Data Science and Big Data Analytics Laboratory Oral Question Bank

Allocation of problem statement to students is on random basis with help of chits. (No change in program once selected)

Sample Oral Questions

1. Complete information of dataset used in given problem statement. Features of datasets, columns etc
   Ans:- **1. Iris**

- **Source & Size:** 150 samples, 4 numeric features, 3 classes.
- **Features:**
  - sepal length (cm)
  - sepal width (cm)
  - petal length (cm)
  - petal width (cm)
- **Target:** species (setosa, versicolor, virginica)

---

2. California Housing

- **Source & Size:** 20,640 blockgroup observations, 8 numeric features, continuous target (median house value in $100 000s).
- **Features:**
  - MedInc – median income in block group
  - HouseAge – median house age
  - AveRooms – average rooms per household
  - AveBedrms – average bedrooms per household
  - Population – block group population
  - AveOccup – average household size
  - Latitude – block group latitude
  - Longitude – block group longitude
- **Target:** MedHouseVal – median house value

---

3. Student Performance (UCI "Student Performance" Data Set)

- **Source & Size:** Two related files ("mat" and "por") with 395 and 649 students respectively, **33** attributes, no missing values.
- **Feature categories:**
  - **Demographics:** school, sex, age, address, famsize, Pstatus, guardian
  - **Parental & Social:** Medu, Fedu, Mjob, Fjob, reason, nursery, internet, romantic
  - **Academic & Support:** traveltime, studytime, failures, schoolsup, famsup, paid, activities
  - **Lifestyle & Health:** famrel, freetime, goout, Dalc, Walc, health
  - **Attendance & Grades:** absences, G1, G2, G3 (first, second, final

period grades)

- **Use cases:** classification (pass/fail) and regression (predict final grade).

---

4. Titanic

- **Source & Size:** 891 passengers, 15 columns.
- **Features:**
  - **Demographic & Class:** survived, pclass, sex, age, sibsp, parch, fare, embarked
  - **Derived & Categorical:** class, who, adult_male, deck, embark_town, alive, alone
- **Use cases:** classification of survival, EDA on class, age, fare distributions.

---

5. Social_Network_Ads

- **Source & Size:** 400 users, 5 columns.
- **Features:**
  - User ID (int)
  - Gender (object: Male/Female)
  - Age (int)
  - EstimatedSalary (int)
  - Purchased (0/1 target)
- **Use cases:** binary classification (will purchase or not) based on demographic and salary.

2. Complete information of python libraries used in Data science. For example pandas, numpy, matplotlib, seaborn.
   a. What is use of these libraries

   Ans:-

   · **pandas**: loading, cleaning, transforming, and aggregating tabular data; ideal for data wrangling and analysis.
   · **NumPy**: fast, multi-dimensional arrays and mathematical operations; underpins pandas and many ML libraries.
   · **Matplotlib**: lowlevel plotting library for creating line, scatter, bar, histogram, and custom charts.
   · **Seaborn**: highlevel interface for statistical graphics; built on Matplotlib, simplifies complex plots (heatmaps, violin, pair plots).

   b. What are different functions of these libraries

Ans:-

- **pandas**
  - read_csv(), read_excel() – import data
  - head(), tail() – preview rows
  - describe(), info() – summary statistics and types
  - isnull(), fillna(), dropna() – missingvalue handling
  - groupby(), pivot_table(), merge() – aggregation and joins
- **NumPy**
  - array(), arange(), linspace() – create arrays
  - reshape(), flatten() – change array shapes
  - mean(), std(), sum(), min(), max() – basic statistics

- dot(), matmul() – linear algebra
- random.rand(), random.choice() – random data

- **Matplotlib** (pyplot API)
  - plot(), scatter(), bar(), hist() – basic chart types
  - xlabel(), ylabel(), title(), legend() – annotate plots
  - subplots() – multiple plots in one figure
  - savefig() – export images
- **Seaborn**
  - histplot(), kdeplot() – univariate distributions
  - boxplot(), violinplot() – distribution comparisons
  - heatmap() – correlation matrices
  - pairplot() – scatterplot matrix
  - countplot(), barplot() – categorical visualization

