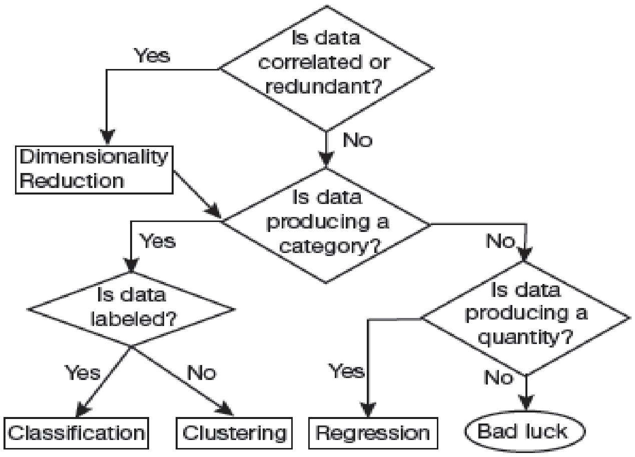


FIGURE 2.8 Summary of ML Algorithms

2.2.1.4 Reinforcement Learning (Rl)

These techniques are different from the previously discussed techniques and are rarely used. In a reinforcement learning algorithm, an agent is trained over a period of time so that it can interact with a specific environment.

Reinforcement learning, is a type of dynamic programming that trains algorithms using a system of **reward and punishment**. The agent receives rewards by performing correctly and penalties for performing incorrectly. In this way, the agent learns without any human intervention to maximize its reward and minimize its penalty. Since RL requires a lot of data, it is mostly used in areas where simulated data is readily available like gameplay and robotics. In these areas, RL is used to find the best possible behaviour or path that can be taken in a particular situation.

Reinforcement learning is different from supervised learning. In supervised learning, the training data has labels, so the model is trained with the correct answer but in case of RL, the reinforcement agent decides what to do to perform the given task. In the absence of a training dataset, it is bound to learn from its own experience.

Example: In the game shown in Fig. 2.9, there is an agent (robot) and a reward (diamond), with many hurdles (fire) in between. The robot has to learn by trying all the possible paths and then choose the path that gives him the reward with the least hurdles. Each right step earns a reward and every wrong step will subtract the reward of the robot. The total reward is calculated when it reaches the final reward that is the diamond.

To summarize technically, in an RL algorithm, note the following:

1. Input is an initial state from which the model will start
2. Output is a list of possible outputs for a particular problem
3. Training is based on the input. The model returns a state and the user will decide to reward or punish the model based on its output. The model keeps learning this way until it finds the best solution (or the solution with maximum reward).

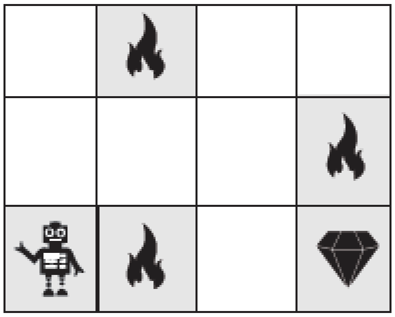


FIGURE 2.9 Robot and the Reward Game

2.2.1.4.1 Types of Reinforcement

There are two types of reinforcement as follows:

Positive reinforcement has a positive effect on behaviour. It occurs when a particular behaviour, increases the strength and the frequency of the behaviour. Such a reinforcement maximizes performance and sustain changes for a long period of time.

Negative reinforcement is defined as strengthening a behaviour by stopping or avoiding a negative condition.

2.2.1.4.2 Applications of Reinforcement Learning

RL is used in large environments when a model of the environment is known, but an analytic solution is not available.

1. RL can be used in robotics for industrial automation.
2. It is used to make machines learn
3. RL is used in data processing applications
4. RL can be used to create training systems that provide custom instruction and materials according to the requirement of students.
5. RL is used for creating game playing software (playing chess).
6. RL is used in a self-driving car where the car (agent) interacts with its environment, receives a reward depending on how it performs, such as driving to destination safely. Similarly, the agent receives a penalty for performing incorrectly, such as going off the road or hitting a hurdle.

An RL agent perceives and interprets its environment, takes actions and learns through trial and error.

Review Questions:

1.What is Experience(E)?

2.What is Performance(P)?

3.What is positive and negativeReinforcement?

L3:

2.3 Regression Analysis in Machine Learning

Regression analysis is a statistical method tool that allow users to study the relationship between a dependent (target) and one or more independent (predictor) variables as shown in Fig. 2.10. This helps the analyst to understand how the value of the dependent variable changes with respect to independent variables. The values to be predicted are usually continuous or real (like **temperature, age, salary, price,** etc.). For example, a company can analyse its data to find out the extent to which expenditure on advertisements increases sales of a particular product. Similarly, we can use regression analysis to predict rainfall using temperature and other factors;

determine market trends; predict road accidents due to rash driving.

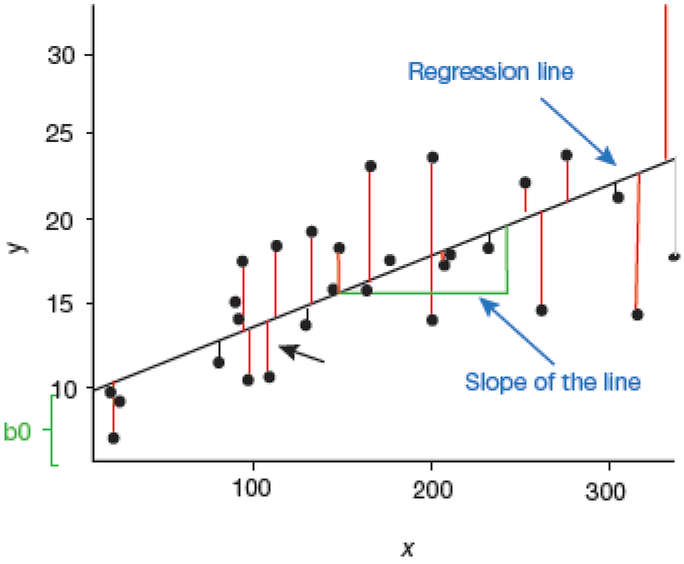


FIGURE 2.10 Plotting the Regression Line

Regression analysis is widely used for **prediction, forecasting, modelling time series data, and inferring the causal-effect relationship between variables.**

In regression analysis, a graph that best fits the given datapoints is plotted. This graph can either have a straight line or a curve that *passes through all the datapoints on the graph that is drawn depicting the dependent and independent variable(s). The vertical distance between the datapoints and the regression line should be minimum as this* line indicates whether the model has captured a strong relationship or not.

2.3.1 How Regression Analysis Works?

Regression analysis creates a mathematical equation that defines y as a function of the x variables. This equation is then used to predict the value of y on the basis of values of the predictor variables (x).

Linear regression is the most basic, simple and widely used technique for predicting values of a continuous variable. It assumes that there exists a linear relationship between the outcome and the predictor variables as shown in Fig. 2.11.

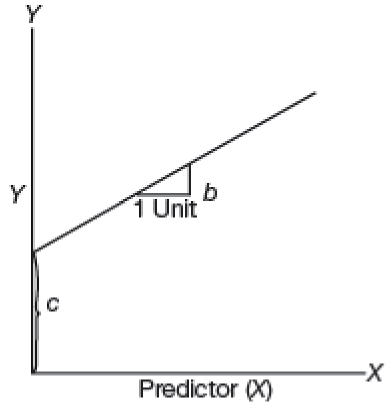


FIGURE 2.11 Relationship between outcome and predictor variable

The linear regression equation can be given as,

$$y = b0 + b*x + e, \text{ where:}$$

b0 is the intercept,
b is the coefficient of x. e is the

residual error

Values of coefficients are determined in such a way that the residual error is minimized. This method of computing the beta coefficients is known as **ordinary least squares** (OLS) method.

In case of multiple predictor variables, say x1 and x2, the regression equation can be written as **y=b0+b1*x1+b2***

x2 +e. There may also exist an **interaction effect** between two or more predictor variables. For example, increasing the value of one predictor variable may in turn increase the effectiveness of the other predictor(s) in explaining the variation in the outcome variable.

When there are multiple predictors in the regression model, then the best combination of predictors needs to be chosen (like **best subsets regression** and **stepwise regression)** to construct the optimal predictive model. In such cases, we use model selection technique that compares multiple models built with different sets of predictors to select the best performing model that minimizes the prediction error.

Linear regression models work well with both continuous and* categorical predictor variables.* However, before applying linear regression model on our data set, we must first make sure that the linear model is suitable for the underlying data. It may happen that the relationship between the outcome

and the predictor variables is not linear. In such cases, a **non-**

linear regression model (like *polynomial and spline regression)*

is preferred over a linear model.

When there is a large multivariate data set containing some correlated predictors, then such variables are summarized into few new variables that are a linear combination of the original variables. These techniques based on p**rincipal components** include **principal component regression** and **partial least squares regression.** Alternatively, we can use penalized regression to simplify a large multivariate model.

Such a regression model penalizes the model for having too many variables. R**idge regression** and L**asso regression** are popular examples of penalized regression model. Thus, different regression models can be applied on the data and their performance can be compared to select the best one that explains the data. Statistical metrics are used to compare the performance of the different models in explaining the data and predicting the outcome of new test data.

Case Study 1: Let us compute auto fare. We know that auto charges are computed by adding a fixed amount and a variable cost. The fixed amount is set by the government. The variable cost depends on the distance travelled. It is usually specified as Rs 11 per km. If the fixed charge of an auto is Rs 30, then a linear regression equation can be used to find the cost of any auto trip. By using “x” to represent the number of kilometers travelled, “y”, the cost of that auto ride can be calculated as, $y = 11x + 30$.

Now, if I say that I took an auto to reach a destination that was 10 km away from my house, how much would I have to pay?

Yes, Rs. 140. Because, y (cost) = 11 X 10 + 30.

Case Study 2: A company decides to carry out its business operations on a rented space. If the cost of the rental space is Rs 20000 plus Rs 500 per employee per day, then compute monthly rental for space given that the company is open 5 days a week.

The linear equation in this scenario, can be given as, y = (500)(5)(4)x

+ 20000

y = 10000x + 20000

If 20 employees are to be present every day, then monthly rental would be

Y (cost) = 10000 X 20 + 20000 = 220000.

Case Study 3: Let us make predictions using simple linear regression equation. If the annual expenditure of a bakery shop is Rs 500000 and the monthly sales is Rs 450000, then the linear equation to compute profit for x months can be given as,

y = 450000x – 500000

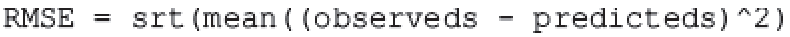
For example, after six months, the shop can expect a profit of, 450000*6 – 500000 =

2200000.

2.3.2 Model Evaluation Metrics

The best regression model is the one with the lowest prediction error. The most widely used metrics for comparing regression models are discussed here:

Root mean squared error measures the model prediction error. It is calculated as the average difference between the observed known values of the outcome and the predicted value by the model. Mathematically,



```
RMSE = srt(mean((observeds - predicteds)^2)
```

Lower the RMSE, better is the model.

Adjusted R-square represents the proportion of variation in the data thereby reflecting the overall quality of the model.

Higher the adjusted R^2, better is the model

These metrics are computed on a new test data that has not been used to train or build the model. In a large data set with many records, rows can be randomly split in 80:20 ratio, where 80% of the rows are used to build the predictive model and rest of the 20% are used as test set or validation set for evaluating the model performance.

One of the best techniques for estimating a model's performance is **k-fold cross-validation**. This can be applied even on a small data set. Step followed to perform k-fold cross- validation are as follows:

Step 1: Randomly split the data set into k-subsets (or k-fold). For example, to generate 5 subsets, value of k = 5.

Step 2: Reserve one subset and call it as test data. Use rest of subsets to the train the model.

Step 3: Test the performance of the model using the test data set and record the prediction error

Step 4: Repeat the above steps until each of the k subsets have been used as the test set.

Step 5: Calculate the average of the k recorded errors. This is also known as cross-validation error. Finally, the best model is the one that has the lowest cross-validation error, RMSE.

2.3.3 Types of Regression

The type of regression analysis that can be applied on a particular data set depends on the attributes, target variables, shape and nature of the regression curve that represent the relationship between dependent and independent variables. Some important regression techniques are as follows:

1. **Linear regression:** It is the simplest regression technique used for predicting values for a dependent variable that has linear relationship between the response and predictors (or descriptive variables). The general equation of linear regression can be given as,

$$Y = bX + C$$

where Y is a dependent variable X is the independent variable

b is the slope of the regression line and C is the intercept.

Though linear regression suffers from overfitting issues, it is still used as it is fast and easy to model and evaluate. It is best used when the target relationship is not complex or enough data is not available. Linear regression (Fig. 2.12) is also very useful for detecting outliers.

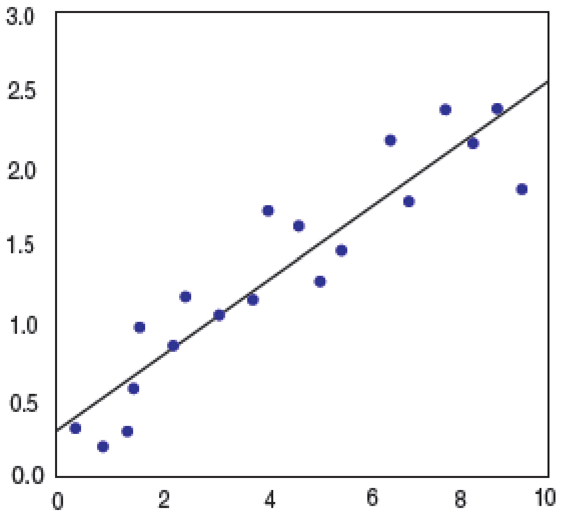


FIGURE 2.12 Linear Regression Graph

2. **Logistic regression:** It is preferred when the dependent variable is binary (dichotomous) in nature (like 0 or 1, true or false, yes or no). That is, logistic regression is preferred when there is a need to ascertain the probability of an event in terms of either success or failure. In such a case, the relationship between the dependent and independent variables are calculated by computing probabilities using the

logit function (refer Fig. 2.13). Logistic regression is mostly used to analyse categorical data.

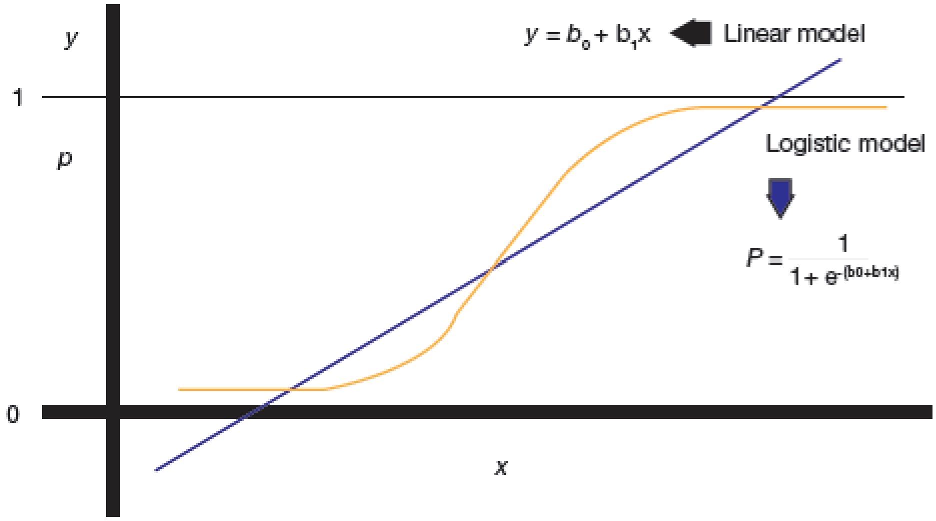


FIGURE 2.13 Logistic Regression Curve

3. **Ridge regression:** Ridge regression is basically used for analysing numerous regression data. In case of multicollinearity, least-square calculations get unbiased, so the regression model becomes too complex and approaches to overfit. In such a case, we need to minimize the variance in the model and save it from overfitting. Ridge regression helps us to do this by correcting the size of the coefficients. A bias degree is affixed to the regression calculations that in turn reduces standard errors.
4. **Lasso (Least Absolute Shrinkage Selector Operator) regression:** It is a widely used regression analysis technique for variable selection and regularization. Lasso regression uses thresholds to select a subset of the covariates given for

the implementation of the final model. **It** reduces the number of dependent variables if the penalty term is huge. This is done by reducing the coefficients to zero so that features can be selected easily. Lasso regression is also known as L1 regularization.

