CBCS SCHEME

USN [redacted] BAI602

Sixth Semester B.E./B.Tech. Degree Examination, June/July 2025
Machine Learning - I

Time: 3 hrs. Max. Marks: 100

Note: 1. Answer any FIVE full questions, choosing ONE full question from each module.
2. M : Marks , L: Bloom's level , C: Course outcomes

		Module – 1	M	L	C
Q.1	a.	Explain the challenges faced in machine learning.	08	L2	CO1
	b.	Explain the types of Data in Big data.	06	L2	CO2
	c.	Describe the four types of data analytics.	06	L2	CO2
		OR			
Q.2	a.	Briefly explain supervised and unsupervised learning.	08	L2	CO1
	b.	Explain skewness and kurtosis.	06	L2	CO2
	c.	Describe the different types of Data visualization techniques.	06	L2	CO2
		Module – 2			
Q.3	a.	Explain the types of continuous probability distribution.	06	L2	CO2
	b.	Briefly explain about confusion matrix and ROC curve.	08	L2	CO3
	c.	Let the data points be $\begin{pmatrix}2\\6\end{pmatrix}$ and $\begin{pmatrix}1\\7\end{pmatrix}$. Apply PCA and find the transformed data	06	L2	CO2
		OR			
Q.4	a.	Explain non-parametric density estimation.	06	L2	CO2
	b.	Explain (i) Training (ii) Testing and (iii) Validation (iv) Unbalanced data set.	08	L2	CO3
	c.	Find LU decomposition of the given matrix. $A = \begin{bmatrix}1 & 2 & 4\\3 & 3 & 2\\3 & 4 & 2\end{bmatrix}$	06	L3	CO2
		Module – 3			
Q.5	a.	Explain about linear regression.	06	L2	CO4
	b.	Consider the training dataset given in Table. Use weighted K-NN and determine the class. Given the test instance (7.6, 60, 8) [Assign K = 3]	08	L3	CO4
	c.	Explain the differences between Instance based learning and Model – based learning.	06	L2	CO4

Sl. No.	CGPA	Assessment	Project submitted	Result
1	9.2	85	8	Pass
2	8	80	7	Pass
3	8.5	81	8	Pass
4	6	45	5	Fail
5	6.5	50	4	Fail
6	8.2	72	7	Pass
7	5.8	38	5	Fail
8	8.9	91	9	Pass

OR

Q.6	a.	Explain about logistic regression.	06	L2	CO4
	b.	Consider the data provided in table and fit it using the second order polynomial.	08	L3	CO4

X	Y
1	1
2	4
3	9
4	15

	c.	Explain nearest centroid classifier using an example.	06	L2	CO4
		Module – 4			
Q.7	a.	Explain pre-prunning and post –prunning. Compare both the methods.	06	L2	CO4
	b.	Assess a student's performance during his course of study and predict whether a student will get a job offer or not in his final year of the course. The training dataset T consists of 10 data instances with attributes such as 'CGPA', 'Interactive', 'Practical knowledge' and 'Communication skills'. The target class attribute is the Job offer.	10	L3	CO4

Sl. No.	CGPA	Interacti veness	Practical knowledge	Communication skills	Job Offer
1	≥ 9	Yes	Very Good	Good	Yes
2	≥ 8	No	Good	Moderate	Yes
3	≥ 9	No	Average	Poor	No
4	< 8	No	Average	Good	No
5	≥ 8	Yes	Good	Moderate	Yes
6	≥ 9	Yes	Good	Moderate	Yes
7	< 8	Yes	Good	Poor	No
8	≥ 9	No	Very Good	Good	Yes
9	≥ 8	Yes	Good	Good	Yes
10	≥ 8	Yes	Average	Good	Yes

Table Q.7(b)

	c.	Explain Entropy and Gini Index.	04	L2	CO4
		OR			
Q.8	a.	Using the above Table – Q.7 (b). Assess a students performance using Naire Baye's algorithm with the dataset. Predict whether a student gets a job offer or not in his final year of the course.	10	L3	CO4
	b.	Explain maximum A Posterios Hypothesis, h_{MAP} and maximum likelihood (M_L) hypothesis, h_{ML}.	06	L2	CO4
	c.	Explain Baye's optimal classifier.	04	L2	CO4
		Module – 5			
Q.9	a.	Explain the types of artificial neural network.	08	L2	CO4
	b.	Explain Grid-Based approach.	08	L2	CO4
	c.	What are the popular applications of artificial neural networks?	04	L1	CO4
		OR			
Q.10	a.	Explain the concept of perceptron and Learning Theory.	08	L2	CO2
	b.	Explain any 4 proximity measures.	08	L2	CO4
	c.	What are the applications of clustering?	04	L1	CO4

* * * * *

VISVEVARAYA TECHNOLOGICAL UNIVERSITY

CBCS Scheme

Sixth Semester B.E. Degree Examination, June/July 2025

MACHINE LEARNING - I (BAI602)

MODULE – 1

1.a. Explain the challenges faced in machine learning. (8M)

1. Problems – Machine learning can deal with the 'well-posed' problems where specifications are complete and available. Computers cannot solve 'ill-posed' problems.
   Consider one simple example (shown in Table 1.3):

   Table 1.3: An Example

Input (x_1, x_2)	Output (y)
1, 1	1
2, 1	2
3, 1	3
4, 1	4
5, 1	5

   Can a model for this test data be multiplication? That is, $y = x_1 \times x_2$. Well! It is true! But, this is equally true that y may be $y = x_1 \div x_2$, or $y = x_1^{x_2}$. So, there are three functions that fit the data. This means that the problem is ill-posed. To solve this problem, one needs more example to check the model. Puzzles and games that do not have sufficient specification may become an ill-posed problem and scientific computation has many ill-posed problems.
2. Huge data – This is a primary requirement of machine learning. Availability of a quality data is a challenge. A quality data means it should be large and should not have data problems such as missing data or incorrect data.
3. High computation power – With the availability of Big Data, the computational resource requirement has also increased. Systems with *Graphics Processing Unit* (GPU) or even *Tensor Processing Unit* (TPU) are required to execute machine learning algorithms. Also, machine learning tasks have become complex and hence time complexity has increased, and that can be solved only with high computing power.
4. Complexity of the algorithms – The selection of algorithms, describing the algorithms, application of algorithms to solve machine learning task, and comparison of algorithms have become necessary for machine learning or data scientists now. Algorithms have become a big topic of discussion and it is a challenge for machine learning professionals to design, select, and evaluate optimal algorithms.
5. Bias/Variance – Variance is the error of the model. This leads to a problem called bias/variance tradeoff. A model that fits the training data correctly but fails for test data, in general lacks generalization, is called overfitting. The reverse problem is called underfitting where the model fails for training data but has good generalization. Overfitting and underfitting are great challenges for machine learning algorithms.

1.b. Explain the types of data in Big Data. (6M)

2.1.1 Types of Data

In Big Data, there are three kinds of data. They are structured data, unstructured data, and semi-structured data.

Structured Data

In structured data, data is stored in an organized manner such as a database where it is available in the form of a table. The data can also be retrieved in an organized manner using tools like SQL.

The structured data frequently encountered in machine learning are listed below:

Record Data A dataset is a collection of measurements taken from a process. We have a collection of objects in a dataset and each object has a set of measurements. The measurements can be arranged in the form of a matrix. Rows in the matrix represent an object and can be called as entities, cases, or records. The columns of the dataset are called attributes, features, or fields. The table is filled with observed data. Also, it is better to note the general jargons that are associated with the dataset. Label is the term that is used to describe the individual observations.

Data Matrix It is a variation of the record type because it consists of numeric attributes. The standard matrix operations can be applied on these data. The data is thought of as points or vectors in the multidimensional space where every attribute is a dimension describing the object.

Graph Data It involves the relationships among objects. For example, a web page can refer to another web page. This can be modeled as a graph. The modes are web pages and the hyperlink is an edge that connects the nodes.

Ordered Data Ordered data objects involve attributes that have an implicit order among them.

The examples of ordered data are:

1. Temporal data – It is the data whose attributes are associated with time. For example, the customer purchasing patterns during festival time is sequential data. Time series data is a special type of sequence data where the data is a series of measurements over time.
2. Sequence data – It is like sequential data but does not have time stamps. This data involves the sequence of words or letters. For example, DNA data is a sequence of four characters – A T G C.
3. Spatial data – It has attributes such as positions or areas. For example, maps are spatial data where the points are related by location.

Unstructured Data

Unstructured data includes video, image, and audio. It also includes textual documents, programs, and blog data. It is estimated that 80% of the data are unstructured data.

Semi-Structured Data

Semi-structured data are partially structured and partially unstructured. These include data like XML/JSON data, RSS feeds, and hierarchical data.

1.c. Describe the four types of data analytics. (6M)

Data analysis and data analytics are terms that are used interchangeably to refer to the same concept. However, there is a subtle difference. Data analytics is a general term and data analysis is a part of it. Data analytics refers to the process of data collection, preprocessing and analysis. It deals with the complete cycle of data management. Data analysis is just analysis and is a part of data analytics. It takes historical data and does the analysis. Data analytics, instead, concentrates more on future and helps in prediction.

There are four types of data analytics:

1. Descriptive analytics
2. Diagnostic analytics
3. Predictive analytics
4. Prescriptive analytics

Descriptive Analytics It is about describing the main features of the data. After data collection is done, descriptive analytics deals with the collected data and quantifies it. It is often stated that analytics is essentially statistics. There are two aspects of statistics – Descriptive and Inference. Descriptive analytics only focuses on the description part of the data and not the inference part.

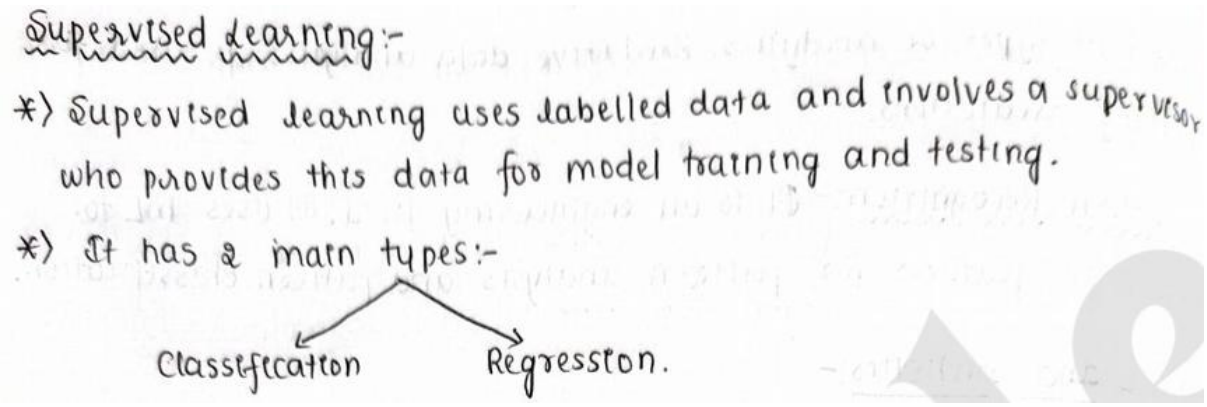
Diagnostic Analytics It deals with the question – 'Why?'. This is also known as causal analysis, as it aims to find out the cause and effect of the events. For example, if a product is not selling, diagnostic analytics aims to find out the reason. There may be multiple reasons and associated effects are analyzed as part of it.

Predictive Analytics It deals with the future. It deals with the question – 'What will happen in future given this data?'. This involves the application of algorithms to identify the patterns to predict the future. The entire course of machine learning is mostly about predictive analytics and forms the core of this book.

Prescriptive Analytics It is about the finding the best course of action for the business organizations. Prescriptive analytics goes beyond prediction and helps in decision making by giving a set of actions. It helps the organizations to plan better for the future and to mitigate the risks that are involved.

OR

2.a. Briefly explain supervised and unsupervised learning. (8M)



Classification:-

*) Classification is a supervised learning method that predicts labels (discrete values) based on input attributes. The relationship b/w inputs and labels is learned using a classification model.

*) Process (2 stages)

The model learns from labelled data.

The model classifies new, unseen data.

*) Common algorithms used are:

→ Decision Tree
→ Random Forest
→ Support Vector Machines (SVM)
→ Naive Bayes
→ Artificial neural networks & Deep Learning (eg., CNN, RNN)

Ex:- *) Identifying images of cats and dogs
*) Classifying diseases (or) plant species.

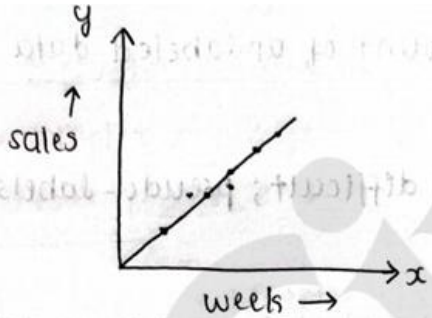
Regression:-

*) Regression predicts continous numerical values rather than discrete labels.

Ex:- Predicting product sales based on weeks of data using a linear regression model of the form:

$$y = 0.66x + 0.54 \quad (y = mx + c)$$

where $x \rightarrow$ independent variable (week)
$y \rightarrow$ dependent variable (sales)



Unsupervised Learning:-

*) Learning occurs through self-instruction without a supervisor.

*) the algorithm identifies patterns and groups similar objects together.

Ex:- i) Cluster analysis

ii) Dimensionality reduction.

Cluster analysis:-

*) Group similar objects into desjoint clusters based on their attributes.

*) Used in image segmentation, medical anomaly detection and gene database clustering.

Ex:- A clustering algorithm groups dog and cat images into separate clusters.