  These four libraries together form the core toolkit for data ingestion, cleaning, analysis, and visualization in Python.

  o4-mini

3. From where do we get the standard datasets

   Ans:-

   Standard datasets are available from:

- **Builtin libraries**: sklearn.datasets (iris, Boston, diabetes), Seaborn (titanic, flights).
- **Online repositories**: Kaggle, UCI Machine Learning Repository, OpenML.
- **Government/Research portals**: data.gov, World Bank, AWS Public Datasets.
- **Academic GitHub** and Google Dataset Search.

4. How do we fill missing values? What are techniques used for same? What are functions given by pandas to fill the missing values

   Ans :- Missing values can be filled using mean, median, mode, or a fixed value. Pandas provides fillna(), interpolate(), dropna() for handling them. For example, df.fillna(df.mean()) replaces NaNs with column means. This ensures models don't fail due to incomplete data.

5. How to handle the inappropriate data and inconsistencies in the preprocessing phase?

   Ans :- Inappropriate data (like negative age) is handled by validating and correcting or removing such entries. Use drop_duplicates(), replace(), and type conversion with astype() to fix inconsistencies. Also, visually inspect data using df.describe() and df.info() to catch outliers and anomalies.

6. How to turn categorical variables into quantitative variables in Python

Ans :- Categorical variables can be converted using:

- LabelEncoder (converts categories to numbers),

- pd.get_dummies() (one-hot encoding),
  This is essential for machine learning models which need numeric inputs. For example, converting 'Male'/'Female' into 0/1 or into separate binary columns.

7. How to do data Normalization
   Ans:- Normalization scales all values between 0 and 1 using MinMaxScaler() from sklearn. Standardization (Z-score scaling) with StandardScaler() makes data have mean 0 and standard deviation 1. It is useful when features have different scales, as in logistic regression or SVM models.

8. How to handle outliers
   Ans:- Outliers can be detected using statistical methods like Z-score (>3 or <-3), IQR method (1.5×IQR rule), or boxplots. Once found, they can be removed, capped, or replaced. Use pandas or seaborn's boxplot() to visualize and df[(z < 3).all(axis=1)] to filter.

9. What is skewness?

Ans:- Skewness measures how asymmetric the data distribution is.

- Positive skew: tail on right
- Negative skew: tail on left
  Use df.skew() in pandas to find skewness. Normally distributed data has skewness ≈ 0. High skew may require transformation (e.g., log, square root).

10. What is mean, median, minimum, maximum, standard deviation? How to handle it in python libraries

Ans:-

· Mean: average (df.mean())
· Median: middle value (df.median())
· Min/Max: smallest/largest values (df.min(), df.max())
· Std: spread from mean (df.std())
  These help understand data distribution and are crucial for normalization and outlier detection

11. How to provide summary statistics of income grouped by the age groups.

Ans:- First, create age bins (e.g. df['age_group'] = pd.cut(df.age, bins=[0,20,40,60,80], labels=…)). Then:

python

```
df.groupby('age_group')['income'].agg(['mean','median','min','max','std','count'])
```

This returns count, mean, median, min, max, and standard deviation of income for each age group.

12. Which libraries are used to display some basic statistical details like percentile, mean, standard deviation

Ans:- · **pandas**: df.describe() gives count, mean, std, min, quartiles, max.

· **numpy**: np.percentile() for custom percentiles, np.mean(), np.std().

· **scipy.stats**: functions like skew(), kurtosis(), and advanced stats.

13. What is Linear Regression Model? How to design it via python library

Ans:- Linear Regression finds the best-fit straight line relating predictors (X) to a continuous target (y). In Python:

python

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

Evaluate with MSE and R^2.

14. What is logistic regression? What is advantage of it?

Ans:- Logistic Regression is a classification algorithm that models the probability of a binary outcome via the sigmoid function.