Ridge regression eases collinearity in between predictors of a model

5. **Polynomial regression:** Polynomial regression is used to construct a model that fits non-linearly separated data. In such a case, the best-fitted line is not a straight line, but a curve.

   The equation of polynomial regression is given as,

   Y=b0+b1x1+b2x22+ bn xnn

   Polynomial regression is widely used to analyse curvilinear form of data (refer Fig. 2.14) and best fitted for least-squares methods.

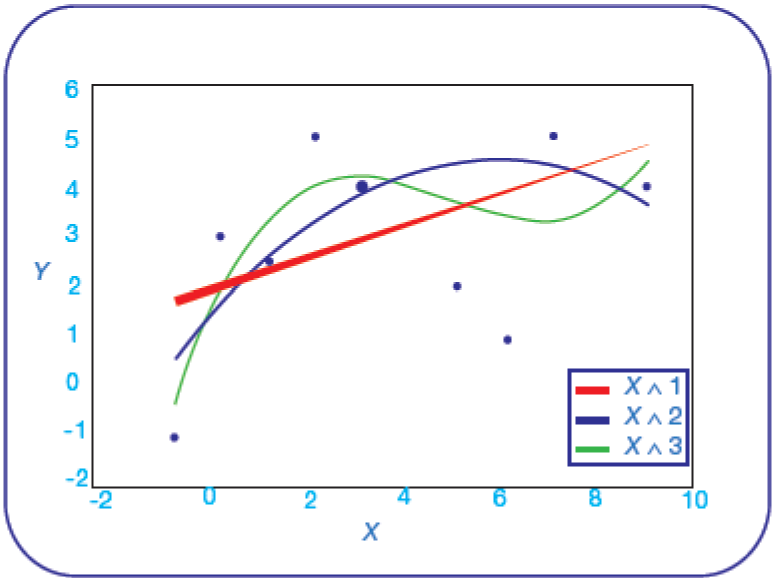


FIGURE 2.14 Polynomial Regression Curve

6. **Stepwise regression:** It is used to build predictive regression models that are carried out naturally. With every forward step, a variable gets added or subtracted from a group of descriptive variables.

   We can use **forward selection**, **backward elimination** or **bidirectional elimination** for adding/removing variables at each step. As the name suggests, forward selection continuously adds variables to the set and reviews the performance of the model. It stops when no further improvement is needed (or being achieved). Backward elimination removes variables until no extra variable is left to be deleted without considerable loss. Bidirectional elimination is a good combination of both the approaches.

7. **Elastic Net regression:** Elastic regression is a mix of ridge and lasso regression. It produces a grouping effect when highly correlated predictors are either in or out in the model combinedly. This technique is preferred when there are too many predictors as compared to the number of observations. ElasticNet regression is usually used in SVM (Support Vector Machine Algorithm), metric training, and document optimizations.

Review Questions:

1.What is Regression?
2.List the types of Regression?
3.What is polynomial Regression?

L4:

2.4 Classification Techniques

In this section, we will read about some widely used classification algorithms. These algorithms predict the probability that the data that follows will fall into one of the predetermined categories.

2.4.1 K-Nearest Algorithm

The k-nearest neighbour algorithm is a supervised learning algorithm in which the output value of data is known but how to get the output is not known. For example, if we have several groups of labeled samples and all items available are homogeneous, then to find which group an item with unknown label belongs, we will find similarities between the item (at hand) and with items in each group. Finally, we will conclude that the item belongs to the group to which it is most similar. The k-nn algorithm works in exactly the same way.

The k nearest neighbours' algorithm stores all available cases and classifies new cases by a majority vote of its k neighbours. The algorithm segregates unlabelled data values into well- defined groups.

Technically speaking, ***k-nn is a non-parametric supervised learning algorithm*** which classifies data into a particular category with the help of training set. Here, the word non- parametric means that it makes assumptions on the underlying data distribution. Non-parametric methods do not have fixed numbers of parameters in the model. The parameters in the model grows with the training data set.

In the k-nn algorithm, the value for a new instance (x) is predicted by searching the training set for the k most similar cases (neighbours) and summarizing the output values for those k cases. In other words, this is the mode (or most common) value.

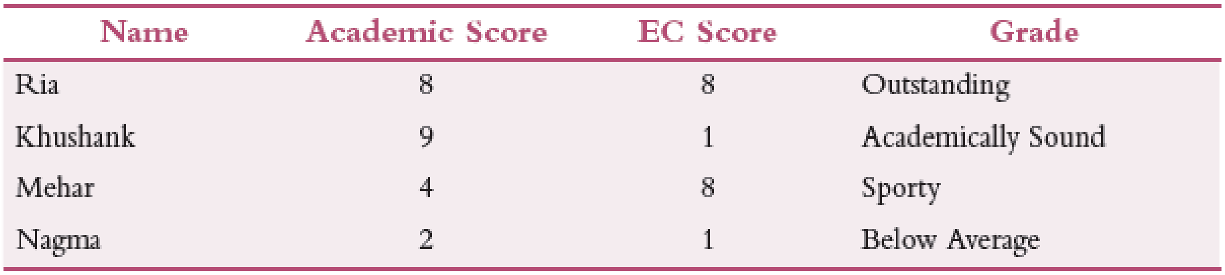
2.4.1.1 Choosing an Appropriate K Value

The most important thing to do in the knn algorithm is to determine an appropriate value of k, that is, the number of nearest neighbours. A large value of k reduces the noisy data. But if we have more data points in one group then we may ignore the smaller patterns which may have useful insights.
k-nn uses all of the data for training while classifying a new data point or instance.

2.4.1.2 Example of Knn Algorithm

Suppose we have n number of students rated on two parameters – Academic Score and EC Score on a scale of 1 to 10 as given in Table 2.1.

TABLE 2.1 Academic Score and EC Score



Name	Academic Score	EC Score	Grade
Ria	8	8	Outstanding
Khushank	9	1	Academically Sound
Mehar	4	8	Sporty
Nagma	2	1	Below Average

Then, academic score is an indication of how well the student performs academically and EC score is the score obtained by the student in extra-curricular activities.

Now, if we have a new student to categorize then we will calculate distance between 'New Student' and its nearest neighbours ('Outstanding', 'Sporty', 'Academically Sound' and 'Below Average') using the Euclidean distance formula as shown in Fig. 2.15.

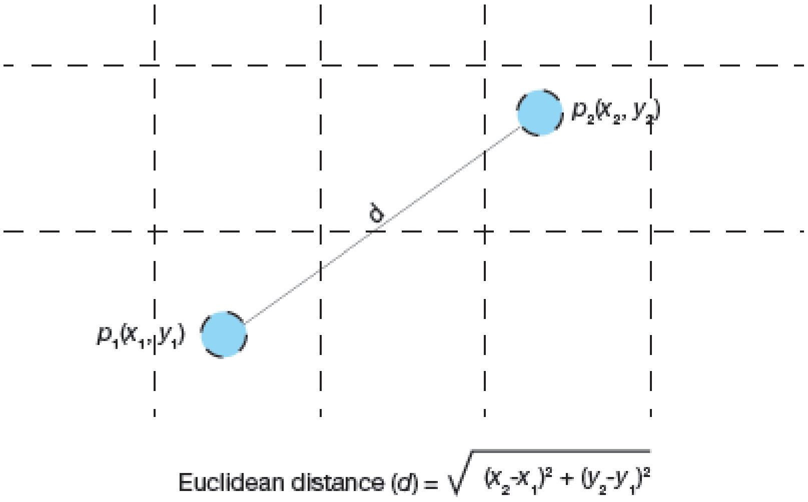

$p_2(x_2, y_2)$
d
$p_1(x_1, y_1)$
Euclidean distance (d) = $\sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$

FIGURE 2.15 Euclidean distance between two points

Assuming that co-ordinates of the New Student are (8,2), we calculate the distance between Below Average (2,1) and New Student as:

dist(Below Average, New Student) = 6.08

Similarly, calculate the distance of New Student from each of its nearest neighbours as shown in Table 2.2.

TABLE 2.2 Scores including New Data

NAME	Academic Score	EC Score	Grade	Distance to new Student
Ria	8	8	Outstanding	6.08
Khushank	9	1	Academically Sound	1.41
Mehar	4	8	Sporty	7.21
Nagma	2	1	Below Average	6.08

We see that the distance between New Student and Academically Sound is the least so the New Student belongs to the group of academically sound students.

k- NN algorithm can also be used for prediction. It is extensively used in pharmaceutical industry to detect the growth of oncogenic (cancer) cells or presence of disease.

Another Example

Consider Fig. 2.16. If we have to two classes – class A denoted by red stars and class B indicated by blue triangles, then to classify a new green square object the value of k plays an important role. If k=3, then according to Fig. 2.16, the 3 nearest neighbours belong to class B. So, the new object is placed in class B. but if we take k = 7, then the new object will be a member of class A. If we take k = 1, then the new object will belong to class B. Hence, value of k greatly influences the classification.

The K-nearest neighbour can be used for regression when dependent variable is continuous. In this case, the predicted value is the average of the values of its k nearest neighbours.

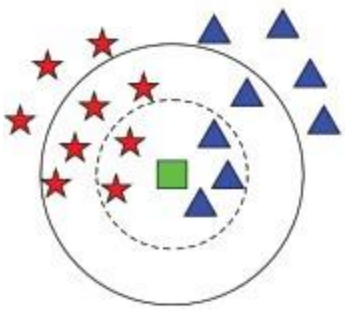


FIGURE 2.16 Finding nearest neighbor for the new data point (green). The new data will either belong to the group represented by red star or to the one represented as blue triangle.

2.4.1.3 Pros and Cons of Knn Algorithm Pros

1. The kNN algorithm is a very simple and effective algorithm.
2. Users can easily implement the algorithm which has made the algorithm quite popular with data professionals.
3. The algorithm is highly unbiased in nature and does not make any prior assumption of the underlying data.
4. The kNN algorithm works well with multi-class algorithms
5. The algorithm can be applied to both classification and regression.

KNN is a non-parametric learning algorithm, which means that it doesn't assume anything about the underlying data.

Cons

1. Usually, simple things are not very powerful; same is the case with kNN algorithm.
2. Although the algorithm trains the model really fast, the prediction time is high
3. Many a time, useful insights may get ignored.
4. The algorithm is sensitive to the scale of data so data must first be standardized
5. The performance of the algorithm deteriorated when there are multiple independent variables
6. The algorithm requires more memory and is computationally expensive. It does not work well if the target variable is skewed
7. The accuracy of the algorithm depends on k value. For any given problem, a small value of k will lead to a large variance in predictions. And setting k to a large value may lead to a large model bias.

Handling categorical variables in KNN

If you have categorical variables, then create k dummy variables out of that categorical variable. For example, if a categorical variable named “Designation” has 3 unique levels / categories then create 3 dummy variables. Each dummy variable has 1 against its designation and 0 otherwise.

2.4.2 Decision Trees

Do you remember the ‘Guess What’ game that we used to play in our childhood? One of us will think of something, and the others had to guess what it is. For that they can ask questions, answers of which will be either ‘Yes’ or ‘No’ based on these clues, the answer is given. For example, consider the tree given in Fig. 2.17. (Note, it is just a sample tree, not a general tree).

According to the rules specified, if you are thinking of an animal that is herbivorous and does not live-in forest then it is a rabbit.

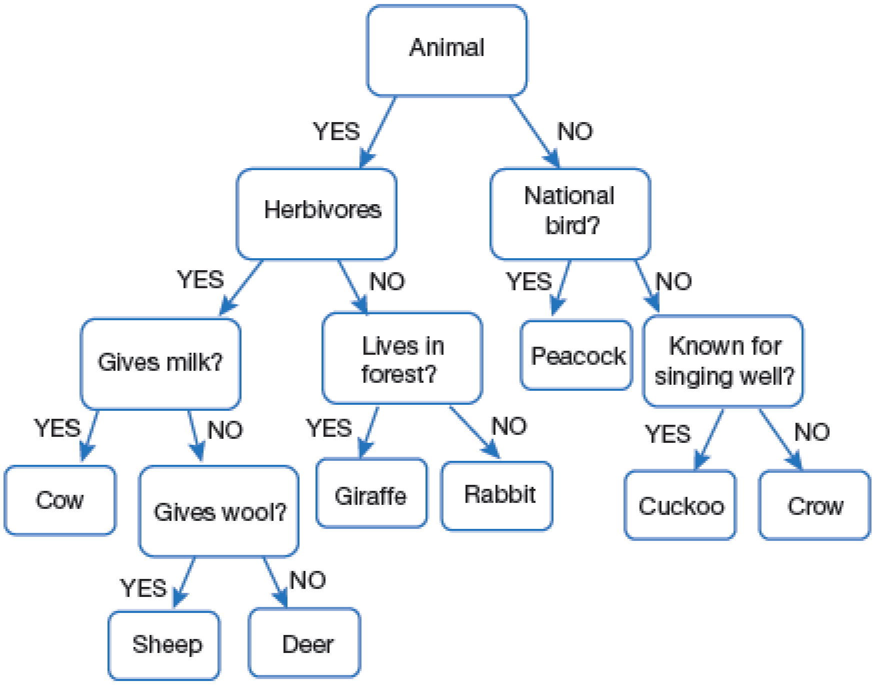


FIGURE 2.17 Decision Tree

The same concept is used to create a decision tree for any data science problem. Now, let us study in detail the math behind it.

2.4.2.1 Basic Introduction

Decision tree is one of the most intuitive and popular techniques in data mining that provides explicit rules for classification. It works well with heterogeneous data and predicts the target value of an item by mapping observations about the item.

A decision tree represents choices and their results in form of a tree. The nodes represent an event or choice and the edges of

the graph represents the decision rules or conditions. Some common applications of decision tree include the following:

1. Predicting an email as spam or not spam
2. Predicting of a tumour is cancerous or not

3. Predicting a loan as a good or bad
4. Predicting credit risk based on certain factors

Basically, decision trees can perform either classification or regression. For example, identifying a credit card transaction as genuine or fraudulent is an example of classification but using a decision tree to forecast prices of stock would be regression task. Unlike linear models, decision trees map non-linear relationships quite well.

2.4.2.2 Principle of Decision Trees

Decision trees can be used to divide a set of items into n predetermined classes based on a specified criterion. They belong to a class of **recursive partitioning algorithms** that are simple to describe and implement. To create a decision tree, the following steps are necessary.

