**1. Time Series Analysis**

**Questions:**

* Write short note on Time series Analysis

**Answer:**

**Time Series Analysis** involves methods to analyze time-ordered data points to extract meaningful statistics and identify characteristics like trends, seasonality, and cyclic patterns. It is widely used in fields like finance, weather forecasting, and stock market analysis.

* **Components of Time Series:**
  + Trend: Long-term movement in the data.
  + Seasonality: Repeated patterns at regular intervals.
  + Noise: Random variations or irregular fluctuations.

**Diagram: Components of Time Series**

| Seasonality

| Trend /\ /\ /\

| / \ / \ / \

|\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\/\_\_\_\_\_\_\_\_\/\_\_\_\_\_\_\_\_ Time

| <---------> <-----> <-----> <----->

**2. TF-IDF (Term Frequency - Inverse Document Frequency)**

**Questions:**

* Write short note on TF-IDF
* Explain the TF/IDF terms in text analysis with suitable example
* What is text processing? Explain TF-IDF with example
* Explain the terms bag of words and TF IDF in text analytics

**Answer:**

**TF-IDF** is a statistical measure used to evaluate the importance of a word in a document relative to a collection (corpus) of documents.

* **Term Frequency (TF):** Number of times a term appears in a document, normalized by total words in that document.

TF(t,d)=Number of times term t appears in document dTotal number of terms in dTF(t,d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in } d}

* **Inverse Document Frequency (IDF):** Measures how common or rare a term is across all documents.

IDF(t)=log⁡N1+Number of documents containing tIDF(t) = \log \frac{N}{1 + \text{Number of documents containing } t}

* **TF-IDF:** The product of TF and IDF.

TFIDF(t,d)=TF(t,d)×IDF(t)TFIDF(t,d) = TF(t,d) \times IDF(t)

**Example:**

* Corpus: 3 documents
  + Doc1: "the cat sat"
  + Doc2: "the cat sat on the mat"
  + Doc3: "the dog sat"

Term: "cat"

* TF in Doc2 = 1/6
* IDF = log(3 / 2) = 0.176
* TF-IDF = 0.166 × 0.176 ≈ 0.029

**Diagram: TF-IDF Process**

Document Collection (Corpus)

↓

Count term frequency (TF)

↓

Compute document frequency (DF) → Calculate IDF

↓

Multiply TF \* IDF → Get TF-IDF score

**3. Clustering & K-means Algorithm**

**Questions:**

* What is clustering? With suitable example explain the steps involved in k-means algorithm.
* Suppose given data points, use k-means algorithm to show the cluster centers after the first round of execution.
* Perform K-means clustering on dataset with initial centroids, compute new centroids after each iteration until convergence.
* Explain how hierarchical clustering can be used for visualizing hierarchical relationships with example and applications.

**Answer:**

**Clustering** is an unsupervised machine learning technique that groups data points such that points in the same group (cluster) are more similar to each other than to those in other groups.

**K-means Algorithm Steps:**

1. Initialize k cluster centroids randomly.
2. Assign each data point to the nearest centroid based on Euclidean distance.
3. Recalculate centroids as the mean of points assigned to each cluster.
4. Repeat steps 2 and 3 until centroids do not change significantly.

**Example:**

Points: A1(2,10), A2(2,5), A3(8,4), B1(5,8), B2(7,5), B3(6,4), C1(1,2), C2(4,9)

Initial centroids: A1(2,10), B1(5,8), C1(1,2)

* Assign points to closest centroid
* Compute new centroids for each cluster

**Diagram: K-means Iteration**

Initial Centroids

C1: (2,10) C2: (5,8) C3: (1,2)

Points assigned to clusters:

Cluster1: A1, A2, C2

Cluster2: B1, B2, B3, A3

Cluster3: C1

New Centroids:

Mean of Cluster1 points → New C1

Mean of Cluster2 points → New C2

Mean of Cluster3 points → New C3

**Hierarchical Clustering:** Builds a tree (dendrogram) showing nested clusters, useful for visualizing data relationships.