Key algorithms:- K-means, PCA, Hierarchical algorithms.

Dimensionality Reduction:-

*) Reduces the no of features while retaining essential information.

*) Converts high-dimensional data into lower-dimensional representations.

2.b. Explain skewness and kurtosis. (6M)

2.5.4 Shape

Skewness and Kurtosis (called moments) indicate the symmetry/asymmetry and peak location of the dataset.

Skewness

The measures of direction and degree of symmetry are called measures of third order. Ideally, skewness should be zero as in ideal normal distribution. More often, the given dataset may not have perfect symmetry (consider the following Figure 2.8).

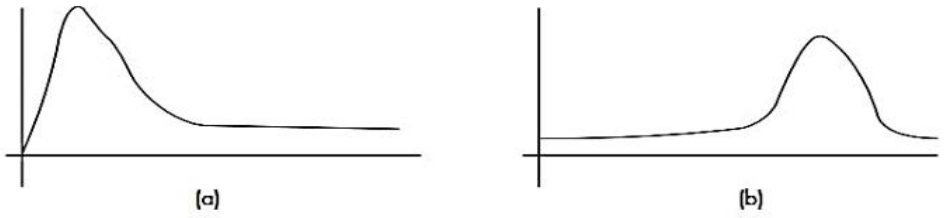


Figure 2.8: (a) Positive Skewed and (b) Negative Skewed Data

The dataset may also either have very high values or extremely low values. If the dataset has far higher values, then it is said to be skewed to the right. On the other hand, if the dataset has far more low values then it is said to be skewed towards left. If the tail is longer on the left-hand side and hump on the right-hand side, it is called positive skew. Otherwise, it is called negative skew.

The given dataset may have an equal distribution of data. The implication of this is that if the data is skewed, then there is a greater chance of outliers in the dataset. This affects the mean and median. Hence, this may affect the performance of the data mining algorithm. A perfect symmetry means the skewness is zero. In the case of skew, the median is greater than the mean. In positive skew, the mean is greater than the median.

Generally, for negatively skewed distribution, the median is more than the mean. The relationship between skew and the relative size of the mean and median can be summarized by a convenient numerical skew index known as Pearson 2 skewness coefficient.

$$\frac{3 \times (\mu - median)}{\sigma} \tag{2.12}$$

Also, the following measure is more commonly used to measure skewness. Let $X_1, X_2, \cdots, X_N$ be a set of 'N' values or observations then the skewness can be given as:

$$\frac{1}{N} \times \sum_{i=1}^{N} \frac{(x_i - \mu)^3}{\sigma^3} \tag{2.13}$$

Here, μ is the population mean and σ is the population standard deviation of the univariate data. Sometimes, for bias correction instead of N, $N-1$ is used.

Kurtosis

Kurtosis also indicates the peaks of data. If the data is high peak, then it indicates higher kurtosis and vice versa.

Kurtosis is the measure of whether the data is heavy tailed or light tailed relative to normal distribution. It can be observed that normal distribution has bell-shaped curve with no long tails. Low kurtosis tends to have light tails. The implication is that there is no outlier data. Let $x_1, x_2, \cdots, x_N$ be a set of 'N' values or observations. Then, kurtosis is measured using the formula given below:

$$\frac{\sum_{i=1}^{N}(x_i - \bar{x})^4 / N}{\sigma^4} \tag{2.14}$$

It can be observed that $N - 1$ is used instead of N in the numerator of Eq. (2.14) for bias correction. Here, $\bar{x}$ and σ are the mean and standard deviation of the univariate data, respectively.

2.c. Describe the different types of data visualization techniques. (6M)

2.5.1 Data Visualization

To understand data, graph visualization is must. Data visualization helps to understand data. It helps to present information and data to customers. Some of the graphs that are used in univariate data analysis are bar charts, histograms, frequency polygons and pie charts.

The advantages of the graphs are presentation of data, summarization of data, description of data, exploration of data, and to make comparisons of data. Let us consider some forms of graphs now:

Bar Chart A Bar chart (or Bar graph) is used to display the frequency distribution for variables. Bar charts are used to illustrate discrete data. The charts can also help to explain the counts of nominal data. It also helps in comparing the frequency of different groups.

The bar chart for students' marks {45, 60, 60, 80, 85} with Student ID = {1, 2, 3, 4, 5} is shown below in Figure 2.3.

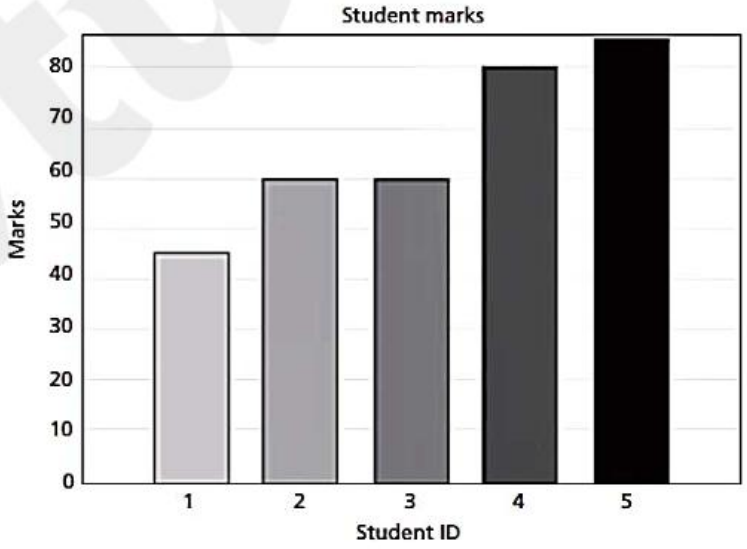


Figure 2.3: Bar Chart

Pie Chart These are equally helpful in illustrating the univariate data. The percentage frequency distribution of students' marks {22, 22, 40, 40, 70, 70, 70, 85, 90, 90} is below in Figure 2.4.

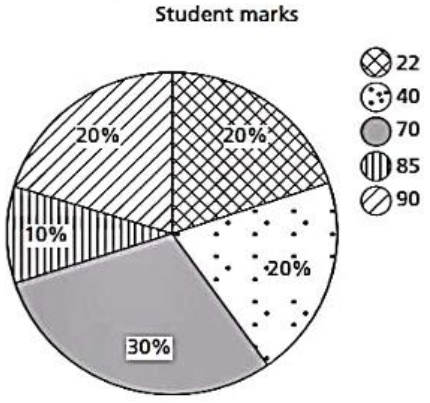


Figure 2.4: Pie Chart

It can be observed that the number of students with 22 marks are 2. The total number of students are 10. So, 2/10 × 100 = 20% space in a pie of 100% is allotted for marks 22 in Figure 2.4.

Histogram It plays an important role in data mining for showing frequency distributions. The histogram for students' marks {45, 60, 60, 80, 85} in the group range of 0–25, 26–50, 51–75, 76–100 is given below in Figure 2.5. One can visually inspect from Figure 2.5 that the number of students in the range 76–100 is 2.

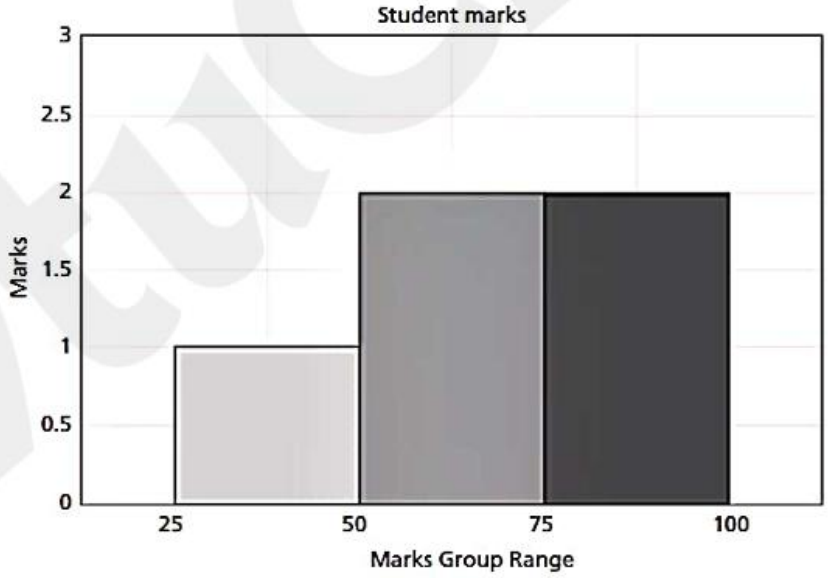


Figure 2.5: Sample Histogram of English Marks

Histogram conveys useful information like nature of data and its mode. Mode indicates the peak of dataset. In other words, histograms can be used as charts to show frequency, skewness present in the data, and shape.

Dot Plots These are similar to bar charts. They are less clustered as compared to bar charts, as they illustrate the bars only with single points. The dot plot of English marks for five students with ID as {1, 2, 3, 4, 5} and marks {45, 60, 60, 80, 85} is given in Figure 2.6. The advantage is that by visual inspection one can find out who got more marks.

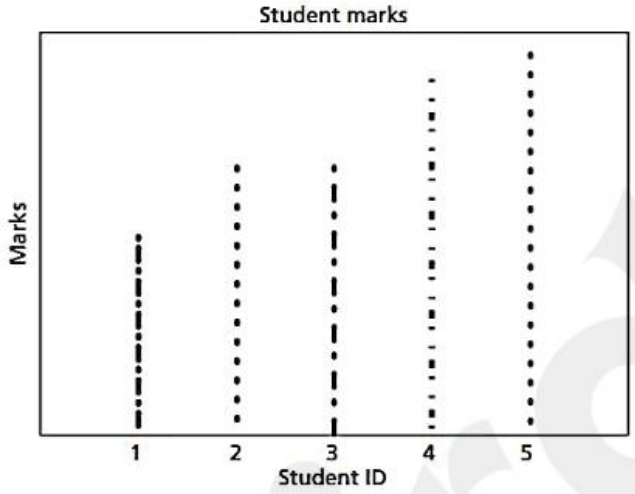


Figure 2.6: Dot Plots

MODULE -2

3.a. Explain the types of continuous probability distribution. (6M)

1. Normal Distribution – Normal distribution is a continuous probability distribution. This is also known as gaussian distribution or bell-shaped curve distribution. It is the most common distribution function. The shape of this distribution is a typical bell-shaped curve. In normal distribution, data tends to be around a central value with no bias on left or right. The heights of the students, blood pressure of a population, and marks scored in a class can be approximated using normal distribution.

   PDF of the normal distribution is given as:

$$f(x, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.24}$$

   Here, μ is mean and σ is the standard deviation. Normal distribution is characterized by two parameters – mean and variance.

   Mostly, one uses the normal distribution curve of mean 0 and a *SD* of 1. In normal distribution, mean, median and mode are same. The distribution extends from $-\infty$ to $+\infty$. Standard deviation is how the data is spread out.

   One important concept associated with normal distribution is z-score. It can be computed as:

   $z = \frac{x - \mu}{\sigma}$. When μ is zero and σ is 1, z-score is same as x. This is useful to normalize the data.

   Most of the statistical tests expect data to follow normal distribution. To check it, normality tests are used. Normality test of the data can be done by Q-Q plot where CDF of one random variable follows CDF of normal distribution. Then, quantity of one distribution is plotted against other distributions. If they are same, then the plot closely follows the straight line from bottom-left to top-right.

2. Rectangular Distribution – This is also known as uniform distribution. It has equal probabilities for all values in the range a, b. The uniform distribution is given as follows:

$$P(X = x) = \begin{cases} \frac{1}{b-a} & \text{for } a \le x \le b \\ 0 & \text{Otherwise} \end{cases} \tag{2.25}$$

3. Exponential Distribution – This is a continuous uniform distribution. This probability distribution is used to describe the time between events in a Poisson process. Exponential distribution is another special case of Gamma distribution with a fixed parameter of 1. This distribution is helpful in modelling of time until an event occurs.

   The PDF is given as follows:

$$f(x, \lambda) = \begin{cases} \lambda e^{-\lambda x} & \text{if } x \ge 0 \quad (\lambda > 0) \\ 0 & \text{if } x < 0 \end{cases} \tag{2.26}$$

   Here, x is a random variable and λ is called rate parameter. The mean and standard deviation of exponential distribution is given as β, where, $\beta = \frac{1}{\lambda}$.

3.b. Briefly explain about confusion matrix and ROC curve. (8M)

The Confusion Matrix

Confusion matrix is a simple table used to measure how well a classification model is performing. It compares the predictions made by the model with the actual results and shows where the model was right or wrong. This helps you understand where the model is making mistakes so you can improve it. It breaks down the predictions into four categories:

- **True Positive (TP):** The model correctly predicted a positive outcome i.e the actual outcome was positive.
- **True Negative (TN):** The model correctly predicted a negative outcome i.e the actual outcome was negative.
- **False Positive (FP):** The model incorrectly predicted a positive outcome i.e the actual outcome was negative. It is also known as a Type I error.
- **False Negative (FN):** The model incorrectly predicted a negative outcome i.e the actual outcome was positive. It is also known as a Type II error.

