**INTERVIEW QUESTIONS**

**DAY 1**

Here are the answers to the questions provided in the document:

**SQL**

1. Are multiple foreign keys possible for a table? Explain with an example.\*\*

Yes, a table can have multiple foreign keys. Each foreign key in a table references the primary key of another table. For example:

CREATE TABLE Orders (

OrderID int PRIMARY KEY,

CustomerID int,

ProductID int,

OrderDate date,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),

FOREIGN KEY (ProductID) REFERENCES Products(ProductID)

);

In this example, the `Orders` table has two foreign keys: `CustomerID` referencing the `Customers` table and `ProductID` referencing the `Products` table.

2. What does SQL's TRUNCATE statement accomplish?

The `TRUNCATE` statement in SQL removes all rows from a table, but the table structure and its columns, constraints, indexes, and so on remain. It is a faster operation compared to `DELETE` since it does not generate individual row delete statements. Example:

TRUNCATE TABLE Employees;

3. What is the difference between a database and a table in SQL?

A database is a collection of organized data and a system to manage it. It contains tables, views, indexes, stored procedures, and other objects. A table is a specific structure within a database that stores data in rows and columns. For example, a database could be an entire customer relationship management system, while a table might store the details of customers.

4. What is the purpose of the ENUM data type in SQL?

The `ENUM` data type in SQL is used to define a column with a list of possible values. It ensures that the column can only contain one value from the list. It is often used for fields with a predefined set of values, such as status or category fields. Example:

CREATE TABLE Orders (

OrderID int,

Status ENUM('Pending', 'Shipped', 'Delivered', 'Cancelled')

);

5. Explain the concept of index fragmentation and its impact on database performance.

Index fragmentation occurs when the logical order of pages in an index does not match the physical order. This can lead to inefficient use of disk space and can slow down query performance because the database engine may need to perform additional I/O operations to read fragmented pages. Reducing fragmentation can improve performance by ensuring that data is stored more contiguously.

6. Write a query for describing the data to get the four moments of business decision.

Assuming "four moments of business decision" refers to basic statistical moments, here is a query that calculates mean, variance, skewness, and kurtosis for a column `Value` in a table `BusinessData`:

SELECT

AVG(Value) AS Mean,

VARIANCE(Value) AS Variance,

SKEWNESS(Value) AS Skewness,

KURTOSIS(Value) AS Kurtosis

FROM

BusinessData;

Note: Functions `VARIANCE`, `SKEWNESS`, and `KURTOSIS` may vary by SQL dialect.

**PowerBI**

1. Explain the concept of calculated measures in Power BI.

Calculated measures in Power BI are calculations used to summarize data. They are created using Data Analysis Expressions (DAX) and are typically used in reports to create values that are not directly stored in the database but are calculated on-the-fly from existing data. Example:

DAX

TotalSales = SUM(Sales[Amount])

2. What is the difference between calculated columns and calculated measures in Power BI?

Calculated columns are computed at the row level and stored in the data model, whereas calculated measures are computed at the time of query and are used to aggregate data dynamically. Calculated columns can be used in slicers and filters, while measures are typically used in aggregations.

3. Explain the concept of data modeling in Power BI.

Data modeling in Power BI involves creating relationships between different tables to build a coherent dataset for analysis. It includes defining keys, establishing relationships, and creating calculated columns and measures to facilitate efficient querying and reporting.

4. How can you implement row-level security in Power BI?

Row-level security (RLS) in Power BI is implemented by defining roles and rules within the Power BI Desktop. These roles filter data based on the logged-in user's permissions. Example:

1. Go to the Modeling tab.

2. Click Manage Roles.

3. Create a new role and define DAX rules to filter data.

4. Publish the report to the Power BI service and assign users to roles.

5. How can you optimize the performance of Power BI reports and dashboards?

- Use appropriate data types and reduce the dataset size.

- Avoid complex calculations in visuals; pre-calculate them.

- Optimize data models by removing unnecessary columns and tables.

- Use aggregations and summarized data.

- Enable query caching and use DirectQuery judiciously.

**Tableau**

1. How would you handle a large dataset in Tableau that exceeds the software's memory limitations?

- Use data extracts instead of live connections to improve performance.

- Aggregate data or create summaries to reduce the dataset size.

- Use filters to limit the amount of data loaded.

- Utilize Tableau’s data engine capabilities to optimize queries.

2. Can you explain how Tableau's level of detail (LOD) expressions work?

LOD expressions allow you to compute values at different levels of detail, independently of the visualization's level of detail. There are three types:

- `FIXED`: Computes values using the specified dimensions, regardless of the view.

- `INCLUDE`: Adds specified dimensions to the existing view level of detail.

- `EXCLUDE`: Removes specified dimensions from the existing view level of detail.

Example:

{ FIXED [Region] : SUM([Sales]) }

3. How can Power BI row-level security be implemented?

This question appears to be repeated. Please refer to the PowerBI section above.

4. Explain the concept of data blending in Tableau and when you would use it.

Data blending in Tableau is used to combine data from different sources, which are not directly related, into a single view. It is useful when you have multiple datasets with a common field, but the datasets cannot be joined directly due to different granularities or other limitations. It allows creating a primary and secondary data source where blending happens based on a common dimension.

5. How would you create a calculated field in Tableau and provide an example of when it might be necessary?

To create a calculated field in Tableau:

1. Go to the Data pane.

2. Right-click and select "Create Calculated Field."

3. Enter the calculation using Tableau’s formula language.

Example:

IF [Sales] > 10000 THEN 'High' ELSE 'Low' END

This could be necessary to categorize sales performance dynamically based on thresholds.

**DAY 2**

**SQL**

1. What is Schema in SQL?

A schema in SQL is a collection of database objects, including tables, views, indexes, and stored procedures, that are associated with a particular database. It helps organize and manage database objects within a database. Each schema belongs to a single database and acts as a namespace for the objects it contains.

2. How many types of Normal Forms do we have in SQL? and what are they?

There are six types of Normal Forms in SQL:

- First Normal Form (1NF): Ensures that the values in each column are atomic (indivisible).

- Second Normal Form (2NF): Achieves 1NF and ensures that all non-key attributes are fully functionally dependent on the primary key.

- Third Normal Form (3NF): Achieves 2NF and ensures that all the attributes are not transitively dependent on the primary key.

- Boyce-Codd Normal Form (BCNF): A stricter version of 3NF where every determinant is a candidate key.

- Fourth Normal Form (4NF):Achieves BCNF and ensures that there are no multi-valued dependencies.

- Fifth Normal Form (5NF): Achieves 4NF and ensures that every join dependency is implied by the candidate keys.

3. Given a table called "customers" with columns for `customer\_id`, `customer\_name`, and `join\_date`, how can you retrieve all customers who joined before January 1, 2022, using an SQL query?

SELECT customer\_id, customer\_name, join\_date

FROM customers

WHERE join\_date < '2022-01-01';

4. In a table called "products," how can you retrieve the product with the highest price using an SQL query?

SELECT \*

FROM products

ORDER BY price DESC

LIMIT 1;

5. How to select UNIQUE records from a table using a SQL Query?

SELECT DISTINCT column\_name

FROM table\_name;

Or, for selecting unique rows based on multiple columns:

SELECT DISTINCT column1, column2, ...

FROM table\_name;

**PowerBI**

1. What is Power BI?

Power BI is a business analytics service by Microsoft that provides interactive visualizations and business intelligence capabilities with an interface simple enough for end-users to create their own reports and dashboards. It connects to a wide range of data sources and enables real-time data analysis and visualization.

2. Which of the following is not a way to create a new visualization in Power BI?

This question appears to be multiple-choice but without options listed. Generally, ways to create visualizations include using the report canvas, selecting fields from the data pane, and using pre-built templates. Options not commonly used might include direct SQL queries within the visualization pane, which is not typically how visualizations are created.

3. In Power BI how do you make a calculated column?

To create a calculated column in Power BI:

1. Go to the Data view.

2. Click on the New Column button in the Modeling tab.

3. Enter the DAX formula for the calculated column.

Example:

DAX

NewColumn = [Column1] + [Column2]

4. What distinguishes a measure in Power BI from a computed column?

A computed column is calculated at the row level and stored in the data model, whereas a measure is calculated at the aggregation level and evaluated at the time of the query. Computed columns can be used in slicers and filters, while measures are typically used in visualizations for dynamic aggregation.

