UNIT–2 : Data Analysis (PYQ Solution)

AKTU · Data Analytics (BADS601) · Solutions by **Chirag Jain**, Ycotes

SECTION A Answers (≈70 Words Each)

Q1. Describe the concept of Lasso Regression. / What is Lasso Regression? (Jan 2025, 2022)

Lasso (Least Absolute Shrinkage and Selection Operator) regression is a linear regression that adds an L1 penalty equal to λ × Σ|β| to the least-squares loss. The penalty shrinks coefficients and forces some of them exactly to zero, so it performs feature selection and reduces overfitting. A larger λ gives a simpler model with more zero coefficients; λ = 0 gives ordinary least squares.

Q2. Differentiate between Univariate & Multivariate Analysis. (2022)

Univariate analysis studies a single variable at a time to describe its distribution using mean, median, mode, spread, histograms and box plots. Multivariate analysis studies three or more variables together to find relationships and joint effects, using methods such as multiple regression, PCA, cluster and factor analysis. Univariate explains *one* variable; multivariate explains *how many variables interact*. (Bivariate analysis lies in between, using two variables.)

Q3. What is decision tree? (2023)

A decision tree is a supervised learning model that looks like a flowchart. Each internal node tests an attribute, each branch is an outcome of the test, and each leaf holds a class label or a value. The tree splits data using measures such as information gain, Gini index or entropy. It is used for classification and regression, and is easy to interpret, though it can overfit unless pruned.

Q4. What is learning rate? (2023)

The learning rate (η or α) is a hyper-parameter that controls how big a step a model takes when it updates its weights during gradient descent: w = w − η × (∂Loss/∂w). If it is too small, training is slow; if it is too large, the algorithm may overshoot the minimum and fail to converge. A suitable value (such as 0.01) or a decaying schedule is chosen by experimentation.

Q5. What is rule induction? (2023)

Rule induction is a machine-learning technique that learns IF–THEN rules directly from data, for example "IF age < 25 AND income = high THEN buys = yes". Rules are generated by covering algorithms (such as CN2, RIPPER, AQ) or by extracting them from decision trees. Each rule is judged by its support and accuracy (confidence). Rule sets are easy for humans to understand and are used for classification and knowledge discovery.

Q6. How can you deal with uncertainty? (2023)

Uncertainty in data and models arises from noise, missing values and vague concepts. It can be handled by: probability theory and Bayesian networks (random uncertainty), fuzzy logic (vagueness), Dempster–Shafer theory (evidence), and statistical methods such as confidence intervals, cross-validation and ensemble models. Data cleaning, imputation of missing values, and collecting more data also reduce uncertainty.

SECTION B Answers (≈250 Words Each)

Q1. Explore PCA. Given data = {4, 8, 13, 7, 11, 4, 5, 14}. Compute the principal component using PCA algorithm. Also use PCA to reduce dimension from 2 to 1. (Jan 2025)

Principal Component Analysis (PCA) is an unsupervised technique that transforms correlated variables into a smaller set of uncorrelated variables called **principal components**, ordered by the variance they capture. Steps: (1) arrange data, (2) subtract the mean, (3) find the covariance matrix, (4) find eigenvalues and eigenvectors, (5) choose the top eigenvector(s), (6) project the data.

The eight numbers are taken as four 2-D points: (x, y) = (4, 11), (8, 4), (13, 5), (7, 14).

Step 1 – Means: $\bar{x} = (4+8+13+7)/4 = 8$, $\bar{y} = (11+4+5+14)/4 = 8.5$.

Step 2 – Deviations: $x - \bar{x} = -4, 0, 5, -1$; $y - \bar{y} = 2.5, -4.5, -3.5, 5.5$.