Step 1: Select a variable which best separate the classes. Set this variable as the root node.

Step 2: Divide each individual variable into the given classes thereby generating new nodes. Note that decision trees are based on forwarding selection mechanism; so, after a split is created, it cannot be re-visited.

Step 3: Again, select a variable which best separates the classes.

Step 4: Repeat steps 2 and 3 for each node generated until further separation of individuals is not possible.

Now, each terminal (or leaf) node consists of individuals of a single class. That is, once the tree is constructed and division criterion is specified, each individual can be assigned to exactly one leaf. This is determined by the values of the independent variables for the individual.

Moreover, a leaf is assigned to a class if the cost of assigning that leaf to any other class is higher than assigning the leaf to the current class. After all the leaf nodes are assigned a class, we calculate the error rate of the tree (also known as the total cost of tree or risk of the tree) by starting with an error rate of each leaf. Thus, we see that a decision tree is a supervised learning predictive model that uses rules to calculate a target value. This target value can be either:

- a continuous variable, for regression trees. Such trees are also known as continuous variable decision trees. For example, predict the price of a product or income of a customer (based on his occupation, experience, etc.).
- a categorical variable, for classification trees. Such trees are also known as categorical variable decision tree. Example, to if the target value is 'Yes' or 'No'- whether a customer will buy a product or not.

Programming Tip: Decision tree works for both categorical and continuous input and output variables.

Pruning the Tree

When the decision tree becomes very deep, it must be pruned as it may contain some irrelevant nodes in the leaves.
Therefore, pruning avoids creating very small nodes with no real statistical significance.

An algorithm based on decision tree is said to be good if it creates the largest tree and automatically prunes it after detecting the optimal pruning threshold.

The algorithm should also use Cross-validation technique and combine error rates found for all possible sub-trees to choose the best sub-tree. A tree with each node having no more than two child nodes is called binary tree.

For example, if information gain of a node is –10 (loss of 10) and then the next split on that gives us IG of 20, then a simple decision tree will stop at step 1 but in pruning, the overall gain would be taken as +10 and both the leaves will be retained.

2.4.2.3 Key Terminology

Some important terms that are frequently used in decision trees (and demonstrated in Fig. 2.18) are as follows:

1. **Root node:** This is the node that performs the first split.
2. **Terminal nodes/Leaves:** These nodes predict the outcome.
3. **Branches:** They are depicted by arrows that connect nodes and shows the flow from question to answer. Technically, a branch is a sub section of entire tree
4. ***Splitting:* This is the** process of dividing a node into two or more sub-nodes. In a decision tree, splitting done until a user-defined stopping-criteria is reached. For example, the programmer may specify that the algorithm should stop once the number of items per node becomes less than 30.

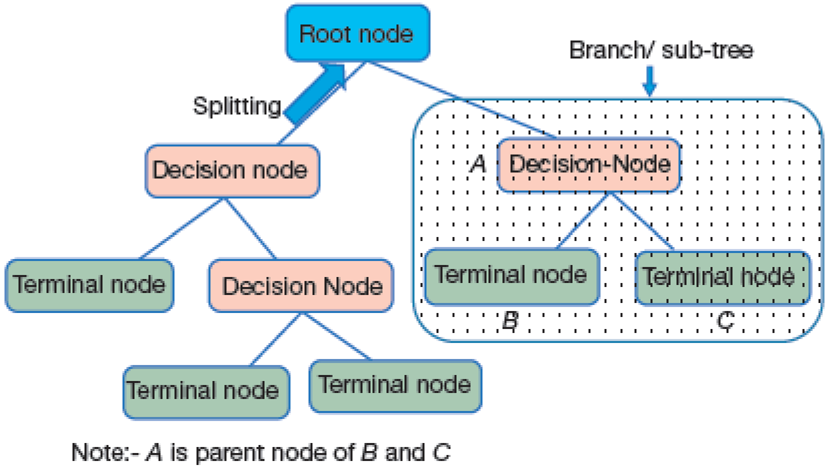


FIGURE 2.18

5. ***Decision node:*** This is a sub-node that splits into further sub- nodes
6. ***Terminal or Leaf node:*** It is a sub-node that does not split further.
7. ***Parent node:*** A node which splits into sub-nodes is called a parent node of the sub-nodes (or child of the parent node).

2.4.2.4 Advantages of Decision Trees

Decision trees are one of the most popular statistical techniques because it provides following advantages to users.

1. It is easy to implement
2. Results can be easily understood. It does not require any statistical knowledge to read and interpret them.
3. Programmers can code the resulting model
4. It executes faster when the model is applied to new individuals.
5. Outliers or extreme individuals can not affect decision trees.
6. Some decision tree algorithms can deal with missing data.
7. A decision tree allows users to visualize the results and represents all factors that are important in decision making.
8. Decision trees provide a clear picture of the underlying structure in data and relationships between variables.
9. They can also be used to identify the most significant variables in the data-set
10. Decision trees closely mirror human decision-making compared to other regression and classification approaches.
11. Decision trees requires less data cleaning as compared to other techniques.
12. They are not influenced by missing values to a fair degree.
13. They can handle both numerical as well as categorical variables.
14. Decision trees employs n**on parametric method as they** have no assumptions about the space distribution and the classifier structure.

OUTLIERS

An outlier is an individual data item or observation that lies at an abnormal distance from other data values in a random sample of a data set. It is the work of the data analyst to decide what will be considered abnormal. Outliers should be carefully investigated as they may contain valuable information about the data being processed. Data analyst must be able to answer questions like the reasons for the presence of outliers in the data set, probability that such values will continue to appear, etc. for example, if an examination was held of maximum marks 100, then one or more students getting more than 100 will be treated as an outlier.

Outlier analysis is extensively used for detecting fraudulent cases. In some cases, outliers may be contextual outliers. This means that a particular data value is an outlier only under a specific condition. For example, in hot summer season when temperature is 40 degrees and above, then an unexpected windstorm and rain may bring it to 32 degrees. Another example could be that a bank customer may not be withdrawing more than 50K in a month but because of a family wedding may withdraw 2.5 lakhs in a day.

2.4.2.5 Limitations of Decision Trees

The disadvantages of using decision trees are as follows:

1. The tree detects local, not global, optima.
2. All independent variables are evaluated sequentially, not simultaneously.
3. The modification of a single variable may change the whole tree if this variable is located near the top of the tree.
4. Decision trees lack robustness. Though this can be overcome by resampling, in which one can construct the trees on many successive samples and can aggregate by mean, but this will lead to a complex tree that is difficult to read and interpret.

   The problem of overfitting can be done by tree pruning and setting constraints on tree size.
5. Since each split reduces the number of remaining records, later splits are based on very few records have less statistical power.
6. The property of forwarding variable selection and constant splits of nodes makes prediction by trees not as accurate as done by other algorithms.
7. Overfitting is a key challenge in modelling decision trees. If there is no limit set of a decision tree, it will give 100% accuracy on training set because in the worst case, there will be a single leaf for each observation.

2.4.3 Random Forests

Like decision trees, random forests are also a versatile machine learning technique that can perform both regression and classification. They give better performance than decision trees as it does everything for reducing the number of dimensions (or variables), treating missing values, outlier values and exploring data.

Random forests perform better than bagged trees as it *de- correlates* the trees. Like in bagging, random forests also create a number of decision trees on training data sets. For creating these trees, each split considers a *random sample of m predictors* as split candidates from the full set of predictors. A split uses only one of those m predictors. Therefore, the main difference between bagging and random forest is that while bagging chooses all predictors, random forest selects only one of the m predictors.

Let us summarize the possibilities.

1. If the number of cases in the training set is K, then a random sample of these K cases is taken as the training set.
2. Out of p input variables, specify a number m, such that m < p. At each node, m random variables out of p predictors are chosen. The best split on these m variables is used to split the node.
3. Each tree is subsequently grown to the largest extent possible
4. Take the average of all the predictions made by the target trees to finally predict new data.
5. To classify a new object based on attributes, each tree gives a classification and finally the classification occurring most frequently is chosen.

In simpler terms, we can understand the working of a random forest in the following steps.

Step 1: Random samples are selected from a given dataset.

Step 2: A decision tree is constructed for each sample to obtain a prediction result from each decision tree as shown in Fig. 2.19.

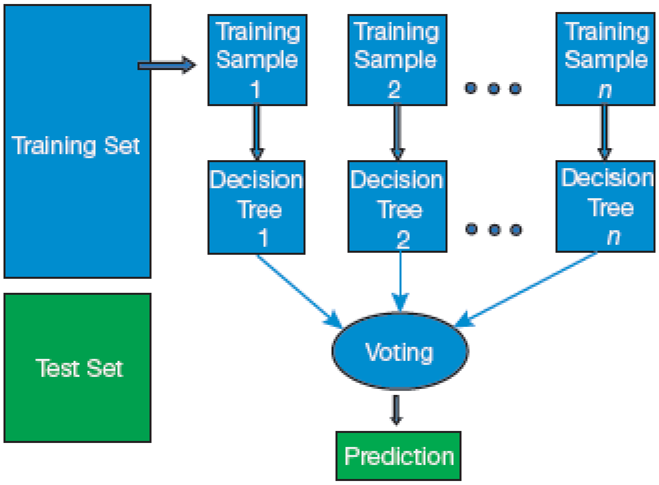


FIGURE 2.19 Random Forest

Step 3: Voting for each predicted result is performed.

Step 4: Prediction result with the maximum votes is selected as the final prediction.

Other major advantages of using Random forests include the following:

1. It can be very effectively used to estimate missing data
2. It maintains accuracy when a large proportion of the data is missing
3. It can balance errors in datasets where the classes are imbalanced
4. It can handle huge datasets with large number of dimensions

Limitations

1. Random forests might easily overfit noisy datasets, especially when performing regression.
2. Each tree is grown to the largest extent possible without pruning.
3. Random forests are slow in generating predictions as multiple decision trees are constructed and then the process of voting selects the best prediction result.
4. Random forests are difficult to interpret as and when compared to a decision tree model.

Review Questions:

1.What is Decision Tree?

2.What do you mean by Random Forests?

3.What is K-NN?

L5:

2.5 Clustering Techniques

Clustering is a set of techniques used to partition data into a number of groups, also called clusters. A c**luster** can be considered as a group of data objects that are similar to other objects in the same cluster and dis-similar to data objects in other clusters. Clustering helps data scientists to identify two main qualities of data—meaningfulness and usefulness.

Meaningful clusters enhance domain knowledge. For example, in the healthcare industry, clustering is used to make groups of patients whose bodies responded differently to the same medical treatment.

Useful clusters serve as an intermediate step in a data pipeline. For example, businesses use this technique for making different clusters of customers depending on their purchasing patterns to create targeted advertising campaigns.

2.5.1 Overview of Clustering Techniques

Data objects can be clustered using a variety of techniques. Each of these techniques has its own strengths and weaknesses. Depending upon the input data, we need to select the most appropriate technique that result in more natural cluster assignment. To select the most suitable clustering algorithm for the dataset is not a trivial task. Some important factors that affect this decision include the characteristics of the clusters, the features of the dataset, the number of outliers, and the number of data objects. Based on these features, three popular categories of clustering algorithms include:

1. Partitional clustering
2. Hierarchical clustering
3. Density-based clustering

Partitional Clustering

It divides data objects into non-overlapping groups in such a way that every object belongs to one and only one cluster, and every cluster must have at least one object.In such algorithms, users can specify the number of clusters (indicated by the variable *k*). These algorithms iteratively assign subsets of data points into *k* clusters. Two examples of partitional clustering algorithms are *k*-means and *k*-medoids.
Note that, these algorithms are **nondeterministic.** That is, such algorithms produce different results from two separate runs even if the same data was used in every run.

Advantages

1. Works well when clusters have a **spherical shape**.
2. Sc**alable** with respect to algorithm complexity.

Limitations

1. Does not perform well with **complex cluster shapes** having different sizes.
2. Does not work with clusters of different **densities**.

Hierarchical Clustering

It creates clusters in a hierarchy either by using a bottom-up or a top-down approach.

In agglomerative clustering, bottom-up approach is used to merge two points that are most similar. This process is repeated until all points have been merged into a single cluster.

In divisive clustering, top-down approach is used in which the algorithm starts with all points as one cluster. In each iteration, the data objects that are least similar in the cluster are split until only single data points remain.

In such an algorithm, a tree-based hierarchy of points called a **dendrogram** is created. The number of clusters to be created, that is, k is determined by the user. Clusters are created by cutting the dendrogram at a specified depth that results in *k* groups of smaller dendrograms.

Hierarchical clustering is a **deterministic** process. Therefore, cluster assignments do not change when the algorithm is executed two or more times using the same input data.

Advantages

1. They highlight the **relationships** between data objects at a finer level.
2. Results are easily **interpretable.**

Limitations

1. They're **computationally expensive** with respect to algorithm complexity.
2. They are adversely affected by **noise** and **outliers** in the data set.

Density-Based Clustering

This technique creates clusters based on the density of data points in a region. A cluster is created where there are high densities of data points separated by low-density regions.

In contrast to other clustering categories, density-based approach does not require the user to specify the number of clusters. It uses a distance-based parameter that acts as a threshold. The threshold value helps the algorithm to determine how close points must be to be considered a cluster member.

Density-Based Spatial Clustering of Applications with Noise (or DBSCAN) and Ordering Points To Identify the Clustering Structure (or **OPTICS**) are popular examples of density-based clustering algorithms.

Advantages

1. Works well when clusters are of **non-spherical shapes**.
2. Performs well even on a data set having noise or outliers.

Limitations

1. Not well suited for clustering in **high-dimensional spaces**.
2. Difficult to identify clusters of **varying densities**.

2.5.2 K-Means Algorithm

We have seen that the clustering technique is used for finding sub-groups of observations within a data set. Clusters are created in such a way that observations in the same group are similar. Correspondingly, observations in different groups are dissimilar. K-means is an unsupervised learning technique as it finds relationships between n observations without being trained by a response variable. K-means clustering is the simplest and the most commonly used clustering method for splitting a dataset into a set of k groups.

2.5.2.1 Properties of Clusters

To understand the concept of clusters, let us take an example of a banking system that forms segments of its customers based on their income and debt. To start with, a scatter plot is used to visualize the customer data.