5. How can you schedule data refresh in Power BI Service?

To schedule a data refresh in Power BI Service:

1. Go to the workspace containing the dataset.

2. Click on the dataset to open its settings.

3. Under the Schedule Refresh section, configure the frequency and time of the data refresh.

4. Ensure data source credentials are provided and valid.

**Tableau**

1. What are the two main products of Tableau?

The two main products of Tableau are:

- Tableau Desktop: A desktop application for data visualization and analysis.

- Tableau Server/Tableau Online: Platforms for sharing and collaborating on Tableau reports and dashboards. Tableau Server is hosted on-premises, while Tableau Online is a cloud-based service.

2. What is the purpose of a hierarchy in Tableau and how can it be created?

A hierarchy in Tableau allows users to organize and drill down into data across different levels of granularity. It is created by dragging one field on top of another in the Data pane to create parent-child relationships. Hierarchies help in analyzing data at various levels, such as year -> quarter -> month -> day.

3. What is a dashboard in Tableau?

A dashboard in Tableau is a collection of multiple visualizations, arranged on a single screen, that provide a comprehensive view of the data. Dashboards allow users to compare and analyze related metrics and trends simultaneously. They can include charts, graphs, filters, and other interactive elements.

4. What does the image indicate if we have a double database in Tableau?

A double database icon in Tableau typically indicates data blending, where data from multiple sources is combined in a single view. The primary data source is indicated with a single database icon, while secondary data sources are indicated with a double database icon.

5. What sorts of charts are available in Tableau?

Tableau offers a wide range of charts, including:

- Bar Chart

- Line Chart

- Pie Chart

- Scatter Plot

- Area Chart

- Heat Map

- Tree Map

- Bubble Chart

- Gantt Chart

- Histogram

- Box Plot

- Bullet Chart

- Packed Bubbles

- Waterfall Chart

- Dual-Axis Chart

**DAY 3**

**Python**

1. What are Python Keywords?

Python keywords are reserved words that have a special meaning and purpose in the language. They are part of the syntax and cannot be used as identifiers (variable names, function names, etc.). Examples of Python keywords include `if`, `else`, `for`, `while`, `class`, `def`, `try`, `except`, and many others.

2. How do you handle a specific exception in your code? If you encounter a ‘FileNotFoundError’ how would you catch and handle it gracefully in your code?\*\*

To handle a specific exception, you use the `try` and `except` blocks. For a `FileNotFoundError`, you can catch and handle it like this:

try:

with open('nonexistent\_file.txt', 'r') as file:

content = file.read()

except FileNotFoundError as e:

print(f"Error: {e}")

This code attempts to open a file that doesn't exist, and if a `FileNotFoundError` is raised, it catches the exception and prints an error message.

3. What is a lambda function in Python and where is it useful?

A lambda function in Python is a small anonymous function defined with the `lambda` keyword. It can have any number of arguments but only one expression. Lambda functions are useful for short, simple functions that are used temporarily, such as in functional programming contexts like `map()`, `filter()`, and `reduce()`. Example:

square = lambda x: x \*\* 2

print(square(5)) # Output: 25

4. How do you handle exceptions in Python and what is the reason for using the exceptions?

Exceptions in Python are handled using `try`, `except`, `else`, and `finally` blocks. The reason for using exceptions is to handle error conditions gracefully without crashing the program. This allows for better error management and debugging. Example:

try:

result = 10 / 0

except ZeroDivisionError as e:

print(f"Error: {e}")

else:

print("No error occurred")

finally:

print("This block always executes")

5. What distinguishes the Python '==' and 'is' operators?

The `==` operator checks for value equality, meaning it compares whether the values of two objects are the same. The `is` operator checks for identity equality, meaning it compares whether two objects are actually the same object in memory. Example:

a = [1, 2, 3]

b = [1, 2, 3]

print(a == b) # Output: True

print(a is b) # Output: False

**EDA (Exploratory Data Analysis)**

1. Explain the concept of correlation and which function is used to check the correlation between features?

Correlation measures the statistical relationship between two variables, indicating how changes in one variable are associated with changes in another. The `corr()` function in pandas is commonly used to check the correlation between features in a DataFrame. Example:

import pandas as pd

df = pd.DataFrame({'A': [1, 2, 3], 'B': [4, 5, 6]})

print(df.corr())

2. Explain the different types of transformation?

Transformations in data processing include:

- Normalization: Scaling data to a specific range (e.g., 0 to 1).

- Standardization: Scaling data to have a mean of 0 and a standard deviation of 1.

- Log Transformation: Applying the natural logarithm to reduce skewness.

- Square Root Transformation: Applying the square root function to stabilize variance.

- Box-Cox Transformation: Transforming data to follow a normal distribution.

3. What is the formula for calculating Skewness and which Python function is used to get the skewness value?

The formula for skewness is:

\[

\text{Skewness} = \frac{n}{(n-1)(n-2)} \sum \left(\frac{x\_i - \bar{x}}{\sigma}\right)^3

\]