Step 3 – Covariance matrix (divide by n − 1 = 3):

- var(x) = (16 + 0 + 25 + 1)/3 = 14
- var(y) = (6.25 + 20.25 + 12.25 + 30.25)/3 = 23
- cov(x,y) = (−10 + 0 − 17.5 − 5.5)/3 = −11

S = [[14, −11], [−11, 23]]

Step 4 – Eigenvalues: $\det(S - \lambda I) = 0$ gives $\lambda^2 - 37\lambda + 201 = 0$, so $\lambda = (37 \pm \sqrt{565})/2$ → $\mathbf{\lambda_1 = 30.385}$, $\mathbf{\lambda_2 = 6.615}$.

Step 5 – Eigenvector for λ_1: $(14 - 30.385)e_1 - 11e_2 = 0 \rightarrow e_2 = -1.4896\, e_1$. Normalising: **e = (0.5574, −0.8303)**. This is the **first principal component**; it carries 30.385/37 = **82.1%** of total variance. The second (λ_2 = 6.615, 17.9%) is (0.8303, 0.5574).

Step 6 – Reduce 2-D to 1-D: project each mean-centred point on e: $Z = 0.5574(x - 8) - 0.8303(y - 8.5)$

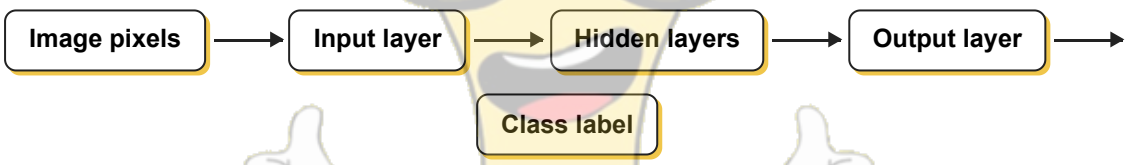
Point	Z
(4, 11)	−2.230 − 2.076 = **−4.31**
(8, 4)	0 + 3.736 = **3.74**
(13, 5)	2.787 + 2.906 = **5.69**
(7, 14)	−0.557 − 4.567 = **−5.12**

The four points are now represented by one number each (sum ≈ 0), losing only about 18% of the variance.

Conclusion: PC1 = 0.5574 x − 0.8303 y is the direction of maximum variance, and the data can be reduced from two dimensions to one.

Q2. Explain the Working of an Artificial Neural Network for Image Classification Task. (2022)

An **Artificial Neural Network (ANN)** is a model made of layers of connected artificial neurons that learns to map input to output by adjusting weights. For image classification (e.g. cat vs dog), it works as follows.



ANN for image classification

1. Input layer. The image is converted into numbers. A 28 × 28 grey image gives 784 pixel values (0-255), scaled to 0-1 and fed as 784 input neurons. (In a CNN the convolution layers first extract edges and shapes.)

2. Hidden layers. Each neuron computes a weighted sum of its inputs plus a bias, z = Σ(w·x) + b, and applies an **activation function** (ReLU, sigmoid) to add non-linearity: a = f(z). Early layers detect simple features such as edges; deeper layers combine them into shapes and objects.

3. Output layer. One neuron per class. A **softmax** function converts the scores into probabilities that add up to 1; the largest probability is the predicted class.

4. Training (learning).

- **Forward pass:** compute the prediction with current weights.
- **Loss:** compare with the true label using cross-entropy loss.

- **Backpropagation:** find the gradient of the loss with respect to every weight.
- **Gradient descent:** update w = w − η × gradient (η = learning rate).
- Repeat over many **epochs** with mini-batches until the loss is small.

5. Testing. New unseen images go through the forward pass only; accuracy is measured on a test set. Regularisation (dropout, early stopping) and data augmentation prevent overfitting.

Conclusion: the ANN learns useful features automatically from labelled pixels, and after training it classifies a new image with a single forward pass.

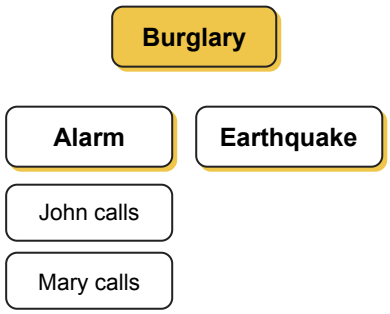
Q3. Explain multivariate analysis and Bayesian network. (2023)

Multivariate analysis is the set of statistical methods that analyse **more than two variables simultaneously** to understand patterns, relationships and joint effects.

