**Comprehensive Viva Questions and Answers for Data Mining Practicals**

**1. Data Handling**

**Q1. What is a DataFrame in Pandas? How is it different from a Series?** A DataFrame is a 2-dimensional labeled data structure with columns of potentially different types. A Series is a 1D labeled array capable of holding any data type.

**Q2. How do you load different types of files (CSV, Excel, JSON) into a DataFrame?**

* CSV: pd.read\_csv('file.csv')
* Excel: pd.read\_excel('file.xlsx')
* JSON: pd.read\_json('file.json')

**Q3. What are different ways to inspect a DataFrame?**

* df.head() – first 5 rows
* df.tail() – last 5 rows
* df.columns – list of column names
* df.shape – dimensions of the DataFrame
* df.info() – metadata
* df.describe() – statistical summary

**Q4. How do you select rows and columns?**

* By label: df.loc[row\_label, column\_label]
* By index: df.iloc[row\_index, column\_index]
* Entire column: df['column']

**Q5. How do you add or delete a column?**

* Add: df['new\_col'] = data
* Delete: df.drop('col\_name', axis=1, inplace=True)

**Q6. How do you perform statistical operations?**

* Mean: df.mean()
* Median: df.median()
* Mode: df.mode()
* Std Deviation: df.std()
* Variance: df.var()
* Skewness: df.skew()
* Kurtosis: df.kurt()

**Q7. How do you sort and filter data?**

* Sort: df.sort\_values(by='column')
* Filter: df[df['column'] > value]

**Q8. How do you perform grouping and aggregation?**

df.groupby('column').mean()

df.groupby('column').agg({'col1': 'sum', 'col2': 'mean'})

**2. Data Preprocessing 01: Data Cleaning**

**Q1. What are common types of data quality issues?**

* Missing values
* Duplicate entries
* Inconsistent formatting
* Outliers
* Typographical errors

**Q2. How do you detect and handle missing values?**

* Detect: df.isnull().sum()
* Replace: df.fillna(value)
* Interpolate: df.interpolate()
* Drop: df.dropna()

**Q3. What is the difference between isnull() and notnull()?**

* isnull(): Detects missing values (returns True for NaN)
* notnull(): Opposite, returns True for non-null values

**Q4. What is data imputation? List techniques.**

* Mean, median, or mode imputation
* Forward fill (ffill) and backward fill (bfill)
* Interpolation
* Model-based imputation (using regression or KNN)

**Q5. How do you detect and handle duplicates?**

* Detect: df.duplicated()
* Drop: df.drop\_duplicates()

**Q6. How do you handle inconsistent data types or formats?**

* Convert types: df['column'] = df['column'].astype(type)
* Parse dates: pd.to\_datetime(df['date\_column'])

**3. Data Preprocessing 02: Integration, Normalization, Reduction**

**Q1. What is data integration? Give an example.** Combining data from multiple sources into a single dataset. E.g., merging customer details from CRM and sales records.

pd.merge(df1, df2, on='common\_column')

**Q2. What are the challenges in data integration?**

* Redundancy
* Schema mismatch
* Data inconsistency
* Conflicting formats

**Q3. What is normalization? Why is it important?** Rescaling data to a common scale to avoid dominance of one feature over others.

**Q4. Types of normalization techniques:**

* Min-Max scaling
* Z-score standardization
* Decimal scaling

**Q5. How do you implement Min-Max and Z-score normalization in Python?**

from sklearn.preprocessing import MinMaxScaler, StandardScaler

MinMaxScaler().fit\_transform(df[['col']])

StandardScaler().fit\_transform(df[['col']])

**Q6. What is data reduction? What are its types?** Reducing volume but maintaining integrity.

* Dimensionality reduction (PCA, LDA)
* Data compression
* Numerosity reduction (histograms, clustering)

**Q7. What is PCA and how does it work?** Principal Component Analysis reduces features while preserving variance. It projects data to a lower-dimensional space.

**4. Data Warehousing & OLAP**

**Q1. Define OLAP. What are its operations?** OLAP: Online Analytical Processing for multidimensional analysis. Operations:

* Roll-up (increasing aggregation)
* Drill-down (more detail)
* Slice (select one dimension)
* Dice (select multiple dimensions)
* Pivot (re-orientation of data)

**Q2. What is a data warehouse?** A subject-oriented, integrated, time-variant, and non-volatile collection of data for decision-making.