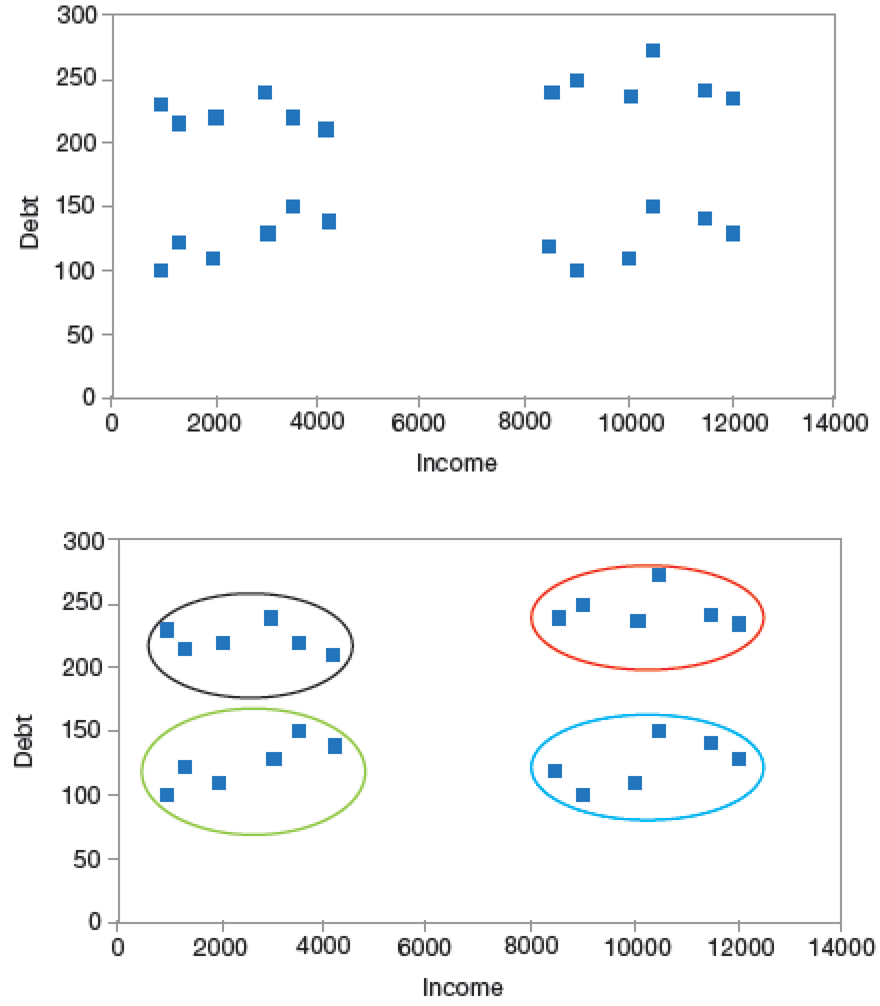


FIGURE 2.20 Cluster of Customers

Income is plotted on the X-axis and the amount of debt is plotted on the Y-axis. A very simple approach is to make four clusters as shown in Fig. 2.20. Looking at Fig. 2.20, we can conclude the following properties of a cluster.

All the data points in a cluster should be similar to each other. In the bank example, such a cluster allows the bank to perform targeted marketing. **The data points from different clusters should be as different as possible.**

2.5.3 Applications of Clustering in Real-World Scenarios

Clustering is a widely used technique across industry in almost every domain, ranging from banking to recommendation engines, document clustering to image segmentation.

Customer Segmentation

It is to target a specific set of customers or audience in the industry (telecom, e-commerce, sports, advertising, sales, etc.).

Document Clustering

Another very common application of clustering is document clustering in which multiple similar documents are clustered in a similar way as shown in Fig. 2.21.

FIGURE 2.21 Document clustering

Image Segmentation

In image segmentation, similar pixels in the image are clubbed together in the same group (refer Fig. 2.22).

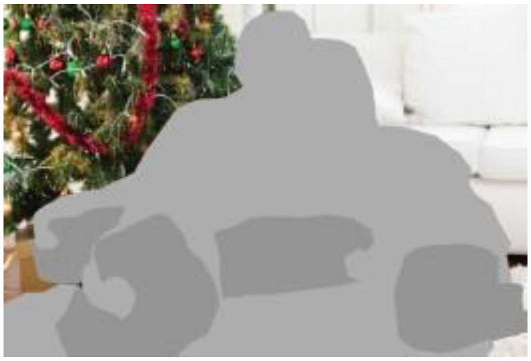


FIGURE 2.22 Image Segmentation

Recommendation Engines

Clustering when used in recommendation engines can help us to recommend songs / friends/ products to our friends.The algorithm makes a list of things liked by a person and then finds similar things to recommend them to that person.

2.5.4 Evaluation Metrics for Clustering

The goal of any clustering algorithm is not just to make clusters, but to make good and meaningful ones. For example, consider the two cases given in Fig. 2.23.

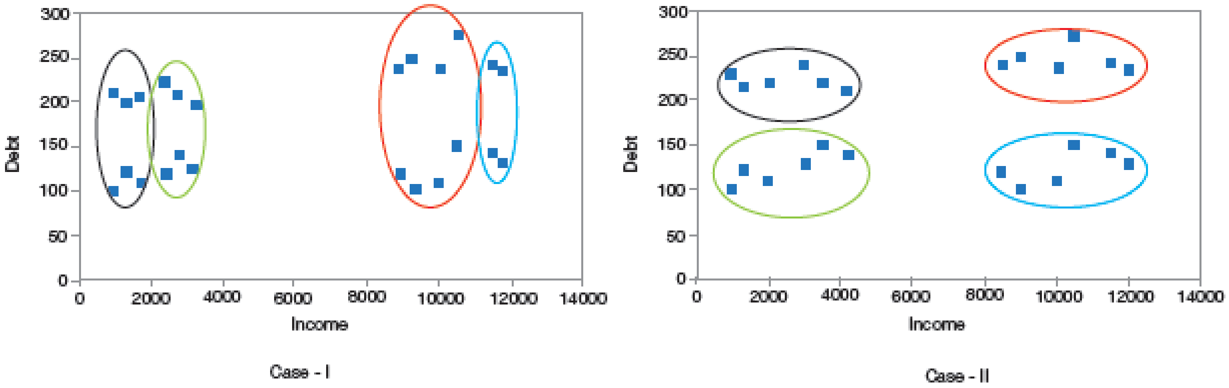


FIGURE 2.23 Different ways of clustering

Since here we have just two features, but in real-world applications, where we have too many features, visualizing all the features together and deciding which clusters are meaningful is just not possible. Therefore, in such a situation, we use evaluation metrics to evaluate the quality of the clusters.

Inertia

Inertia indicates how far the points within a cluster are. For this**, sum of distances of all the points within a cluster from the centroid of that cluster is calculated. Once inertia is calculated for all the clusters, the final value of inertia is obtained by adding all these values. This final value that gives the** distance within the clusters is known as **intra-cluster distance.** So, inertia gives us the sum of intra-cluster distances. For best results (or clusters), the inertial value **should be as low as possible.**

Dunn Index

If the distance between the centroid of a cluster and the points in that cluster is small, it means that the points are closer to each other. So, the inertial value ensures that the first property of clusters is satisfied. However, it does not say anything about the second property – that different clusters should be as different from each other as possible. So, for this, we use the Dunn index.

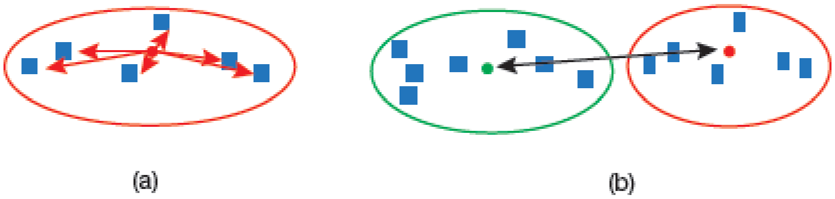


FIGURE 2.24 (a) Intra cluster distance; (b) Inter cluster distance

Along with the distance between the centroid and points, **the Dunn index also considers the distance between two clusters. The d**istance between the centroids of two different clusters is known as **inter-cluster distance**. The expression for Dunn Index can be given as,

$$\text{Dunn index} = \frac{\min(\text{inter cluster distance})}{\max(\text{intra cluster distance})}$$

That is, Dunn index is the ratio of the minimum inter-cluster distances and maximum of intra-cluster distances. Its value must be high for getting better clusters.

We must maximize the value of Dunn index. More the value of the Dunn index, better will be the clusters. For a higher Dunn index value, the numerator should be maximum. So, the distance between even the closest clusters should be more so that every cluster is far away from each other. Correspondingly, denominator should be minimum. So, intra-cluster distances should be more.

2.5.5 How K-Means Algorithm Works?

When implementing the k-means clustering algorithm, the following steps are performed.

First specify the number of clusters (k) that will be generated in the final solution. The algorithm starts by randomly selecting k observations (also known as centroids) to serve as the initial centres for the clusters.

In the next step, every remaining observation is assigned to its closest cluster, where closest is a term calculated using the Euclidean distance between the centroid of the cluster and the observation. This step is called 'cluster assignment step'.

The algorithm then computes the new mean value of each cluster. Therefore, this step is known as 'centroid update'. Once the centers are re-calculated, every observation is rechecked to determine its closeness with the same and other clusters. All the observations may have to be reassigned a cluster based on the updated cluster means.

The cluster assignment and centroid update steps are repeatedly performed until the cluster assignments remains the same over iterations (technically, until *convergence* is achieved).

2.5.6 Pros and Cons of K-Means Algorithm Pros

1. K-means clustering algorithm is very simple and fast.
2. The algorithm can efficiently deal with very large data sets.

Cons

1. Number of clusters must be pre-specified.
2. The algorithm is sensitive to outliers
3. Changing the order of data will give different results.
4. Create a random data set and then apply the k-means clustering algorithm.

2.5.7 Density-Based Spatial Clustering of Applications with Noise (Dbscan)

We have seen that clustering is an unsupervised learning technique that forms groups of the data points. All points in the same group have similar properties and those in different groups exhibit a different set of properties.

Clusters or groups in data are formed using a variety of distance measures like K-Means (distance between points), affinity propagation (graph distance), mean-shift (distance between points), DBSCAN (distance between nearest points), Gaussian mixtures (Mahalanobis distance to centres), spectral clustering (graph distance), etc.

Though all clustering analysis techniques use the same approach (calculating similarities and then clustering data points into groups), here in this section, we will explore **Density-based spatial clustering of applications with noise** (**DBSCAN**) technique.

Why DBSCAN? K-Means clustering, clusters loosely related observations together. Every observation or data point is eventually a part of some cluster even if the observations are scattered far away in the vector space.

The cluster to which an observation will be associated with depends on the mean value of cluster elements. Even a small change in data points *may alter* the clusters formed. However, this problem is greatly reduced in DBSCAN.

Another issue with *k*-means algorithm is that you need to specify the number of clusters ('*k*') to be formed and we often err to specify the most optimal value for k. In DBSCAN, the value of k need not be specified. We just need to have an idea about the distance between values and some guidance for what amount of distance is considered 'close'. DBSCAN produce better results than *k*-means. This is evident from Fig. 2.25.

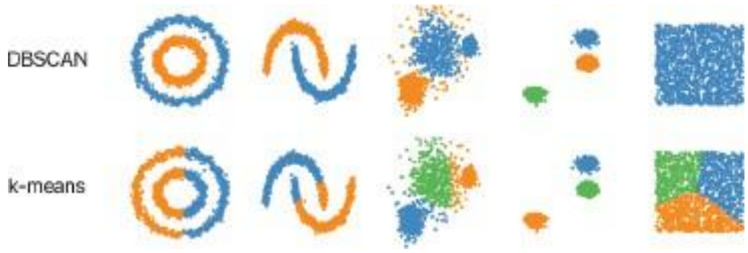


FIGURE 2.25 DBSCAN and k-means Clustering Algorithms

2.5.7.1 Understanding Density-Based Clustering Algorithms

DBSCAN is an unsupervised learning technique that identifies distinct clusters in the data. A cluster is identified as a contiguous region of high point density, separated from other such clusters by contiguous regions of low point density.

The algorithm discovers clusters of different shapes and sizes from a large amount of data that may contain several outlier values.

Two important parameters of the DBSCAN algorithm are discussed here:

minPts which indicates the minimum number of points (a threshold) clustered together for a region to be considered dense.

As a rule of thumb, value of *minPts* can be derived from the number of dimensions *D* in the data set. Generally, $minPts \geq D$ **+**
1. Setting *minPts* **= 1** is of no use as then every point will form a cluster of its own. With *minPts* **≤ 2**, the result will be similar to hierarchical clustering. Therefore, minimum value of *minPts* should be at least 3. However, for noisy data sets we must set it to a larger value to get more significant clusters. To be more specific, we can choose *minPts* **= 2·***dim* to choose larger values for dataset that is very large, or is noisy or contains duplicate values.

eps (ε) is the distance measure that will be used to locate the points in the neighbourhood of any point.

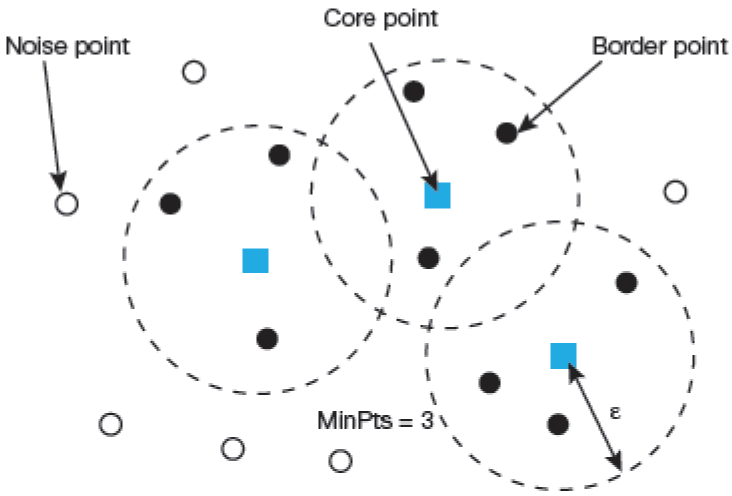


FIGURE 2.26 Terminology used in DBSCAN

The value for ε can be chosen by using a k-distance graph, in which the distance is plotted to the *k* nearest neighbour ordered from the largest to the smallest value. Here, the value of k can be set to *minPts*-**1.** Optimal values of ε can be noted from this plot from the location where an 'elbow' is present. If ε is too small, a large part of the data will not be clustered; whereas for a large value of ε, clusters will merge and most of the points will be in the same cluster. However, small values of ε should be preferred to have only a small fraction of points within this distance of each other.

These parameters can be understood by exploring the concepts of density reachability and density connectivity.

Density reachability establishes a point to be reachable from another if it lies within a particular distance (eps) from it.

Connectivity is a transitivity-based chaining-approach that is used to determine points that are located in a particular cluster. For example, a and e points could be connected if a->b->c->d->e, where a->b means b is in the neighbourhood of a.

Once DBSCAN clustering is done, we can recognize three types of points.

Core, it is a point having at least *m* points within distance *n*
from itself.

Border, is a point having at least one Core point at a distance *n*.

Noise, is a point that does not fall in the above two categories. This means that noise has less than *m* points within distance *n* from itself (refer Fig. 2.26).

2.5.7.2 Algorithmic Steps for Dbscan Clustering

Repeat until all points have been visited.

1. Arbitrarily pick up a point in the dataset ().
2. If there are at least 'minPoint' points within a radius of 'ε' to the point then all these points are considered to be a part of the same cluster.
3. Expand the clusters by recursively computing the neighbourhood for each point.

Review Questions:

1.What is Dendogram?

2.What is clustering?

3.List different clustering Techniques?

L6:

2.6 Naïve Bayes Classification

Data science has progressed from simple linear regression models to complex techniques but practitioners still prefer the models that are simple and easy to interpret. In this widely used category of algorithms Naïve Bayes algorithm is one of the prominent names as it is not only simple but so powerful that it outperforms complex algorithms for very large datasets.

Naive Bayes is a probabilistic machine learning algorithm based on the Bayes Theorem.

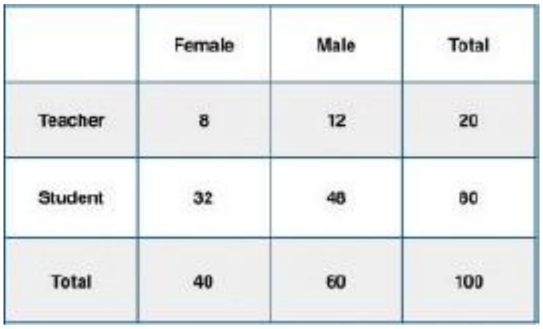
This algorithm is extensively used in a wide range of classification tasks varying from filtering spam, classifying documents to predicting sentiments.