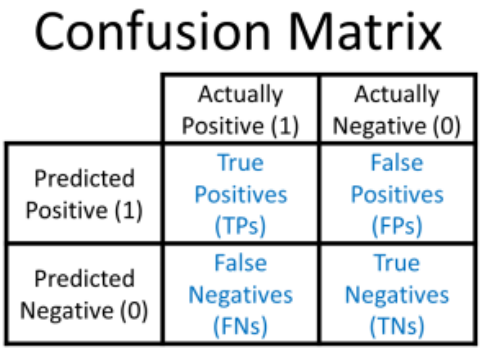


Fig. Confusion Matrix

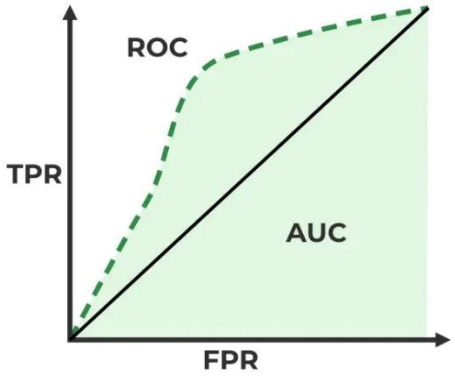


Fig. ROC Curve

The Receiver Operator Characteristic (ROC) Curve

AUC-ROC curve is a graph used to check how well a binary classification model works. It helps us to understand how well the model separates the positive cases like people with a disease from the negative cases like people without the disease at different threshold level. It is a plot of the percentage of true positives on the Y-axis against false positives on the X-axis.

It shows how good the model is at telling the difference between the two classes by plotting:

- **True Positive Rate (TPR):** how often the model correctly predicts the positive cases also known as **Sensitivity or Recall**.
- **False Positive Rate (FPR):** how often the model incorrectly predicts a negative case as positive.
- **Specificity:** measures the proportion of actual negatives that the model correctly identifies. It is calculated as 1 – FPR.

The higher the curve the better the model is at making correct predictions. AUC-ROC is effective when:

- The dataset is balanced and the model needs to be evaluated across all thresholds.

- False positives and false negatives are of similar importance.

These terms are derived from the confusion matrix which provides the following values:

- **True Positive (TP)**: Correctly predicted positive instances
- **True Negative (TN)**: Correctly predicted negative instances
- **False Positive (FP)**: Incorrectly predicted as positive
- **False Negative (FN)**: Incorrectly predicted as negative

Accuracy Metrics used in both Confusion Matrix and ROC Curve

1. **Accuracy:** It shows how many predictions the model got right out of all the predictions. It gives idea of overall performance but it can be misleading when one class is more dominant over the other. For example, a model that predicts the majority class correctly most of the time might have high accuracy but still fail to capture important details about other classes.
2. **Precision:** It focuses on the quality of the model's positive predictions. It tells us how many of the "positive" predictions were actually correct. It is important in situations where false positives need to be minimized such as detecting spam emails or fraud**.**
3. **Recall:** It measures how good the model is at predicting positives. It shows the proportion of true positives detected out of all the actual positive instances. High recall is essential when missing positive cases has significant consequences like in medical tests. It is also known as **Sensitivity** or **True Positive Rate**.
4. **F1-Score:** It combines precision and recall into a single metric to balance their trade-off. It is the harmonic mean of precision and recall. It provides a better sense of a model's overall performance particularly for imbalanced datasets. It is helpful when both false positives and false negatives are important though it assumes precision and recall are equally important but, in some situations, one might matter more than the other.
5. **Specificity:** It is another important metric in the evaluation of classification models particularly in binary classification. It measures the ability of a model to correctly identify negative instances. Specificity is also known as the **True Negative Rate**.
6. **False Positive Rate:** It shows how many of the actual negative cases were predicted as positive.

Metric	Formula
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision	$\frac{TP}{TP+FP}$
Recall/Sensitivity/True Positive Rate (TPR)	$\frac{TP}{TP+FN}$
False Positive Rate (FPR)	$\frac{FP}{FP+TN}$
Specificity/True Negative Rate (TNR)	$\frac{TN}{TN+FP}$ OR $1-FPR$
F1-Score/F-Measure	$\frac{2\times Precision\times Recall}{Precision+Recall}$ OR $\frac{2TP}{2TP+FP+FN}$

3.c. Let the data points be $\binom{2}{6}$ and $\binom{1}{7}$. Apply PCA and find the transformed data.(6M)

Example 2.12: Let the data points be $\begin{pmatrix}2\\6\end{pmatrix}$ and $\begin{pmatrix}1\\7\end{pmatrix}$. Apply PCA and find the transformed data. Again, apply the inverse and prove that PCA works.

Solution: One can combine two vectors into a matrix as follows:

The mean vector can be computed as Eq. (2.53) as follows:

$$\mu = \begin{pmatrix}\frac{2+1}{2}\\ \frac{6+7}{2}\end{pmatrix} = \begin{pmatrix}1.5\\6.5\end{pmatrix}$$

As part of PCA, the mean must be subtracted from the data to get the adjusted data:

$$x_1 = \begin{pmatrix}2-1.5\\6-6.5\end{pmatrix} = \begin{pmatrix}0.5\\-0.5\end{pmatrix}$$
$$x_2 = \begin{pmatrix}1-1.5\\7-6.5\end{pmatrix} = \begin{pmatrix}-0.5\\0.5\end{pmatrix}$$

One can find the covariance for these data vectors. The covariance can be obtained using Eq. (2.54):

$$m_1 = \begin{pmatrix}0.5\\-0.5\end{pmatrix}\begin{pmatrix}0.5 & -0.5\end{pmatrix} = \begin{pmatrix}0.25 & -0.25\\-0.25 & 0.25\end{pmatrix}$$
$$m_2 = \begin{pmatrix}-0.5\\0.5\end{pmatrix}\begin{pmatrix}-0.5 & 0.5\end{pmatrix} = \begin{pmatrix}0.25 & -0.25\\-0.25 & 0.25\end{pmatrix}$$

The final covariance matrix is obtained by adding these two matrices as:

$$C = \begin{pmatrix}0.5 & -0.5\\-0.5 & 0.5\end{pmatrix}$$

The eigen values and eigen vectors of matrix C can be obtained (left as an exercise) as $\lambda_1 = 1$, $\lambda_1 = 0$. The eigen vectors are $\begin{pmatrix}-1\\1\end{pmatrix}$ and $\begin{pmatrix}1\\1\end{pmatrix}$. The matrix A can be obtained by packing the eigen vector of these eigen values (after sorting it) of matrix C. For this problem, $A = \begin{pmatrix}-1 & 1\\1 & 1\end{pmatrix}$. The transpose of A, $A^T = \begin{pmatrix}-1 & 1\\1 & 1\end{pmatrix}$ is also the same matrix as it is an orthogonal matrix. The matrix can be normalized by diving each elements of the vector, by the norm of the vector to get:

$$A = \begin{pmatrix}-\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\end{pmatrix}$$

One can check that the PCA matrix A is orthogonal. A matrix is orthogonal is $A^{-1} = A$ and $AA^{-1} = I$.

$$AA^T = \begin{pmatrix}-\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\end{pmatrix}\begin{pmatrix}-\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\end{pmatrix}$$
$$= \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}$$

The transformed matrix y using Eq. (2.55) is given as:

$$y = A \times (x - m)$$

Recollect that $(x–m)$ is the adjusted matrix.

$$y = A(x-m) = \begin{pmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 0.5 & -0.5 \\ -0.5 & 0.5 \end{pmatrix}$$

$$= \begin{pmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix} \left(\text{for convenience } 0.5 = \frac{1}{2} \right)$$

$$= \begin{pmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 \end{pmatrix}$$

One can check the original matrix can be retrieved from this matrix as:

$$\{(A)^T \times y\} + m$$

$$x = A^T y + m = \begin{pmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 1.5 \\ 6.5 \end{pmatrix}$$

$$= \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix} + \begin{pmatrix} 1.5 \\ 6.5 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 6 & 7 \end{pmatrix}$$

Therefore, one can infer the original is obtained without any loss of information.

OR

4.a. Explain non-parametric density estimation. (6M)

Density Estimation

Let there be a set of observed values $x_1, x_2, \cdots, x_n$ from a larger set of data whose distribution is not known. Density estimation is the problem of estimating the density function from an observed data. The estimated density function, denoted as, $p(x)$ can be used to value directly for any unknown data, say x_t as $p(x_t)$. If its value is less than ε, then x_t is not an outlier or anomaly data. Else, it is categorized as an anomaly data.

There are two types of density estimation methods, namely parametric density estimation and non-parametric density estimation.

Non-parametric Density Estimation A non-parametric estimation can be generative or discriminative. Parzen window is a generative estimation method that finds $p(x \mid \Theta)$ as conditional density. Discriminative methods directly compute $p(\Theta \mid x)$ as posteriori probability. Parzen window and k-Nearest Neighbour (KNN) rule are examples of non-parametric density estimation. Let us discuss about them now.

Parzen Window Let there be 'n' samples, $X = \{x_1, x_2, \cdots, x_n\}$

The samples are drawn independently, called as identically independent distribution. Let R be the region that covers 'k' samples of total 'n' samples. Then, the probability density function is given as:

$$p = k/n \tag{2.38}$$

The estimate is given as:

$$p(x) = \frac{k/n}{V} \tag{2.39}$$

where, V is the volume of the region R. If R is the hypercube centered at x and h is the length of the hypercube, the volume V is h^2 for 2D square cube and h^3 for 3D cube.

The Parzen window is given as follows:

$$\varphi\left(\frac{x_i - x}{h}\right) = \begin{cases} 1 \text{ if } \dfrac{\left|x_{ik} - x_k\right|}{h} < \dfrac{1}{2} \\ 0 \text{ otherwise} \end{cases} \tag{2.40}$$

The window indicates if the sample is inside the region or not. The Parzen probability density function estimate using Eq. (2.40) is given as:

$$\begin{aligned} p(x) &= \frac{k/n}{V} \\ &= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{V_n}\varphi\left(\frac{x_i - x}{h}\right) \end{aligned} \tag{2.41}$$

This window can be replaced by any other function too. If Gaussian function is used, then it is called Gaussian density function.

KNN Estimation The KNN estimation is another non-parametric density estimation method. Here, the initial parameter k is determined and based on that k-neighbours are determined. The probability density function estimate is the average of the values that are returned by the neighbours.

4.b. Explain (i) Training (ii) Testing and (iii) Validation Sets and (iv) Unbalanced Datasets. (8M)

2.2.2 Training, Testing, and Validation Sets

We now need three sets of data: the training set to actually train the algorithm, the validation set to keep track of how well it is doing as it learns, and the test set to produce the final results. This is becoming expensive in data, especially since for supervised learning it all has to have target values attached (and even for unsupervised learning, the validation and test sets need targets so that you have something to compare to), and it is not always easy to get accurate labels (which may well be why you want to learn about the data). The area of semi-supervised learning attempts to deal with this need for large amounts of labelled data; see the Further Reading section for some references.

Clearly, each algorithm is going to need some reasonable amount of data to learn from (precise needs vary, but the more data the algorithm sees, the more likely it is to have seen examples of each possible type of input, although more data also increases the computational time to learn). However, the same argument can be used to argue that the validation and

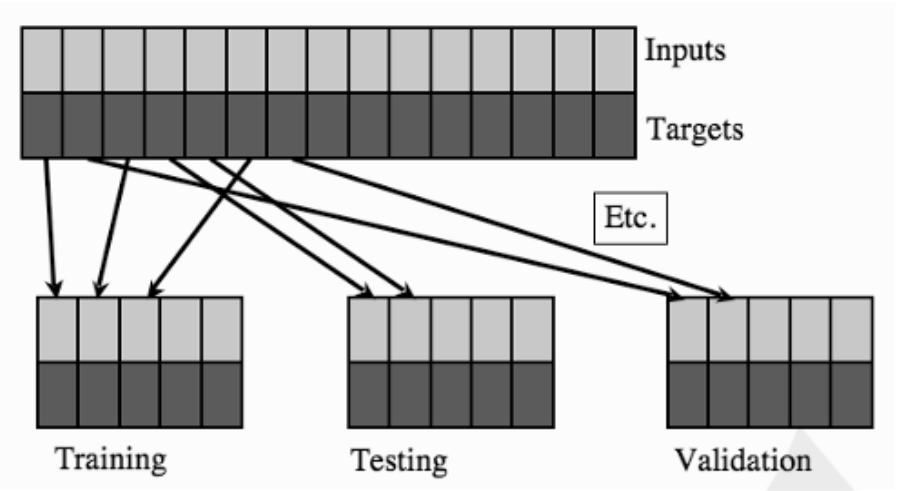


FIGURE 2.6 The dataset is split into different sets, some for training, some for validation, and some for testing.

test sets should also be reasonably large. Generally, the exact proportion of training to testing to validation data is up to you, but it is typical to do something like 50:25:25 if you have plenty of data, and 60:20:20 if you don't. How you do the splitting can also matter. Many datasets are presented with the first set of datapoints being in class 1, the next in class 2, and so on. If you pick the first few points to be the training set, the next the test set, etc., then the results are going to be pretty bad, since the training did not see all the classes. This can be dealt with by randomly reordering the data first, or by assigning each datapoint randomly to one of the sets, as is shown in Figure 2.6.

If you are really short of training data, so that if you have a separate validation set there is a worry that the algorithm won't be sufficiently trained; then it is possible to perform leave-some-out, multi-fold cross-validation. The idea is shown in Figure 2.7. The dataset is randomly partitioned into K subsets, and one subset is used as a validation set, while the algorithm is trained on all of the others. A different subset is then left out and a new model is trained on that subset, repeating the same process for all of the different subsets. Finally, the model that produced the lowest validation error is tested and used. We've traded off data for computation time, since we've had to train K different models instead of just one. In the most extreme case of this there is leave-one-out cross-validation, where the algorithm is validated on just one piece of data, training on all of the rest.