**Q3. Explain star and snowflake schema with examples.**

* Star: Fact table in the center connected to dimension tables directly
* Snowflake: Dimension tables are normalized

**Q4. What are fact and dimension tables?**

* Fact: Contains numeric performance data (e.g., revenue)
* Dimension: Describes context (e.g., time, region, product)

**Q5. What is a factless fact table?** A table that captures events but has no measurable data. Example: student attendance.

**Q6. What are OLAP types?**

* ROLAP: Relational OLAP (uses relational DBMS)
* MOLAP: Multidimensional OLAP (uses cubes)
* HOLAP: Hybrid OLAP (combination of both)

**5. ETL using Tableau/PowerBI/Statistica**

**Q1. What is ETL and why is it important?**

* Extract: Fetch data
* Transform: Clean, normalize, filter
* Load: Move into warehouse or BI tool

**Q2. How do you connect data sources in Tableau/PowerBI?** Use connectors like Excel, CSV, MySQL, SQL Server, Google Sheets, APIs.

**Q3. What is a worksheet, dashboard, and story in Tableau?**

* Worksheet: Single view with charts
* Dashboard: Collection of worksheets
* Story: Sequence of dashboards with narration

**Q4. Explain filters and parameters in Tableau.**

* Filters: Restrict data in views
* Parameters: Dynamic input controls that change filter values

**Q5. What is data blending in Tableau?** Combining data from multiple sources with a common field using primary and secondary data sources.

**Q6. How do you perform transformation in Power BI?** Using Power Query Editor: remove columns, pivot/unpivot, replace values, format columns.

**6. A-Priori Algorithm & Association Rules**

**Q1. What is market basket analysis?** A technique to understand customer purchase patterns using association rules.

**Q2. Define support, confidence, lift, and conviction.**

* Support: Frequency of occurrence
* Confidence: Strength of implication
* Lift: Ratio of observed support to expected if independent
* Conviction: Measure of implication strength

**Q3. Steps in A-Priori algorithm:**

1. Set min support & confidence
2. Generate frequent itemsets
3. Generate association rules from them

**Q4. Python implementation with MLxtend:**

from mlxtend.frequent\_patterns import apriori, association\_rules

frequent\_items = apriori(df, min\_support=0.2, use\_colnames=True)

rules = association\_rules(frequent\_items, metric="lift", min\_threshold=1)

**Q5. Limitations of Apriori:**

* Generates large number of candidates
* Time-consuming on large datasets

**7. Classification**

**Q1. What is classification in machine learning?** Predicting categorical labels based on input features.

**Q2. What are popular classification algorithms?**

* Logistic Regression
* Decision Tree
* Random Forest
* KNN
* Naive Bayes
* SVM

**Q3. What is overfitting and underfitting?**

* Overfitting: Model too complex, fits training data too well
* Underfitting: Model too simple, fails to capture patterns

**Q4. What is cross-validation?** Technique to validate model performance by splitting data into train-test multiple times (e.g., K-Fold)

**Q5. Explain confusion matrix terms:**

* TP: True Positive
* TN: True Negative
* FP: False Positive
* FN: False Negative

**Q6. Evaluation metrics:**

* Accuracy = (TP+TN)/(TP+TN+FP+FN)
* Precision = TP / (TP + FP)
* Recall = TP / (TP + FN)
* F1-Score = 2 \* (Precision \* Recall) / (Precision + Recall)

**Q7. Example of Decision Tree using sklearn:**

from sklearn.tree import DecisionTreeClassifier

model = DecisionTreeClassifier()

model.fit(X\_train, y\_train)

pred = model.predict(X\_test)

**8. Clustering**

**Q1. What is clustering?** Grouping data points into clusters where members are similar.

**Q2. Types of clustering:**

* Partitioning (K-Means)
* Hierarchical (Agglomerative/Divisive)
* Density-based (DBSCAN)

**Q3. How does K-Means work?**

1. Choose K centers
2. Assign points to nearest cluster
3. Recalculate centroids
4. Repeat until convergence

**Q4. How to choose K in K-means?**

* Elbow method
* Silhouette score

**Q5. Hierarchical Clustering types?**

* Agglomerative (bottom-up)
* Divisive (top-down)

**Q6. What is DBSCAN?** Density-based clustering algorithm. Forms clusters based on density and detects noise.

**Q7. Python implementation of K-Means:**

from sklearn.cluster import KMeans

kmeans = KMeans(n\_clusters=3)

kmeans.fit(X)

labels = kmeans.labels\_