Data Mining Viva Prep

Sure Aditya, here's your **viva prep** with each topic explained simply using bullet points and examples where needed:

---

1. K-Means

- K-Means is a **clustering algorithm** used in unsupervised learning.
- It groups data into **K number of clusters** based on similarity.
- Each cluster has a **centroid**, and data points are grouped by nearest centroid.
- It minimizes the **intra-cluster variance**.

---

2. K-Means Algorithm

- **Choose K** cluster centers randomly (initial centroids).
- **Assign** each data point to the nearest centroid using distance (usually Euclidean).
- **Recalculate** centroids as the mean of all assigned points.
- **Repeat** steps 2-3 until centroids don't change or reach a max iteration.

Example: Cluster students based on marks in 2 subjects.

---

3. Confusion Matrix

- A table to evaluate classification model performance.
- Rows: **Actual values**, Columns: **Predicted values**.
- Shows **TP, TN, FP, FN** clearly.
- Helps calculate metrics like accuracy, precision, recall.

Example: 2x2 matrix for binary classification.

---

4. Accuracy

- Tells how many predictions were correct out of total.
- Formula:
  $\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$
- Good when classes are balanced.
- Misleading if class imbalance exists.

---

5. Precision

- Tells out of predicted positives, how many were correct.
- Formula:
  $\text{Precision} = \frac{TP}{TP+FP}$
- Useful when false positives are costly.
- Example: In spam detection, don't mark important mail as spam.

---

6. TP, TN, FP, FN

- **TP (True Positive):** Correctly predicted positive (actual = 1, predicted = 1).
- **TN (True Negative):** Correctly predicted negative (actual = 0, predicted = 0).
- **FP (False Positive):** Incorrectly predicted positive (actual = 0, predicted = 1).
- **FN (False Negative):** Incorrectly predicted negative (actual = 1, predicted = 0).

---

7. Preprocessing

- Steps to clean and prepare data before mining.
- Common steps:
  - **Missing value handling** (mean, drop)
  - **Normalization** (min-max scaling)
  - **Encoding** categorical values

- **Noise removal** and **outlier detection**

---

8. F1 and Recall Formula

- **Recall**:
  $\text{Recall} = \frac{TP}{TP+FN}$
  - Measures actual positives that are correctly predicted.
- **F1 Score**:
  $F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$
  - Harmonic mean of precision and recall.
  - Useful for imbalanced datasets.

---

9. Outlier, Noise

- **Outlier:**
  - Data point far from others.
  - May be real or error.
  - Example: A person earning ₹1 crore in a ₹30k avg income dataset.
- **Noise:**
  - Random error or variance in data.
  - Affects accuracy.
  - Example: Typing mistake in sensor data or user entry.

---

10. Data Visualization Tools

- Used to visually represent data and patterns.
- Popular tools:
  - **Tableau** – drag-drop interface, dashboards
  - **Power BI** – Microsoft tool, real-time analytics

- **Matplotlib / Seaborn (Python)** – for plots, graphs
- **Excel** – easy charts, pivot tables

Let me know if you want me to convert this into a PDF or add expected viva questions for revision.

Sure Aditya! Here's the continuation of the viva prep — answering only the questions **not covered earlier**. Everything is explained briefly with examples where helpful:

🔶 1. Association Rule Mining (Apriori)

1. What is the Apriori algorithm? How does it work?

- Used to find **frequent itemsets** and generate association rules.
- **Apriori property**: If an itemset is frequent, all its subsets are also frequent.
- Steps:
  - Find frequent itemsets using min support.
  - Generate rules from those itemsets using min confidence.

2. Define support, confidence, and lift.

- **Support**: Frequency of itemset in dataset.
  $\text{Support}(A) = \frac{\text{Transactions containing A}}{\text{Total transactions}}$
- **Confidence**: How often B appears when A does.
  $\text{Conf}(A \Rightarrow B) = \frac{\text{Support}(A \cup B)}{\text{Support}(A)}$
- **Lift**: Strength of rule compared to random co-occurrence.
  $\text{Lift}(A \Rightarrow B) = \frac{\text{Confidence}(A \Rightarrow B)}{\text{Support}(B)}$

3. What is the role of minimum support in association rule mining?

- **Filters out** rare itemsets.
- Reduces number of candidates → improves efficiency.
- Low min-support = more rules, but higher processing time.

4. Why do we sort rules by lift?

- Lift shows **strength beyond chance**.
- Higher lift = more meaningful rule.
- Lift > 1 → items occur **together more** than independently.

5. How do you interpret a rule like {butter} ⇒ {bread}?

- If someone buys **butter**, they are **likely to buy bread**.
- Use support/confidence/lift to validate rule.
- Helpful in **market basket analysis**.

6. What are frequent itemsets?

- Sets of items that **appear together often**.
- Satisfy **minimum support threshold**.
- Basis for rule generation.

7. What does a lift value > 1 signify?

- A and B **occur together more often** than by chance.
- Rule is **positively correlated**.