Advantage: It outputs class probabilities, is simple to implement, interpretable (weights indicate feature impact), and works well as a baseline for binary classification.

15. What is confusion matrix? How to calculate TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall?

Ans:- A Confusion Matrix is a 2×2 table of predicted vs actual classes:

- **TP** (true positives): predicted 1 & actual 1
- **TN** (true negatives): predicted 0 & actual 0
- **FP**: predicted 1 & actual 0
- **FN**: predicted 0 & actual 1

  Compute via confusion_matrix(y_true, y_pred).
- **Accuracy** = (TP + TN) / Total
- **Error rate** = (FP + FN) / Total
- **Precision** = TP / (TP + FP)
- **Recall** = TP / (TP + FN)

16. What is Simple Naïve Bayes classification algorithm

Ans:- Naïve Bayes applies Bayes' Theorem with a strong assumption that features are independent. It computes class probabilities as less

$$P(\text{Class}|\text{Features}) \propto P(\text{Class}) \times \prod P(\text{Feature}_i|\text{Class})$$

In Python, use GaussianNB() from sklearn.naive_bayes for continuous data like Iris.

17. What is Tokenization, POS Tagging, stop words removal, Stemming and Lemmatization, Term Frequency and Inverse Document Frequency.

- Ans:- **Tokenization**: splitting text into words/sentences.
- **POS Tagging**: labeling tokens with grammatical tags (noun, verb, etc.).
- **Stopword Removal**: dropping common words ("the", "is").
- **Stemming**: chopping words to root form (e.g. "fishing" → "fish").
- **Lemmatization**: converting words to valid dictionary base forms.
- **Term Frequency (TF)**: count of a term in a document.
- **Inverse Document Frequency (IDF)**: log-scaled inverse proportion of documents containing the term.
  Their product (TF–IDF) weights terms by importance.

18. What is histogram? How to draw histogram using seaborn library

Ans:- A histogram visualizes the distribution of a numeric variable by binning values into bars. In Seaborn:

python

```
import seaborn as sns
sns.histplot(data=df, x='fare', bins=30)
```

This plots the frequency of ticket prices in the Titanic dataset.

19. What is box plot? How to implement it

Ans:- A box plot (box-and-whisker) shows a distribution's quartiles and outliers. In Seaborn:

python

```
sns.boxplot(x='sex', y='age', hue='survived', data=df)
```

This displays age distribution by gender and survival status, highlighting median, IQR, and outliers.

20. What are attribute types? (e.g., numeric, nominal)

Ans:- · **Numeric**: quantitative values (integers, floats).

· **Nominal**: categorical without order (colors, gender).

· **Ordinal**: categorical with order (small, medium, large).

· **Binary**: two-category variables (yes/no).
  Understanding types guides encoding and analysis methods.

21. How to read text input using python libraries

- Ans:- **Built-in**:

python

```
with open('file.txt') as f:
    text = f.read()
```

- **pandas** for structured text:

```
python

df = pd.read_csv('data.tsv', sep='\t')
```

- **NLTK**: nltk.corpus or file readers for NLP preprocessing.

22. How to do map-reduce programming? What is advantage of it

Ans:- MapReduce divides tasks into **Map** (process/transform data chunks in parallel) and **Reduce** (aggregate results). Frameworks like Hadoop or PySpark implement it.
Advantage: scales to massive datasets across clusters, handles failures, and simplifies parallel computation.

23. What is a scala programming? features of scala programming

Ans:- Scala is a JVM language integrating object-oriented and functional paradigms.
Features:

- Statically typed with type inference
- Immutable collections and pattern matching
- Concurrency support via Actors/Akka
- Full Java interoperability
- Suited for big data (e.g., Apache Spark).

24. What is scikit-learn library?

Ans:- scikitlearn is a Python library offering simple, efficient tools for machine learning and data mining. It includes algorithms for classification, regression, clustering, dimensionality reduction, preprocessing, and model selection, all with a consistent API and good documentation for rapid prototyping and production use.