1) What is Pandas library in Python?

Pandas is a powerful open-source data manipulation and analysis library. It provides data structures like Series (1D) and DataFrame (2D) for handling structured data easily.

✅ **2) Key Features of Pandas:**

- **Fast and efficient DataFrame object**
- **Tools for reading/writing data**
- **Handling of missing data**
- **Label-based slicing, indexing**
- **Data alignment and reshaping**

✅ **3) What is Numpy?**

NumPy (Numerical Python) is a library for numerical computing. It supports large, multi-dimensional arrays and provides mathematical functions to operate on them efficiently.

✅ **4) What is matplotlib?**

Matplotlib is a data visualization library that allows creating static, animated, and interactive plots in Python.

✅ **5) Seaborn vs Matplotlib**

- **Matplotlib is a low-level plotting library.**
- **Seaborn is built on top of Matplotlib, and provides more attractive and informative statistical graphics.**

✅ **6) Is Sklearn same as Scikit-learn?**

Yes, Sklearn is the import name of the Scikit-learn library. It's used for machine learning tasks like classification, regression, clustering, etc.

✅ 7) Functions in Pandas and NumPy:

Pandas: read_csv(), head(), tail(), info(), describe(), groupby(), merge()
NumPy: array(), mean(), std(), reshape(), linspace(), random.rand()

✅ 8) What is a DataFrame in Python?

A DataFrame is a 2D labeled data structure with rows and columns. It's like a table or Excel sheet in memory.

✅ 9) How to find duplicates?

python

CopyEdit

df.duplicated()

df[df.duplicated()]

✅ 10) Use of describe()

Gives summary statistics of numerical columns – count, mean, std, min, max, etc.

✅ 11) Naive Bayes algorithms used:

- **GaussianNB**
- **MultinomialNB**
- **BernoulliNB**
  From sklearn.naive_bayes

✅ 12) Significance of Confusion Matrix

It evaluates the performance of a classification model by comparing predicted vs actual labels.

✅ 13) TP, TN, FP, FN

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

✅ 14) What is Recall?

Recall = TP / (TP + FN)
It measures how many actual positives were correctly identified.

✅ 15) What is Precision?

Precision = TP / (TP + FP)
It measures how many predicted positives were actually correct.

✅ 16) What is F1 Score?

Harmonic mean of Precision and Recall.
F1 = 2 × (Precision × Recall) / (Precision + Recall)

✅ 17) Why is data visualization needed?

To explore, understand, and communicate patterns and trends in data.

✅ 18) What is an Outlier?

An extreme data point that differs significantly from others. Can skew results and affect model performance.

✅ 19) Histogram vs Pie Chart

- **Histogram: To show frequency distribution of numerical data.**

- **Pie Chart: To show percentage or proportion of categorical data.**

✅ **20) Challenges in Big Data Visualization**

- **High volume and variety**
- **Real-time updates**
- **Performance and scalability**
- **User interactivity**

✅ **21) Jointplot and Distplot**

- **jointplot(): Shows bivariate plot with marginal histograms**
- **distplot() (deprecated): Shows distribution of a variable**

✅ **22) Tools for Data Visualization**

Matplotlib, Seaborn, Plotly, Power BI, Tableau, D3.js

✅ **23) What is Data Wrangling?**

Cleaning and transforming raw data into usable format for analysis.

✅ **24) What is Data Transformation?**

Converting data from one format or structure to another (e.g., scaling, encoding).

✅ **25) StandardScaler in Python?**

It standardizes features by removing the mean and scaling to unit variance.

python

CopyEdit

from sklearn.preprocessing import StandardScaler

✅ 26) What is Hadoop?

An open-source framework for distributed storage and processing of big data using MapReduce.

✅ 27) HDFS and MapReduce

- **HDFS: Hadoop Distributed File System**
- **MapReduce: Programming model for distributed computation**

✅ 28) Components of Hadoop Ecosystem

HDFS, MapReduce, YARN, Hive, Pig, HBase, Sqoop, Flume, Zookeeper

---

✅ 29) What is Scala?

