IAE 1 MLBC Answers Bank

Q.1. Outline the key conceptual steps involved in performing Principal Component Analysis (PCA) on a dataset.

Principal Component Analysis (PCA) is a statistical technique used for dimensionality reduction.
It transforms a dataset with many correlated features into a smaller set of uncorrelated features called Principal Components (PCs),
while preserving as much information as possible. The key conceptual steps are:

1. Standardize the data: Features often have different scales (marks out of 10 vs income in lakhs). Standardization ensures all variables contribute equally.
2. Compute the covariance matrix: This matrix captures how different variables vary with respect to each other. High covariance indicates strong relationships.
3. Calculate eigenvalues and eigenvectors: Eigenvectors show directions of maximum variance, while eigenvalues indicate the magnitude of variance in those directions.
4. Sort and select principal components: Rank eigenvalues from largest to smallest. Select top-k eigenvectors to reduce dimensions while retaining maximum variance.
5. Transform the dataset: Multiply the original dataset with chosen eigenvectors to obtain new reduced-dimension data.

This results in a compact representation of data, making models faster and less prone to overfitting while still capturing essential patterns.

Q.2. Explain the role of the covariance matrix in Principal Component Analysis (PCA).

The covariance matrix is central to PCA as it quantifies relationships between variables.
It tells us whether features increase together (positive covariance), move in opposite directions (negative covariance), or are unrelated (zero covariance).

In PCA, the covariance matrix is decomposed to find eigenvalues and eigenvectors.
- Eigenvectors represent the directions of new axes (principal components).
- Eigenvalues represent the amount of variance carried along each component.

Thus, the covariance matrix helps PCA identify directions of maximum variance and ensures dimensionality reduction is meaningful and not random.

Q.3. Explain the core principle of Linear Regression. How does the algorithm determine the 'best-fit line,' and why do we minimize the sum of squared errors?

Linear regression models the relationship between independent variables (features) and a dependent variable (target) assuming linearity.
The equation is usually written as: $y = mx + c$ (or $y = \beta 0 + \beta 1x$).

- Best-fit line: The algorithm tries different slopes (m) and intercepts (c) and checks how well they predict outcomes.
- Error calculation: Error is the difference between actual (y) and predicted ($\hat{y}$).
- Sum of Squared Errors (SSE): Squaring ensures positive errors and penalizes larger deviations more strongly.

Minimizing SSE ensures the line is as close as possible to all data points.

Thus, the best-fit line is the one where the overall squared error is minimized, resulting in the most accurate predictions.

Q.4. Why is Linear Regression generally unsuitable for binary classification tasks? Explain how Logistic Regression overcomes this limitation.

Linear regression is unsuitable for binary classification because it outputs continuous values ($-\infty$ to $+\infty$), while classification requires probabilities (0 to 1).
It may predict invalid probabilities (like −0.5 or 1.2), making it unreliable for classification.

Logistic Regression solves this using the sigmoid function:

$$f(z) = 1 / (1 + e^{(-z)})$$

This function maps any input into the range (0,1), giving valid probabilities.
A threshold (commonly 0.5) is applied: values ≥ 0.5 are classified as class 1, others as class 0.

Hence, logistic regression is a natural fit for binary classification tasks unlike linear regression.

Q.5. In what scenarios would you choose Polynomial Regression over Simple Linear Regression?

Simple linear regression fits a straight line, but not all real-world data follows a straight pattern.
Polynomial regression is preferred when the relationship between input and output is non-linear but still continuous.

Example: Predicting marks vs hours studied. Initially, marks increase with hours studied, but after a point, more hours may not give equal improvement.
A straight line underfits this trend, while a polynomial curve (quadratic or cubic) captures it more accurately.

Thus, polynomial regression is chosen when data shows curves, peaks, or non-linear variations that cannot be captured by a straight line.

Q.6. Compare and contrast Ridge Regression (L2) and Lasso Regression (L1).

Both Ridge and Lasso are regularization methods that prevent overfitting by penalizing large coefficients.

- Ridge Regression (L2): Adds squared penalty term $\lambda\Sigma\beta^2$. It shrinks coefficients closer to zero but never fully eliminates them.
It is useful when features are highly correlated since it distributes importance among them.

- Lasso Regression (L1): Adds absolute penalty term $\lambda\Sigma|\beta|$. It can reduce some coefficients exactly to zero, effectively performing feature selection.
This makes Lasso useful when we want a simpler model with fewer variables.

In short: Ridge = shrinkage, keeps all features; Lasso = shrinkage + selection, removes irrelevant features.

Q.7. Briefly explain the purpose of Elastic Net Regression and describe how it combines the strengths of both Ridge and Lasso Regression.

Elastic Net Regression combines Ridge (L2) and Lasso (L1) penalties.
The cost function includes both $\lambda 1\Sigma|\beta| + \lambda 2\Sigma\beta^2$.

- From Ridge, it inherits the ability to handle multicollinearity by distributing weights across correlated variables.
- From Lasso, it inherits the ability to perform feature selection by shrinking some coefficients to zero.

Elastic Net is especially useful when:
1. The dataset has many features, some of which are correlated.
2. The number of features is greater than the number of data points.

Thus, it balances stability (Ridge) with sparsity (Lasso).

Q.8. State the formula for Bayes' Theorem. What is the 'naive' assumption that the Naive Bayes classifier makes about the features in a dataset, and why is this assumption critical for the algorithm's simplicity and efficiency?

Bayes' Theorem:

$$P(A|B) = [P(B|A) * P(A)] / P(B)$$

Where:
- P(A|B) = Probability of A given B (posterior probability)
- P(B|A) = Probability of B given A (likelihood)
- P(A) = Prior probability of A
- P(B) = Probability of B

Naive Bayes assumption: All features are independent given the class label.
For example, in spam classification, presence of words like "free" and "offer" are treated as independent.

Importance of assumption:
- Makes computations simple and efficient.
- Works surprisingly well in practice, even if independence is not fully true, making Naive Bayes popular in text and spam classification tasks.

Q.9. Compare and contrast the three variants of the Naive Bayes algorithm.

Aspect	Gaussian Naive Bayes	Multinomial Naive Bayes	Bernoulli Naive Bayes
Type of Data	Continuous values (e.g., height, weight, temperature)	Discrete counts (e.g., word frequencies in text)	Binary values (0/1, word present or not)
Distribution Assumption	Assumes normal (Gaussian) distribution of features	Assumes multinomial distribution of features	Assumes Bernoulli distribution for binary outcomes
Application Domain	Medical diagnosis, sensor readings, continuous attributes	Text classification, NLP tasks, spam detection	Document classification based on word presence/absence
Strengths	Handles continuous data effectively	Works very well with frequency-based features	Best when presence/absence of features is more important than counts
Example	Predicting disease based on patient age, weight	Spam filtering using word counts	Spam filtering based on whether words occur in an email