- **Multiple regression:** predicts one dependent variable from several independent variables.
- **PCA and factor analysis:** reduce many correlated variables to a few components.
- **Cluster analysis:** groups similar observations.
- **Discriminant analysis, MANOVA, canonical correlation:** classify or compare groups on several variables.
- Uses: market segmentation, medical diagnosis, finance and quality control.

Bayesian network (belief network) is a **directed acyclic graph (DAG)** in which each node is a random variable and each edge shows a conditional dependency. Every node has a **conditional probability table (CPT)** giving P(node | parents). The joint probability is the product of local probabilities:

$P(X_1, \ldots, X_n) = \Pi\, P(X_i \mid \text{Parents}(X_i))$

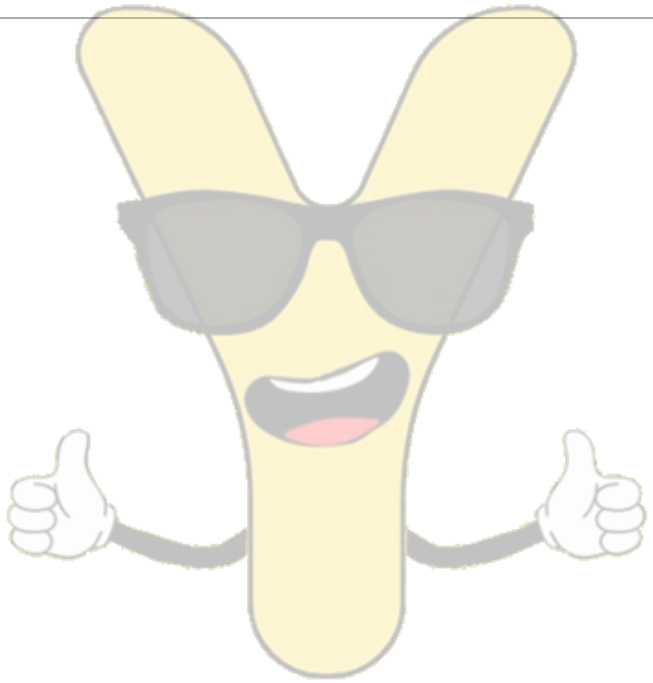


Example Bayesian network

Example: Burglary and Earthquake both affect Alarm, and the Alarm causes John or Mary to call. Given that John called, we can compute the probability of a burglary using **Bayes' theorem**: P(A|B) = P(B|A)·P(A)/P(B).

Point	Multivariate analysis	Bayesian network
Nature	Family of statistical methods	Probabilistic graphical model
Purpose	Find relationships, reduce, group	Reason under uncertainty
Output	Components, clusters, equations	Probabilities of events

Conclusion: multivariate analysis explains how many variables behave together, while Bayesian networks model their cause-and-effect and handle uncertain, incomplete information.



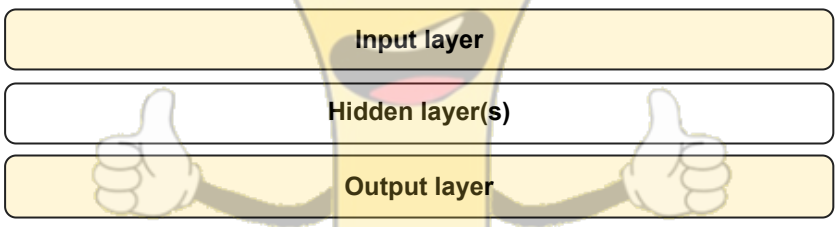
SECTION C Answers (≈500 Words Each)

Q1. Differentiate between Neural Network and Artificial Neural Network. (Jan 2025)

A **neural network** in the general sense is any network of interconnected processing units that passes signals and learns from experience. The best-known one is the **biological neural network** – the brain, with about 86 billion neurons connected by synapses. An **Artificial Neural Network (ANN)** is a computer model, inspired by that biological network, built from mathematical neurons arranged in layers and trained on data.