The `skew()` function from the pandas library is used to calculate skewness. Example:

```python

import pandas as pd

df = pd.DataFrame({'A': [1, 2, 3, 4, 5]})

print(df['A'].skew())

4. What does X-axis and Y-axis represent in a Histogram?

In a histogram:

- The X-axis represents the bins or intervals into which the data is divided.

- The Y-axis represents the frequency or count of data points within each bin.

5. Which function is used to get a horizontal bar plot?

The `barh()` function in matplotlib is used to create a horizontal bar plot. Example:

import matplotlib.pyplot as plt

plt.barh(['A', 'B', 'C'], [3, 4, 5])

plt.show()

**Data Preprocessing**

1. How do you handle skewed distributions in data preprocessing?

Skewed distributions can be handled using various transformations such as log transformation, square root transformation, or Box-Cox transformation to make the data more normal-like.

2. Sometimes in data null values play hide and seek. How will you identify null values?

Null values can be identified using functions like `isnull()` or `isna()` combined with `sum()` to get the count of null values. Example:

import pandas as pd

df = pd.DataFrame({'A': [1, None, 3], 'B': [4, 5, None]})

print(df.isnull().sum())

3. What challenges can arise during the data preprocessing phase and how do you overcome those challenges?

Challenges in data preprocessing include handling missing values, dealing with outliers, scaling and normalization, and encoding categorical variables. These can be overcome using techniques like imputation for missing values, robust scaling for outliers, and one-hot encoding or label encoding for categorical variables.

4. In One Hot Encoding technique sometimes results come as sparse Matrix. What is the reason for it?

One-hot encoding results in a sparse matrix when there are many unique categories because each category is represented as a separate column with binary values (0 or 1). This can lead to a large number of columns, with most of them containing zeros.

5. What is the purpose of data normalization and what methods can we use to normalize the data?

The purpose of data normalization is to scale the data to a specific range, usually 0 to 1, to ensure that all features contribute equally to the model. Methods for normalization include Min-Max scaling and Z-score normalization. Example using Min-Max scaling:

from sklearn.preprocessing import MinMaxScaler

import pandas as pd

df = pd.DataFrame({'A': [1, 2, 3], 'B': [4, 5, 6]})

scaler = MinMaxScaler()

normalized\_df = pd.DataFrame(scaler.fit\_transform(df), columns=df.columns)

print(normalized\_df)

**DAY 4**

**SQL**

1. Can you explain the concept of query execution plans and how they can be analyzed and optimized?

A query execution plan is a sequence of operations that the database management system (DBMS) will perform to execute a SQL query. It is a roadmap that shows how the query will be executed, including details about table scans, joins, and indexes used. Execution plans can be analyzed using tools like `EXPLAIN` in MySQL, `EXPLAIN PLAN` in Oracle, and `SET SHOWPLAN` in SQL Server. Optimization involves:

- Ensuring indexes are used efficiently.

- Avoiding full table scans when possible.

- Rewriting queries to reduce complexity.

- Checking for proper join order and method (nested loops, hash joins, etc.).

2. What are the three degrees of normalization and how is normalization done in each degree?

The three degrees of normalization are:

- First Normal Form (1NF): Ensures each column contains atomic (indivisible) values and each record is unique.

- Second Normal Form (2NF): Achieved by meeting 1NF requirements and removing partial dependencies (i.e., no non-key attribute should depend on a part of the primary key).

- Third Normal Form (3NF): Achieved by meeting 2NF requirements and removing transitive dependencies (i.e., non-key attributes should not depend on other non-key attributes).

3. Can you explain a scenario where using derived columns can improve query performance?

Using derived columns can improve query performance by precomputing complex expressions or aggregations. For example, if a query frequently calculates the total price as `quantity \* unit\_price`, adding a derived column `total\_price` can save computation time and improve performance by reducing the need for repeated calculations.

4. What are the differences between OLTP and OLAP? Can you provide an example of a real-world application that requires OLTP processing and another that requires OLAP processing?

- OLTP (Online Transaction Processing):\*\* Handles day-to-day transaction data, characterized by a large number of short online transactions (INSERT, UPDATE, DELETE). Example: A retail POS (point of sale) system.

- OLAP (Online Analytical Processing):\*\* Handles historical data for analysis and decision making, characterized by complex queries and data analysis (SELECT). Example: A business intelligence system analyzing sales data trends.

5. How would you determine which type of join to use based on the requirements of a specific query?

The type of join depends on the relationship between the tables and the data you need:

- INNER JOIN: Use when you need records that have matching values in both tables.

- LEFT JOIN (or LEFT OUTER JOIN): Use when you need all records from the left table and the matched records from the right table (nulls if no match).

- RIGHT JOIN (or RIGHT OUTER JOIN): Use when you need all records from the right table and the matched records from the left table (nulls if no match).

- FULL JOIN (or FULL OUTER JOIN): Use when you need all records when there is a match in one of the tables.

- CROSS JOIN: Use when you need a Cartesian product of both tables.

6. How to delete DUPLICATE records from a table Using ROWID and ROW\_NUMBER Analytic Function?

To delete duplicate records using `ROWID` and `ROW\_NUMBER` in SQL:

DELETE FROM your\_table

WHERE ROWID IN (

SELECT ROWID

FROM (

SELECT ROWID,

ROW\_NUMBER() OVER (PARTITION BY column1, column2 ORDER BY column1) AS row\_num

FROM your\_table

)

WHERE row\_num > 1

);

**Power BI**

1. How can you create a slicer in Power BI that dynamically filters multiple visuals?

In Power BI, you can create a slicer by adding a slicer visual to the report and selecting the field you want to filter by. To filter multiple visuals, ensure that the visuals are using the same dataset and that the interactions are set to filter. This can be done by clicking on the slicer, selecting "Edit interactions," and ensuring that the relevant visuals are set to filter mode.

2. When would you choose to use Power BI DirectQuery mode instead of Import mode?

Use DirectQuery mode when:

- The dataset is too large to fit into memory.

- Real-time or near-real-time data updates are required.

- You need to enforce row-level security based on the underlying database.

3. Can you explain the concept of row context and filter context in DAX and how they affect calculations?

- Row Context: Refers to the current row in a table where a calculation is being performed. It is relevant in row-by-row operations, like in calculated columns.

- Filter Context:Refers to the set of filters applied to the data model that affects calculations. It is relevant in measures and when using functions like `CALCULATE()` which can modify filter context.

4. What is the difference in behavior between the SUM function and the SUMX function when working with related tables in Power BI?

- SUM:Aggregates a single column of numbers.

- SUMX: Iterates over a table, evaluates an expression for each row, and then sums the results. It is useful when calculations depend on related tables or complex expressions.

5. What are some considerations you need to keep in mind to ensure query folding occurs during data loading in Power Query?

- Avoid complex transformations in Power Query that cannot be translated into the source system's query language.

- Use simple transformations like filtering, grouping, and column selection.

- Keep transformations in the early steps simple to maximize the chance of query folding.

**Tableau**

1. What is the difference between a live connection and an extract in Tableau?\*\*

- Live Connection: Connects directly to the data source, reflecting real-time data changes.

- Extract: A snapshot of the data at a specific point in time, stored locally, and used for faster performance and offline access.

2. How can a metadata table improve the performance and user experience in Tableau?

A metadata table can centralize and simplify data definitions, reduce redundancy, and improve performance by minimizing the need for complex joins and calculations at the visualization level.

3. When would you use a global filter and when would you use a context filter in your Tableau visualizations?

- Global Filter: Use when you want a filter to apply across all worksheets in a dashboard.

- Context Filter: Use when you need to filter data before applying other filters, creating a dependent relationship between filters.

4. Explain the purpose of the Data Source tab, Worksheet tab, and Dashboard tab in Tableau Desktop.

- Data Source Tab: Used to connect to and prepare data for analysis, including defining relationships, joins, and data cleaning.

- Worksheet Tab: Used to create individual visualizations (charts, graphs) based on the data.

- Dashboard Tab: Used to combine multiple worksheets into a cohesive dashboard for comprehensive analysis and presentation.

5. How can users interact with Tableau dashboards and visualizations using the Tableau Mobile app?

Users can interact with Tableau dashboards on the Mobile app by:

- Viewing and navigating through dashboards.

- Using filters, parameters, and highlights to explore data.

- Accessing dashboards offline if they have been made available for offline viewing.

DAY 5

**Python**:

1. What is the purpose of the `\_\_init\_\_` method in Python classes? How is it different from other methods?

The `\_\_init\_\_` method in Python classes is known as the constructor. It is called when an instance (object) of the class is created. The primary purpose of the `\_\_init\_\_` method is to initialize the instance attributes with the values passed to it when the object is instantiated.

- Example:

class Person:

def \_\_init\_\_(self, name, age):

self.name = name

self.age = age

person = Person("Alice", 30)