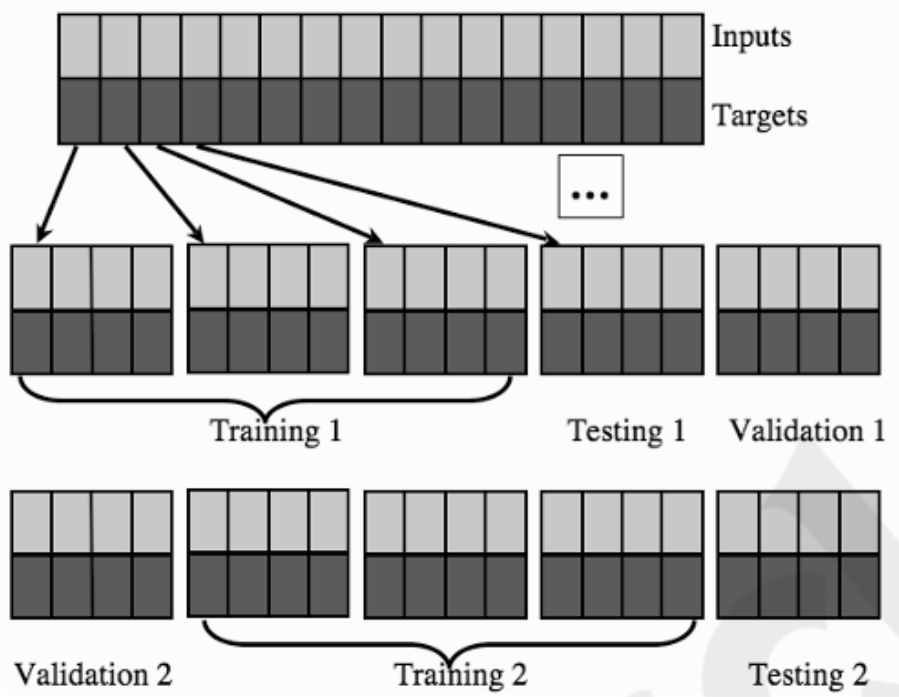


FIGURE 2.7 Leave-some-out, multi-fold cross-validation gets around the problem of data shortage by training many models. It works by splitting the data into sets, training a model on most sets and holding one out for validation (and another for testing). Different models are trained with different sets being held out.

2.2.6 Unbalanced Datasets

Note that for the accuracy we have implicitly assumed that there are the same number of positive and negative examples in the dataset (which is known as a balanced dataset). However, this is often not true (this can potentially cause problems for the learners as well, as we shall see later in the book). In the case where it is not, we can compute the balanced accuracy as the sum of sensitivity and specificity divided by 2. However, a more correct measure is Matthew's Correlation Coefficient, which is computed as:

$$MCC = \frac{\#TP \times \#TN - \#FP \times \#FN}{\sqrt{(\#TP + \#FP)(\#TP + \#FN)(\#TN + \#FP)(\#TN + \#FN)}} \tag{2.9}$$

If any of the brackets in the denominator are 0, then the whole of the denominator is set to 1. This provides a balanced accuracy computation.

4.c. Find the LU decomposition of the given matrix. $\mathbf{A} = \begin{bmatrix} 1 & 2 & 4 \\ 3 & 3 & 2 \\ 3 & 4 & 2 \end{bmatrix}$ **(6M)**

Example 2.9: Find *LU* decomposition of the given matrix:

$$A = \begin{pmatrix} 1 & 2 & 4 \\ 3 & 3 & 2 \\ 3 & 4 & 2 \end{pmatrix}$$

Solution: First, augment an identity matrix and apply Gaussian elimination. The steps are as shown in:

$$\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\begin{bmatrix}1&2&4\\3&3&2\\3&4&2\end{bmatrix}$$ Initial Matrix

$$\begin{bmatrix}1&0&0\\3&1&0\\0&0&1\end{bmatrix}\begin{bmatrix}1&2&4\\0&-3&-10\\3&4&2\end{bmatrix}$$ $R_2 = R_2 - 3R_1$

$$\begin{bmatrix}1&0&0\\3&1&0\\3&0&1\end{bmatrix}\begin{bmatrix}1&2&4\\0&-3&-10\\0&-2&-10\end{bmatrix}$$ $R_3 = R_3 - 3R_1$

$$\begin{bmatrix}1&0&0\\3&1&0\\3&\frac{2}{3}&1\end{bmatrix}\begin{bmatrix}1&2&4\\0&-3&-10\\0&0&\frac{-10}{3}\end{bmatrix}$$ $R_3 = R_3 - \frac{2}{3}R_2$

Now, it can be observed that the first matrix is L as it is the lower triangular matrix whose values are the determiners used in the reduction of equations above such as 3, 3 and 2/3. The second matrix is U, the upper triangular matrix whose values are the values of the reduced matrix because of Gaussian elimination.

$$L = \begin{pmatrix}1&0&0\\3&1&0\\3&\frac{2}{3}&1\end{pmatrix} \text{ and } U = \begin{pmatrix}1&2&4\\0&-3&-10\\0&0&-\frac{10}{3}\end{pmatrix}.$$

MODULE -3

5.a. Explain about linear regression. (6M)

5.3 INTRODUCTION TO LINEAR REGRESSION

In the simplest form, the linear regression model can be created by fitting a line among the scattered data points. The line is of the form given in Eq. (5.2).

$$y = a_0 + a_1 \times x + e \tag{5.2}$$

Here, a_0 is the intercept which represents the bias and a_1 represents the slope of the line. These are called regression coefficients. e is the error in prediction.

The assumptions of linear regression are listed as follows:

1. The observations (y) are random and are mutually independent.
2. The difference between the predicted and true values is called an error. The error is also mutually independent with the same distributions such as normal distribution with zero mean and constant variables.
3. The distribution of the error term is independent of the joint distribution of explanatory variables.
4. The unknown parameters of the regression models are constants.

The idea of linear regression is based on Ordinary Least Square (OLS) approach. This method is also known as ordinary least squares method. In this method, the data points are modelled using a straight line. Any arbitrarily drawn line is not an optimal line. In Figure 5.4, three data points and their errors (e_1, e_2, e_3) are shown. The vertical distance between each point and the line (predicted by the approximate line equation $y = a_0 + a_1x$) is called an error. These individual errors are added to compute the total error of the predicted line. This is called sum of residuals. The squares of the individual errors can also be computed and added to give a sum of squared error. The line with the lowest sum of squared error is called line of best fit.

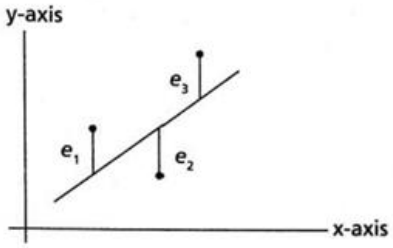


Figure 5.4: Data Points and their Errors

In another words, OLS is an optimization technique where the difference between the data points and the line is optimized.

Mathematically, based on Eq. (5.2), the line equations for points (x_1, x_2, ..., x_n) are:

$$y_1 = (a_0 + a_1x_1) + e_1$$
$$y_2 = (a_0 + a_1x_2) + e_2$$
$$\vdots$$
$$y_n = (a_0 + a_1x_n) + e_n \quad (5.3)$$

In general, the error is given as: $e_i = y_i - (a_0 + a_1x_i)$ (5.4)

This can be extended into the set of equations as shown in Eq. (5.3).

5.b. Consider the training dataset given in the below table. Use weighted k-NN to determine the class. Given the test instance (7.6, 60, 8) [Assign k=3]. (8M)

Sl. No.	CGPA	Assessment	Project Submitted	Result
1	9.2	85	8	Pass
2	8	80	7	Pass
3	8.5	81	8	Pass
4	6	45	5	Fail
5	6.5	50	4	Fail
6	8.2	72	7	Pass
7	5.8	38	5	Fail
8	8.9	91	9	Pass

Example 4.2: Consider the same training dataset given in Table 4.1. Use Weighted k-NN and determine the class.

Solution:

Step 1: Given a test instance (7.6, 60, 8) and a set of classes {Pass, Fail}, use the training dataset to classify the test instance using Euclidean distance and weighting function.

Assign $k = 3$. The distance calculation is shown in Table 4.5.

Table 4.5: Euclidean Distance

S.No.	CGPA	Assessment	Project Submitted	Result	Euclidean Distance
1.	9.2	85	8	Pass	$\sqrt{(9.2-7.6)^2+(85-60)^2+(8-8)^2}$ $= 25.05115$
2.	8	80	7	Pass	$\sqrt{(8-7.6)^2+(80-60)^2+(7-8)^2}$ $= 20.02898$
3.	8.5	81	8	Pass	$\sqrt{(8.5-7.6)^2+(81-60)^2+(8-8)^2}$ $= 21.01928$
4.	6	45	5	Fail	$\sqrt{(6-7.6)^2+(45-60)^2+(5-8)^2}$ $= 15.38051$
5.	6.5	50	4	Fail	$\sqrt{(6.5-7.6)^2+(50-60)^2+(4-8)^2}$ $= 10.82636$
6.	8.2	72	7	Pass	$\sqrt{(8.2-7.6)^2+(72-60)^2+(7-8)^2}$ $= 12.05653$
7.	5.8	38	5	Fail	$\sqrt{(5.8-7.6)^2+(38-60)^2+(5-8)^2}$ $= 22.27644$
8.	8.9	91	9	Pass	$\sqrt{(8.9-7.6)^2+(91-60)^2+(9-8)^2}$ $= 31.04336$

Step 2: Sort the distances in the ascending order and select the first 3 nearest training data instances to the test instance. The selected nearest neighbors are shown in Table 4.6.

Table 4.6: Nearest Neighbors

Instance	Euclidean Distance	Class
4	15.38051	Fail
5	10.82636	Fail
6	12.05653	Pass

Step 3: Predict the class of the test instance by weighted voting technique from the 3 selected nearest instances.

- Compute the inverse of each distance of the 3 selected nearest instances as shown in Table 4.7.

Table 4.7: Inverse Distance

Instance	Euclidean Distance	Inverse Distance	Class
4	15.38051	0.06502	Fail
5	10.82636	0.092370	Fail
6	12.05653	0.08294	Pass

- Find the sum of the inverses.

  Sum = 0.06502 + 0.092370 + 0.08294 = 0.24033

- Compute the weight by dividing each inverse distance by the sum as shown in Table 4.8.

Table 4.8: Weight Calculation

Instance	Euclidean Distance	Inverse Distance	Weight = Inverse distance/Sum	Class
4	15.38051	0.06502	0.270545	Fail
5	10.82636	0.092370	0.384347	Fail
6	12.05653	0.08294	0.345109	Pass

- Add the weights of the same class.

  Fail = 0.270545 + 0.384347 = 0.654892

  Pass = 0.345109

- Predict the class by choosing the class with the maximum vote.

The class is predicted as '**Fail**'.

5.c. Explain the differences between Instance-based Learning and Model-based Learning. (6M)

4.1.1 Differences Between Instance- and Model-based Learning

An *instance* is an entity or an example in the training dataset. It is described by a set of features or attributes. One attribute describes the class label or category of an instance. *Instance-based methods* learn or predict the class label of a test instance only when a new instance is given for classification and until then it delays the processing of the training dataset.

It is also referred to as *lazy learning* methods since it does not generalize any model from the training dataset but just keeps the training dataset as a knowledge base until a new instance is given. In contrast, *model-based learning*, generally referred to as *eager learning*, tries to generalize the training data to a model before receiving test instances. Model-based machine learning describes all assumptions about the problem domain in the form of a model. These algorithms basically learn in two phases, called training phase and testing phase. In training phase, a model is built from the training dataset and is used to classify a test instance during the testing phase. Some examples of models constructed are decision trees, neural networks and Support Vector Machines (SVM), etc.

The differences between Instance-based Learning and Model-based Learning are listed in Table 4.1.

Table 4.1: Differences between Instance-based Learning and Model-based Learning

Instance-based Learning	Model-based Learning
Lazy Learners	Eager Learners
Processing of training instances is done only during testing phase	Processing of training instances is done during training phase

(Continued)

Instance-based Learning	Model-based Learning
No model is built with the training instances before it receives a test instance	Generalizes a model with the training instances before it receives a test instance
Predicts the class of the test instance directly from the training data	Predicts the class of the test instance from the model built
Slow in testing phase	Fast in testing phase
Learns by making many local approximations	Learns by creating global approximation

Instance-based learning also comes under the category of memory-based models which normally compare the given test instance with the trained instances that are stored in memory. Memory-based models classify a test instance by checking the similarity with the training instances.

Some examples of Instance-based learning algorithms are:

1. *k*-Nearest Neighbor (*k*-NN)
2. Variants of Nearest Neighbor learning
3. Locally Weighted Regression
4. Learning Vector Quantization (LVQ)
5. Self-Organizing Map (SOM)
6. Radial Basis Function (RBF) networks

In this chapter, we will discuss about certain instance-based learning algorithms such as *k*-Nearest Neighbor (*k*-NN), Variants of Nearest Neighbor learning, and Locally Weighted Regression learning.

Self-Organizing Map (SOM) and Radial Basis Function (RBF) networks are discussed along with the concepts of artificial neural networks discussed in Chapter 10 since they could be referred only after the understanding of neural networks.

These instance-based methods have serious limitations about the range of feature values taken. Moreover, they are sensitive to irrelevant and correlated features leading to misclassification of instances.

OR

6.a. Explain about logistic regression. (6M)

Definition: Logistic Regression is a supervised learning algorithm used for classification problems, particularly binary classification, where the output is a categorical variable with two possible outcomes (e.g., yes/no, pass/fail, spam/not spam). It can be viewed as an extension of linear regression.

Purpose: Logistic Regression predicts the probability of a categorical outcome and maps the prediction to a value between 0 and 1. It works well when the dependent variable is binary.

Core Concept: Logistic Regression models the probability of a particular response variable. For instance, if the predicted probability of an email being spam is 0.7, there is a 70% chance the email is spam.