Working of the two

- *Biological:* dendrites receive signals, the cell body adds them, and if the total crosses a threshold the axon fires a signal through synapses to the next neuron. Learning changes synaptic strength.
- *Artificial:* inputs $x_1 \ldots x_n$ are multiplied by weights $w_1 \ldots w_n$, summed with a bias, and passed through an activation function: $y = f(\Sigma\, w_i x_i + b)$. Learning changes the weights by backpropagation and gradient descent.



Layers of an Artificial Neural Network

Basis	Neural network (biological)	Artificial neural network
Origin	Natural, in living beings	Man-made, in software or hardware
Basic unit	Neuron with dendrites, axon, synapse	Node / perceptron with weights and activation
Number of units	About 10^{11} neurons	Hundreds to billions of nodes
Signals	Electro-chemical	Numerical values
Speed	Slow (milliseconds)	Very fast (nanoseconds)
Learning	Experience, adapts through plasticity	Training algorithm on labelled data
Connections	Highly complex, about 10,000 per neuron	Simple, structured layers
Fault tolerance	Very high	Moderate
Energy	About 20 watts	High, needs GPUs
Processing	Massively parallel, noisy	Parallel but deterministic
Purpose	Thinking, perception, control	Classification, prediction, pattern recognition

Types of ANN: feed-forward (single-layer perceptron, multi-layer perceptron), convolutional (images), recurrent (sequences), and radial basis networks.

Applications of ANN: image and speech recognition, medical diagnosis, stock forecasting, fraud detection and self-driving cars.

Conclusion: an ANN is only a simplified mathematical imitation of the biological neural network; the biological one is far more complex and adaptive, while the ANN is faster for specific, well-defined tasks.

Q2. A = {(10,0,2), (20,0,4), (25,0,7), (30,0,9), (40,1), (50,0,4)}, B = {(10,0,4), (20,0,1), (25,0,9), (30,0,2), (40,0,6), (50,0,6)}. Apply Union, Intersection, Complement, Bold Union and Bold Intersection operations on above listed Fuzzy Sets. (Jan 2025)

Each pair is (element, membership value); the printed "0,2" means 0.2 (decimal comma), and (40,1) means membership 1. So:

A = {10/0.2, 20/0.4, 25/0.7, 30/0.9, 40/1.0, 50/0.4}
B = {10/0.4, 20/0.1, 25/0.9, 30/0.2, 40/0.6, 50/0.6}

Formulas (for membership μ):

- Union: μ(A∪B) = max(μA, μB)
- Intersection: μ(A∩B) = min(μA, μB)
- Complement: μ(A′) = 1 − μA
- Bold union (bounded sum): μ(A⊕B) = min(1, μA + μB)
- Bold intersection (bounded difference): μ(A⊙B) = max(0, μA + μB − 1)

Step-wise calculation

x	μA	μB	Union (max)	Intersection (min)	A′ (1−μA)	B′ (1−μB)	Bold union	Bold intersection
10	0.2	0.4	0.4	0.2	0.8	0.6	min(1, 0.6) = 0.6	max(0, −0.4) = 0
20	0.4	0.1	0.4	0.1	0.6	0.9	0.5	max(0, −0.5) = 0
25	0.7	0.9	0.9	0.7	0.3	0.1	min(1, 1.6) = 1	1.6 − 1 = 0.6
30	0.9	0.2	0.9	0.2	0.1	0.8	min(1, 1.1) = 1	1.1 − 1 = 0.1
40	1.0	0.6	1.0	0.6	0	0.4	1	0.6
50	0.4	0.6	0.6	0.4	0.6	0.4	min(1, 1.0) = 1	max(0, 0) = 0