Programming Tip: The tern naïve means that features that are used in the model are independent of each other. So when one feature change, the other is not affected.

2.6.1 Understanding Conditional Probability

To understand conditional probability, consider an example of flipping a coin. There are equal chances of getting either heads or tails. So, the probability of getting heads is 50%.

Now, in case we are supposed to find the probability of getting a queen spade from a deck of 52 cards then the denominator is 13 (the eligible population) and not 52. Since there is only one queen in 13 spades, the probability that we get a queen spade is 1/13.



	Female	Male	Total
Teacher	8	12	20
Student	32	48	80
Total	40	60	100

FIGURE 2.27 Sample Dataset

Therefore, we can say that the conditional probability of A given B, denotes the probability of occurring A given that B has

already occurred. Mathematically, it can be expressed as, P(A|B) = P(A AND B) / P(B).

Let's apply this concept on the data set given in Fig. 2.27 and calculate the conditional probability that the teacher is a male. So, the required conditional probability P(Teacher | Male) = 12 / 60 = 0.2.

Programming Tip: If Y has two categories, then calculate the probability of each class of Y and select the one with higher value.

$$P(\text{Teacher} \mid \text{Male}) = \frac{P(\text{Teacher} \cap \text{Male})}{P(\text{Male})} = \frac{12}{60} = 0.2$$

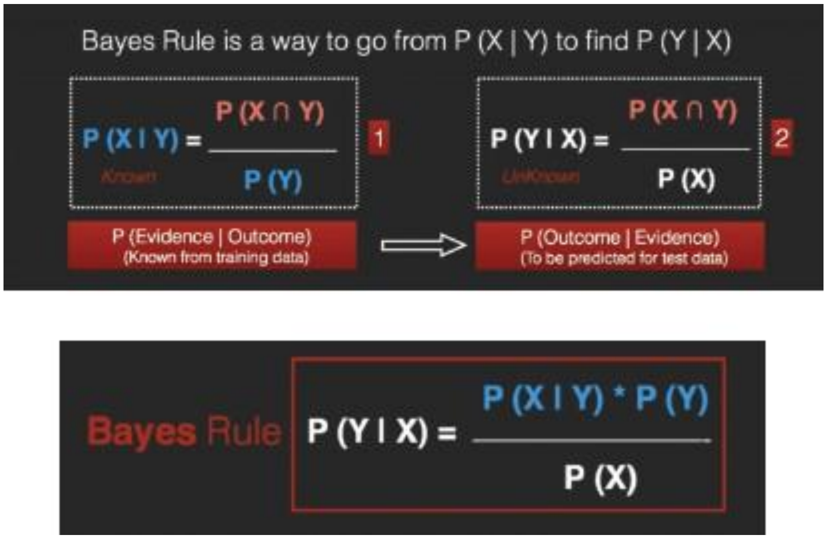
Likewise, the conditional probability of B given A can be computed as,

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} \qquad (1)$$

$$P(B \mid A) = \frac{P(A \cap B)}{P(A)} \qquad (2)$$

2.6.2 The Bayes Rule

The Bayes Rule uses P(X|Y), known from the training dataset, to find P(Y|X). For observations in test dataset, X would be known while Y is unknown. And for each row of the test dataset, we have to find the probability of Y given that X has already happened (refer Fig. 2.28).



Bayes Rule is a way to go from P (X | Y) to find P (Y | X)

$$P(X \mid Y) = \frac{P(X \cap Y)}{P(Y)} \quad 1$$

Known

P (Evidence | Outcome)
(Known from training data)

$$P(Y \mid X) = \frac{P(X \cap Y)}{P(X)} \quad 2$$

UnKnown

P (Outcome | Evidence)
(To be predicted for test data)

Bayes Rule

$$P(Y \mid X) = \frac{P(X \mid Y) * P(Y)}{P(X)}$$

FIGURE 2.28 Bayes Rule for Conditional Probability

2.6.3 Types of Events

If two probabilities, P(B) and P(B|A) are the same, then it means that the occurrence of event A had no effect on event B. Therefore, event A and event B are *independent events*. If the conditional probability is zero, then it means that the occurrence of event A implies that event B cannot occur. If the reverse is also true, then the events are said to be mutually exclusive events. In such a case, only one event can take place at a time. All other cases are classified as dependent events where the conditional probability can be either lower or higher than the original.

For example, if we toss a coin twice and we want to calculate the probability of getting a head both times, then we must consider that the second event is independent of the first. The desired probability is P(Heads in first throw)*P(Heads in second

throw)= ½ X ½ = ¼

Mathematically, we can say that if the events are not independent, we multiply the probability of one event with the probability of the second event.

For expressing dependent events mathematically, we multiply the probability of any one event with the probability of the second event after the first has happened.

P(A and B)=P(A)*P(B|A) or P(A and B)=P(B)*P(A|B)

For example, while drawing two cards (King and Queen) without replacement, the probability of the first event is dependent of 52 cards whereas the probability of the second event is dependent on 51 cards.

This means that **P(King and Queen)=P(Queen)* P(King|Queen)**

So if P(Queen) is 4/52 then once the King is drawn, P(King|Queen) is 4/51 as now only 51 cards are left. Therefore, the desired probability = 4/52 X 4/51

The probability of two non-mutually exclusive events is given as, **P(A OR B)=P(A) + P(B) - P(A AND B)**.

For example, in a throw of dice, the probability of getting a number that is multiple of 2 or 3 is a scenario of events which are not mutually exclusive since 6 is both a multiple of 2 and 3 and is counted twice. Therefore, **P(multiple of 2 or 3) = P(Multiple of 2) + P(Multiple of 3) - P(Multiple of 2 AND 3) = P(2, 4, 6) + P(3, 6) - P(6) = 3/6 + 2/6 -1/6 = 4/6 = 2/3**

2.6.4 Naive Bayes Algorithm

In real-world problems, there are multiple X variables. So, when the features are independent, the Bayes Rule can be extended to what is called Naive Bayes. It is called 'Naive' because of the naive assumption that the Xs are independent of each other.

Naive Bayes classification algorithm can be used with continuous features but performs best for categorical variables. When dealing with numeric features, the algorithm assumes that numerical values are normally distributed.

2.6.5 Laplace Correction

We can better understand the importance of Laplace correction with an example. If we were to classify fruits as mango, orange or banana, then the value of P(Long | Orange) will be zero as there are no 'Long' oranges.

When we use this probability in the model having multiple features, then the entire probability will become zero as anything multiplied by zero is zero. To avoid this, we need to increase the count of the variable with zero to a small value (usually 1) in the numerator, so that the overall probability does not become zero.

This correction is called 'Laplace Correction and the function for building the Naive Bayes model will accept this correction as a parameter.

2.6.6 Pros and Cons of Naive Bayes Algorithm Pros

1. It is a simple algorithm that is easy to build and understand.
2. It predicts classes faster than many other classification algorithms.
3. It can be easily trained using a small data set.
4. It can be used to predict values for even large data sets.

Cons

- If a given class and a feature has 0 frequency, then the conditional probability will become 0 resulting in the 'Zero Conditional Probability Problem.' This is a serious issue as it wipes out all other vital information about probabilities. However, to avoid such situations there are several sample correction techniques like the 'Laplacian Correction.'
- Another disadvantage is the very strong assumption of independence class features that it makes. It is near to impossible to find such data sets in real life.

2.6.7 Applications

Some of the real-world scenarios where Naive Bayes algorithm is used include the following:

1. **Text classification**: Naïve Bayes classification algorithm is one of the most preferred algorithms to classify whether a text document belongs to one or more categories (classes).
2. **Spam filtration**: Many popular email services uses the Bayesian spam filtering to distinguish spam email from legitimate email. Many server-side email filters like DSPAM, SpamBayes, SpamAssassin, Bogofilter, and ASSP are all based on Naïve Bayes classification technique.
    1. **Sentiment Analysis**: The Naïve Bayes algorithm is also used to analyse the analyse tweets, comments, and reviews posted on social networking site to classify as being negative, positive or neutral.
    2. **Recommendation System**: Companies uses recommendation systems to predict whether a particular user will like their product and/ or buy their product or not. For this, the Naive Bayes algorithm is used in conjunction with collaborative filtering techniques to build hybrid recommendation systems.

Review Questions:

1.What is conditional probability?

2.What is Naïve Bayes algorithm?

L7:

2.7 Neural Network

Neural Network (NN) or Artificial Neural Network (ANN) is a machine learning algorithm that is inspired by the biological neuron system and learns by examples. It consists of a large number of highly interconnected processing elements called neurons to solve problems. The algorithm follows a non-linear path and information is processed in parallel throughout the nodes. The best part of neural network is that it can change its internal structure by adjusting weights of inputs.

Neural network algorithms were developed to solve problems which are easy for humans but difficult for machines. These problems include pattern recognition in which we need to identify similar pictures or patterns. Because of its ability to do pattern recognition, neural networks application ranges from optical character recognition to object detection.

2.7.1 Working of Neural Networks

Do you in human's brain, dendrites of a neuron receive input signals from another neuron and gives output based on those inputs to an axon of some other neuron. The human brain comprises billions of interconnected web of neurons that process information makes the brain to think and take decisions.

Similarly, the neural network algorithm works using a set of connected input/output units. In this structure, each connection has a weight associated with it. In the learning phase, the network learns by adjusting the weights to predict the correct class label of the given inputs.

The learning phase is used along with back propagation error method. Therefore, when error is calculated at the output unit, it is back propagated to all the units because error at each unit contributes to total error at the output unit. The errors at each unit are then used to optimize the weight at each connection.

The output of a neuron may vary from −inf to +inf. So, we need a mapping function also known as the Activation function that maps inputs to the outputs.

A Simple Neural Network Model

In Fig. 2.29, we can see that it is a feedback neural network in which information is passed in both directions (forward as well as backwards). The structure of the neural network can change over time based on the inputs. Though the figure shows that there is one input layer, one hidden layer and one output layer but we can have multiple layers of hidden layers as shown in Fig. 2.30.

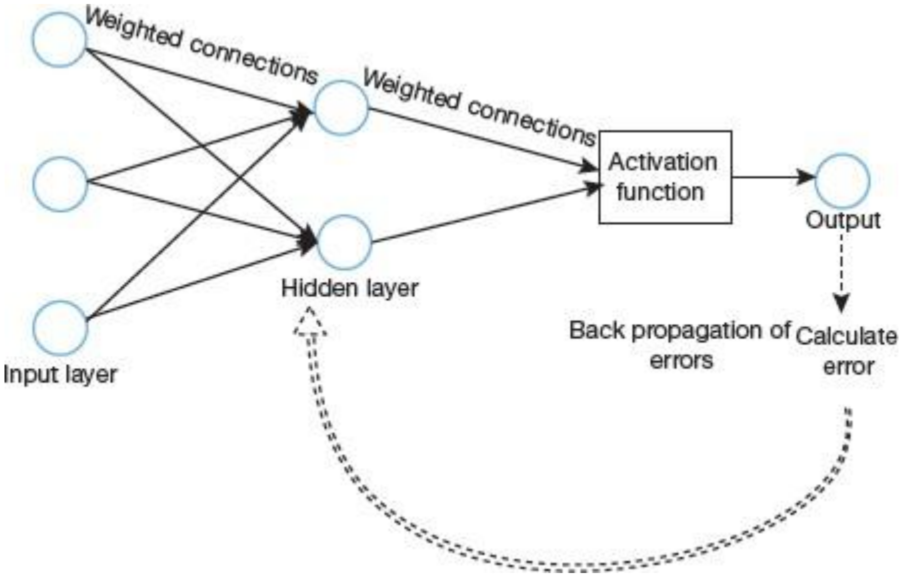


FIGURE 2.29 Feedback neural network

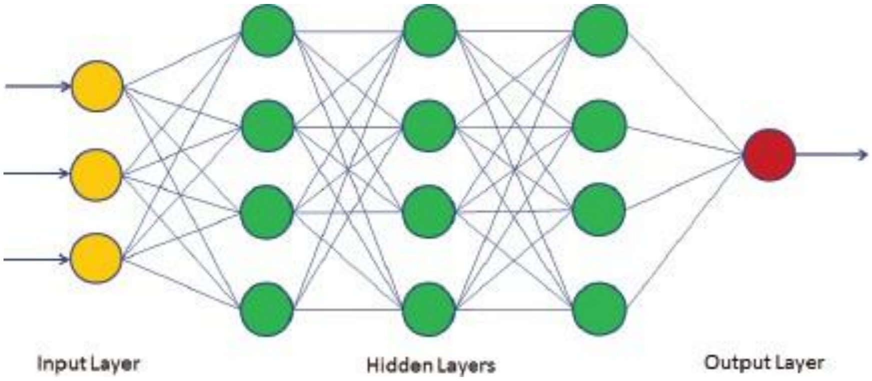


FIGURE 2.30 Neural Network with multiple hidden layers

Here, the input layer is the first layer of the neural network and receives the raw input. It processes the input and passes the processed information to the hidden layers. The hidden layer passes the information to the last layer, which gives the final output.

The output of a neuron can range from −inf to +inf. The neuron doesn't know the boundary. So, we need a mapping mechanism between the input and output of the neuron. This mechanism of mapping inputs to output is known as Activation Function.

2.7.2 Pros and Cons Pros

1. Neural network is a flexible algorithm that can be used to solve both regression and classification problems.
2. Neural networks perform well on nonlinear dataset with a large number of inputs such as images.
3. Neural networks can work with any number of inputs and layers.
4. Neural network works very fast as compared to other classification algorithms since it performs calculations in parallel.

Cons

1. Algorithms like Decision Tree and Regression that are simple, fast, easy to train, and provide better performance are seen as a preferred choice for classification and regression problems because
2. Neural networks require more time for development and needs more computation power.
3. Neural Networks needs more data than any other Machine Learning algorithm.
4. Neural networks can be performed only on numerical inputs and non-missing value datasets.

2.7.3 Applications of Neural Networks

Neural network algorithms are extensively used in the following fields.

1. **Pattern recognition** applications like facial recognition, object detection, fingerprint recognition, etc. uses neural network algorithms.
2. **Anomaly detection** applications that are specifically designed to detect unusual patterns that don't fit in the general patterns use neural networks because of their ability to perform well in pattern recognition tasks.
3. **Time series prediction** applications like predicting stock prices, forecasting weather prefer to use neural networks.
4. **Natural language processing** applications including text classification, Named Entity Recognition (NER), Part-of- Speech Tagging, Speech Recognition, and Spell Checking

   which are used in a wide range of applications makes use of neural networks algorithms.

2.7.4 How Neural Networks Work?

We have seen that neural networks take several inputs, processes it through multiple neurons from multiple hidden layers, and returns the final result using an output layer. This process is known as '**forward propagation**'.

To make a perfect model, we need to minimize the value or weight of neurons that are contributing more to the error. For this, we need to travel back to the neurons of the neural network and find where the error lies. This process is known as '**backward propagation**'. We can reduce the number of iterations to minimize the error by using the 'gradient descent' algorithm.

A simple strategy of creating input-output relationships is discussed below.

1. **Use a subset of inputs to calculate the output.** Take only those inputs that

satisfy a given threshold value. For example, if the threshold value is 0, then if x1 + x2 + x3 > 0, the output is 1 and 0 otherwise.