Applications:

- Email classification: Is the email spam or not?
- Student admission prediction: Should a student be admitted or not based on scores?
- Exam result classification: Will the student pass or fail based on marks?

Challenges:

- Linear regression can predict values outside the range of 0 to 1, which is unsuitable for probabilities.
- Logistic Regression overcomes this by using a sigmoid function to map values to the range [0, 1].

Sigmoid Function: The sigmoid function (also called the logit function) is used to map any real number to the range [0, 1]. It is an S-shaped curve and is mathematically represented as:

$$logit(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$$

Here, x is the independent variable and e is the Euler number ($e \approx 2.71828$).

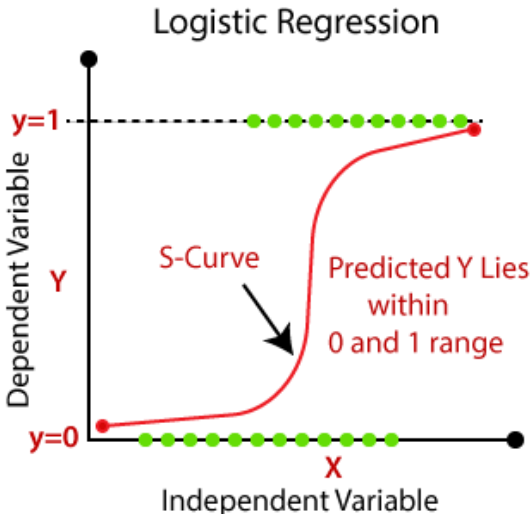


Fig: Logistic Regression Curve/Sigmoid Curve

6.b. Consider the data provided in the below table and fit it using the second order polynomial. (8M)

X	Y
1	1
2	4
3	9
4	15

Table 5.8: Sample Data

x	y
1	1
2	4
3	9
4	15

Solution: For applying polynomial regression, computation is done as shown in Table 5.9. Here, the order is 2 and the sample i ranges from 1 to 4.

Table 5.9: Computation Table

x_i	y_i	$x_i y_i$	x_i^2	$x_i^2 y$	x_i^3	x_i^4
1	1	1	1	1	1	1
2	4	8	4	16	8	16
3	9	27	9	81	27	81
4	15	60	16	240	64	256
$\Sigma x_i = 10$	$\Sigma y_i = 29$	$\Sigma x_i y_i = 96$	$\Sigma x_i^2 = 30$	$\Sigma x_i^2 y_i = 338$	$\Sigma x_i^3 = 100$	$\Sigma x_i^4 = 354$

It can be noted that, $N = 4$, $\Sigma y_i = 29$, $\Sigma x_i y_i = 96$, $\Sigma x_i^2 y_i = 338$. When the order is 2, the matrix using Eq. (5.28) is given as follows:

$$\begin{bmatrix} 4 & 10 & 30 \\ 10 & 30 & 100 \\ 30 & 100 & 354 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 29 \\ 96 \\ 338 \end{bmatrix}$$

Therefore, using Eq. (5.29), one can get coefficients as:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 4 & 10 & 30 \\ 10 & 30 & 100 \\ 30 & 100 & 354 \end{bmatrix}^{-1} \times \begin{bmatrix} 29 \\ 96 \\ 338 \end{bmatrix} = \begin{pmatrix} -0.75 \\ 0.95 \\ 0.75 \end{pmatrix}.$$

This leads to the regression equation using Eq. (5.26) as:

$$y = -0.75 + 0.95\,x + 0.75\,x^2$$

6.c. Explain nearest centroid classifier with an example. (6M)

4.4 NEAREST CENTROID CLASSIFIER

A simple alternative to *k*-NN classifiers for similarity-based classification is the Nearest Centroid Classifier. It is a simple classifier and also called as Mean Difference classifier. The idea of this classifier is to classify a test instance to the class whose centroid/mean is closest to that instance.

Algorithm 4.3: Nearest Centroid Classifier

Inputs: Training dataset *T*, Distance metric *d*, Test instance *t*

Output: Predicted class or category

1. Compute the mean/centroid of each class.
2. Compute the distance between the test instance and mean/centroid of each class (Euclidean Distance).
3. Predict the class by choosing the class with the smaller distance.

Example 4.3: Consider the sample data shown in Table 4.9 with two features x and y. The target classes are 'A' or 'B'. Predict the class using Nearest Centroid Classifier.

Table 4.9: Sample Data

X	Y	Class
3	1	A
5	2	A
4	3	A
7	6	B
6	7	B
8	5	B

Solution:

Step 1: Compute the mean/centroid of each class. In this example there are two classes called 'A' and 'B'.

Centroid of class 'A' = (3 + 5 + 4, 1 + 2 + 3)/3 = (12, 6)/3 = (4, 2)

Centroid of class 'B' = (7 + 6 + 8, 6 + 7 + 5)/3 = (21, 18)/3 = (7, 6)

Now given a test instance (6, 5), we can predict the class.

Step 2: Calculate the Euclidean distance between test instance (6, 5) and each of the centroid.

$$\text{Euc\_Dist}[(6,5);(4,2)] = \sqrt{(6-4)^2+(5-2)^2} = \sqrt{13} = 3.6$$

$$\text{Euc\_Dist}[(6,5);(7,6)] = \sqrt{(6-7)^2+(5-6)^2} = \sqrt{2} = 1.414$$

The test instance has smaller distance to class B. Hence, the class of this test instance is predicted as 'B'.

MODULE – 4

7.a. Explain pre-pruning and post-pruning. Compare both methods. (6M)

Pre-Pruning (Early Stopping)

Sometimes, the growth of the decision tree can be stopped before it gets too complex, this is called pre-pruning. It is important to prevent the overfitting of the training data, which results in a poor performance when exposed to new data. Pre-pruning results in a simpler tree that is less likely to overfit the training facts. Some common pre-pruning techniques include:

- **Maximum Depth**: It limits the maximum level of depth in a decision tree.
- **Minimum Samples per Leaf**: Set a minimum threshold for the number of samples in each leaf node.
- **Minimum Samples per Split**: Specify the minimal number of samples needed to break up a node.
- **Maximum Features:** Restrict the quantity of features considered for splitting.

Post-Pruning (Reducing Nodes)

After the tree is fully grown, post-pruning involves removing branches or nodes to improve the model's ability to generalize. Post-pruning simplifies the tree while preserving its accuracy. Some common post-pruning techniques include:

- **Cost-Complexity Pruning (CCP)**: This method assigns a price to each subtree primarily based on its accuracy and complexity, then selects the subtree with the lowest fee.
- **Reduced Error Pruning**: Removes branches that do not significantly affect the overall accuracy.
- **Minimum Impurity Decrease**: Prunes nodes if the decrease in impurity (Gini impurity or entropy) is beneath a certain threshold.
- **Minimum Leaf Size**: Removes leaf nodes with fewer samples than a specified threshold.

Comparison: Pre-Pruning vs. Post-Pruning

Aspect	Pre-Pruning	Post-Pruning
Tree Growth	Tree stops growing early based on conditions	Tree grows fully, and then unnecessary branches are pruned
Risk	Risk of underfitting if stopping criteria are too strict	Risk of overfitting initially, but pruning reduces complexity
Computation	Faster, as the tree is not grown fully	Slower, requires growing and pruning the tree
Control	Requires careful tuning of stopping criteria	More control over pruning decisions based on validation results
Interpretability	Smaller and simpler tree from the start	Tree is simplified after pruning, balancing complexity and accuracy

7.b. Assess a student's performance during his course of study and predict whether a student will get a job offer or not in his final year of the course. The training dataset *T* consists of 10 data instances with attributes such as 'CGPA', 'Interactiveness', 'Practical Knowledge' and 'Communication Skills'. The target class attribute is 'Job Offer'. (10M)

Sl. No.	CGPA	Interactiveness	Practical Knowledge	Communication Skills	Job Offer
1	≥ 9	Yes	Very Good	Good	Yes
2	≥ 8	No	Good	Moderate	Yes
3	≥ 9	No	Average	Poor	No
4	< 8	No	Average	Good	No
5	≥ 8	Yes	Good	Moderate	Yes
6	≥ 9	Yes	Good	Moderate	Yes
7	< 8	Yes	Good	Poor	No
8	≥ 9	No	Very Good	Good	Yes
9	≥ 8	Yes	Good	Good	Yes
10	≥ 8	Yes	Average	Good	Yes

Solution:

Step 1:

Calculate the Entropy for the target class 'Job Offer'.

Entropy_Info(Target Attribute = Job Offer) = Entropy_Info(7, 3) =

$$= -\left[\frac{7}{10}\log_2\frac{7}{10} + \frac{3}{10}\log_2\frac{3}{10}\right] = -(-0.3599 + -0.5208) = 0.8807$$

Iteration 1:

Step 2:

Calculate the Entropy_Info and Gain(Information_Gain) for each of the attribute in the training dataset.

Table 6.4 shows the number of data instances classified with Job Offer as Yes or No for the attribute CGPA.

Table 6.4: Entropy Information for CGPA

CGPA	Job Offer = Yes	Job Offer = No	Total	Entropy
≥9	3	1	4	
≥8	4	0	4	0
<8	0	2	2	0

Entropy_Info(T, CGPA)

$$= \frac{4}{10}\left[-\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4}\right] + \frac{4}{10}\left[-\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4}\right] + \frac{2}{10}\left[-\frac{0}{2}\log_2\frac{0}{2} - \frac{2}{2}\log_2\frac{2}{2}\right]$$

$$= \frac{4}{10}(0.3111 + 0.4997) + 0 + 0$$

$$= 0.3243$$

Gain (CGPA) = 0.8807 − 0.3243

= 0.5564

Table 6.5 shows the number of data instances classified with Job Offer as Yes or No for the attribute Interactiveness.

Table 6.5: Entropy Information for Interactiveness

Interactiveness	Job Offer = Yes	Job Offer = No	Total	Entropy
YES	5	1	6	
NO	2	2	4	

$$\text{Entropy\_Info}(T, \text{Interactiveness}) = \frac{6}{10}\left[-\frac{5}{6}\log_2\frac{5}{6} - \frac{1}{6}\log_2\frac{1}{6}\right] + \frac{4}{10}\left[-\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4}\right]$$

$$= \frac{6}{10}(0.2191 + 0.4306) + \frac{4}{10}(0.4997 + 0.4997)$$

$$= 0.3898 + 0.3998 = 0.7896$$

Gain(Interactiveness) = 0.8807 − 0.7896

= 0.0911

Table 6.6 shows the number of data instances classified with Job Offer as Yes or No for the attribute Practical Knowledge.

Table 6.6: Entropy Information for Practical Knowledge

Practical Knowledge	Job Offer = Yes	Job Offer = No	Total	Entropy
Very Good	2	0	2	**0**
Average	1	2	3	
Good	4	1	5	

Entropy_Info(T, Practical Knowledge)

$$= \frac{2}{10}\left[-\frac{2}{2}\log_2\frac{2}{2} - \frac{0}{2}\log_2\frac{0}{2}\right] + \frac{3}{10}\left[-\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3}\right] + \frac{5}{10}\left[-\frac{4}{5}\log_2\frac{4}{5} - \frac{1}{5}\log_2\frac{1}{5}\right]$$

$$= \frac{2}{10}(0) + \frac{3}{10}(0.5280 + 0.3897) + \frac{5}{10}(0.2574 + 0.4641)$$

$$= 0 + 0.2753 + 0.3608$$

$$= 0.6361$$

Gain(Practical Knowledge) = 0.8807 – 0.6361

= 0.2446

Table 6.7 shows the number of data instances classified with Job Offer as Yes or No for the attribute Communication Skills.

Table 6.7: Entropy Information for Communication Skills

Communication Skills	Job Offer = Yes	Job Offer = No	Total
Good	4	1	5
Moderate	3	0	3
Poor	0	2	2

Entropy_Info(T, Communication Skills)

$$= \frac{5}{10}\left[-\frac{4}{5}\log_2\frac{4}{5} - \frac{1}{5}\log_2\frac{1}{5}\right] + \frac{3}{10}\left[-\frac{3}{3}\log_2\frac{3}{3} - \frac{0}{3}\log_2\frac{0}{3}\right] + \frac{2}{10}\left[-\frac{0}{2}\log_2\frac{0}{2} - \frac{2}{2}\log_2\frac{2}{2}\right]$$

$$= \frac{5}{10}(0.5280 + 0.3897) + \frac{3}{10}(0) + \frac{2}{10}(0)$$

$$= 0.3609$$

Gain(Communication Skills) = 0.8813 – 0.36096

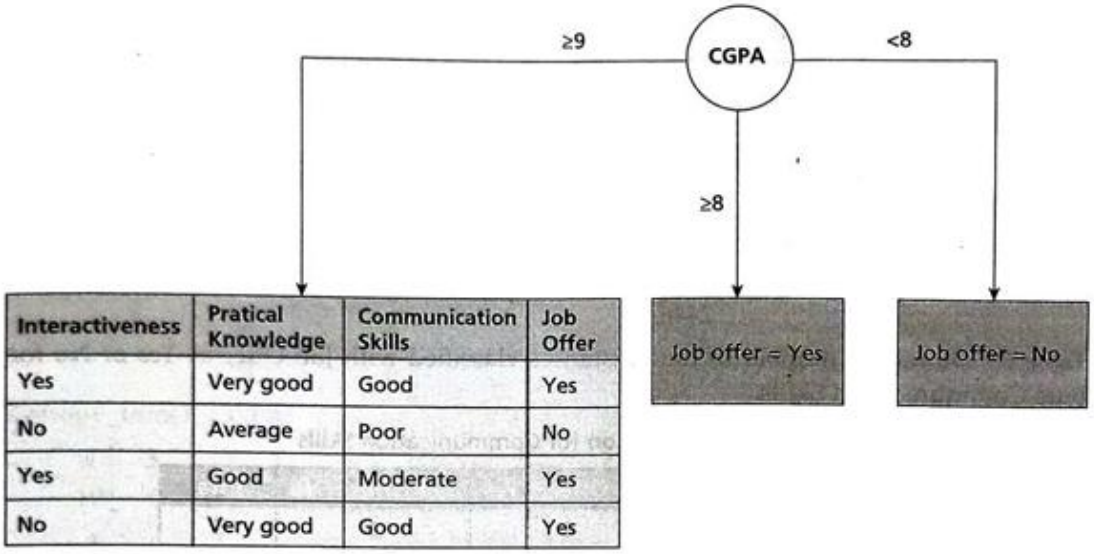
= 0.5203

The Gain calculated for all the attributes is shown in Table 6.8:

Table 6.8: Gain

Attributes	Gain
CGPA	**0.5564**
Interactiveness	0.0911
Practical Knowledge	0.2246
Communication Skills	0.5203

The best split attribute is CGPA since it has the maximum gain. So, we choose CGPA as the root node. There are three distinct values for CGPA with outcomes ≥9, ≥8 and <8. The entropy value is 0 for ≥8 and <8 with all instances classified as Job Offer = Yes for ≥8 and Job Offer = No for <8. Hence, both ≥8 and <8 end up in a leaf node. The tree grows with the subset of instances with CGPA ≥9 as shown in Figure 6.3.