A hybrid functional and object-oriented programming language that runs on the JVM.

✅ 30) Features of Scala:

- **Type inference**
- **Immutability**
- **Concurrency support**
- **Functional programming**

✅ 31) Scala vs Java

- **Scala is concise, supports functional programming, has better concurrency.**
- **Java is verbose and purely OOP.**

✅ 32) Applications of Scala

- **Big Data (Spark)**
- **Web development**
- **Machine Learning**
- **Distributed systems**

✅ 33) Steps to run Scala in Spark (Windows)

1. **Copy .scala file to Spark folder**
2. **Open CMD in that folder**
3. **Run spark-shell**
4. **Load file: :load filename.scala**

✅ 34) What is Data Science?

Field of extracting knowledge from structured/unstructured data using statistics, ML, and computing.

✅ 35) What is Big Data?

Extremely large datasets that require special tools for storage and analysis.

✅ 36) Characteristics of Big Data

- **Volume**
- **Velocity**
- **Variety**
- **Veracity**
- **Value**

✅ 37) Phases in Data Science Life Cycle

1. **Data Collection**

2. **Data Cleaning**
3. **Data Exploration**
4. **Modeling**
5. **Evaluation**
6. **Deployment**

✅ 38) Central Tendency

A measure to find the center of data: Mean, Median, Mode

✅ 39) Dispersion

Measures spread of data: Range, Variance, Standard Deviation

✅ 40) Calculate for 10, 22, 13, 10, 21, 43, 77, 21, 10:

- **Mean: 25.22**
- **Mode: 10**
- **Median: 21**
- **Midrange: (10+77)/2 = 43.5**

✅ 41) Variance

Measure of how far values spread from mean.
Use: np.var(data)

✅ 42) Standard Deviation

Square root of variance. Shows average distance from mean.
Use: np.std(data)

✅ 43) Posterior Probability

P(H|E): Probability of hypothesis H given the evidence E.

✅ 44) Likelihood Probability

P(E|H): Probability of evidence E given hypothesis H.

✅ 45) Handling Missing Values

- **Drop missing: df.dropna()**
- **Fill with mean/median: df['col'].fillna(df['col'].mean())**
- **Forward/backward fill: df.fillna(method='ffill')**

✅ 46) What is NLTK?

Natural Language Toolkit – a library for processing human language data (text).

✅ 47) Tokenization

Breaking text into words or sentences.

✅ 48) Stemming

Reduces words to root form (e.g., "running" → "run")

✅ 49) Lemmatization

Reduces words to meaningful base form using vocabulary (e.g., "better" → "good")

✅ 50) Corpus in NLP

A collection of text data used for training NLP models.

Let me know if you want this as a PDF file for easy printing or studying offline!

1. What is Data Wrangling? Steps involved?

Data Wrangling is the process of cleaning, transforming, and preparing raw data into a usable format.

Steps:

1. Data Collection
2. Data Cleaning
3. Data Transformation
4. Data Reduction
5. Data Integration
6. Data Visualization

✅ 2. What are different types of data?

- **Categorical (Qualitative)**
  - *Nominal*: No order (e.g., gender, city)
  - *Ordinal*: Ordered (e.g., education level)
- **Numerical (Quantitative)**
  - *Discrete*: Whole numbers (e.g., no. of siblings)
  - *Continuous*: Real values (e.g., height, salary)

✅ 3. Difference between DataFrame and Series in Pandas

Feature	DataFrame	Series
Structure	2D (rows and columns)	1D (single column)
Use	Table of data	Single column or row
Example	pd.DataFrame()	pd.Series()

✅ 4. Difference between .loc[] and .iloc[]

Feature	.loc[] (Label-based)	.iloc[] (Integer-based)
Access by	Index label	Row/column number
Example	df.loc[3], df.loc['row3']	df.iloc[3]

✅ 5. What are missing values? How to handle them?