**4. Confusion Matrix and Related Metrics**

**Questions:**

* Write short note on Confusion matrix
* Given a confusion matrix, calculate accuracy, precision, recall, error rate
* Define accuracy, precision, recall, AUC-ROC with respect to confusion matrix
* Explain AVC-ROC curve (likely AUC-ROC curve)

**Answer:**

**Confusion Matrix** is a table to evaluate classification models, showing actual vs predicted classes.

|  | **Predicted Positive** | **Predicted Negative** |
| --- | --- | --- |
| Actual Positive | True Positive (TP) | False Negative (FN) |
| Actual Negative | False Positive (FP) | True Negative (TN) |

* **Accuracy:** TP+TNTP+TN+FP+FN\frac{TP + TN}{TP + TN + FP + FN}  
  Overall correctness.
* **Precision:** TPTP+FP\frac{TP}{TP + FP}  
  Correct positive predictions.
* **Recall (Sensitivity):** TPTP+FN\frac{TP}{TP + FN}  
  Ability to find all positive samples.
* **Error Rate:** FP+FNTotal\frac{FP + FN}{Total}
* **AUC-ROC Curve:**  
  Plots True Positive Rate (Recall) vs False Positive Rate, area under curve (AUC) indicates model quality.

**Diagram: Confusion Matrix**

Predicted

+ -

Actual + TP FN

- FP TN

**5. Text Analysis / Text Processing Steps**

**Questions:**

* Explain text analysis steps: Part of speech (POS) tagging, Lemmatization, Stemming
* Explain Text Preprocessing steps with suitable example

**Answer:**

* **POS Tagging:** Assigning each word a part of speech (noun, verb, adjective, etc.) based on context.
* **Stemming:** Reducing words to their root by chopping off suffixes (e.g., "running" → "run").
* **Lemmatization:** Reducing words to their dictionary form (lemma), considering context (e.g., "better" → "good").

**Example:**

Sentence: "The cats are running faster."

* POS tagging: The (DT), cats (NNS), are (VB), running (VBG), faster (JJR)
* Stemming: cats → cat, running → run
* Lemmatization: running → run, faster → fast

**6. Model Validation Techniques**

**Questions:**

* Discuss Holdout method and Random Sub Sampling methods
* Explain k-fold Cross Validation & Random Subsampling
* What is the holdout method, how does it work? Explain training set, validation set, test set differences

**Answer:**

* **Holdout Method:** Dataset is split into training and test sets. Model trained on training set, tested on test set.
* **Random Subsampling:** Repeatedly split data randomly into training and test sets multiple times, average results.
* **K-fold Cross Validation:** Dataset is divided into k folds. Model trained on k-1 folds and tested on remaining fold. Process repeated k times, average accuracy is computed.

**Diagram: K-Fold Cross Validation**

Dataset → Split into k folds: Fold1, Fold2, ..., Foldk

Iteration 1: Train on Fold2,...Foldk, Test on Fold1

Iteration 2: Train on Fold1, Fold3,...Foldk, Test on Fold2

...

Iteration k: Train on Fold1,...Fold(k-1), Test on Foldk

Average results → Final performance

**7. Noise & Irrelevant Data Handling in Text Data**

**Questions:**

* How do you handle noise and irrelevant information in text data during preprocessing?

**Answer:**

**Handling Noise in Text Data:**

* Remove stop words (common words like "the", "is")
* Remove punctuation, special characters, numbers
* Normalize text (lowercase, remove accents)
* Handle misspellings and typos
* Use stemming or lemmatization to reduce word forms
* Filter out irrelevant or redundant words

**Bag of Words:** Represents text as frequency of words disregarding order.

**Diagram: Text Preprocessing Flow**

Raw Text → Remove Noise (stop words, punctuation) → Normalize (lowercase)

→ Tokenization → Stemming / Lemmatization → Vectorize (Bag of Words / TF-IDF)

Sure! Here are concise keywords for each topic to help you quickly identify and recall them:

**1. Time Series Analysis**

* Trend
* Seasonality
* Noise
* Forecasting
* Temporal data
* Patterns

**2. TF-IDF (Term Frequency - Inverse Document Frequency)**

* Term Frequency (TF)
* Inverse Document Frequency (IDF)
* Text weighting
* Document relevance
* Vector space model
* Keyword importance

**3. Clustering & K-means Algorithm**

* Unsupervised learning
* Euclidean distance
* Centroid
* Iterative refinement
* Partitioning
* Cluster assignment