2. **Add weights to the inputs.** Weights give importance to an input. For example, if w1=4, w2=5, and w3=6 then on assigning these weights to x1, x2 and x3, we will multiply input with their weights and compare the result with threshold value. That is, w1*x1 + w2*x2 + w3*x3 > threshold.

   As per the details provided, more importance has been given to x3 in comparison to x1 and x2.

3. **Add bias:** Each perceptron has a bias that indicates how flexible the perceptron is. Therefore, linear representation of input will now be, w1*x1 + w2*x2 + w3* x3 + 1*bias.

Neuron applies non-linear transformations (activation function) to the inputs and biases.

Now, back-propagation (BP) algorithms determine the loss (or error) at the output and then propagate it back into the network. The weights are updated to minimize the error resulting from each neuron. In this process, the first step is to determine the gradient (derivatives) of each node with respect to the final output.

Note that one round of forwarding and backpropagation iteration is known as one training iteration or **Epoch**.

2.7.5 What is an Activation Function?

Activation function takes the sum of weighted input (w1*x1 + w2*x2 + w3*x3 + 1*b) as an argument and returns the output of the neuron.

$$a = f\left(\sum_{i=0}^{n} w_i x_i\right)$$

The activation function is usually used to make a non-linear transformation that allow us to fit nonlinear hypotheses or to estimate the complex functions. Some commonly used activation functions include 'Sigmoid', 'Tanh', ReLu, etc.

2.7.6 Gradient Descent

There are two variants of gradient descent algorithm- Full Batch Gradient Descent and Stochastic Gradient Descent (SGD). Both these variants update the weights of the neurons. Although the underlying technique is the same, the only difference lies in the number of training samples used to update the weights and biases.

While Full Batch Gradient Descent Algorithm uses all the training data points to update each of the weights at once, the Stochastic Gradient uses 1 or more data points but never the entire training data for updating at once.

For example, if we have a dataset of 10 data points with two weights **w1** and **w2**, then

In a **Full Batch** algorithm, all 10 data points are used to calculate the change in w1 (Δw1) and change in w2 (Δw2) and update w1 and w2 accordingly.

However, with SGD algorithm, initially the first data point is used to calculate the change in w1 (Δw1) and change in w2(Δw2) and update the weights w1 and w2. Then, the second data point is used to update weights. An activation function is what makes a neural network capable of learning complex non-linear functions.

Neural Networks and Deep Learning

Majority of AI systems are supported by breakthroughs in machine learning and deep learning techniques. These techniques together go hand-in-hand to deliver an efficient system that sometimes it is just difficult to understand the difference between artificial intelligence, machine learning and deep learning. However, venture capitalist Frank Chen highlighted this difference by stating that 'A*rtificial intelligence is a set of algorithms and intelligence to try to mimic human intelligence. Machine learning is one of them, and deep learning is one of those machine learning techniques.'*

Deep learning is inspired from human brain and the neurons in the human brain. Therefore, to understand deep learning, we need to first recapitulate how neurons in our brans work. Have you wondered,

How a child distinguishes between a car and a bike? How do we perform

complex pattern recognition tasks?

How do we differentiate between a male voice and a female voice?

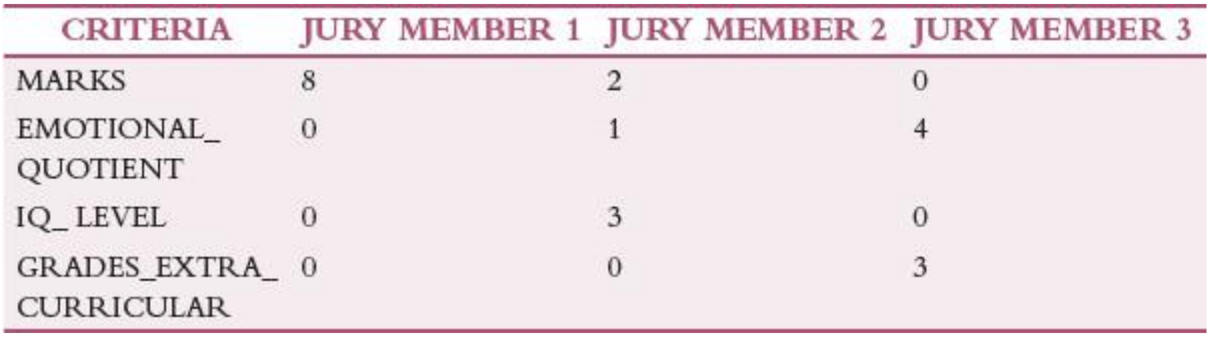
The answer is that our brains have a huge, connected and a complex network of about 10 billion neurons which are, in turn, connected to another 10,000 neurons. This network of neurons, or what we call as, a neural network, lays the foundation of deep learning. So, let us first get an introduction on neural networks.

Example: A school has to select a student for the prestigious title, 'STUDENT OF THE YEAR'. The principal forms a jury of three people for this task. The jury decides that the selection criteria would be- MARKS, EMOTIONAL_QUOTIENT, IQ_ LEVEL, GRADES_EXTRA_CURRICULAR.

The school has a history of fair selection procedure and to continue the same standard, the principal shares with the jury, data of about 10 students previously selected to give the jury an opportunity to practice, which will eventually help them make a fair selection.

Every jury member is given a maximum of 10 points (weight) on which they would rate a student. These 10 points can be distributed across the four criteria. The cut-off average to qualify the selection process is '6'. So, a student must have an average score of ≥ 6.

SCORE OF STUDENT 1



CRITERIA	JURY MEMBER 1	JURY MEMBER 2	JURY MEMBER 3
MARKS	8	2	0
EMOTIONAL_ QUOTIENT	0	1	4
IQ_ LEVEL	0	3	0
GRADES_EXTRA_ CURRICULAR	0	0	3

Jury Member 1: The first jury member gives the entire prominence to marks. Therefore, allots 8 points to Student 1 just on this criterion.

Jury Member 1: The second member distributes the points across all criteria except grades in extra-curricular activities.

Jury Member 3: The third jury member finds emotional quotient and grades in extracurricular activities more important and thus allocated points based on performance in these areas.

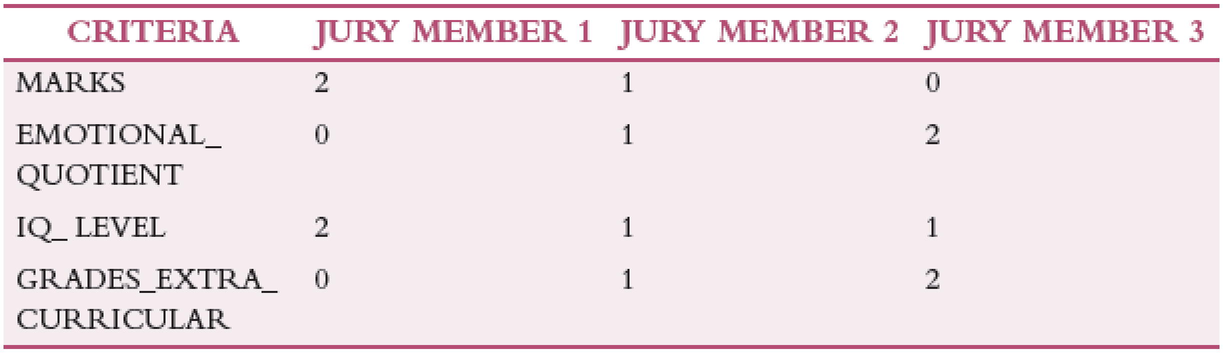
Finally, the average score of Student 1 is calculated as- (8 + 6 + 7) / 3 = 7.

Since this score is > 6, Student 1 qualifies round 1. But the principal revealed that this student actually did not get selected in round 1 as per the original decision.

Now, the jury members were asked to adjust their scoring process to see if they can correctly predict Student 2's selection or not.

JURY MEMBER 1: Member thinks that excess weightage has been assigned to Marks, so now considers IQ_Level also to assign points.

JURY MEMBER 2: Now distributes, points across all four criteria.
JURY MEMBER 3: Now, allots points to all criteria except MARKS.



CRITERIA	JURY MEMBER 1	JURY MEMBER 2	JURY MEMBER 3
MARKS	2	1	0
EMOTIONAL_ QUOTIENT	0	1	2
IQ_ LEVEL	2	1	1
GRADES_EXTRA_ CURRICULAR	0	1	2

The average sore of Student 2 is (4 + 4 + 5) / 3 = 4.3 Since this score is

less than 6, Student 2 is not selected.

However, this time principal reveals that decision of the jury was indeed correct. Student 2 was rejected in the original selection process.

In this way, the jury evaluates records of other candidates and in the process learns a pattern for assigning right 'weightage' for each criterion. Of course, the weightage that gives the highest number of correct predictions is kept for evaluating the upcoming contest.

This entire process of learning and developing an accuracy is nothing but Artificial Neural Networks (ANN).

Deep Learning

Deep learning (also known as deep neural learning or deep neural network) is an AI technique that imitates the working of human brain in processing data and creating patterns for use in decision making. It is a subset of machine learning that uses unsupervised learning to learn from unstructured or unlabelled data.

In today's scenario we generate massive amount of data (approx..2.6 quintillion bytes) every day. This huge data paves the way for application of deep learning techniques (refer Fig. 2.31). Deep learning algorithms flourish when applied on tonnes of data processed using stronger computing power.

When applied on very diverse, unstructured and inter- connected data, deep learning can be used to solve complex problems. The deeper learning algorithms learn, the better they perform.

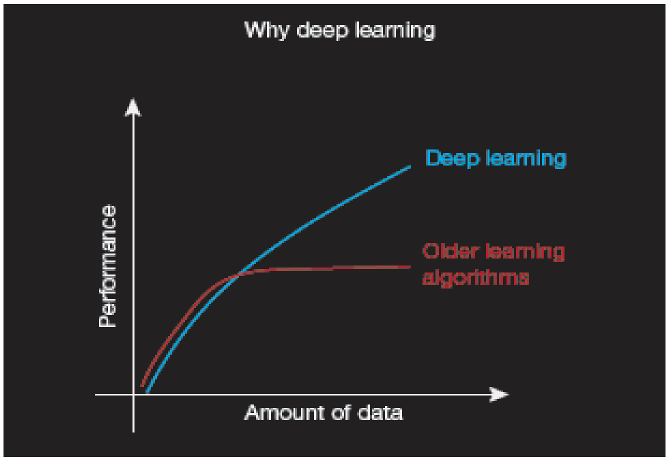


FIGURE 2.31 How do data science techniques scale with amount of data?

Deep learning is usually used to detect objects, recognize speech, translate languages, recognize similar images, recommend products, make important decisions, detect fraud or money laundering, and explore the possibility of reusing drugs for new ailments among other functions. It's use became unprecedented with massive explosion of data world-wide. The big data that we collect from social media, internet search engines, e-commerce platforms, online cinemas, etc are unstructured. Such data when analysed using deep learning techniques, can unravel wealth of information and can be used with AI systems for automated support. Figure 2.32 highlights the relation between AI and DL

Deep learning is the key technology behind driverless cars. Such cars are designed to recognize a stop sign, or distinguish a pedestrian from a lamppost. Basically, deep learning model learns to perform classification directly from images, text, or sound. For better accuracy, such models are trained by using a large set of labelled data and neural network architectures containing several layers.

Deep learning uses a hierarchical level of artificial neural networks to carry out the process of machine learning.

How Does Deep Learning Work?

We have studied that neural networks are layers of nodes. It is designed in the same way the human brain is made up of neurons. Nodes within individual layers are connected to adjacent layers. While in the human brain, a single neuron receives thousands of signals from other neurons, neural networks work in the same way. A message or a signal travels between nodes and weights are assigned to every node.

A node with a higher weight will have more weightage on the next layer of nodes. The final layer computes the weighted inputs to produce the final output.

A neural network is said to be a deep neural network depending on the number of layers it has. A deep learning system needs more powerful hardware as it has more layers and perform several complex mathematical calculations on a large amount of data. Large data sets ensure accurate results. For example, a facial recognition program starts with detecting and recognizing edges and lines of faces and then gradually learns to detect other significant parts of the faces. The process continues till the overall representations of faces are identified. Over time, the program trains itself with every iteration and thus the probability of correct answers increases. **Neural networks can be used to perform clustering, classification or regression.** The individual layers of neural networks can be thought of as a filter that increases the likelihood of giving the correct result as output.

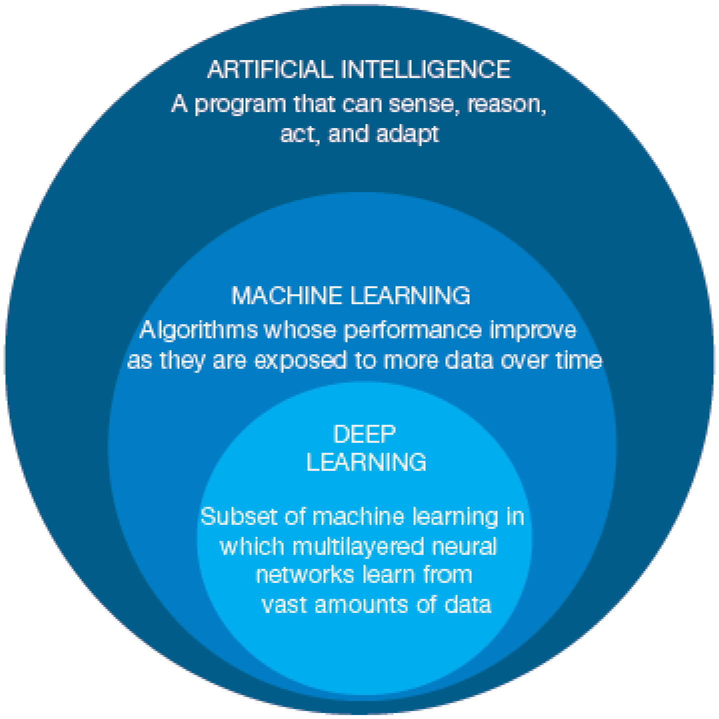

ARTIFICIAL INTELLIGENCE
A program that can sense, reason, act, and adapt
MACHINE LEARNING
Algorithms whose performance improve as they are exposed to more data over time
DEEP LEARNING
Subset of machine learning in which multilayered neural networks learn from vast amounts of data

FIGURE 2.32 DL is a subset of ML which in turn is just a sub- domain of AI

For example, when creating a neural network that identifies images having a dog, we need to train the network with pictures showing dogs at different angles and with varying amounts of light and shadow. These images in the training data set are converted into data, which moves through different nodes in the network. Prediction, identification or classification result is finally produced by the last layer, that is, the output layer of the neural network. The output produced by the neural network is then matched with the labels provided by any human. If the two values match, the output is confirmed otherwise the neural network notes the error and adjusts the weights provided to each node.

Some popularly used open-source deep learning libraries include Google Tensorflow, Facebook open-source modules for Torch, Amazon DSSTNE on GitHub, and Microsoft CNTK.

Machine Learning Vs Deep Learning

In a machine learning algorithm, relevant features are manually extracted from images and a particular classifier has to be manually implemented. The extracted features are then used to create a model that categorizes the objects in the image. In contrast, relevant features are automatically extracted from images by the deep learning model. Even the modelling steps are automatic (refer Figs 2.33 and 2.34).