Interactiveness	Pratical Knowledge	Communication Skills	Job Offer
Yes	Very good	Good	Yes
No	Average	Poor	No
Yes	Good	Moderate	Yes
No	Very good	Good	Yes

Figure 6.3: Decision Tree After Iteration 1

Now, continue the same process for the subset of data instances branched with CGPA ≥ 9.

Iteration 2:

In this iteration, the same process of computing the Entropy_Info and Gain are repeated with the subset of training set. The subset consists of 4 data instances as shown in the above Figure 6.3.

Entropy_Info(T) = Entropy_Info(3, 1) =

$$= -\left[\frac{3}{4}\log_2\frac{3}{4} + \frac{1}{4}\log_2\frac{1}{4}\right]$$

$$= -(-0.3111 + -0.4997)$$

$$= 0.8108$$

$$\text{Entropy\_Info}(T, \text{Interactiveness}) = \frac{2}{4}\left[-\frac{2}{2}\log_2\frac{2}{2} - \frac{0}{2}\log_2\frac{0}{2}\right] + \frac{2}{4}\left[-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}\right]$$

$$= 0 + 0.4997$$

$$\text{Gain(Interactiveness)} = 0.8108 - 0.4997$$

$$= 0.3111$$

Entropy_Info(T, Practical Knowledge)

$$= \frac{2}{4}\left[-\frac{2}{2}\log_2\frac{2}{2} - \frac{0}{2}\log_2\frac{0}{2}\right] + \frac{1}{4}\left[-\frac{0}{1}\log_2\frac{0}{1} - \frac{1}{1}\log_2\frac{1}{1}\right] + \frac{1}{4}\left[-\frac{1}{1}\log_2\frac{1}{1} - \frac{0}{1}\log_2\frac{0}{1}\right]$$

$$= 0$$

Gain(Practical Knowledge) = 0.8108

Entropy_Info(T, Communication Skills)

$$= \frac{2}{4}\left[-\frac{2}{2}\log_2\frac{2}{2} - \frac{0}{2}\log_2\frac{0}{2}\right] + \frac{1}{4}\left[-\frac{0}{1}\log_2\frac{0}{1} - \frac{1}{1}\log_2\frac{1}{1}\right] + \frac{1}{4}\left[-\frac{1}{1}\log_2\frac{1}{1} - \frac{0}{1}\log_2\frac{0}{1}\right]$$

$= 0$

Gain(Communication Skills) = 0.8108

The gain calculated for all the attributes is shown in Table 6.9.

Table 6.9: Total Gain

Attributes	Gain
Interactiveness	0.3111
Practical Knowledge	**0.8108**
Communication Skills	**0.8108**

Here, both the attributes 'Practical Knowledge' and 'Communication Skills' have the same Gain. So, we can either construct the decision tree using 'Practical Knowledge' or 'Communication Skills'. The final decision tree is shown in Figure 6.4.

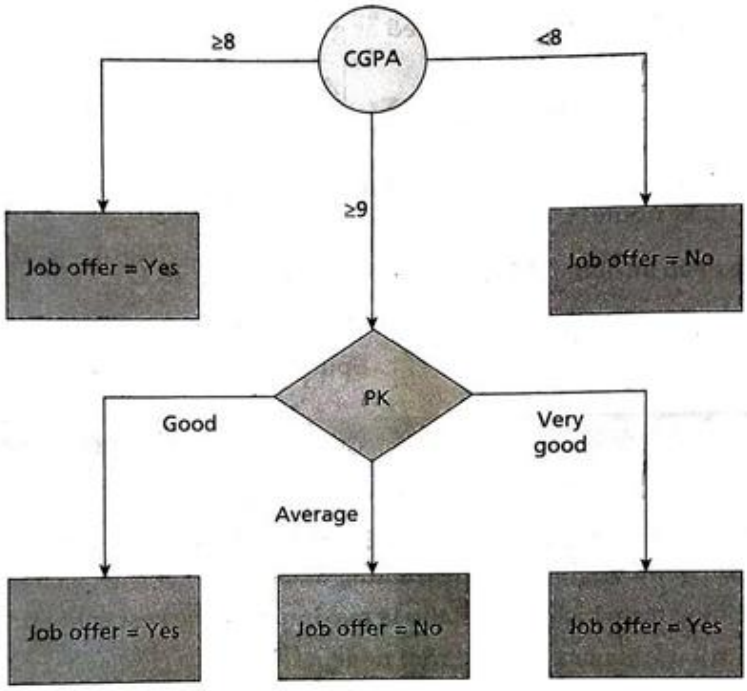


Figure 6.4: Final Decision Tree

7.c. Explain Entropy and Gini Index. (4M)

Entropy

Entropy is the measure of randomness in data. Randomness signifies the heterogeneity of labels. Decision trees split the data in manner that leads to decrease in entropy. Thus Decision Trees aim to divide the data with heterogenous labels into subsets/sub-regions of data with homogenous labels. Thus with each division level of homogeneity increases and entropy decreases. In fact entropy is the cost function that decision trees employ as basis of splitting the data, if the the split leads to decrease in entropy then it's carried out else not.

For a dataset with 4 labels- say a, b, c, d – with probability of occurrence of each label being p, q, r, s respectively, then the entropy of the data would be given by the following equation:

$$E = -P * \log(p) - q * \log(q) - r * \log(r) - s * \log(s)$$

For a dataset with n classes, the formula would be:

$$E = -\sum_{i=1}^{n} p * \log(p)$$

Where p is the probability of occurrence of each class.

Gini Impurity

According to Wikipedia, *'Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labelled if it was randomly labelled according to the distribution of labels in the subset.'*

Like entropy, Gini Impurity too is a measure of randomness of data. Randomness signifies the heterogeneity of labels. Decision trees split the data in manner that leads to decrease in Gini Impurity. Thus Decision Trees aim to divide the data with heterogenous labels into subsets/sub-regions of data with homogenous labels. Thus with each division level of homogeneity increases and Gini Impurity decreases. In fact Gini Impurity is the cost function that decision trees employ as basis of splitting the data, if the the split leads to decrease in Gini Impurity then it's carried out else not.

Higher the GINI value, higher is the homogeneity of the data instances.

Gini_Index(T) is computed as given in Eq. (6.13).

$$\text{Gini\_Index}(T) = 1 - \sum_{i=1}^{m} P_i^2 \qquad (6.13)$$

where,

P_i be the probability that a data instance or a tuple 'd' belongs to class C_i. It is computed as:

P_i = |No. of data instances belonging to class i|/|Total no of data instances in the training dataset T|

GINI Index assumes a binary split on each attribute, therefore, every attribute is considered as a binary attribute which splits the data instances into two subsets S_1 and S_2.

OR

8.a. Using the above Table – 7.b., assess a student's performance using Naïve Bayes' algorithm with the dataset. Predict whether a student gets a job offer or not in his final year of the course. (10M)

Example 8.2: Assess a student's performance using Naïve Bayes algorithm with the dataset provided in Table 8.1. Predict whether a student gets a job offer or not in his final year of the course.

Table 8.1: Training Dataset

S.No.	CGPA	Interactiveness	Practical Knowledge	Communication Skills	Job Offer
1.	≥9	Yes	Very good	Good	Yes
2.	≥8	No	Good	Moderate	Yes
3.	≥9	No	Average	Poor	No
4.	<8	No	Average	Good	No
5.	≥8	Yes	Good	Moderate	Yes
6.	≥9	Yes	Good	Moderate	Yes
7.	<8	Yes	Good	Poor	No
8.	≥9	No	Very good	Good	Yes
9.	≥8	Yes	Good	Good	Yes
10.	≥8	Yes	Average	Good	Yes

Solution: The training dataset T consists of 10 data instances with attributes such as 'CGPA', 'Interactiveness', 'Practical Knowledge' and 'Communication Skills' as shown in Table 8.1. The target variable is Job Offer which is classified as Yes or No for a candidate student.

Step 1: Compute the prior probability for the target feature 'Job Offer'. The target feature 'Job Offer' has two classes, 'Yes' and 'No'. It is a binary classification problem. Given a student instance, we need to classify whether 'Job Offer = Yes' or 'Job Offer = No'.

From the training dataset, we observe that the frequency or the number of instances with 'Job Offer = Yes' is 7 and 'Job Offer = No' is 3.

The prior probability for the target feature is calculated by dividing the number of instances belonging to a particular target class by the total number of instances.

Hence, the prior probability for 'Job Offer = Yes' is 7/10 and 'Job Offer = No' is 3/10 as shown in Table 8.2.

Table 8.2: Frequency Matrix and Prior Probability of Job Offer

Job Offer Classes	No. of Instances	Probability Value
Yes	7	P (Job Offer = Yes) = 7/10
No	3	P (Job Offer = No) = 3/10

Step 2: Compute Frequency matrix and Likelihood Probability for each of the feature.

Step 2(a): Feature – CGPA

Table 8.3 shows the frequency matrix for the feature CGPA.

Table 8.3: Frequency Matrix of CGPA

CGPA	Job Offer = Yes	Job Offer = No
≥9	3	1
≥8	4	0
<8	0	2
Total	7	3

Table 8.4 shows how the likelihood probability is calculated for CGPA using conditional probability.

Table 8.4: Likelihood Probability of CGPA

CGPA	P (Job Offer = Yes)	P (Job Offer = No)
≥9	P (CGPA ≥9 | Job Offer = Yes) = 3/7	P (CGPA ≥9 | Job Offer = No) = 1/3
≥8	P (CGPA ≥8 | Job Offer = Yes) = 4/7	P (CGPA ≥8 | Job Offer = No) = 0/3
<8	P (CGPA <8 | Job Offer = Yes) = 0/7	P (CGPA <8 | Job Offer = No) = 2/3

As explained earlier the Likelihood probability is stated as the sampling density for the evidence given the hypothesis. It is denoted as P (Evidence | Hypothesis), which says how likely is the occurrence of the evidence given the parameters.

It is calculated as the number of instances of each attribute value and for a given class value divided by the number of instances with that class value.

For example P (CGPA ≥9 | Job Offer = Yes) denotes the number of instances with 'CGPA ≥9' and 'Job Offer = Yes' divided by the total number of instances with 'Job Offer = Yes'.

From the Table 8.3 Frequency Matrix of CGPA, number of instances with 'CGPA ≥9' and 'Job Offer = Yes' is 3. The total number of instances with 'Job Offer = Yes' is 7. Hence, P (CGPA ≥9 | Job Offer = Yes) = 3/7.

Similarly, the Likelihood probability is calculated for all attribute values of feature CGPA.

Step 2(b): Feature – Interactiveness

Table 8.5 shows the frequency matrix for the feature Interactiveness.

Table 8.5: Frequency Matrix of Interactiveness

Interactiveness	Job Offer = Yes	Job Offer = No
YES	5	1
NO	2	2
Total	7	3

Table 8.6 shows how the likelihood probability is calculated for Interactiveness using conditional probability.

Table 8.6: Likelihood Probability of Interactiveness

Interactiveness	P (Job Offer = Yes)	P (Job Offer = No)
YES	P (Interactiveness = Yes | Job Offer = Yes) = 5/7	P (Interactiveness = Yes | Job Offer = No) = 1/3
NO	P (Interactiveness = No | Job Offer = Yes) = 2/7	P (Interactiveness = No | Job Offer = No) = 2/3

Step 2(c): Feature – Practical Knowledge

Table 8.7 shows the frequency matrix for the feature Practical Knowledge.

Table 8.7: Frequency Matrix of Practical Knowledge

Practical Knowledge	Job Offer = Yes	Job Offer = No
Very Good	2	0
Average	1	2
Good	4	1
Total	7	3

Table 8.8 shows how the likelihood probability is calculated for Practical Knowledge using conditional probability.

Table 8.8: Likelihood Probability of Practical Knowledge

Practical Knowledge	P (Job Offer = Yes)	P (Job Offer = No)
Very Good	P (Practical Knowledge = Very Good | Job Offer = Yes) = 2/7	P (Practical Knowledge = Very Good | Job Offer = No) = 0/3
Average	P (Practical Knowledge = Average | Job Offer = Yes) = 1/7	P (Practical Knowledge = Average | Job Offer = No) = 2/3
Good	P (Practical Knowledge = Good | Job Offer = Yes) = 4/7	P (Practical Knowledge = Good | Job Offer = No) = 1/3

Step 2(d): Feature – Communication Skills

Table 8.9 shows the frequency matrix for the feature Communication Skills.