- Difference from other methods: Other methods in a class are used to define behaviors or functionalities of the class, and they can be called explicitly using the object of the class. The `\_\_init\_\_` method is automatically invoked when the object is created.

2. Explain the usage of `\*args` and `\*\*kwargs` in Python function definitions. Provide an example.

- `\*args` allows a function to accept any number of positional arguments. The arguments are passed as a tuple.

- `\*\*kwargs` allows a function to accept any number of keyword arguments. The arguments are passed as a dictionary.

- Example: def example\_function(\*args, \*\*kwargs):

print("Positional arguments:", args)

print("Keyword arguments:", kwargs)

example\_function(1, 2, 3, a=4, b=5)

Output:

Positional arguments: (1, 2, 3)

Keyword arguments: {'a': 4, 'b': 5}

3. How does Python handle exceptions? What are the differences between try/except and finally blocks?

- Exception Handling: Python handles exceptions using `try`, `except`, `else`, and `finally` blocks. When an exception occurs, it is caught by the `except` block.

- Example:

result = 10 / 0

except ZeroDivisionError:

print("You can't divide by zero!")

- Differences: - `try/except`: The `try` block contains the code that might raise an exception. The `except` block contains the code to handle the exception.

- `finally`: The `finally` block contains code that will be executed no matter what, whether an exception was raised or not.

- Example:

try:

result = 10 / 2

except ZeroDivisionError:

print("You can't divide by zero!")

finally:

print("This will always be executed.")

4. How can you sort a dictionary in Python based on its values?

- Using `sorted()` function:

my\_dict = {'apple': 10, 'banana': 5, 'cherry': 20}

sorted\_dict = dict(sorted(my\_dict.items(), key=lambda item: item[1]))

print(sorted\_dict)

Output:

{'banana': 5, 'apple': 10, 'cherry': 20}

5. Explain negative indexing in lists and provide an example.

- Negative Indexing: Negative indexing allows you to access elements from the end of a list. The index `-1` refers to the last item, `-2` to the second last item, and so on.

- Example:

my\_list = [10, 20, 30, 40, 50]

print(my\_list[-1]) # Output: 50

print(my\_list[-2]) # Output: 40

6. Define a palindrome. Write a Python program to check if a string is a palindrome.

- Palindrome:

A palindrome is a string that reads the same backward as forward.

- Python Program:

def is\_palindrome(s):

return s == s[::-1]

word = "radar"

print(is\_palindrome(word)) # Output: True

**EDA (Exploratory Data Analysis):**

1. What insights can you gather from a histogram in EDA?

- Insights from a Histogram:

- Distribution of data

- Frequency of different ranges of values

- Skewness (left-skewed or right-skewed)

- Presence of outliers

- Mode(s) of the data

2. Which data visualization technique is commonly used to display the relationship between a categorical variable and a numerical variable?

- Common Visualization Technique:

- Bar Plot

- Box Plot

3. What are the standard names for positive, negative, and normal kurtosis curves?

- Kurtosis Types:

- Positive Kurtosis: Leptokurtic

- Negative Kurtosis: Platykurtic

- Normal Kurtosis: Mesokurtic

4. What is the formula for calculating the expected value?

- Formula for Expected Value:

\[

E(X) = \sum (x\_i \cdot p\_i)

\]

Where \(x\_i\) are the values and \(p\_i\) are the probabilities of the values.

5. What is the default value for the number of bins in a histogram using matplotlib and seaborn libraries?

- Default Value for Number of Bins:

- Matplotlib: `10`

- Seaborn: `auto` (which determines the number of bins based on the data)

Data Preprocessing:

1. What is the difference between data normalization and data standardization? When would you use each technique?

- Normalization: - Rescaling the data to a range of [0, 1] or [-1, 1].

- Use Case:When you want to transform data to fit within a specific range.

- Standardization:

- Rescaling the data to have a mean of 0 and a standard deviation of 1.

- Use Case: When data needs to have a standard normal distribution for certain algorithms.

2. Explain the difference between One Hot Encoding and Label Encoding. When would you use each method?

- One Hot Encoding: - Converts categorical variables into binary vectors.

- Use Case: When the categorical variables are nominal and there is no ordinal relationship.

- Label Encoding: - Converts categorical variables into integer codes.

- Use Case: When the categorical variables are ordinal and there is an inherent order.

3. Define discretization. How can you discretize continuous data?

- Discretization: - The process of converting continuous data into discrete bins or intervals.

- Methods to Discretize Continuous Data:

- Equal-width binning

- Equal-frequency binning

- Clustering-based methods (e.g., k-means)

4. How would you normalize text data for natural language processing (NLP)?

- Normalization Techniques for Text Data:

- Lowercasing

- Removing punctuation and special characters

- Removing stop words

- Stemming or lemmatization

- Tokenization

5. What is meant by Imbalanced data? How do you handle imbalanced datasets in data preprocessing?

- Imbalanced Data:

- A situation where the classes in a dataset are not represented equally.

- Handling Imbalanced Datasets:

- Resampling techniques (over-sampling the minority class or under-sampling the majority class)

- Using different evaluation metrics (e.g., F1 score, precision-recall curve)

- Applying algorithms that are robust to imbalanced data (e.g., decision trees, ensemble methods)

- Synthetic data generation (e.g., SMOTE)

**DAY 6**

**SQL:**

1. What is the purpose of the HAVING clause in SQL queries? Provide an example.

The `HAVING` clause is used in SQL to filter records that work on aggregated data. It is used with the `GROUP BY` clause to restrict the groups of returned rows to those groups for which the specified condition is true.

- Example:

SELECT department, COUNT(\*)

FROM employees

GROUP BY department

HAVING COUNT(\*) > 10;

This query selects the departments with more than 10 employees.

2. How do you calculate the average, sum, and count of a column in SQL? Provide example queries for each.

- Average:

SELECT AVG(salary)

FROM employees;

- Sum:

SELECT SUM(salary)

FROM employees;

- Count:

SELECT COUNT(salary)

FROM employees;

3. Explain the difference between UNION and UNION ALL in SQL.

- UNION:

Combines the results of two or more SELECT statements and removes duplicate rows.

- UNION ALL:

Combines the results of two or more SELECT statements without removing duplicates.

- Example:

-- Using UNION

SELECT name FROM employees

UNION

SELECT name FROM managers;

-- Using UNION ALL

SELECT name FROM employees

UNION ALL

SELECT name FROM managers;

4. What distinguishes a subquery from a join in SQL? When would you use each approach?

- Subquery:

A query within another query. Used when the result of one query is used as a condition in another query.

- Example:

l

SELECT name FROM employees

WHERE department\_id = (SELECT department\_id FROM departments WHERE name = 'Sales');

- Join: Combines rows from two or more tables based on a related column between them.

- Example:

SELECT employees.name, departments.name

FROM employees

JOIN departments ON employees.department\_id = departments.department\_id;

- Usage:

Use subqueries when you need to filter based on the result of another query. Use joins when you need to combine related data from multiple tables.

5. What is the purpose of the TRIGGER statement in SQL? Provide an example of how it can be used.

A `TRIGGER` is a set of SQL statements that automatically executes when an event (INSERT, UPDATE, DELETE) occurs in a table.

- Example:

CREATE TRIGGER update\_timestamp

BEFORE UPDATE ON employees

FOR EACH ROW

SET NEW.last\_modified = NOW();

This trigger updates the `last\_modified` timestamp column whenever a row in the `employees` table is updated.

6. Write an SQL query to find the third maximum salary from a table.

- Example:

SELECT salary

FROM (

SELECT salary, DENSE\_RANK() OVER (ORDER BY salary DESC) AS rank

FROM employees

) AS ranked\_salaries

WHERE rank = 3;

**Power BI:**

1. What distinguishes computed columns from calculated tables in Power BI? When would you use each?

- Computed Columns:

Calculated at the row level within the existing table.

- Use Case: When you need new columns in your existing table based on other columns.

- Calculated Tables: Calculated at the table level and result in a new table.

- Use Case: When you need a new table for your analysis that is derived from existing tables.

2. Explain the purpose of the CALCULATE function in Power BI. Provide an example of how it can be used.

The `CALCULATE` function evaluates an expression in a modified filter context.

- Example:

DAX

CALCULATE(SUM(sales[amount]), sales[year] = 2020)

This calculates the sum of sales for the year 2020.

3. How do DAX and Power Query differ from each other in Power BI?

- DAX (Data Analysis Expressions):

Used for data analysis and creating calculated columns and measures.

- Usage: Within the data model for calculations.

- Power Query:

Used for data extraction, transformation, and loading (ETL).

- Usage: In the query editor to prepare and clean data before loading it into the data model.

4. What is Power Query in Power BI and what are its key functionalities?

Power Query is a data connection technology that enables users to discover, connect, combine, and refine data across a wide variety of sources.

- Key Functionalities:

- Data transformation and cleaning

- Data shaping

- Data combining

- Data importing from various sources

5. Name and briefly describe the three views in Power BI.

- Report View:

Allows users to create visualizations and reports.

- Data View:

Shows the data in tables and allows users to manage relationships and perform calculations.

- Model View:

Allows users to view and manage the relationships between tables.

**Tableau:**

1. What does the notion of data densification mean in Tableau?

Data densification refers to the process of generating additional data points in a visualization to ensure continuous lines or areas, filling in gaps where data might be missing.

2. How do joining and blending data sources differ in Tableau? Explain with an example.

- Joining:

Combines tables within the same data source based on a common field.

- Example:

SELECT \*

FROM orders

JOIN customers ON orders.customer\_id = customers.customer\_id;

- Blending:

Combines data from different data sources on a common dimension.

- Example: Creating a relationship between sales data from an Excel file and customer data from a SQL database on the common field `customer\_id`.

3. Define a Gantt chart and explain its use in Tableau.

A Gantt chart is a type of bar chart that represents a project schedule. It is used to show the start and finish dates of elements within a project.

- Use in Tableau:

Gantt charts in Tableau are used to track project schedules and timelines, visualize duration and overlap of tasks, and manage resources.

4. What are the different types of filters available in Tableau? Provide examples of each.

- Dimension Filter:

Filters data based on dimension values.

- Example: Filtering by region (North, South, East, West).

- Measure Filter:

Filters data based on measure values.

- Example:Filtering sales greater than $1000.

- Date Filter:

Filters data based on date ranges.

- Example: Filtering data for the year 2023.

- Context Filter:

Sets a context for other filters to work on a subset of data.

- Example: Setting a context filter on a specific country to apply other filters only within that country.

- Top N Filter:

Filters data to show the top N values.

- Example: Showing the top 10 products by sales.

5. What do the terms "Blue Mark" and "Blue Pill" indicate in Tableau?

- Blue Mark:

Represents discrete fields in Tableau that segment data.

- Blue Pill:

Refers to discrete fields placed on the Rows or Columns shelf that create headers in the view.

**DAY 7**

**SQL:**

1. What is the purpose of the HAVING clause in SQL queries? Provide an example.

The `HAVING` clause is used in SQL to filter records that work on aggregated data. It is used with the `GROUP BY` clause to restrict the groups of returned rows to those groups for which the specified condition is true.

- Example:

SELECT department, COUNT(\*)

FROM employees

GROUP BY department

HAVING COUNT(\*) > 10;

This query selects the departments with more than 10 employees.

2. How do you calculate the average, sum, and count of a column in SQL? Provide example queries for each.

- Average:

SELECT AVG(salary)

FROM employees;

- Sum:

SELECT SUM(salary)

FROM employees;

- Count:

SELECT COUNT(salary)

FROM employees;

3. Explain the difference between UNION and UNION ALL in SQL.

- UNION: Combines the results of two or more SELECT statements and removes duplicate rows.

- UNION ALL:

Combines the results of two or more SELECT statements without removing duplicates.

- Example:

-- Using UNION

SELECT name FROM employees

UNION

SELECT name FROM managers;

-- Using UNION ALL

SELECT name FROM employees

UNION ALL

SELECT name FROM managers;

4. What distinguishes a subquery from a join in SQL? When would you use each approach?

- Subquery: A query within another query. Used when the result of one query is used as a condition in another query.

- Example:

SELECT name FROM employees

WHERE department\_id = (SELECT department\_id FROM departments WHERE name = 'Sales');

- Join:

Combines rows from two or more tables based on a related column between them.

- Example:

SELECT employees.name, departments.name

FROM employees

JOIN departments ON employees.department\_id = departments.department\_id;

- Usage:

Use subqueries when you need to filter based on the result of another query. Use joins when you need to combine related data from multiple tables.

5. What is the purpose of the TRIGGER statement in SQL? Provide an example of how it can be used.

A `TRIGGER` is a set of SQL statements that automatically executes when an event (INSERT, UPDATE, DELETE) occurs in a table.

- Example:

CREATE TRIGGER update\_timestamp

BEFORE UPDATE ON employees

FOR EACH ROW

SET NEW.last\_modified = NOW();

This trigger updates the `last\_modified` timestamp column whenever a row in the `employees` table is updated.

6. Write an SQL query to find the third maximum salary from a table.

- Example:

SELECT salary

FROM (

SELECT salary, DENSE\_RANK() OVER (ORDER BY salary DESC) AS rank

FROM employees

) AS ranked\_salaries

WHERE rank = 3;

Power BI:

1. What distinguishes computed columns from calculated tables in Power BI? When would you use each?

- Computed Columns:

Calculated at the row level within the existing table.

- Use Case:When you need new columns in your existing table based on other columns.

- Calculated Tables:

Calculated at the table level and result in a new table.

- Use Case: When you need a new table for your analysis that is derived from existing tables.

2. Explain the purpose of the CALCULATE function in Power BI. Provide an example of how it can be used.

The `CALCULATE` function evaluates an expression in a modified filter context.

- Example:

DAX

CALCULATE(SUM(sales[amount]), sales[year] = 2020)

This calculates the sum of sales for the year 2020.

3. How do DAX and Power Query differ from each other in Power BI?

- DAX (Data Analysis Expressions):

Used for data analysis and creating calculated columns and measures.

- Usage: Within the data model for calculations.

- Power Query:

Used for data extraction, transformation, and loading (ETL).

- Usage: In the query editor to prepare and clean data before loading it into the data model.

4. What is Power Query in Power BI and what are its key functionalities?

Power Query is a data connection technology that enables users to discover, connect, combine, and refine data across a wide variety of sources.

-Key Functionalities:

- Data transformation and cleaning

- Data shaping

- Data combining

- Data importing from various sources

5. Name and briefly describe the three views in Power BI.

- Report View:

Allows users to create visualizations and reports.

- Data View: Shows the data in tables and allows users to manage relationships and perform calculations.

- Model View:

Allows users to view and manage the relationships between tables.

Tableau:

1. What does the notion of data densification mean in Tableau?

Data densification refers to the process of generating additional data points in a visualization to ensure continuous lines or areas, filling in gaps where data might be missing.

2. How do joining and blending data sources differ in Tableau? Explain with an example.

-Joining:

Combines tables within the same data source based on a common field.

- Example:

SELECT \*

FROM orders

JOIN customers ON orders.customer\_id = customers.customer\_id;

- Blending: Combines data from different data sources on a common dimension.

- Example: Creating a relationship between sales data from an Excel file and customer data from a SQL database on the common field `customer\_id`.

3. Define a Gantt chart and explain its use in Tableau.

A Gantt chart is a type of bar chart that represents a project schedule. It is used to show the start and finish dates of elements within a project.

- Use in Tableau:

Gantt charts in Tableau are used to track project schedules and timelines, visualize duration and overlap of tasks, and manage resources.

4. What are the different types of filters available in Tableau? Provide examples of each.

- Dimension Filter: Filters data based on dimension values.

- Example: Filtering by region (North, South, East, West).

- Measure Filter:

Filters data based on measure values.

- Example: Filtering sales greater than $1000.

- Date Filter: Filters data based on date ranges.

- Example: Filtering data for the year 2023.

- Context Filter: Sets a context for other filters to work on a subset of data.

- Example: Setting a context filter on a specific country to apply other filters only within that country.

- Top N Filter:

Filters data to show the top N values.

- Example: Showing the top 10 products by sales.

5. What do the terms "Blue Mark" and "Blue Pill" indicate in Tableau?

- Blue Mark: Represents discrete fields in Tableau that segment data.

- Blue Pill:

Refers to discrete fields placed on the Rows or Columns shelf that create headers in the view.

**DAY 8**

**Python:**

1. What is the difference between shallow copy and deep copy in Python? Provide an example for each.

- Shallow Copy: A shallow copy creates a new object but inserts references into it to the objects found in the original. It copies the collection but not the elements within the collection.

import copy

original\_list = [1, 2, [3, 4]]

shallow\_copy = copy.copy(original\_list)

shallow\_copy[2][0] = 'changed'

print(original\_list) # Output: [1, 2, ['changed', 4]]

print(shallow\_copy) # Output: [1, 2, ['changed', 4]]

- Deep Copy: A deep copy creates a new object and recursively adds the copies of nested objects present in the original elements.

import copy

original\_list = [1, 2, [3, 4]]

deep\_copy = copy.deepcopy(original\_list)

deep\_copy[2][0] = 'changed'

print(original\_list) # Output: [1, 2, [3, 4]]

print(deep\_copy) # Output: [1, 2, ['changed', 4]]

2. Explain what metaclasses are in Python and how they can be used.

Metaclasses are classes of classes, meaning that a metaclass defines the behavior of a class. In Python, everything is an object, including classes. Metaclasses allow you to customize class creation. You can use metaclasses to enforce certain behaviors or constraints on a group of classes.

Example:

class Meta(type):

def \_\_new\_\_(cls, name, bases, dct):

print(f'Creating class {name}')

return super().\_\_new\_\_(cls, name, bases, dct)

class MyClass(metaclass=Meta):

pass

# Output: Creating class MyClass

3. Describe the Global Interpreter Lock (GIL) in Python and its impact on multi-threading.

The Global Interpreter Lock (GIL) is a mutex that protects access to Python objects, preventing multiple native threads from executing Python bytecodes at once. This means that even in multi-threaded programs, only one thread can execute Python code at a time. The GIL simplifies memory management but can be a bottleneck in CPU-bound multi-threaded programs. To overcome this, Python developers often use multi-processing or other concurrency methods.

4. How do Python decorators work? Give an example of how decorators can be used.

Decorators are a way to modify or enhance functions or methods without changing their actual code. They are higher-order functions that take another function and extend its behavior.

Example:

def decorator\_func(func):

def wrapper():

print("Something before the function.")

func()

print("Something after the function.")

return wrapper

@decorator\_func

def say\_hello():

print("Hello!")

say\_hello()