8. How is Apriori different from FP-Growth?

- Apriori: generates **candidates** → scans DB multiple times.
- FP-Growth: uses **tree structure** → avoids candidate generation.
- FP-Growth is **faster and memory-efficient**.

9. What type of datasets are suitable for association rule mining?

- **Transactional datasets**.
- Examples: Retail purchases, web clickstreams.
- Works well when items are **discrete and categorical**.

10. Can we generate rules from 1-item frequent sets?

- No, rules require **at least 2 items** to form **antecedent ⇒ consequent**.
- Need itemsets with **2 or more items** to make a rule.

🔶 2. Naive Bayes Classifier (SMS Spam Detection)

1. What is the Naive Bayes assumption?

- Features are **independent given the class**.
- Example: Words in a message are assumed independent.

2. Why is it called "naive"?

- Because of the **strong assumption** of feature independence.
- Real-world data often violates this.

3. Which Naive Bayes variant is used for text classification?

- **Multinomial Naive Bayes**.
- Works well with **word counts and frequency**.

4. What is the formula for Bayes' Theorem?

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

- Used to compute **posterior probability**.

5. What is Laplace smoothing and why is it used?

- Adds **1 to each count** to avoid zero probabilities.
- Especially useful if a word never appears in training for a class.

6. Why is text preprocessing important in spam classification?

- Reduces **noise and dimensionality**.
- Steps: lowercasing, removing stop words, stemming.
- Improves classifier accuracy.

7. How do we evaluate a spam classifier? What metrics do we use?

- **Confusion matrix**, **precision**, **recall**, **F1-score**, and **accuracy**.
- Precision important to **avoid mislabeling good messages as spam**.

10. What will your model predict for the message: "Congratulations, you've won a free trip!"?

- Likely to classify as **spam**.

- Words like “congratulations”, “free”, “won” have high spam probability.

🔶 3. Decision Tree Classifier (Iris Dataset)

1. What is a decision tree and how does it work?

- A **tree-like model** for classification or regression.
- Splits data on features using **conditions**.
- Ends in **leaf nodes** showing predicted class.

2. What’s the difference between Gini index and entropy?

- Both measure **impurity**.
- **Entropy**:
  $-p \log_2 p - q \log_2 q$
- **Gini index**:
  $1 - (p^2 + q^2)$
- Gini is **faster**, entropy more **information-theoretic**.

3. How does a decision tree decide which feature to split?

- Based on **information gain** (using entropy) or **Gini gain**.
- Chooses feature that best **separates classes**.

4. What is overfitting in decision trees?

- Model becomes **too complex**.
- Performs well on training but **poor on test data**.

5. How can we avoid overfitting?

- **Pruning**, **limit depth**, **minimum samples** for split/leaf.
- Use **cross-validation**.

6. What is pruning? What are its types?

- Removing **extra nodes** to avoid overfitting.
- **Pre-pruning**: Stop early (max depth, min samples).
- **Post-pruning**: Cut back after full growth.

7. What is the depth of a tree?

- **Number of levels** from root to deepest leaf.
- High depth = more complexity.

8. What does a leaf node represent?

- A **class label** (in classification).
- Final prediction for that path.

9. How do you interpret a decision tree diagram?

- **Follow conditions** from root to leaf.
- Each node = split based on feature value.

10. What will be the predicted class for sepal=5.1, petal=1.5?

- Depends on **trained tree path**.
- Likely class: **setosa**, since petal length is low.

🔶 5. Hierarchical Clustering (Wholesale Customers)

1. What is hierarchical clustering?

- Creates **nested clusters** in a tree format.
- Doesn't need pre-defined number of clusters.
- Can be **agglomerative** (bottom-up) or **divisive** (top-down).

2. What is the difference between agglomerative and divisive clustering?

- **Agglomerative**: Start with single points, merge step-by-step.
- **Divisive**: Start with all points, split recursively.

3. What is a dendrogram?

- A **tree diagram** showing clustering process.
- Y-axis = distance at which clusters merge.

4. How do you interpret a dendrogram?

- **Cut at a certain height** to get desired clusters.

- Lower height = more fine clusters.

5. What is linkage? Name types of linkage methods.

- Linkage = how to measure **distance between clusters**.
- Types:
  - **Single** (min)
  - **Complete** (max)
  - **Average**
  - **Ward's method** (minimize variance)

6. What is the difference between complete and single linkage?

- **Single**: Minimum distance between any two points in clusters.
- **Complete**: Maximum distance between any two points.

7. What distance metric did you use and why?

- Common: **Euclidean** for continuous data.
- Chosen for **simplicity and interpretability**.

8. What happens if we cut the dendrogram at a higher height?

- Fewer clusters are formed.
- Merges clusters that are more distant.

9. How does hierarchical clustering differ from K-Means?

- No need to predefine **k**.
- Produces **hierarchy**, not flat clusters.
- Slower on large datasets.

10. Is hierarchical clustering suitable for large datasets?

- Not very efficient for **large datasets**.
- **Time complexity is high**, use K-means instead.

Let me know if you want this in a summarized or tabular format for last-minute revision!