Table 8.9: Frequency Matrix of Communication Skills

Communication Skills	Job Offer = Yes	Job Offer = No
Good	4	1
Moderate	3	0
Poor	0	2
Total	7	3

Table 8.10 shows how the likelihood probability is calculated for Communication Skills using conditional probability.

Table 8.10: Likelihood Probability of Communication Skills

Communication Skills	P (Job Offer = Yes)	P (Job Offer = No)
Good	P (Communication Skills = Good | Job Offer = Yes) = 4/7	P (Communication Skills = Good | Job Offer = No) = 1/3
Moderate	P (Communication Skills = Moderate | Job Offer = Yes) = 3/7	P (Communication Skills = Moderate | Job Offer = No) = 0/3
Poor	P (Communication Skills = Poor | Job Offer = Yes) = 0/7	P (Communication Skills = Poor | Job Offer = No) = 2/3

Step 3: Use Bayes theorem Eq. (8.1) to calculate the probability of all hypotheses.

Given the test data = (CGPA ≥9, Interactiveness = Yes, Practical knowledge = Average, Communication Skills = Good), apply the Bayes theorem to classify whether the given student gets a Job offer or not.

P (Job Offer = Yes | Test data) = (*P*(CGPA ≥9 | Job Offer = Yes) *P* (Interactiveness = Yes | Job Offer = Yes) *P* (Practical knowledge = Average | Job Offer = Yes) *P* (Communication Skills = Good | Job Offer = Yes) *P* (Job Offer = Yes)))/(*P* (Test Data))

We can ignore *P* (Test Data) in the denominator since it is common for all cases to be considered.

Hence, *P* (Job Offer = Yes | Test data) = (*P*(CGPA ≥9 |Job Offer = Yes) *P* (Interactiveness = Yes | Job Offer = Yes) *P* (Practical knowledge = Average | Job Offer = Yes) *P* (Communication Skills = Good | Job Offer = Yes) *P* (Job Offer = Yes)

= 3/7 × 5/7 × 1/7 × 4/7 × 7/10

= 0.0175

Similarly, for the other case 'Job Offer = No',

We compute the probability,

P (Job Offer = No| Test data) = (*P*(CGPA ≥9 |Job Offer =No) *P* (Interactiveness = Yes | Job Offer = No) *P* (Practical knowledge = Average | Job Offer = No) *P* (Communication Skills = Good | Job Offer =No) *P* (Job Offer = No))/(*P*(Test Data)).

P (CGPA ≥9 |Job Offer = No) *P* (Interactiveness = Yes | Job Offer = No) *P* (Practical knowledge = Average | Job Offer = No) *P* (Communication Skills = Good | Job Offer = No) *P* (Job Offer = No)

= 1/3 × 1/3 × 2/3 × 1/3 × 3/10

= 0.0074

Step 4: Use Maximum A Posteriori (MAP) Hypothesis, h_{MAP} Eq. (8.2) to classify the test object to the hypothesis with the highest probability.

Since *P* (Job Offer = Yes | Test data) has the highest probability value, the test data is classified as **'Job Offer = Yes'**.

8.b. Explain Maximum A Posteriori (MAP) Hypothesis, h_{MAP} and Maximum Likelihood (ML) Hypothesis, h_{ML}. (6M)

Maximum A Posteriori (MAP) Hypothesis, h_{MAP}

Given a set of candidate hypotheses, the hypothesis which has the maximum value is considered the *maximum probable hypothesis* or *most probable hypothesis*. This most probable hypothesis is called the Maximum A Posteriori Hypothesis h_{MAP}. Bayes theorem Eq. (8.1) can be used to find the h_{MAP}.

$$h_{MAP} = \max_{h \in H} P(Hypothesish \mid Evidence\ E)$$

$$= \max_{h \in H} \frac{P(Evidence\ E \mid Hypothesis\ h) P(Hypothesis\ h)}{P(Evidence\ E)}$$

$$= \max_{h \in H} P(Evidence\ E \mid Hypothesis\ h) P(Hypothesis\ h) \quad (8.2)$$

Maximum Likelihood (ML) Hypothesis, h_{ML}

Given a set of candidate hypotheses, if every hypothesis is equally probable, only $P(E \mid h)$ is used to find the *most probable hypothesis*. The hypothesis that gives the maximum likelihood for $P(E \mid h)$ is called the Maximum Likelihood (ML) Hypothesis, h_{ML}.

$$h_{ML} = \max_{h \in H} P(Evidence\ E \mid Hypothesis\ h) \quad (8.3)$$

8.c. Explain Bayes' optimal classifier. (4M)

8.3.3 Bayes Optimal Classifier

Bayes optimal classifier is a probabilistic model, which in fact, uses the Bayes theorem to find the most probable classification for a new instance given the training data by combining the predictions of all posterior hypotheses, This is different from Maximum A Posteriori (MAP) Hypothesis, h_{MAP} which chooses the maximum probable hypothesis or the most probable hypothesis.

Here, a new instance can be classified to a possible classification value C_i by the following Eq. (8.4).

$$= \max_{C_i} \sum_{h_i \in H} P(C_i \mid h_i)\, P(h_i \mid T) \quad (8.4)$$

Example 8.3: Given the hypothesis space with 4 hypothesis h_1, h_2, h_3 and h_4. Determine if the patient is diagnosed as COVID positive or COVID negative using Bayes Optimal classifier.

Solution: From the training dataset T, the posterior probabilities of the four different hypotheses for a new instance are given in Table 8.12.

Table 8.12: Posterior Probability Values

$P(h_i \mid T)$	P (COVID Positive $\mid h_i$)	P (COVID Negative $\mid h_i$)
0.3	0	1
0.1	1	0
0.2	1	0
0.1	1	0

h_{MAP} chooses h_1 which has the maximum probability value 0.3 as the solution and gives the result that the patient is COVID negative. But Bayes Optimal classifier combines the predictions of h_2, h_3 and h_4 which is 0.4 and gives the result that the patient is COVID positive.

$$\sum_{h_i \in H} P(\text{COVID Negative} \mid h_i) P(h_i \mid T) = 0.3 \times 1 = 0.3$$

$$\sum_{h_i \in H} P(\text{COVID Positive} \mid h_i) P(h_i \mid T) = 0.1 \times 1 + 0.2 \times 1 + 0.1 \times 1 = 0.4$$

Therefore, $\max_{C_i \in \{\text{COVID Positive, COVID Negative}\}} \sum_{h_i \in H} P(C_i \mid h_i) P(h_i \mid T)$ = COVID Positive.

Thus, this algorithm, diagnoses the new instance to be COVID positive.

MODULE – 5

9.a. Explain the types of artificial neural networks. (8M)

10.5 TYPES OF ARTIFICIAL NEURAL NETWORKS

ANNs consist of multiple neurons arranged in layers. There are different types of ANNs that differ by the network structure, activation function involved and the learning rules used. In an ANN, there are three layers called input layer, hidden layer and output layer. Any general ANN would consist of one input layer, one output layer and zero or more hidden layers.

10.5.1 Feed Forward Neural Network

This is the simplest neural network that consists of neurons which are arranged in layers and the information is propagated only in the forward direction. This model may or may not contain a hidden layer and there is no back propagation. Based on the number of hidden layers they are further classified into single-layered and multi-layered feed forward networks. These ANNs are simple to design and easy to maintain. They are fast but cannot be used for complex learning. They are used for simple classification and simple image processing, etc. The model of a Feed Forward Neural Network is shown in Figure 10.7.

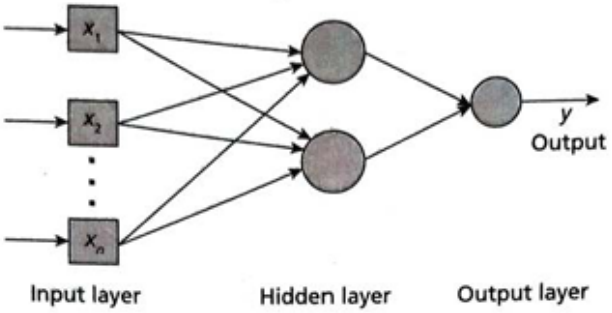


Figure 10.7: Model of a Feed Forward Neural Network

10.5.2 Fully Connected Neural Network

Fully connected neural networks are the ones in which all the neurons in a layer are connected to all other neurons in the next layer. The model of a fully connected neural network is shown in Figure 10.8.

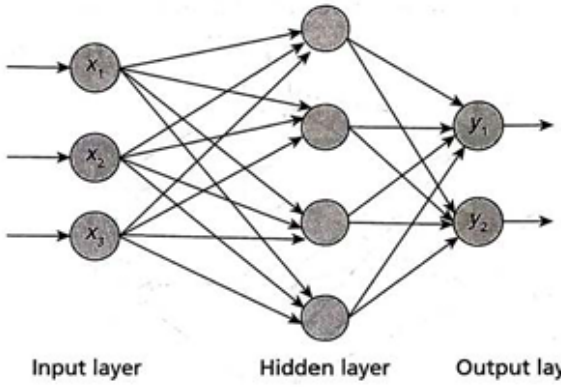


Figure 10.8: Model of a Fully Connected Neural Network

10.5.3 Multi-Layer Perceptron (MLP)

This ANN consists of multiple layers with one input layer, one output layer and one or more hidden layers. Every neuron in a layer is connected to all neurons in the next layer and thus they are fully connected. The information flows in both the directions. In the forward direction, the inputs are multiplied by weights of neurons and forwarded to the activation function of the

neuron and output is passed to the next layer. If the output is incorrect, then in the backward direction, error is back propagated to adjust the weights and biases to get correct output. Thus, the network learns with the training data. This type of ANN is used in deep learning for complex classification, speech recognition, medical diagnosis, forecasting, etc. They are comparatively complex and slow. The model of an MLP is shown in Figure 10.9.

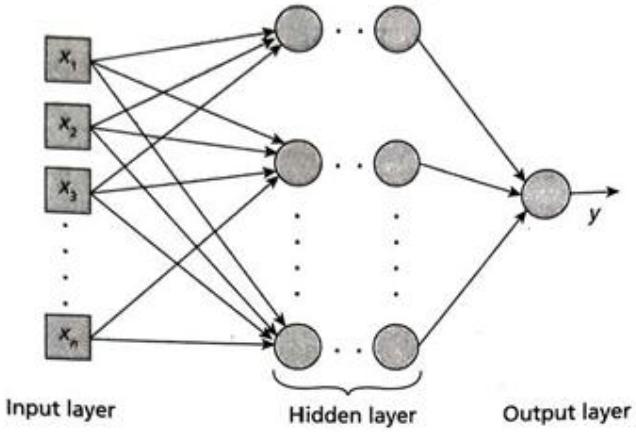


Figure 10.9: Model of a Multi-Layer Perceptron

10.5.4 Feedback Neural Network

Feedback neural networks have feedback connections between neurons that allow information flow in both directions in the network. The output signals can be sent back to the neurons in the same layer or to the neurons in the preceding layers. Hence, this network is more dynamic during training. The model of a feedback neural network is shown in Figure 10.10.

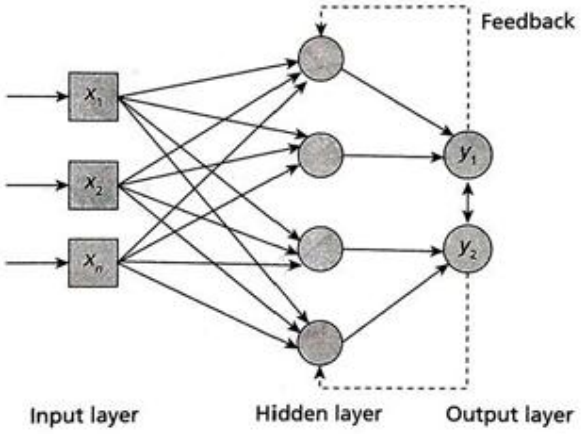


Figure 10.10: Model of a Feedback Neural Network

9.b. Explain Grid-based approach. (8M)

13.6 GRID-BASED APPROACH

Grid-based approach is a space-based approach. It partitions space into cells, the given data is fitted on the cells for cluster formation.

There are three important concepts that need to be mastered for understanding the grid-based schemes. They are:

1. Subspace clustering
2. Concept of dense cells
3. Monotonicity property

Subspace Clustering

Grid-based algorithms are useful for clustering high-dimensional data, that is, data with many attributes. Some data like gene data may have millions of attributes. Every attribute is called a dimension. But all the attributes are not needed, as in many applications one may not require all the attributes. For example, an employee's address may not be required for profiling his diseases. Age may be required in that case. So, one can conclude that only a subset of features is required. For example, one may be interested in grouping gene data with similar characteristics or organs that have similar functions.

Finding subspaces is difficult. For example, N dimensions may have 2^{N-1} subspaces. Exploring all the subspaces is a difficult task. Here, only the CLIQUE algorithms are useful for exploring the subspaces. CLIQUE (Clustering in Quest) is a grid-based method for finding clustering in subspaces. CLIQUE uses a multiresolution grid data structure.

Concept of Dense Cells

CLIQUE partitions each dimension into several overlapping intervals and intervals it into cells. Then, the algorithm determines whether the cell is dense or sparse. The cell is considered dense if it exceeds a threshold value, say τ. Density is defined as the ratio of number of points and volume of the region. In one pass, the algorithm finds the number of cells, number of points, etc. and then combines the dense cells. For that, the algorithm uses the contiguous intervals and a set of dense cells.