**4. Confusion Matrix & Metrics**

* True Positive (TP)
* False Positive (FP)
* True Negative (TN)
* False Negative (FN)
* Accuracy
* Precision
* Recall (Sensitivity)
* Error Rate
* AUC-ROC

**5. Text Analysis / Processing**

* POS tagging
* Stemming
* Lemmatization
* Tokenization
* Natural Language Processing (NLP)
* Text normalization

**6. Model Validation Techniques**

* Holdout method
* Random subsampling
* K-fold cross validation
* Training set
* Validation set
* Test set
* Model evaluation

**7. Noise Handling in Text Data**

* Stop words removal
* Punctuation removal
* Lowercasing
* Spelling correction
* Bag of Words
* Text vectorization

Absolutely! Here’s a **side-by-side** view of **common keywords** and their **concise answers** for quick comparison and study:

| **Common Keywords** | **Common Answer** |
| --- | --- |
| **Time Series Analysis:**Trend, Seasonality, Noise, Forecasting | Studies data over time to find patterns like trends (long-term movement), seasonality (repeating cycles), and noise (random variation). Used for forecasting. |
| **TF-IDF:**Term Frequency, Inverse Document Frequency, Text weighting | Measures word importance in documents by combining term frequency (how often a word appears in a document) and inverse document frequency (rarity across documents). |
| **Clustering:**Unsupervised learning, K-means, Centroids, Euclidean distance | Groups similar data points into clusters. K-means initializes centroids, assigns points by nearest centroid using Euclidean distance, recalculates centroids, repeats. |
| **Confusion Matrix:**TP, FP, TN, FN, Accuracy, Precision, Recall, Error Rate | Evaluates classification results: accuracy = correct predictions / total, precision = TP/(TP+FP), recall = TP/(TP+FN), error rate = wrong predictions / total. |
| **Text Processing:**POS tagging, Stemming, Lemmatization | Prepares text for analysis. POS tagging labels word types, stemming cuts words to root forms, lemmatization converts words to dictionary base form considering context. |
| **Model Validation:**Holdout, Random Subsampling, K-fold CV, Train/Test/Validation sets | Techniques to assess model accuracy: holdout splits data once, random subsampling repeats splits, k-fold CV splits data into k parts for iterative training/testing. |
| **Noise Handling in Text:**Stop words removal, Punctuation removal, Lowercasing, Bag of Words, TF-IDF | Cleans text by removing irrelevant tokens and normalizing. Bag of Words counts word frequencies; TF-IDF weights words by importance for better text analysis. |

If you want, I can make this into a table image or PDF for easy reference. Would you like that?

Certainly! Here’s **one common comprehensive answer** that covers all the major concepts from your questions and topics, summarized into **9 clear points**:

**Comprehensive Common Answer (9 Points)**

1. **Time Series Analysis** studies data points ordered in time to identify trends (long-term movements), seasonality (periodic patterns), and noise (random fluctuations), helping in forecasting future values.
2. **TF-IDF (Term Frequency-Inverse Document Frequency)** is a text analysis technique that measures the importance of a word in a document relative to a corpus by combining how often the word appears in the document (TF) and how rare it is across all documents (IDF).
3. **Clustering** is an unsupervised learning method that groups data points based on similarity. The **k-means algorithm** initializes k cluster centers (centroids), assigns points to nearest centroids using Euclidean distance, recalculates centroids as the mean of assigned points, and repeats until centroids stabilize.
4. **Confusion Matrix** is a tool for evaluating classification models by comparing actual vs predicted labels and computing metrics such as Accuracy (correct predictions), Precision (correct positive predictions), Recall (sensitivity), and Error Rate (incorrect predictions).
5. **Text Processing** involves preparing text data for analysis through steps like Part-of-Speech (POS) tagging (labeling word types), stemming (reducing words to roots), and lemmatization (finding dictionary base forms considering context).
6. **Model Validation Techniques** include the Holdout method (single train-test split), Random Subsampling (multiple random splits), and K-Fold Cross Validation (splitting data into k folds and iteratively training/testing) to reliably evaluate model performance.
7. **Noise Handling in Text Data** removes irrelevant or misleading information such as stop words, punctuation, and case differences, improving analysis accuracy.
8. **Bag of Words** represents text as a vector of word frequencies without considering order, while **TF-IDF** weights these words to emphasize important and rare terms, enhancing text representation.
9. These combined techniques form the foundation for **data analytics, machine learning, and natural language processing**, enabling efficient data clustering, classification, and text mining for practical applications like disease risk prediction, customer segmentation, and document retrieval.