Results

- A ∪ B = {10/0.4, 20/0.4, 25/0.9, 30/0.9, 40/1.0, 50/0.6}
- A ∩ B = {10/0.2, 20/0.1, 25/0.7, 30/0.2, 40/0.6, 50/0.4}
- A′ = {10/0.8, 20/0.6, 25/0.3, 30/0.1, 40/0, 50/0.6}
- B′ = {10/0.6, 20/0.9, 25/0.1, 30/0.8, 40/0.4, 50/0.4}
- Bold union A ⊕ B = {10/0.6, 20/0.5, 25/1, 30/1, 40/1, 50/1}
- Bold intersection A ⊙ B = {10/0, 20/0, 25/0.6, 30/0.1, 40/0.6, 50/0}

Check: the intersection is always ≤ the union, and the bold intersection ≤ the intersection ≤ the union ≤ the bold union, which holds for every element above.

Conclusion: the standard operations (max, min, 1 − μ) are the usual fuzzy-set operations, while the bold (bounded) operations are stricter or looser versions that add or subtract the membership values.

Q3. Discuss different types of Time Series Data Analysis along with its Major Application area. (2022)

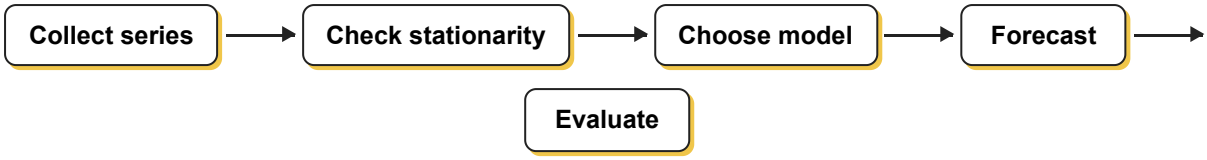
A **time series** is a sequence of observations recorded at regular time intervals, such as daily stock prices, monthly sales or hourly temperature. **Time series analysis** studies the past pattern to understand the data and to forecast the future.

Components of a time series

- **Trend (T):** long-term upward or downward movement.
- **Seasonality (S):** regular pattern repeating in a fixed period, e.g. festival sales every year.
- **Cyclic (C):** longer, irregular ups and downs, such as business cycles.
- **Irregular / random (I):** unpredictable noise.
  The model can be additive (Y = T + S + C + I) or multiplicative (Y = T × S × C × I).

Types of time series analysis

1. **Descriptive analysis:** plots and summarises the series; finds trend, seasonality and outliers.
2. **Exploratory / decomposition analysis:** splits the series into trend, seasonal and residual parts (STL decomposition).
3. **Forecasting analysis:** predicts future values using **moving average, exponential smoothing, AR, MA, ARMA, ARIMA, SARIMA**, Holt-Winters, and LSTM neural networks.
4. **Smoothing and curve fitting:** moving averages and regression remove noise and fit trend lines.
5. **Classification analysis:** assigns a label to a whole series, e.g. normal or abnormal ECG.
6. **Segmentation / anomaly detection:** splits the series into segments or finds unusual points, e.g. fraud, machine faults.
7. **Intervention analysis:** measures the effect of an event (policy, advertisement) on the series.
8. **Univariate vs multivariate:** one variable over time versus several related series (e.g. sales with price and weather).



Time series forecasting steps

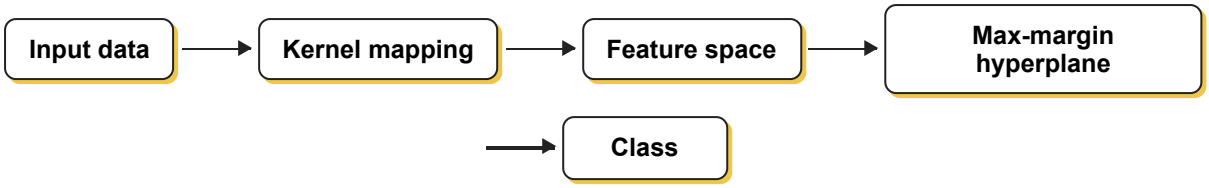
Major application areas

Area	Use
Finance	Stock price and exchange-rate forecasting, risk
Retail and business	Demand and sales forecasting, inventory
Weather and climate	Temperature, rainfall, cyclone prediction
Healthcare	ECG/EEG monitoring, epidemic spread
Energy	Electricity load forecasting
Manufacturing / IoT	Predictive maintenance from sensor data
Economics	GDP, inflation, unemployment
Web	Traffic and server load prediction

Conclusion: time series analysis finds the trend, seasonality and randomness in time-ordered data, and is the basis of forecasting in business, science and engineering.