# Output:

# Something before the function.

# Hello!

# Something after the function.

5. What is a generator in Python and how is it different from a regular function?

A generator is a function that returns an iterator that produces a sequence of values when iterated over. Unlike regular functions that use `return` to return a value and terminate, generators use `yield` to return a value and suspend the function's state, allowing it to resume where it left off.

Example:

def simple\_generator():

yield 1

yield 2

yield 3

gen = simple\_generator()

print(next(gen)) # Output: 1

print(next(gen)) # Output: 2

print(next(gen)) # Output: 3

6. Write a Python program to check if a number is prime.

def is\_prime(n):

if n <= 1:

return False

for i in range(2, int(n\*\*0.5) + 1):

if n % i == 0:

return False

return True

# Test the function

print(is\_prime(11)) # Output: True

print(is\_prime(4)) # Output: False

**EDA:**

1. If a dataset has more than two mode values, what does it indicate about the data?

If a dataset has more than two mode values, it indicates that the data is multimodal. This means that there are multiple values that occur with the highest frequency.

2. What is the disadvantage of using variance as a measure of dispersion in data analysis?

The disadvantage of using variance as a measure of dispersion is that it is sensitive to outliers. Variance squares the differences from the mean, so larger differences (outliers) have a disproportionately large effect on the variance.

3. What value should be assigned to the "vert" parameter in the boxplot function to visualize it horizontally?

To visualize a boxplot horizontally, the "vert" parameter should be set to `False`.

Example:

import matplotlib.pyplot as plt

plt.boxplot(data, vert=False)

4. What is the formula for calculating excess kurtosis in a dataset?

Excess kurtosis is calculated using the formula:

\[ \text{Excess Kurtosis} = \frac{n(n+1)}{(n-1)(n-2)(n-3)} \sum\_{i=1}^{n} \left( \frac{x\_i - \bar{x}}{s} \right)^4 - \frac{3(n-1)^2}{(n-2)(n-3)} \]

where \( n \) is the number of data points, \( x\_i \) are the data points, \( \bar{x} \) is the mean, and \( s \) is the standard deviation.

5. Explain the primary and secondary uses of a scatter plot in data analysis.

- Primary use: To visualize the relationship or correlation between two numerical variables.

- Secondary use: To identify patterns, clusters, or outliers within the data.

6. How is Anomaly Detection performed in Models of Time-Varying Processes?

Anomaly detection in time-varying processes can be performed using methods such as:

- Statistical methods: Analyzing deviations from a statistical model (e.g., z-scores, moving averages).

- Machine learning models: Training models like ARIMA, LSTM, or other time-series forecasting models to detect deviations from predicted values.

- Density-based methods: Using algorithms like DBSCAN to identify points that are sparsely distributed.

7. What technique would you use to prevent Swamping and Masking for Isolation Forest Anomaly Detection?

To prevent swamping and masking in Isolation Forest Anomaly Detection, you can:

- Tune the hyperparameters, such as the number of estimators and contamination factor.

- Use ensemble methods to combine the results of multiple models.

- Apply preprocessing steps like feature scaling and transformation to ensure that features are on comparable scales.

**Data preprocessing:**

1. What are the common techniques used for handling missing data in a dataset? Provide examples for each.

- Deletion: Removing rows or columns with missing values.

df.dropna()

- Imputation: Filling missing values with a statistical measure (mean, median, mode) or using algorithms like k-NN.

df.fillna(df.mean())

- Forward/Backward Fill: Filling missing values with the previous or next value.

df.fillna(method='ffill')

2. Explain the difference between standardization and normalization in data preprocessing.

- Standardization: Transforms data to have a mean of 0 and a standard deviation of 1.

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

standardized\_data = scaler.fit\_transform(data)

- Normalization: Scales data to a fixed range, usually [0, 1].

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

normalized\_data = scaler.fit\_transform(data)

3. How do you handle outliers in a dataset? Describe some methods or techniques.

- Removal: Removing data points that are significantly different from others.

df = df[(df['col'] > lower\_bound) & (df['col'] < upper\_bound)]

- Transformation: Applying transformations like log or square root to reduce the effect of outliers.

df['col'] = np.log(df['col'])

- Capping: Limiting extreme values to a certain threshold.

df['col'] = np.where(df['col'] > upper\_bound, upper\_bound, df['col'])

4. How do outliers affect when data is based on time series? What methods and techniques are available to detect outliers in time series data?

Outliers in time series data can distort analysis, leading to incorrect forecasts or model parameters. Methods to detect outliers include:

- Statistical methods: Z-scores, moving averages.

- Model-based methods: Residual analysis from ARIMA models.

- Machine learning methods: Using models like LSTM to identify deviations from predicted patterns.

5. What is dimensionality reduction and what are some techniques that can be used for it?

Dimensionality reduction is the process of reducing the number of input variables in a dataset. Techniques include:

- Principal Component Analysis (PCA): Transforms data to new axes of maximum variance.

from sklearn.decomposition import PCA

pca = PCA(n\_components=2)

reduced\_data = pca.fit\_transform(data

**DAY 9**

1. What is the difference between shallow copy and deep copy in Python? Provide an example for each.

Shallow Copy:

A shallow copy creates a new object, but inserts references into it to the objects found in the original.

Example:

import copy

original = [1, 2, [3, 4]]

shallow\_copy = copy.copy(original)

print(shallow\_copy) # Output: [1, 2, [3, 4]]

shallow\_copy[2][0] = 'X'

print(original) # Output: [1, 2, ['X', 4]]

Deep Copy:

A deep copy creates a new object and recursively adds copies of nested objects found in the original.

Example:

deep\_copy = copy.deepcopy(original)

deep\_copy[2][0] = 'Y'

print(original) # Output: [1, 2, ['X', 4]]

2. Explain what metaclasses are in Python and how they can be used.

Metaclasses are classes of classes that define how classes behave. A class is an instance of a metaclass. Metaclasses allow you to customize class creation.

Example:

class Meta(type):

def \_\_new\_\_(cls, name, bases, dct):

dct['custom\_attribute'] = 'custom\_value'

return super().\_\_new\_\_(cls, name, bases, dct)

class MyClass(metaclass=Meta):

pass

print(MyClass.custom\_attribute) # Output: custom\_value

3. Describe the Global Interpreter Lock (GIL) in Python and its impact on multi-threading.

The GIL is a mutex that protects access to Python objects, preventing multiple threads from executing Python bytecodes at once. This can lead to performance issues in CPU-bound multi-threaded programs.

4. How do Python decorators work? Give an example of how decorators can be used.

Decorators are functions that modify the behavior of another function. They allow for reusable and modular code.

Example:

def decorator\_function(original\_function):

def wrapper\_function(\*args, \*\*kwargs):

print('Wrapper executed this before {}'.format(original\_function.\_\_name\_\_))

return original\_function(\*args, \*\*kwargs)

return wrapper\_function

@decorator\_function

def display():

print('Display function ran')

display()

5. What is a generator in Python and how is it different from a regular function?

Generators are functions that return an iterable set of items, one at a time, in a special way using the `yield` keyword. They are memory efficient.

Example:

def generator\_function():

for i in range(10):

yield i

for value in generator\_function():

print(value)

6. Write a Python program to check if a number is prime.

def is\_prime(n):

if n <= 1:

return False

for i in range(2, int(n\*\*0.5) + 1):

if n % i == 0:

return False

return True

number = 29

print(f"{number} is prime: {is\_prime(number)}")

7. If a dataset has more than two mode values, what does it indicate about the data?

It indicates that the data is multimodal, meaning there are multiple values that appear with the same highest frequency.

8. What is the disadvantage of using variance as a measure of dispersion in data analysis?

Variance is sensitive to outliers because it squares the differences from the mean, giving more weight to larger deviations.

9. What value should be assigned to the 'vert' parameter in the boxplot function to visualize it horizontally?

Set the `vert` parameter to `False` to visualize the boxplot horizontally.

Example:

import matplotlib.pyplot as plt

plt.boxplot(data, vert=False)

10. What is the formula for calculating excess kurtosis in a dataset?

Excess kurtosis is calculated as:

\[ \text{Excess Kurtosis} = \frac{n(n+1)}{(n-1)(n-2)(n-3)} \sum \left( \frac{(x\_i - \bar{x})^4}{s^4} \right) - \frac{3(n-1)^2}{(n-2)(n-3)} \]

Where \( n \) is the number of observations, \( \bar{x} \) is the mean, and \( s \) is the standard deviation.

11. Explain the primary and secondary uses of a scatter plot in data analysis.

Primary Use:\*To identify the relationship or correlation between two variables.

Secondary Use:To detect outliers and the distribution of data points.

12. How is Anomaly Detection performed in Models of Time-Varying Processes?

Anomaly detection in time-varying processes can be performed using techniques such as moving averages, ARIMA models, and machine learning algorithms like LSTM networks.

13. What technique would you use to prevent Swamping and Masking for Isolation Forest Anomaly Detection?

Adjust the contamination parameter and use ensemble methods to improve the robustness of the model against swamping and masking effects.

14. What are the common techniques used for handling missing data in a dataset? Provide examples for each.

- Mean/Median Imputation: Replace missing values with the mean or median.

Example:

df['column'].fillna(df['column'].mean(), inplace=True)

- Mode Imputation: Replace with the most frequent value.

- Forward/Backward Fill: Use adjacent values to fill missing data.

Example:

df['column'].fillna(method='ffill', inplace=True)