Got it! Here’s a **year-wise grouping** of questions into **topics**, showing which questions belong to which topic group along with their a/b parts.

**Year 1**

**Group 1: Text Analysis & Clustering**

* **Q5 (a) i)** Time series Analysis (Topic: Time Series)
* **Q5 (a) ii)** TF-IDF (Topic: Text Analytics)
* **Q5 (b)** What is clustering? Explain k-means algorithm (Topic: Clustering)

**Group 2: Model Evaluation & Validation**

* **Q6 (a) i)** Confusion matrix (Topic: Confusion Matrix & Metrics)
* **Q6 (a) ii)** AUC-ROC curve (Topic: Confusion Matrix & Metrics)
* **Q6 (b)** Holdout method and Random Sub Sampling methods (Topic: Model Validation Techniques)

**Year 2**

**Group 1: Clustering & Text Processing**

* **Q5 (a)** K-means clustering with given points (Topic: Clustering)
* **Q5 (b)** Part-of-speech tagging, Lemmatization, Stemming (Topic: Text Processing)

**Group 2: Confusion Matrix & TF-IDF**

* **Q6 (a)** Calculate accuracy, precision, recall, error rate from confusion matrix (Topic: Confusion Matrix & Metrics)
* **Q6 (b)** Explain TF-IDF in text analysis (Topic: Text Analytics)

**Year 3**

**Group 1: Text Processing & Clustering**

* **Q5 (a)** What is text processing? Explain TF-IDF (Topic: Text Analytics)
* **Q5 (b)** Steps involved in k-means algorithm (Topic: Clustering)

**Group 2: Confusion Matrix & Validation**

* **Q6 (a)** Define accuracy, precision, recall, AUC-ROC (Topic: Confusion Matrix & Metrics)
* **Q6 (b)** Explain k-fold Cross Validation & Random Subsampling (Topic: Model Validation Techniques)

**Year 4**

**Group 1: Clustering & Text Analysis**

* **Q5 (a)** K-means clustering first round with given points (Topic: Clustering)
* **Q5 (b)** Part-of-speech tagging and Lemmatization (Topic: Text Processing)

**Group 2: Confusion Matrix & Text Preprocessing**

* **Q6 (a)** Calculate accuracy, precision, recall, error rate from confusion matrix (Topic: Confusion Matrix & Metrics)
* **Q6 (b)** Explain text preprocessing steps (Topic: Text Processing)

**Year 5**

**Group 1: Clustering & Text Preprocessing**

* **Q5 (a)** Perform k-means clustering on 2D points, find new centroids (Topic: Clustering)
* **Q5 (b)** Handle noise and irrelevant info in text; explain Bag of Words and TF-IDF (Topic: Text Preprocessing & Analytics)

**Group 2: Hierarchical Clustering & Model Validation**

* **Q6 (a)** Hierarchical clustering for visualization and applications (Topic: Clustering - Hierarchical)
* **Q6 (b)** Holdout method and training/validation/test set explanation (Topic: Model Validation Techniques)

If you want, I can also prepare the detailed answers for each group!

**Summary Table**

| **Rank** | **Topic** | **Years Appeared** | **Main Questions Example** |
| --- | --- | --- | --- |
| 1 | Clustering | 1,2,3,4,5 | K-means steps and examples |
| 2 | Confusion Matrix & Metrics | 1,2,3,4 | Accuracy, Precision, Recall |
| 3 | Text Analysis / Processing | 1,2,3,4,5 | TF-IDF, POS tagging, Lemmatization |
| 4 | Model Validation Techniques | 1,3,5 | Holdout, Random subsampling, K-fold CV |
| 5 | Time Series Analysis | 1 | Definition and usage |
| 6 | Noise Handling / Bag of Words | 5 | Text preprocessing techniques |
| 7 | Hierarchical Clustering | 5 | Visualization & applications |

**Predicted que and ans in group for this year**

**🔹 Group 1: Clustering & Text Analytics**

**A. Clustering**

**1. What is clustering? Explain k-means steps with example.**

**Clustering is an unsupervised learning technique that partitions data into groups (clusters) so that points within a cluster are similar and points in different clusters are dissimilar.  
k-Means Algorithm Steps:**