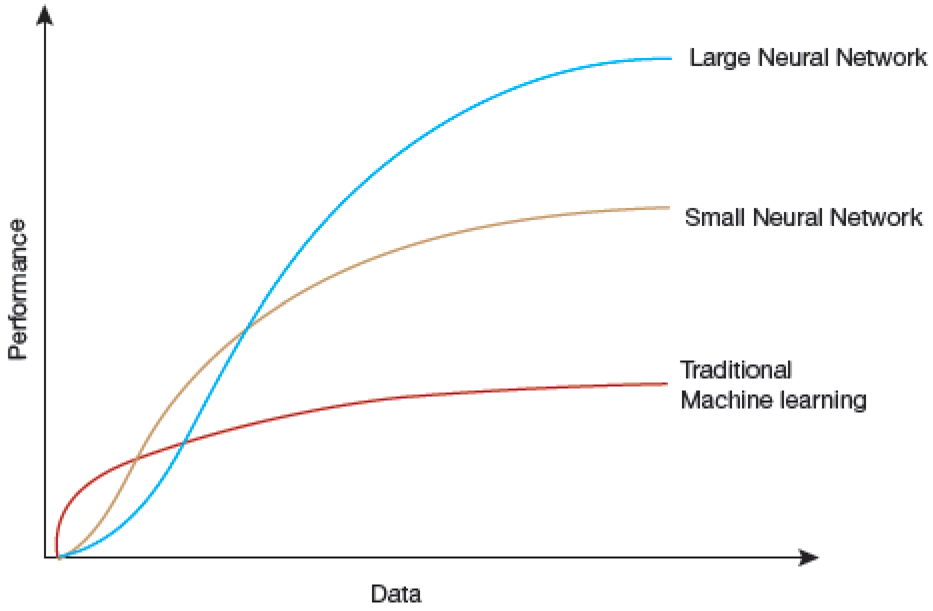


FIGURE 2.33 Comparison between ML and DL

While a typical machine learning neural network model may have two to three hidden layers, deep networks on the other hand have as many as 150 hidden layers.

Moreover, deep learning performs 'end-to-end learning'. This means that a neural network when given raw data and a task (like classification), the model automatically learns how to do this task.

Last but not the least, deep learning algorithms scale well with data. It is free from shallow learning converges problems that a typical machine learning model may face. Shallow learning refers to creation of plateau at a certain level of performance. Even after adding more data for training, the accuracy and performance of the model does not change.

While the biggest advantage of deep learning over machine learning techniques is that it frees users from the worry about trimming down the number of features used, but the main drawback is that a deep learning model is very expensive to train and needs commercial-grade GPUs. Many data scientists, treat deep learning model as a ‘black box’ because the model is very complex and extremely difficult to understand.

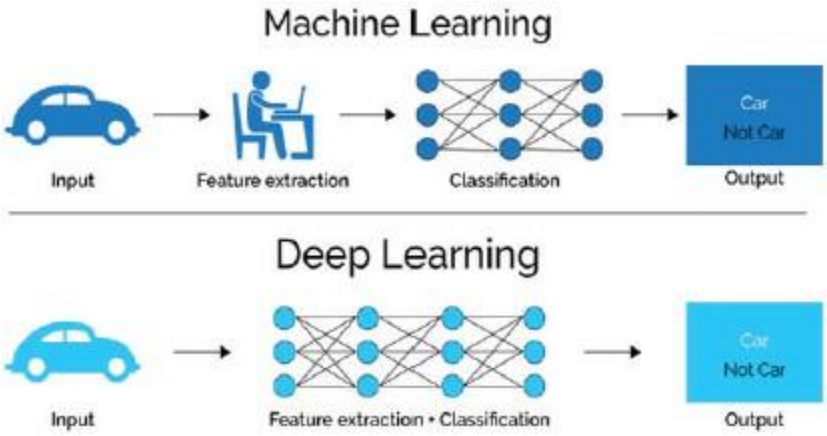
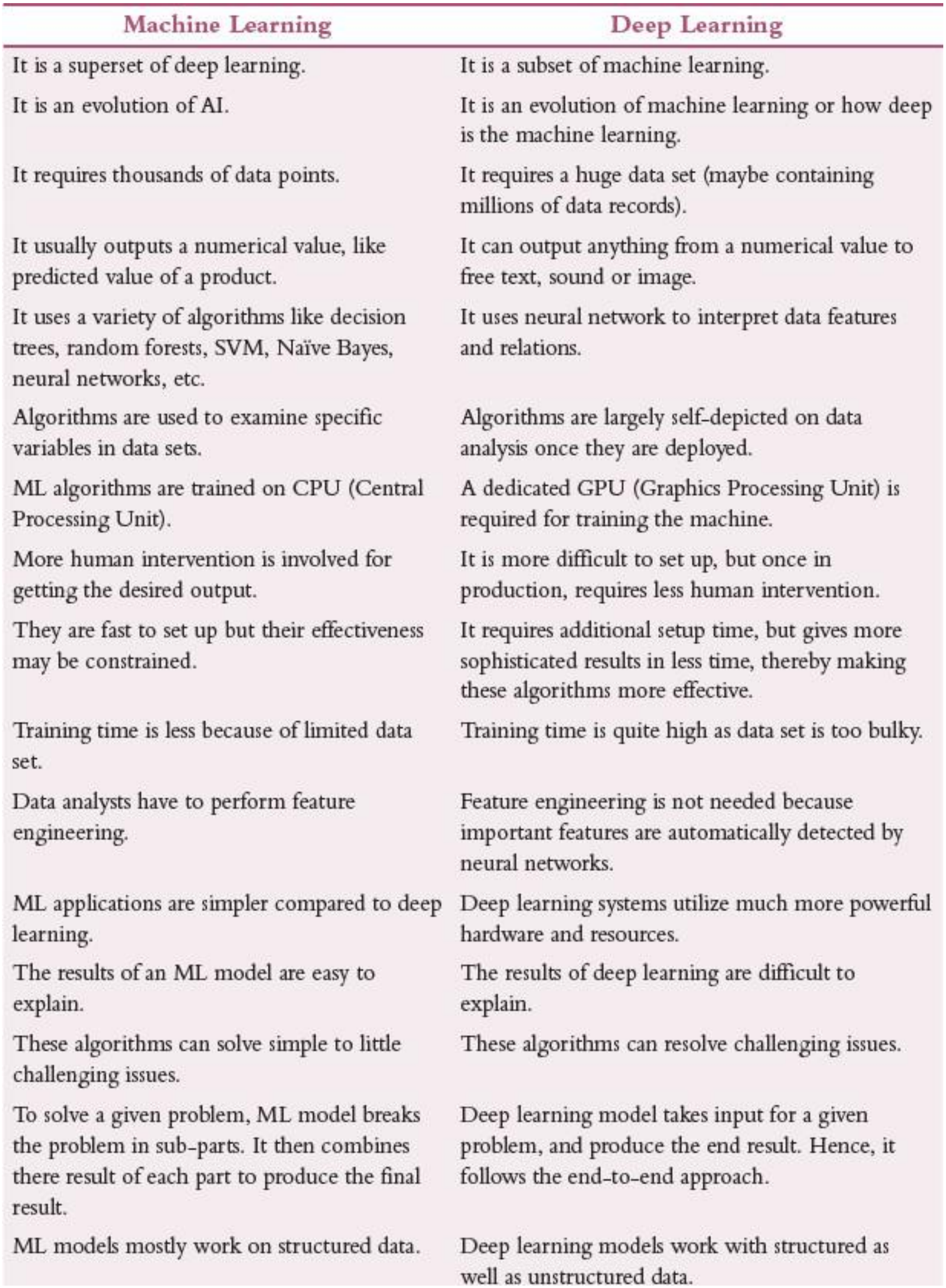


FIGURE 2.34 Deep Learning supports more automation using large number of hidden layers)

TABLE 2.3 Differences between machine learning and deep learning

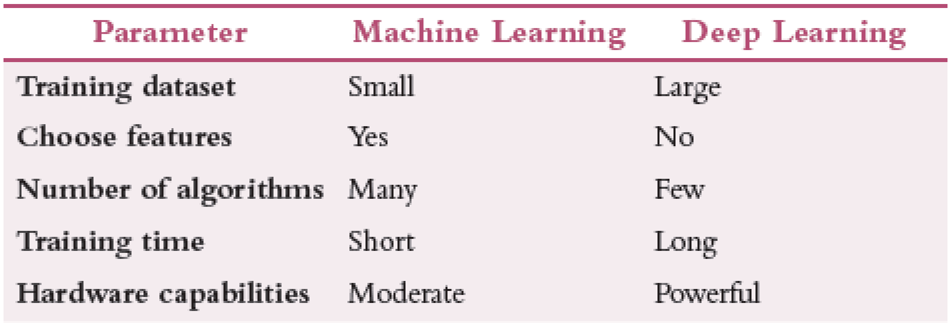


Machine Learning	Deep Learning
It is a superset of deep learning.	It is a subset of machine learning.
It is an evolution of AI.	It is an evolution of machine learning or how deep is the machine learning.
It requires thousands of data points.	It requires a huge data set (maybe containing millions of data records).
It usually outputs a numerical value, like predicted value of a product.	It can output anything from a numerical value to free text, sound or image.
It uses a variety of algorithms like decision trees, random forests, SVM, Naïve Bayes, neural networks, etc.	It uses neural network to interpret data features and relations.
Algorithms are used to examine specific variables in data sets.	Algorithms are largely self-depicted on data analysis once they are deployed.
ML algorithms are trained on CPU (Central Processing Unit).	A dedicated GPU (Graphics Processing Unit) is required for training the machine.
More human intervention is involved for getting the desired output.	It is more difficult to set up, but once in production, requires less human intervention.
They are fast to set up but their effectiveness may be constrained.	It requires additional setup time, but gives more sophisticated results in less time, thereby making these algorithms more effective.
Training time is less because of limited data set.	Training time is quite high as data set is too bulky.
Data analysts have to perform feature engineering.	Feature engineering is not needed because important features are automatically detected by neural networks.
ML applications are simpler compared to deep learning.	Deep learning systems utilize much more powerful hardware and resources.
The results of an ML model are easy to explain.	The results of deep learning are difficult to explain.
These algorithms can solve simple to little challenging issues.	These algorithms can resolve challenging issues.
To solve a given problem, ML model breaks the problem in sub-parts. It then combines there result of each part to produce the final result.	Deep learning model takes input for a given problem, and produce the end result. Hence, it follows the end-to-end approach.
ML models mostly work on structured data.	Deep learning models work with structured as well as unstructured data.

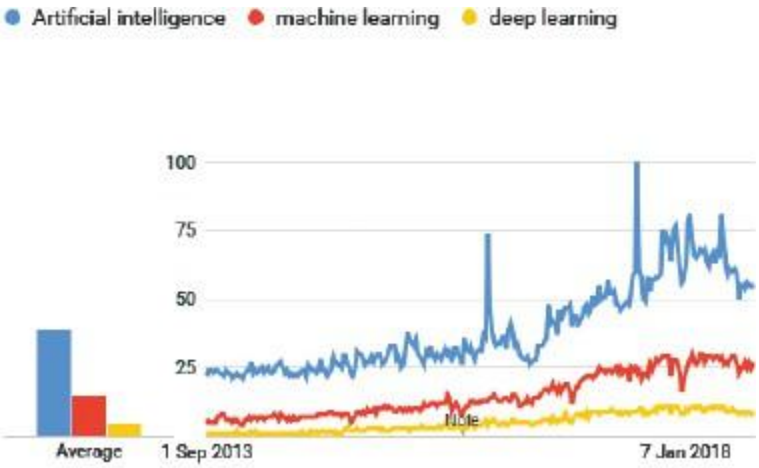
When to Use Ml or Dl?

If you have a complex problem to solve with huge amounts of data and powerful hardware capabilities, then you must go for implementing a deep learning solution. But in case any of them is missing, choose the ML model to solve your problem. Table 2.4 compares the two techniques in more detail.

TABLE 2.4 Comparison between machine learning and deep learning



Parameter	Machine Learning	Deep Learning
Training dataset	Small	Large
Choose features	Yes	No
Number of algorithms	Many	Few
Training time	Short	Long
Hardware capabilities	Moderate	Powerful



Note that in case you want to automate feature extraction, especially when a data set contains hundreds and thousands of features (or attributes) then, use deep learning. Since, only few features are actually useful, a deep learning system learns about the relevance of those features to select ones that will help the system to learn useful information. Feature extraction uses PCA, T-SNE or any other dimensionality reduction algorithms.

For example, an image processing application extracts the features in the image (like the eyes, the nose, lips and so on) and feeds it to the classification model. The deep learning model has layers of neural networks. The first layer learns small details from the picture; the next layers will combine the previous knowledge to make more complex information.

Applications of Deep Learning

Deep learning techniques have already outperformed humans in some tasks like classifying objects in images. For example, deep learning uses millions of images and thousands of hours of video to develop driverless cars. For this, it requires substantial computing power. High-performance GPUs (Graphics Processing Units) having a parallel architecture when combined with cloud computing allows development teams to reduce training time for a deep learning network from weeks to hours or even less. Besides, driverless cars, deep learning is also used in the following areas.

In aerospace and defence sector, deep learning is used to ***identify objects from satellites*** that locate areas of interest, and safe or unsafe zones for the military troops. In medical field, cancer researchers use deep learning to automatically ***detect cancer*** cells.

Industrial automation uses deep learning to improve ***worker safety*** around heavy machinery. It automatically detects when people or objects comes within an unsafe distance of machines.

In electronic gadgets, deep learning is used in ***automated hearing and speech translation***. For example, home assistance devices that recognizes and responds to voice commands and knows user's preferences are trained using deep learning algorithms.

Digital assistants like Siri, Cortana, Alexa, and Google Now use deep learning for natural language processing and ***speech recognition***.

Deep learning assist in ***translation*** between languages. This can be very helpful for travellers, business people and those in government sectors. Skype and Google

Translate uses deep neural networks to translate spoken conversations in real-time.

Several email systems including Gmail uses deep learning techniques to identify ***spam messages*** before they even reach the inbox.

PayPal uses deep learning to ***prevent fraudulent*** payments.
Apps like CamFind implements deep neural networks to allow its users to take a picture of any object and ***discover details*** of that object using mobile visual search technology.

Google Deepmind’s WaveNet can generate speech ***mimicking human voice*** that sounds more natural than speech systems presently on the market.

Facebook uses deep learning identify and ***tag friends*** a user uploads a new picture.

Example, How Deep Learning is Used for Fraud Detection

To detect fraud, deep learning uses time, geographic location, IP address, type of retailer, and other features. In such a scenario, the first layer of the neural network processes raw data input (like the amount of the transaction) and passes it as an output to the next layer. The second layer accepts the output of the first layer as its input. This input is then processed by including additional information like the user’s IP address and passes its result to the third layer.

The third layer may incorporate geographic location to make the results even better. This process continues across all levels of the neuron network. The final layer generates the output that may signal the analyst to freeze the user’s account. Some applications of AI that makes extensive use of Deep Learning are illustrated in Fig. 2.35.

Deep learning is used by ***chatbots*** and service bots for providing ***customer service***. Many companies use these techniques to respond in an intelligent and helpful way to text- based questions. ***Image colourization*** activities use deep learning techniques to transform black-and-white images into coloured ones. This work was earlier done manually. But these days, deep learning algorithms generate impressive and accurate results by using the context and objects in the images to colour them to recreate the black-and-white image in colour.

Deep learning algorithms when used in medicine and pharmaceuticals, can be used to diagnose diseases and tumour in the body. They are also used to prescribe personalized medicines created specifically for an individual's genome.