- Interpolation: Use interpolation methods to estimate missing values.

- Dropping Missing Values: Remove rows or columns with missing values.

Example:

df.dropna(inplace=True)

15. Explain the difference between standardization and normalization in data preprocessing.

- Standardization: Transforms data to have a mean of zero and a standard deviation of one.

Example:

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

standardized\_data = scaler.fit\_transform(data)

- Normalization: Scales data to a fixed range, typically 0 to 1.

Example:

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

normalized\_data = scaler.fit\_transform(data)

16. How do you handle outliers in a dataset? Describe some methods or techniques.

- Removing Outliers: Drop the data points that are outliers.

- Capping: Set a maximum and minimum threshold to limit the effect of outliers.

- Transformation: Apply log or square root transformations to reduce the impact of outliers.

- Imputation: Replace outliers with a calculated value, such as the mean or median.

17. How do outliers affect when data is based on time series? What methods and techniques are available to detect outliers in time series data?

Outliers in time series can distort trend and seasonality patterns. Detection methods include moving averages, ARIMA models, and machine learning techniques such as isolation forests and seasonal decomposition.

18. What is dimensionality reduction, and what are some techniques that can be used for it?

Dimensionality reduction is the process of reducing the number of features in a dataset. Techniques include:

- Principal Component Analysis (PCA)

- Linear Discriminant Analysis (LDA)

- t-Distributed Stochastic Neighbor Embedding (t-SNE)

- Autoencoders

19. Discuss the role of data preparation in the context of artificial intelligence and machine learning.

Data preparation is crucial as it ensures the quality and relevance of data fed into machine learning models. It includes cleaning, transforming, and encoding data, handling missing values, and feature engineering, which significantly impacts the model's performance and accuracy.

**DAY 10**

**SQL:**

1. Do you know about ‘LOAD DATA INFILE’? What challenges occur while using this method?

LOAD DATA INFILE is a SQL command used to import data from a file into a table. It is particularly efficient for bulk data loading. However, there are several challenges associated with using this method:

- File Path Restrictions: The file must be accessible to the server, which can be problematic in secure environments.

- Data Format: The data must be formatted correctly to match the table structure.

- Error Handling: Errors in the data file (e.g., incorrect data types) can cause the entire load operation to fail.

- Permissions: The user must have the necessary permissions to execute the command and access the file.

- Security Risks: There is a risk of SQL injection if the file content is not properly sanitized.

2. NTILE() is one of the window functions. Where do we use this function and why?

NTILE() is a window function that distributes rows of a result set into a specified number of approximately equal groups. This is useful for:

- Data Segmentation: Splitting data into quantiles, such as quartiles or percentiles, for statistical analysis.

- Performance Analysis: Comparing performance metrics across different groups.

- Balanced Sampling: Creating balanced samples for further analysis or machine learning.

Example:

SELECT column,

NTILE(4) OVER (ORDER BY column) AS quartile

FROM table;

3. What is a correlated subquery in SQL?

A correlated subquery is a subquery that references columns from the outer query. It is executed once for each row processed by the outer query.

Example:

SELECT e1.name,

e1.salary

FROM employees e1

WHERE e1.salary > (SELECT AVG(e2.salary)

FROM employees e2

WHERE e1.department\_id = e2.department\_id);

4. How will you build a connection between a primary key and a foreign key?

A foreign key is used to establish a link between the data in two tables. The foreign key in one table points to the primary key in another table.

Example:

CREATE TABLE departments (

department\_id INT PRIMARY KEY,

department\_name VARCHAR(100)

);

CREATE TABLE employees (

employee\_id INT PRIMARY KEY,

employee\_name VARCHAR(100),

department\_id INT,

FOREIGN KEY (department\_id) REFERENCES departments(department\_id)

);

5. What is the purpose of the HAVING clause in SQL?

The HAVING clause is used to filter groups of rows after the GROUP BY clause has been applied. It is similar to the WHERE clause, but operates on aggregated data.

Example:

SELECT department\_id,

COUNT(employee\_id) AS num\_employees

FROM employees

GROUP BY department\_id

HAVING COUNT(employee\_id) > 10;

6. Write a SQL query that detects outliers.

Outliers can be detected using statistical methods, such as calculating the Z-score or using percentiles.

Example using Z-score:

SELECT \*,

(value - AVG(value) OVER ()) / STDDEV(value) OVER () AS z\_score

FROM table

WHERE ABS((value - AVG(value) OVER ()) / STDDEV(value) OVER ()) > 3;

**PowerBI:**

1. How would you use a Power Query parameter to filter data based on a specific date range?

You can create parameters in Power Query and use them to filter data.

Steps:

1. In Power Query, go to Manage Parameters > New Parameter.

2. Define the parameter (e.g., StartDate, EndDate).

3. Use these parameters in your query to filter data:

let

Source = ...,

FilteredRows = Table.SelectRows(Source, each [Date] >= StartDate and [Date] <= EndDate)

in

FilteredRows

2. Explain a scenario where composite models would be beneficial in a Power BI project considering different data sources and performance requirements.

Composite models in Power BI allow you to combine DirectQuery and Import mode in a single report. This is beneficial when:

- You need real-time data from certain sources (DirectQuery).

- You want to optimize performance by importing less frequently changing data.

For example, a retail dashboard that combines real-time sales data (DirectQuery) with static product information (Import).

3. Explain how you would use the CALCULATE function to apply a specific filter condition to a measure calculation.

The CALCULATE function evaluates an expression in a modified filter context.

Example:

TotalSales = CALCULATE(SUM(Sales[Amount]), Sales[Region] = "North")

This calculates the total sales amount only for the "North" region.

4. Provide an example of using Power Apps to fetch data from an external API and display it in a Power BI report.

1. Create a Power App that uses a connector to fetch data from an external API.

2. Embed the Power App in a Power BI report using the PowerApps visual.

3. Pass data between Power BI and Power Apps using context parameters.

5. Provide an example of using the first two lines of visualization to present real-time data or highlight current trends in a Power BI report.

You can use a line chart with two series:

1. One series for the current year’s data.

2. Another series for the previous year’s data to highlight trends and comparisons.

**Tableau:**

1. Demonstrate the usage of a hierarchy to perform drill-down analysis on a specific dimension exploring data at different levels of detail.

1. Create a hierarchy in Tableau by dragging dimensions into each other (e.g., Country > State > City).

2. Use this hierarchy in a visualization to allow users to drill down from Country to State to City by clicking on the plus (+) icon in the view.

2. Explain the concept of an aggregated measure versus a non-aggregated measure in Tableau and provide an example of each.

- Aggregated Measure: A measure that is summarized (e.g., SUM, AVG).

Example: `SUM(Sales)`

- Non-Aggregated Measure: A raw measure value.

Example: `Sales`

3. Demonstrate the process of blending data from a local Excel file and a remote SQL database creating a visualization that provides insights from both sources.

1. Connect to the Excel file and the SQL database in Tableau.

2. Define a relationship between the datasets using a common field.

3. Use fields from both data sources in a single worksheet to create a blended visualization.

4. Demonstrate how to use a table calculation to calculate the moving average of a measure over a specific period of time.

1. Create a line chart with the measure and date.

2. Add a table calculation for moving average:

- Right-click the measure > Add Table Calculation.

- Select Moving Average and specify the period.

5. Explain the difference between fixed LOD expressions and include/exclude LOD expressions in Tableau and provide examples of each.

- Fixed LOD Expression: Calculates values using the specified dimensions, ignoring the view context.

Example: `{ FIXED [Region] : SUM([Sales]) }`

- Include LOD Expression: Includes additional dimensions to the view context.

Example: `{ INCLUDE [Category] : SUM([Sales]) }`

- Exclude LOD Expression: Excludes dimensions from the view context.

Example: `{ EXCLUDE [Sub-Category] : SUM([Sales]) }`

**DAY 11**

1. Explain the concept of decorators in Python. Discuss how decorators can be used to create reusable code patterns and provide examples of advanced use cases such as creating class decorators or decorators with arguments.

Decorators in Python are a powerful tool to modify the behavior of a function or a method. They allow you to wrap another function to extend its behavior without permanently modifying it. This is done using the @decorator syntax above a function definition. Decorators can help create reusable code patterns by abstracting repetitive tasks.

Example of a simple decorator:

def my\_decorator(func):

def wrapper():

print("Something is happening before the function is called.")

func()

print("Something is happening after the function is called.")

return wrapper

@my\_decorator

def say\_hello():

print("Hello!")

say\_hello()

Advanced use cases include class decorators and decorators with arguments:

def repeat(n):

def decorator(func):

def wrapper(\*args, \*\*kwargs):

for \_ in range(n):

func(\*args, \*\*kwargs)

return wrapper

return decorator

@repeat(3)

def greet(name):

print(f"Hello, {name}!")

greet("Alice")

2. What are the similarities and differences between Python and R programming Language?

Similarities:

- Both are high-level, interpreted languages.

- Both support procedural, object-oriented, and functional programming paradigms.

- Extensive libraries and packages for data analysis, machine learning, and statistical computing.