1. **Choose k, the number of clusters.**
2. **Initialize k centroids (randomly or by selecting k points).**
3. **Assign each point to the nearest centroid (Euclidean distance).**
4. **Recompute each centroid as the mean of all points assigned to it.**
5. **Repeat steps 3–4 until assignments no longer change (or centroids stabilize).**

**Illustrative Example (k = 2):  
Points: (2,3), (3,3), (6,7), (7,8)  
Initial centroids: C₁=(2,3), C₂=(7,8)**

**Iteration 1:**

**Assign (2,3)→C₁, (3,3)→C₁, (6,7)→C₂, (7,8)→C₂**

**New C₁ = mean[(2,3),(3,3)] = (2.5,3)**

**New C₂ = mean[(6,7),(7,8)] = (6.5,7.5)**

**Iteration 2:**

**Reassign using updated centroids → same assignments → STOP**

**2. First-round k-means on given points**

**Data:  
A1(2,10), A2(2,5), A3(8,4), B1(5,8), B2(7,5), B3(6,4), C1(1,2), C2(4,9)  
Initial centroids: A1(2,10), B1(5,8), C1(1,2)**

**Step 1: Assign to nearest centroid  
Compute Euclidean distances quickly:**

| **Point** | **to A1(2,10)** | **to B1(5,8)** | **to C1(1,2)** | **Assigned Cluster** |
| --- | --- | --- | --- | --- |
| **A1** | **0** | **√13≈3.6** | **√65≈8.1** | **A1** |
| **A2** | **5** | **√18≈4.2** | **√10≈3.2** | **C1** |
| **A3** | **√52≈7.2** | **√25=5** | **√73≈8.5** | **B1** |
| **B1** | **√13≈3.6** | **0** | **√41≈6.4** | **B1** |
| **B2** | **√29≈5.4** | **√9=3** | **√45≈6.7** | **B1** |
| **B3** | **√36=6** | **√16=4** | **√29≈5.4** | **B1** |
| **C1** | **√65≈8.1** | **√41≈6.4** | **0** | **C1** |
| **C2** | **√5≈2.2** | **√10≈3.2** | **√50≈7.1** | **A1** |

**Clusters after 1st round:**

* **Cluster A1: {A1, C2}**
* **Cluster B1: {A3, B1, B2, B3}**
* **Cluster C1: {A2, C1}**

**Step 2: Compute new centroids:**

* **New A1 = mean[(2,10),(4,9)] = (3,9.5)**
* **New B1 = mean[(8,4),(5,8),(7,5),(6,4)] = (6.5,5.25)**
* **New C1 = mean[(2,5),(1,2)] = (1.5,3.5)**

**3. Full k-means convergence example**

**Data: A2(3, ?), B4(?,7), C3(3,5), D6(6,9), E8(8,6), F7(7,8) *(assuming missing coords: A2(3,5))*  
Initial centroids: µ₁=(2,3), µ₂=(8,6)**

**Iterate:**

1. **Assign points to nearest µ**
2. **Recompute µ**
3. **Repeat until µ stabilize**

***(You’d show two–three iterations similarly to above.)***

**4. Hierarchical Clustering**

**Definition: Builds nested clusters by either merging (agglomerative) or splitting (divisive), producing a dendrogram.**

**Agglomerative Steps:**

1. **Start each point as its own cluster.**
2. **Merge the two closest clusters.**
3. **Update distance matrix.**
4. **Repeat until one cluster remains.**

**Dendrogram ASCII:**

**┌─────────┐**

**│ Cluster │**

**│ of │**

**│ All Pt. │**

**└───┬───┬─┘**

**│ │**

**┌──┴┐ ┌┴───┐**

**C1 C2 C3 C4**

**Applications:**

* **Gene expression analysis**
* **Document clustering (e.g., similar news articles)**
* **Market segmentation**