Q4. Differentiate different types of support vector and kernel methods of Data Analysis. / What are the different kernel methods of Data Analytics? (2022, 2023)

Support Vector Machine (SVM) is a supervised algorithm that finds the **optimal separating hyperplane** with the **maximum margin** between classes. The training points that lie on the margin are the **support vectors**. When data cannot be separated by a straight line, a **kernel method** maps it into a higher-dimensional space where it becomes linearly separable, without computing that mapping explicitly (the **kernel trick**): $K(x, y) = \varphi(x) \cdot \varphi(y)$.



SVM with a kernel

Types of support vector machines

Type	Description
Linear SVM (hard margin)	Perfectly separable data; no errors allowed
Soft-margin SVM	Allows some misclassification, controlled by parameter C
Non-linear (kernel) SVM	Uses a kernel for curved boundaries
Support Vector Regression (SVR)	Fits a function within an ε-tube for numeric prediction
One-class SVM	Learns one class; used for anomaly detection
Multi-class SVM	One-vs-one or one-vs-rest combination

Kernel functions

1. **Linear kernel:** K(x, y) = x·y. Fast; best for linearly separable and high-dimensional (text) data.
2. **Polynomial kernel:** K(x, y) = (x·y + c)^d. Captures feature interactions of degree d.
3. **Radial Basis Function (Gaussian) kernel:** K(x, y) = exp(−γ‖x − y‖²). Most popular; handles complex boundaries; parameter γ.
4. **Sigmoid kernel:** K(x, y) = tanh(αx·y + c). Behaves like a neural network.
5. **Laplacian / exponential kernels:** similar to RBF with the absolute distance.
6. **String and graph kernels:** for text, DNA sequences and networks.

Kernel	Boundary	Parameters	Typical use
Linear	Straight line	C	Text classification
Polynomial	Curved, degree d	d, c	Image processing
RBF	Very flexible	γ, C	General purpose default
Sigmoid	Neural-like	α, c	Some neural tasks

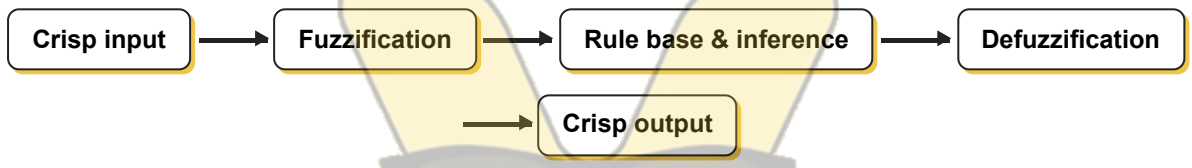
Kernel-based methods other than SVM: kernel PCA, kernel ridge regression, Gaussian processes and kernel k-means.

Conclusion: SVMs differ in margin type and task, while kernels differ in the shape of the decision boundary; choosing the kernel and its parameters by cross-validation is the key to a good model.

Q5. Differentiate between Crisp logic and Fuzzy logic. (2023)

Crisp (classical, Boolean) logic treats every statement as completely **true (1) or false (0)**. An element either belongs to a set or does not. Example: "A person is tall if height ≥ 6 ft" – someone of 5 ft 11 in is simply "not tall".

Fuzzy logic, proposed by **Lotfi Zadeh (1965)**, allows **degrees of truth** between 0 and 1. A membership function μ(x) tells how strongly an element belongs to a set. The same person of 5 ft 11 in may be "tall" with membership 0.8. It models vague human terms such as *warm, fast, young*.