Personalized shopping and entertainment can use deep learning techniques to suggest what users should buy or watch next.

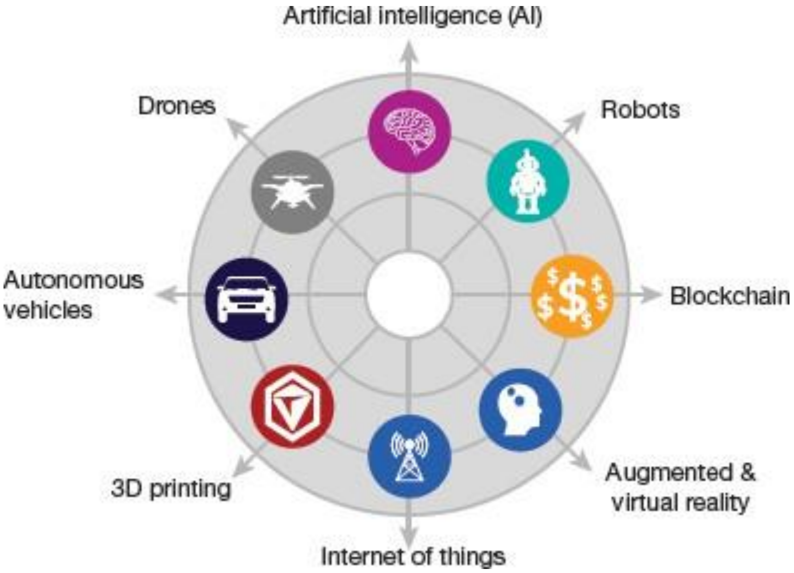


FIGURE 2.35 Applications of Deep Learning

Review Questions:

1.What is neural networks?
2.What is SGD algorithm?
3. what is Deep learning?

L8:

2.8 Support Vector Machine (SVM)

A support vector machine (SVM) is a classification algorithm that classifies data based on its features. An SVM will classify any new element into one of the two classes as shown in Fig. 2.36.

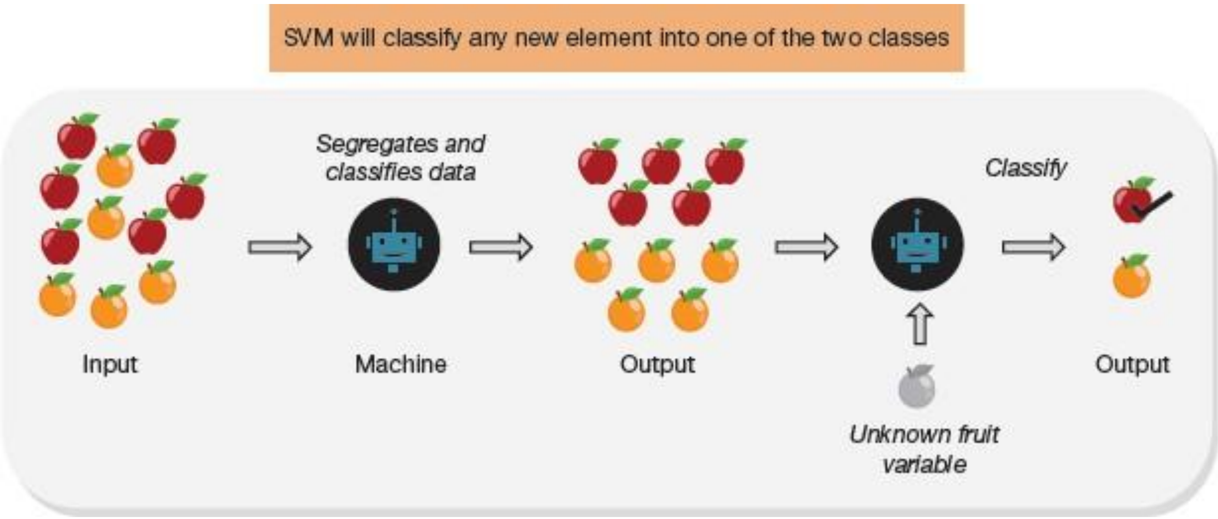

SVM will classify any new element into one of the two classes
Input
Segregates and classifies data
Machine
Output
Classify
Output
Unknown fruit variable

FIGURE 2.36 Classification Process using SVM

To make our machine learning model learn, we supply input data to it. The SVM algorithm will automatically extract features from the input data. This knowledge is used to segregate and classify the input data to generate the desired output. Once the model has learnt how to classify, any new data inputted to it can be classified (with a specific accuracy).

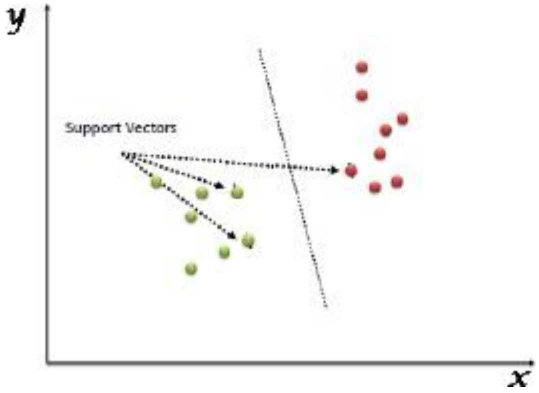


FIGURE 2.37 Identifying Best Line on the Plot for Classification

An SVM is therefore a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used to classify data. In the SVM algorithm, each data item is plotted as a point in n- dimensional space (where n is number of features). In this plot, support vectors are simply the co-ordinates of individual observation. For classification, we need to identify the best hyper plane/line or class (refer Fig. 2.37). For example, consider the scenarios given below.

Scenario 1: In Fig. 2.38, there are three hyper planes A, B and

C. However, hyper planes B and C better segregates the two classes.

Scenario 2: When all three hyper-planes (A, B and C) are segregating the classes well, we must choose one that

maximizes the distances between nearest data point (either class) and hyper-plane. This distance is called m**argin**.
Note: Hyper plane with low margin can result in mis- classification.

In Fig. 2.38, we see that the margin for hyper-plane C is high as compared to both A and B. Hence, hyper plane C is chosen as the right decision boundary or hyper plane.

Scenario 3: In this scenario, hyper plane A is selected as it maximizes the margin and reduces classification error.

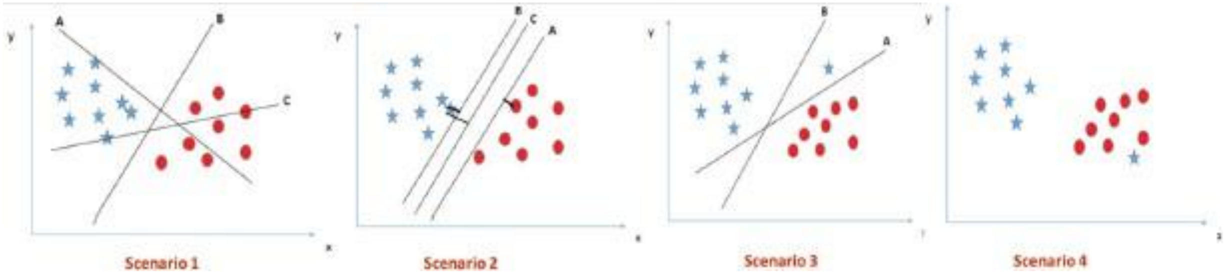


FIGURE 2.38 Identifying Hyperplanes on the Plot for Classification

Scenario 4: In Fig. 2.38, segregation is difficult as one star lies in the territory of circle class.

Scenario-5: In the scenario, linear hyper-plane between the two classes is not possible. However, SVM can solve this solution in two ways.

Solution 1: Adding a new feature z=x^2+y^2 and plotting the data points on axis x and z.

In the plot,

1. all values for z would be positive always because z is the squared sum of both x and y
2. In the original plot, red circles appear close to the origin of x and y axes because of lower value of z. Stars appear away from the origin due to higher value of z.

Solution 2: Use the kernel trick of the SVM algorithm. The SVM kernel is a function that takes low dimensional input space and transforms it to a higher dimensional space, that is, it converts not separable problem to separable problem. It is usually used in non-linear separation problem. In the figure, we see that the hyper-plane in original input space looks like a circle.

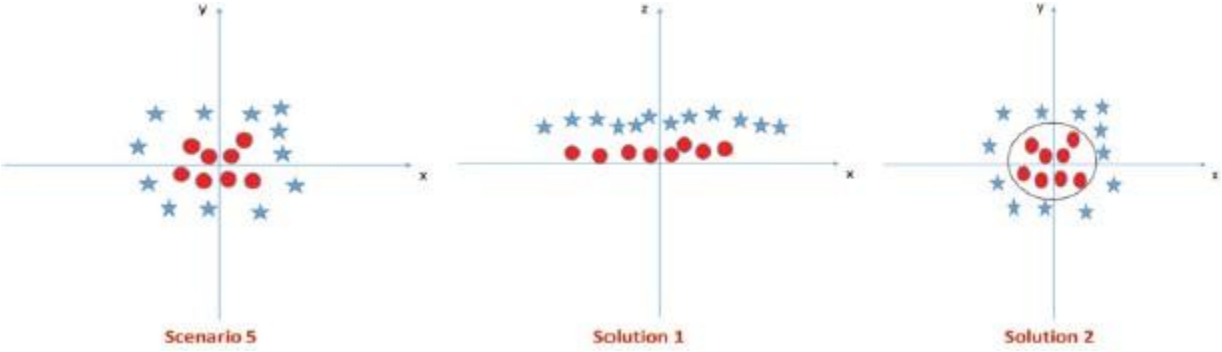


FIGURE 2.39 Evaluating Hyperplanes

Example:

Cricket players can be classified as batsmen or bowlers using the runs-to-wicket ratio. A player with more runs is a batsman and the one with more wickets is a bowler. In this example (refer Fig. 2.40), we can create a two- dimensional plot showing a clear separation between bowlers and batsmen.

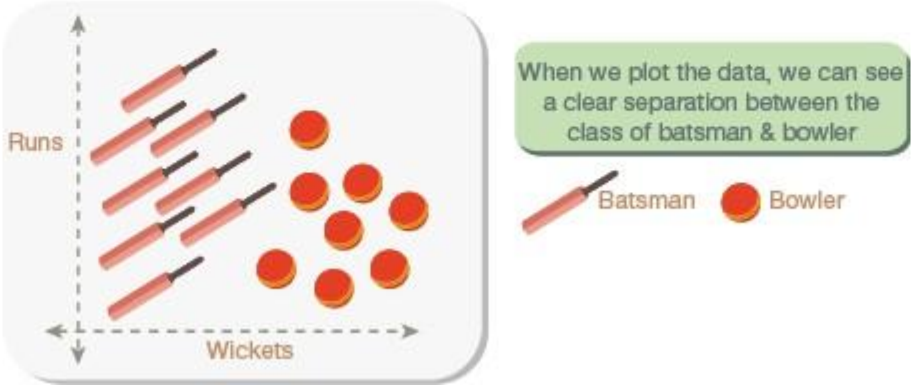


FIGURE 2.40 2D Plot representing Batsmen and Bowlers Data

Whenever we get data for a new player that is not yet classified, we draw a decision boundary, or a line separating the two classes to help classify the new data points. We can draw multiple decision boundaries as shown in Fig. 2.41. Therefore, the aim is to find the line of best fit that clearly separates those two groups. The correct line will help to classify the new data point.

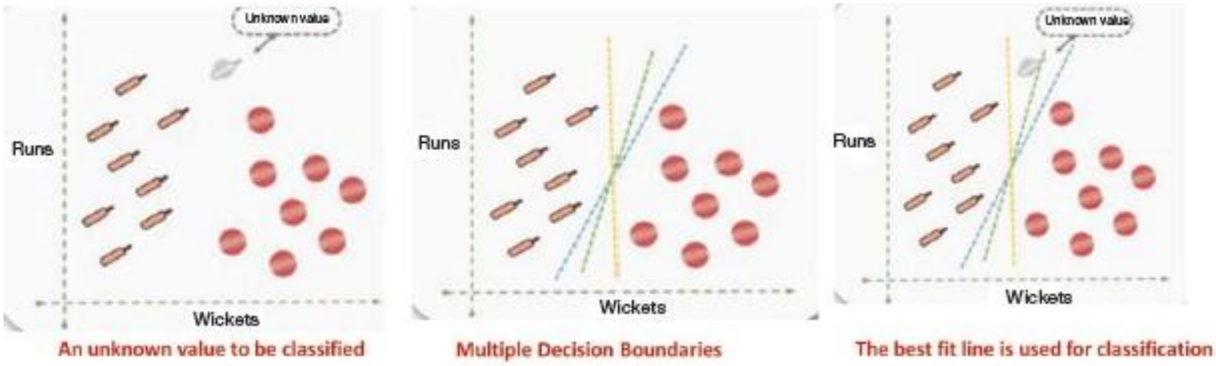


FIGURE 2.41 Identifying Line of Best Fit

The best fit line is computed by evaluating the maximum margin from equidistant support vectors. Support vectors mean two points (one from each class) that are closest together, but that maximize the distance between them or the margin.

Points to Remember

1. **Hyperplane** is a line in 2D, plane in 3D and hyper plane in higher dimensions (above 3 dimensions).

2. **Margin** is the distance between the hyperplane and the closest data point. We need to maximize this margin for better classification.
3. The kernel function is used to handle non-linear separable data. It does this by transforming data into a higher dimensional feature space to make it possible to perform the linear separation. Different kernel functions include the following:
    1. Gaussian RBF kernel
    2. Sigmoid kernel
    3. Polynomial kernel

Any of these functions can be chosen depending on the dimensions and way data has to be transformed.

Tuning Parameters

Tuning parameters can be divided into two groups—linear kernel SVM and non-linear kernel SVM.

For linear kernel SVM, there is only one parameter—cost (C) which implies misclassification cost on training data.

A large C gives you low bias and high variance; low bias, as there is a penalty (cost) for misclassification. A large C means the cost of misclassification is high. This ensures that the algorithm strictly explains the input data stricter (and potentially overfits).

A small C gives higher bias and lower variance. This is because a small value of C lowers the cost of misclassification.

Ideally, a good balance must be maintained between being 'not too strict' and 'not too loose'. For finding an optimal value of C, cross-validation and resampling techniques along with grid search have proved to be excellent.

For non-linear kernel (Radial), Two parameters for fine tuning in radial kernel are cost and gamma. We have already discussed cost (C) in case of linear kernel.

Gamma explains how far the influence of a single training example reaches. A small value of gamma indicates that the model is too constrained and cannot capture the complexity or 'shape' of the data.

SVM algorithm is very sensitive to the choice of the kernel parameters.

2.8.1 How Does SVM Work?

Step 1: Select an optimal hyperplane that maximizes margin

Step 2: Apply a penalty or cost 'c' for misclassification.

Step 3: Use kernel trick on non-linearly separable data points by transforming data to high dimensional space where it is easier to classify with linear decision surfaces.

Data Standardization

All kernel methods are based on distance. Hence, values of all variables must be scaled. If we do not standardize our variables to comparable ranges, then the variable with the largest range will completely dominate the computation of the kernel function. For example, if we have two variables—X1 and X2 where values of variable X1 lies in the range 0 to 100 while that of X2 lies in range of 100 and 10000, then values of X2 will dominate variable X1. Therefore, we must standardize each

Review Questions:

1.What is SVM?

2.What is Hyperplane and margin?

3. What is linear kernel SVM?