**B. Text Analytics & Processing**

**1. TF-IDF**

* **Term Frequency (TF):**

**TF(t,d)=# t in dtotal terms in d TF(t,d) = \frac{\#\,t\text{ in }d}{\text{total terms in }d}**

* **Inverse Document Frequency (IDF):**

**IDF(t)=log⁡N1+# docs containing t IDF(t) = \log\frac{N}{1 + \#\,\text{docs containing }t}**

* **TF-IDF:**

**TFIDF(t,d)=TF(t,d)×IDF(t) TFIDF(t,d) = TF(t,d)\times IDF(t)**

**Example:  
Corpus of 3 docs, term “cat”:**

* **TF in Doc2 (“the cat sat on the mat”): 1/6**
* **IDF = log(3/2)=0.176**
* **TF-IDF ≈ 0.166×0.176≈0.029**

**2. Bag of Words vs. TF-IDF**

* **Bag of Words: Vector of raw word counts.**
* **TF-IDF: Weighted vector emphasizing rare-but-important terms.**

**3. Text Preprocessing Pipeline**

**Raw Text**

**↓ Lowercase**

**↓ Remove punctuation & digits**

**↓ Tokenization**

**↓ Remove stop-words**

**↓ Stemming / Lemmatization**

**↓ Vectorization (BOW / TF-IDF)**

**4. Noise & Irrelevant Data Handling**

* **Remove stop-words (“the”, “is”…)**
* **Strip punctuation, HTML tags, numbers**
* **Correct misspellings**
* **Normalize contractions (“don’t”→“do not”)**

**5. POS Tagging, Stemming, Lemmatization**

* **POS Tagging: Label tokens by part of speech (NN, VB, JJ…)**
* **Stemming: Heuristic chopping (e.g., “running”→“run”)**
* **Lemmatization: Dictionary-based root (“better”→“good”)**

**🔹 Group 2: Model Evaluation & Validation**

**A. Confusion Matrix & Metrics**

**1. Confusion Matrix**

|  | **Predicted Positive** | **Predicted Negative** |
| --- | --- | --- |
| **Actual Positive** | **True Positive (TP)** | **False Negative (FN)** |
| **Actual Negative** | **False Positive (FP)** | **True Negative (TN)** |

**2. Metrics Formulas**

* **Accuracy: (TP+TN)/(TP+FP+FN+TN)(TP + TN)/(TP+FP+FN+TN)**
* **Precision: TP/(TP+FP)TP/(TP + FP)**
* **Recall: TP/(TP+FN)TP/(TP + FN)**
* **Error Rate: (FP+FN)/(TP+FP+FN+TN)(FP + FN)/(TP+FP+FN+TN)**

**3. Example Calculation**

**Given matrix for disease risk:**

**Pred Yes Pred No**

**Actual Yes 80 220**

**Actual No 150 9500**

* **TP=80, FP=150, FN=220, TN=9500**
* **Accuracy = (80+9500)/ (80+150+220+9500) ≈ 0.963**
* **Precision = 80/(80+150) ≈ 0.348**
* **Recall = 80/(80+220) = 0.267**
* **Error rate = (150+220)/ (total) ≈ 0.037**

**4. AUC-ROC Curve**

* **ROC Curve: plots True Positive Rate (Recall) vs. False Positive Rate (FP/(FP+TN)).**
* **AUC: Area under ROC; closer to 1 ⇒ better model.**

**TPR**

**1 ┌────────╮**

**│ ╰───**

**│ ╱**

**0 └─┬──┬──┬─┬─> FPR**

**0 .2 .5 1**

**B. Model Validation Techniques**

**1. Holdout Method**

* **Split data once into Training and Test sets (e.g., 70/30).**
* **Train on training; evaluate on test.**

**2. Random Subsampling**

* **Repeat holdout split multiple times with different random splits.**
* **Average performance metrics.**

**3. K-Fold Cross Validation**

1. **Split data into k equal folds.**
2. **For each fold i=1..k:**
   * **Train on k−1 folds, test on fold i.**
3. **Average the k results.**

**Folds: [1][2][3][4][5]**

**Iter1: train 2,3,4,5 → test 1**

**Iter2: train 1,3,4,5 → test 2**

**...**

**4. Train / Validation / Test Sets**

* **Training Set: used to learn model parameters.**
* **Validation Set: used to tune hyperparameters and prevent overfitting.**
* **Test Set: final, untouched evaluation of model generalization.**

**🎓 With these detailed explanations and diagrams, you’ll be well prepared for any clustering, text-analytics, evaluation-metrics, or validation-technique question!**