Algorithm 13.5: Dense Cells

Step 1: Define a set of grid points and assign the given data points on the grid.

Step 2: Determine the dense and sparse cells. If the number of points in a cell exceeds the threshold value τ, the cell is categorized as dense cell. Sparse cells are removed from the list.

Step 3: Merge the dense cells if they are adjacent.

Step 4: Form a list of grid cells for every subspace as output.

Monotonicity Property

CLIQUE uses anti-monotonicity property or apriori property of the famous apriori algorithm. It means that all the subsets of a frequent item should be frequent. Similarly, if the subset is infrequent, then all its supersets are infrequent as well. Based on the apriori property, one can conclude that a k-dimensional cell has r points if and only if every $(k - 1)$ dimensional projections of this cell have atleast r points. So like association rule mining that uses apriori rule, the candidate dense cells are generated for higher dimensions. The algorithm works in two stages as shown below.

Algorithm 13.6: CLIQUE

Stage 1:

Step 1: Identify the dense cells.

Step 2: Merge dense cells c_1 and c_2 if they share the same interval.

(*Continued*)

Step 3: Generate Apriori rule to generate $(k + 1)^{th}$ cell for higher dimension. Then, check whether the number of points cross the threshold. This is repeated till there are no dense cells or new generation of dense cells.

Stage 2:

Step 1: Merging of dense cells into a cluster is carried out in each subspace using maximal regions to cover dense cells. The maximal region is an hyperrectangle where all cells fall into.

Step 2: Maximal region tries to cover all dense cells to form clusters.

In stage two, CLIQUE starts from dimension 2 and starts merging. This process is continued till the n-dimension.

Advantages of CLIQUE

1. Insensitive to input order of objects
2. No assumptions of underlying data distributions
3. Finds subspace of higher dimensions such that high-density clusters exist in those subspaces

Disadvantage

The disadvantage of CLIQUE is that tuning of grid parameters, such as grid size, and finding optimal threshold for finding whether the cell is dense or not is a challenge.

10.c. What are the popular applications of artificial neural networks?

10.9 POPULAR APPLICATIONS OF ARTIFICIAL NEURAL NETWORKS

ANN learning mechanisms are used in many complex applications that involve modelling of non-linear processes. ANN is a useful model that can handle even noisy and incomplete data. They are used to model complex patterns, recognize patterns and solve prediction problems like humans in many areas such as:

1. Real-time applications: Face recognition, emotion detection, self-driving cars, navigation systems, routing systems, target tracking, vehicle scheduling, etc.
2. Business applications: Stock trading, sales forecasting, customer behaviour modelling, Market research and analysis, etc.
3. Banking and Finance: Credit and loan forecasting, fraud and risk evaluation, currency price prediction, real-estate appraisal, etc.
4. Education: Adaptive learning software, student performance modelling, etc.
5. Healthcare: Medical diagnosis or mapping symptoms to a medical case, image interpretation and pattern recognition, drug discovery, etc.
6. Other Engineering Applications: Robotics, aerospace, electronics, manufacturing, communications, chemical analysis, food research, etc.

OR

10.a. Explain the concept of perceptron and learning theory.

10.4 PERCEPTRON AND LEARNING THEORY

The first neural network model 'Perceptron', designed by Frank Rosenblatt in 1958, is a linear binary classifier used for supervised learning. He modified the McCulloch & Pitts Neuron model by combining two concepts, McCulloch-Pitts model of an artificial neuron and Hebbian learning rule of adjusting weights. He introduced variable weight values and an extra input that represents *bias* to this model. He proposed that artificial neurons could actually learn weights and thresholds from data and came up with a supervised learning algorithm that enabled the artificial neurons to learn the correct weights from training data by itself. The perceptron model (shown in Figure 10.5) consists of 4 steps:

1. Inputs from other neurons
2. Weights and bias
3. Net sum
4. Activation function

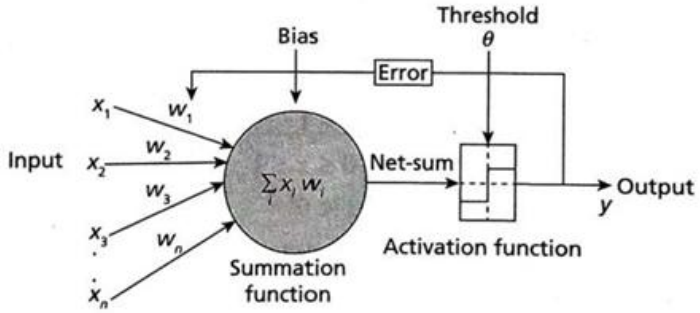


Figure 10.5: Perceptron Model

Thus, the modified neuron model receives a set of inputs $x_1, x_2, \ldots, x_n$, their associated weights $w_1, w_2, \ldots, w_n$ and a bias. The summation function 'Net-sum' Eq. (10.13) computes the weighted sum of the inputs received by the neuron.

$$\text{Net-sum} = \sum_{i=1}^{n} x_i w_i \tag{10.13}$$

After computing the 'Net-sum', bias value is added to it and inserted in the activation function as shown below:

$$f(x) = \text{Activation function } (\textit{Net-sum} + \textit{bias}) \tag{10.14}$$

The activation function is a binary step function which outputs a value 1 if $f(x)$ is above the threshold value θ, and a 0 if $f(x)$ is below the threshold value θ. Then, output of a neuron:

$$Y = \begin{cases} 1 \ \textit{if } f(x) \geq \theta \\ 0 \ \textit{if } f(x) < \theta \end{cases} \tag{10.15}$$

Before learning how a neural network works, let us learn about how a perceptron model works.

10.b. Explain any 4 proximity measures.

Proximity measures determine similarity or dissimilarity among objects. Distance measures, also known as dissimilarity measures indicate how different objects are. Similarity measures indicate how alike objects are. Clustering algorithms need proximity measures to find the similarity or dissimilarity among objects to group them. In clustering algorithms more distance equates to less similarity. Some proximity measures are discussed below.

Quantitative Variables

Some of the qualitative variables are discussed below.

Euclidean Distance It is one of the most important and common distance measures. It is also called as L_2 norm. It can be defined as the square root of squared differences between the coordinates of a pair of objects.

The Euclidean distance between objects x_i and x_j with k features is given as follows:

$$\text{Distance}(x_i, x_j) = \sqrt{\sum_{k=1}^{n}(x_{ik} - x_{jk})^2} \tag{13.1}$$

The advantage of Euclidean distance is that the distance does not change with the addition of new objects. But the disadvantage is that if the units change, the resulting Euclidean or squared Euclidean changes drastically. Another disadvantage is that as the Euclidean distance involves a square root and a square, the computational complexity is high for implementing the distance for millions or billions of operations involved.

City Block Distance City block distance is known as Manhattan distance. This is also known as boxcar, absolute value distance, Manhattan distance, Taxicab or L_1 norm. The formula for finding the distance is given as follows:

$$\text{Distance}(x_i, x_j) = \sum_{k=1}^{n}\left|x_{ik} - x_{jk}\right| \tag{13.2}$$

Chebyshev Distance Chebyshev distance is known as maximum value distance. This is the absolute magnitude of the differences between the coordinates of a pair of objects. This distance is called supremum distance or L_{max} or L_∞ norm. The formula for computing Chebyshev distance is given as follows:

$$\text{Distance}(x_i, x_j) = \max_k\left|x_{ik} - x_{jk}\right| \tag{13.3}$$

Example 13.1: Suppose, if the coordinates of the objects are (0, 3) and (5, 8), then what is the Chebyshev distance?

Solution: The Euclidean distance using Eq. (13.1) is given as follows:

$$\text{Distance}(x_i, x_j) = \sqrt{(0-5)^2 + (3-8)^2}$$
$$= \sqrt{50} = 7.07$$

The Manhattan distance using Eq. (13.2) is given as follows:

$$\text{Distance}(x_i, x_j) = \left|(0-5) + (3-8)\right| = 10$$

The Chebyshev distance using Eq. (13.3) is given as follows:

$$\text{Max}\left\{\left|0-5\right|, \left|3-8\right|\right\} = \text{Max}\{5, 5\} = 5$$

Minkowski Distance In general, all the above distance measures can be generalized as:

$$\text{Distance}(x_i, x_j) = \left(\Sigma\left|x_{ik} - x_{jk}\right|^r\right)^{\frac{1}{r}} \tag{13.4}$$

This is called Minkowski distance. Here, r is a parameter. When the value of r is 1, the distance measure is called city block distance. When the value of r is 2, the distance measure is called Euclidean distance. When, r is ∞, then this is Chebyshev distance.

Binary Attributes

Binary attributes have only two values. Distance measures discussed above cannot be applied to find distance between objects that have binary attributes. For finding the distance among objects with binary objects, the contingency Table 13.3 can be used. Let x and y be the objects consisting of N-binary objects. Then, the contingency table can be constructed by counting the number of matching of transitions, 0–0, 0–1, 1–0 and 1–1.

Table 13.3: Contingency Table

Attributes Matching	0	1
0	a	b
1	c	d

In other words, 'a' is the number of attributes where x attribute is 0 and y attribute is 0. 'b' is the number of attributes where x attribute is 0 and y attribute is 1, 'c' is the number of attributes where x attribute is 1 and y attribute is 0 and 'd' is the number of attributes where x attribute is 1 and y attribute is 1.

Simple Matching Coefficient (SMC) SMC is a simple distance measure and is defined as the ratio of number of matching attributes and the number of attributes. The formula is given as:

$$\frac{a+d}{a+b+c+d} \tag{13.5}$$

The values of a, b, c, and d can be observed from the Table 13.4.

Jaccard Coefficient Jaccard coefficient is another useful measure for and is given as follows:

$$J = \frac{d}{b+c+d} \tag{13.6}$$

Example 13.2: If the given vectors are $x = (1, 0, 0)$ and $y = (1, 1, 1)$ then find the SMC and Jaccard coefficient?

Solution: It can be seen from Table 13.2 that, $a = 0$, $b = 2$, $c = 0$ and $d = 1$.

The SMC using Eq. (13.5) is given as $\frac{a+d}{a+b+c+d} = 0+1/3 = 0.33$

Jaccard coefficient using Eq. (13.6) is given as $J = \frac{d}{b+c+d} = 1/3 = 0.33$

Hamming Distance Hamming distance is another useful measure that can be used for knowing the sequence of characters or binary values. It indicates the number of positions at which the characters or binary bits are different.

For example, the hamming distance between $x = (1\ 0\ 1)$ and $y = (1\ 1\ 0)$ is 2 as x and y differ in two positions. The distance between two words, say wood and hood is 1, as they differ in only one character. Sometimes, more complex distance measures like edit distance can also be used.

Categorical Variables

In many cases, categorical values are used. It is just a code or symbol to represent the values. For example, for the attribute Gender, a code 1 can be given to female and 0 can be given to male.

To calculate the distance between two objects represented by variables, we need to find only whether they are equal or not. This is given as:

$$\text{Distance}(x, y) = \begin{cases} 1 & \text{if } x \neq y \\ 0 & \text{if } x = y \end{cases} \tag{13.7}$$

Ordinal Variables

Ordinal variables are like categorical values but with an inherent order. For example, designation is an ordinal variable. If job designation is 1 or 2 or 3, it means code 1 is higher than 2 and code 2 is higher than 3. It is ranked as $1 > 2 > 3$.

Let us assume the designations of office employees are clerk, supervisor, manager and general manager. These can be designated as numbers as clerk = 1, supervisor = 2, manager = 3 and general manager = 4. Then, the distance between employee X who is a clerk and Y who is a manager can be obtained as:

$$\text{Distance}(X, Y) = \frac{|position(X) - position(Y)|}{n - 1} \tag{13.8}$$

Here, position (X) and position(Y) indicate the designated numerical value. Thus, the distance between X (Clerk = 1) and Y (Manager = 3) using Eq. (13.8) is given as:

$$\text{Distance}(X, Y) = \frac{|position(X) - position(Y)|}{n - 1} = \frac{|1 - 3|}{4 - 1} = \frac{2}{3} \approx 0.66$$

Vector Type Distance Measures

For text classification, vectors are normally used. Cosine similarity is a metric used to measure how similar the documents are irrespective of their size. Cosine similarity measures the cosine of the angle between two vectors projected in a multi-dimensional space. The similarity function for vector objects can be defined as:

$$sim(X,Y) = \frac{X \bullet Y}{\|X\| \times \|Y\|} = \frac{\sum_{i=1}^{n} X_i \times Y_i}{\sqrt{\sum_{i=1}^{n} X_i^2} \times \sqrt{\sum_{i=1}^{n} Y_i^2}} \tag{13.9}$$

The numeration is the dot product of the vectors A and B. The denominator is the product of the norm of vectors A and B.

Example 13.3: If the given vectors are $A = \{1, \mathbf{1}, 0\}$ and $B = \{0, 1, 1\}$, then what is the cosine similarity?

Solution: The dot product of the vector is $1 \times 0 + \mathbf{1} \times 1 + 0 \times 1 = 1$. The norm of the vectors A and B is $\sqrt{2}$.

So, the cosine similarity using Eq. (13.9) is given as $\frac{1}{\sqrt{2}\sqrt{2}} = \frac{1}{2} = 0.5$

10.c. What are the applications of clustering? (4M)

Applications of Clustering

1. Grouping based on customer buying patterns
2. Profiling of customers based on lifestyle
3. In information retrieval applications (like retrieval of a document from a collection of documents)
4. Identifying the groups of genes that influence a disease
5. Identification of organs that are similar in physiology functions
6. Taxonomy of animals, plants in Biology
7. Clustering based on purchasing behaviour and demography
8. Document indexing
9. Data compression by grouping similar objects and finding duplicate objects

---

THE END