Missing values are cells with no data.
Handling Methods:

- Detect: df.isnull().sum()
- Drop: df.dropna()
- Fill:
  - Mean: df['col'].fillna(df['col'].mean())
  - Median: fillna(df['col'].median())
  - Forward fill: df.fillna(method='ffill')

✅ 6. What are outliers? How to detect and treat them?

Outliers are extreme values differing from others.

Detection:

- Boxplot: sns.boxplot()
- IQR:

python

CopyEdit

Q1 = df['col'].quantile(0.25)

Q3 = df['col'].quantile(0.75)

IQR = Q3 - Q1

Treatment:

- Remove
- Replace with median/mean
- Capping (limiting to upper/lower bounds)

✅ 7. Difference between apply(), map(), and applymap()

Function	Works on	Used for
apply()	Series, DataFrame	Row/column-wise ops
map()	Series	Element-wise ops
applymap()	DataFrame	Element-wise (entire DF)

✅ 8. How do you normalize or scale data?

Normalization (MinMax Scaling):

python

CopyEdit

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
df_scaled = scaler.fit_transform(df[['col']])
```

Standardization (Z-score):

python

CopyEdit

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
df_scaled = scaler.fit_transform(df[['col']])
```

✅ 9. How to check skewness and kurtosis?

python

CopyEdit

```
df.skew()   # Measure of asymmetry
df.kurt()   # Measure of tails/peakedness
```

✅ 10. Types of joins in Pandas

Join Type	Description
Inner	Only common rows from both tables
Left	All from left, matched from right

Join Type	Description
Right	All from right, matched from left
Outer	All rows from both tables

Code:

python

CopyEdit

```
pd.merge(df1, df2, how='inner')  # Change 'inner' to left/right/outer
```

✅ 11. What are groupby operations?

Used to split data into groups and apply aggregate functions.

Example:

python

CopyEdit

```
df.groupby('gender')['marks'].mean()
```

Groups by gender, then calculates average marks.

✅ 12. What is correlation and covariance?

- **Correlation**: Strength of linear relationship (range: -1 to 1)
  - Code: df.corr()
- **Covariance**: Direction of how two variables change together
  - Code: df.cov()

✅ 13. One-hot encoding vs Label encoding

Encoding Type	Description	Use Case
Label Encoding	Converts categories to numbers	Ordinal data
One-Hot Encoding	Creates binary columns for each category	Nominal data

Code:

python

CopyEdit

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

df['gender'] = le.fit_transform(df['gender'])


df = pd.get_dummies(df, columns=['gender'])
```

Let me know if you'd like this as a downloadable .txt or .pdf file!

d46) What is Spark Framework?

Apache Spark is an open-source, distributed computing system designed for fast computation. It is widely used for **big data processing**, **machine learning**, **real-time analytics**, and **data pipelines**.
It is much faster than Hadoop MapReduce due to in-memory processing.

Key Features:

- In-memory processing (faster than disk-based Hadoop)
- Supports multiple languages: Scala, Java, Python, R
- Supports batch and real-time processing
- Built-in libraries: Spark SQL, MLlib (machine learning), GraphX, and Spark Streaming

---

✅ **Steps to Run Scala Program in Windows using Spark Framework**

These steps assume you have Spark and Scala installed correctly on your Windows system.

1️⃣ Place your Scala file
Copy your .scala program (e.g., sum.scala) to your Spark directory.
Example path:

makefile

CopyEdit

C:\Program Files\Big Data\Spark

2️⃣ Open Command Prompt in Spark Folder
In Windows Explorer, go to the Spark folder → click on the address bar → type cmd and press Enter.
This opens the command prompt directly in that folder.

3️⃣ Start Spark Shell

bash

CopyEdit

spark-shell

This will launch the interactive Scala shell with Spark.

4️⃣ Load and Run Your Scala File
Use the :load command to run your Scala program in the Spark shell:

scala

CopyEdit

:load sum.scala

Or for another file:

scala

CopyEdit

:load pali.scala

After loading, the Scala program will execute in the Spark shell environment.

Let me know if you want help writing