A fuzzy logic system

Basis	Crisp logic	Fuzzy logic
Truth values	Only 0 or 1	Any value in [0, 1]
Set membership	Full or none	Partial membership, μ(x)
Boundary	Sharp	Gradual / overlapping
Handles	Precise, certain data	Vague, uncertain data
Reasoning	Binary (true/false)	Approximate, human-like
Set operations	AND, OR, NOT on 0/1	min, max, 1 − μ (also bounded sum)
Law of excluded middle	Holds ($A \cup A' = U$)	Does not hold
Example	Water is hot (yes/no)	Water is 70% hot
Hardware	Digital circuits	Fuzzy controllers
Applications	Databases, programming conditions	Washing machines, AC, ABS brakes, cameras

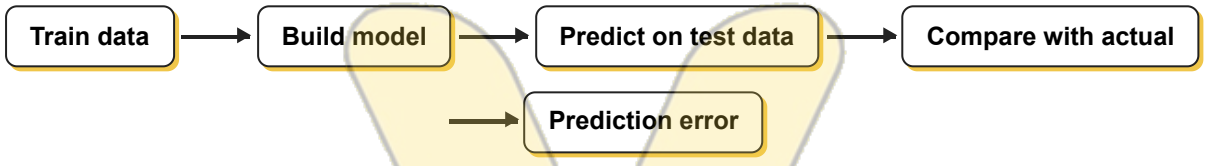
Example – temperature: crisp rule: IF T > 30 then HOT. At 30.1 °C the output is HOT, at 29.9 °C it is not – an abrupt change. Fuzzy: μ(hot) rises smoothly from 0 at

20 °C to 1 at 40 °C, so 30 °C is 0.5 hot, and the controller responds smoothly.

Conclusion: crisp logic suits exact yes/no problems, while fuzzy logic is a generalisation of it that can handle vagueness; a crisp set is just a fuzzy set whose membership values are only 0 or 1.

Q6. What is Prediction error? With the help of suitable example explain prediction error in classification and regression. (2023)

Prediction error is the difference between the value predicted by a model and the actual (true) value. It measures how good a model is, and is mainly estimated on **test data** not used for training. Total expected error is made of **bias² + variance + irreducible noise**: high bias leads to under-fitting, high variance leads to over-fitting.



Measuring prediction error

(a) Prediction error in regression (numeric target). Error for one case = actual − predicted. Common measures:

- **MAE** = (1/n) Σ $|y_i - \hat{y}_i|$
- **MSE** = (1/n) Σ $(y_i - \hat{y}_i)^2$
- **RMSE** = √MSE
- **R^2** = 1 − SSE/SST

Example: a model predicts house prices (in lakh ₹).

House	Actual y	Predicted ŷ	Error	Squared
1	50	45	5	25
2	60	65	−5	25
3	80	76	4	16
4	40	43	−3	9

MAE = (5 + 5 + 4 + 3)/4 = 4.25; MSE = 75/4 = 18.75; RMSE = √18.75 ≈ 4.33 lakh. On average the prediction is wrong by about 4.3 lakh.

(b) Prediction error in classification (category target). The error is a wrong class label. It is summarised by a **confusion matrix**:

	Predicted Yes	Predicted No
Actual Yes	TP	FN
Actual No	FP	TN

- **Error rate** = (FP + FN)/(TP + TN + FP + FN); **Accuracy** = 1 − error rate
- **Precision** = TP/(TP + FP); **Recall** = TP/(TP + FN); **F1** = 2PR/(P + R)

Example: a spam filter on 100 mails gives TP = 40, TN = 45, FP = 5, FN = 10. Error rate = (5 + 10)/100 = 15%; accuracy = 85%; precision = 40/45 = 0.89; recall = 40/50 = 0.80.

Reducing prediction error: use cross-validation, regularisation, more data, feature selection and ensemble methods; compare training error and test error to detect over-fitting.

Conclusion: regression error measures *how far* the prediction is from the true number, while classification error measures *how often* the predicted class is wrong.