- Strong community support and open-source.

Differences:

- Python is a general-purpose language, while R is specifically designed for statistical analysis and data visualization.

- Python uses a more general-purpose syntax, whereas R has a syntax tailored for statistical computing.

- Python is widely used in web development, automation, and software development, whereas R is predominantly used in academia and research for statistical analysis.

3. Suggest some scenarios where we need to use List and Tuple?

- Use Lists when:

- You need a collection of items that may need to change (mutable).

- You need to add, remove, or change elements.

- You need order and duplicate elements.

- Use Tuples when:

- You need a collection of items that should not change (immutable).

- You want to ensure data integrity by preventing changes.

- You need a sequence of elements to be used as dictionary keys.

- You want to optimize performance for large datasets, as tuples are faster than lists.

4. What is the Major difference between while loops and for loops? Give one example where we have to use these both loops.

- while loops are used when the number of iterations is not known beforehand. They continue to execute as long as the condition is true.

- for loops are used when the number of iterations is known or can be determined from the start (e.g., iterating over a range or a collection).

Example where both loops are used:

numbers = []

i = 0

while i < 10:

numbers.append(i)

i += 1

for number in numbers:

print(number)

5. There are duplicate entries in the list How would you remove duplicates without using any loops and libraries?

You can remove duplicates from a list using the set data structure, which inherently removes duplicates because it does not allow duplicate values.

my\_list = [1, 2, 3, 2, 1, 4, 5, 4]

my\_list = list(set(my\_list))

print(my\_list) # Output will be [1, 2, 3, 4, 5]

6. Write a Python program to calculate the factorial of a given number but with a twist. In addition to finding the factorial the program should also display each step of the factorial calculation. For example if the input number is 5 the program output should be:

def factorial\_with\_steps(n):

result = 1

for i in range(n, 0, -1):

if i != n:

print(f"Step {n - i + 1}: {result} \* {i} = {result \* i}")

result \*= i

print(f"Factorial of {n} is {result}")

factorial\_with\_steps(5)

**EDA:**

1. What are the primary and secondary purposes of a box plot? Suggest some alternatives for boxplot?

- Primary purposes:

- To display the distribution of data.

- To show the central tendency (median).

- To visualize the spread and skewness of the data.

- To identify outliers.

- Secondary purposes:

- To compare distributions across different categories or groups.

Alternatives to box plot:

- Violin plot: Combines a box plot with a kernel density plot.

- Strip plot: Displays individual data points.

- Swarm plot: Similar to strip plot but avoids overlapping points.

- Histogram: Shows the frequency distribution of data.

2. How positive skewness affects the distribution of the data.

Positive skewness (right skew) means that the tail on the right side of the distribution is longer or fatter than the left side. This indicates that there are a few unusually large values. As a result:

- The mean is typically greater than the median.

- The distribution is not symmetric.

- Data visualization and summary statistics may need to account for the skew.

3. Explain the difference between covariance and correlation and discuss their implications in analyzing the relationship between variables.

- Covariance:

- Measures the direction of the linear relationship between two variables.

- Can take any value (positive, negative, or zero).

- Magnitude is not standardized, making it difficult to interpret.

- Correlation:

- Measures both the direction and the strength of the linear relationship between two variables.

- Standardized between -1 and 1.

- Easier to interpret due to the standard scale.

Implications:

- Covariance indicates whether variables change together, but not how strongly they are related.

- Correlation provides both the direction and strength, making it more useful for comparing relationships between different pairs of variables.

4. We have a dataset and need to create a histogram. How do you calculate the appropriate number of bins and the bin width for the histogram using a specific formula? Take any example dataset.

One common method to calculate the number of bins is using Sturges' Rule:

Number of bins = log2(n) + 1

where n is the number of data points.

Example:

import numpy as np

import matplotlib.pyplot as plt

data = np.random.randn(1000)

n = len(data)

num\_bins = int(np.ceil(np.log2(n) + 1))

plt.hist(data, bins=num\_bins)

plt.show()

5. Which functions are used for labelling X-axis and Y-axis when using matplotlib library?

- plt.xlabel('X-axis label')

- plt.ylabel('Y-axis label')

**Data preprocessing:**

1. What is data preprocessing and identify and explain the potential challenges or issues that could arise if the dataset was not preprocessed before analysis?

Data preprocessing involves transforming raw data into a clean and usable format. It includes steps such as handling missing values, normalizing data, encoding categorical variables, and feature scaling.

Potential challenges if not preprocessed:

- Inaccurate results due to noise and outliers.

- Inefficient or failed algorithms due to inappropriate data formats.

- Misleading patterns due to unhandled missing values.

- Poor model performance due to unscaled features.

2. How would you handle the missing values using common techniques such as mean imputation forward or backward filling or advanced techniques like multiple imputation or predictive modeling?

- Mean/median/mode imputation:

df['column'].fillna(df['column'].mean(), inplace=True)

- Forward fill:

df['column'].fillna(method='ffill', inplace=True)

- Backward fill:

df['column'].fillna(method='bfill', inplace=True)

- Multiple imputation (e.g., using the mice package):

from sklearn.experimental import enable\_iterative\_imputer

from sklearn.impute import IterativeImputer

imputer = IterativeImputer()

df\_imputed = imputer.fit\_transform(df)

- Predictive modeling:

Train a model to predict missing values based on other features.

3. What is feature scaling and why is it necessary?

Feature scaling is the process of normalizing the range of independent variables or features of data. It is necessary because:

- Many machine learning algorithms are sensitive to the scale of data.

- Features with larger scales can dominate others and affect model performance.

- It speeds up convergence in optimization algorithms.

Common methods include standardization (z-score normalization) and min-max scaling.

4. After applying One Hot Encoding technique how would you assign names to each column?

Using pandas' get\_dummies function, you can assign names to columns:

import pandas as pd

df = pd.DataFrame({'color': ['red', 'blue', 'green']})

one\_hot\_encoded\_df = pd.get\_dummies(df, columns=['color'], prefix=['color'])

print(one\_hot\_encoded\_df)

5. Assuming your data is not stationary what Kind of Transformations would you like to do and In real world which Transformation is the best choice?

For non-stationary data, transformations such as differencing, logarithms, and seasonal decomposition can be applied.

- Differencing:

df['differenced'] = df['data\_column'].diff()

- Log transformation:

df['log\_transformed'] = np.log(df['data\_column'])

- Seasonal decomposition

:

from statsmodels.tsa.seasonal import seasonal\_decompose

result = seasonal\_decompose(df['data\_column'], model='additive', period=12)

result.plot()

plt.show()

The best choice depends on the nature of the data and the specific problem, but differencing and log transformation are commonly used techniques.

**DAY 12**

**SQL:**

1. Discuss the difference between aggregate functions and window functions in SQL and how can you partition data using window functions? Provide an example.

Aggregate functions perform a calculation on a set of values and return a single value, such as SUM, AVG, COUNT, MAX, MIN.

Window functions perform a calculation across a set of table rows related to the current row. Unlike aggregate functions, window functions do not cause rows to become grouped into a single output row. Example:

SELECT employee\_id, department, salary,

RANK() OVER (PARTITION BY department ORDER BY salary DESC) AS rank

FROM employees;

2. Can multiple conditions be used in a join statement? Provide one example.

Yes, multiple conditions can be used in a join statement. Example:

SELECT e.employee\_id, e.name, d.department\_name

FROM employees e

JOIN departments d ON e.department\_id = d.department\_id AND e.location\_id = d.location\_id;

3. What is the difference between DATE, DATETIME, and TIMESTAMP data types?

DATE stores only the date part (YYYY-MM-DD).

DATETIME stores both date and time (YYYY-MM-DD HH:MM:SS).

TIMESTAMP stores date and time with timezone information, and it is affected by the time zone changes.

4. What distinguishes an INNER JOIN from an OUTER JOIN in SQL?

INNER JOIN returns only the rows that have matching values in both tables.

OUTER JOIN returns all rows from one table and the matched rows from the second table. If no match is found, NULL values are returned for columns from the second table.

5. What function does SQL's GROUP BY clause serve?

GROUP BY groups rows that have the same values in specified columns into aggregated data. It is often used with aggregate functions like SUM, COUNT, AVG, etc.

6. How to find the employee with a second and third MAX Salary using a SQL query with and without using analytic functions?

Without analytic functions:

SELECT MAX(salary) AS second\_highest\_salary

FROM employees

WHERE salary < (SELECT MAX(salary) FROM employees);

SELECT MAX(salary) AS third\_highest\_salary

FROM employees

WHERE salary < (SELECT MAX(salary) FROM employees WHERE salary < (SELECT MAX(salary) FROM employees));

With analytic functions:

SELECT employee\_id, salary

FROM (

SELECT employee\_id, salary,

DENSE\_RANK() OVER (ORDER BY salary DESC) AS rank

FROM employees

) AS ranked

WHERE rank = 2 OR rank = 3;

Note: create a table with 10 entries in SQL which contains employee ID, employee name, employee age, and employee salary.

**PowerBI:**

1. Can you demonstrate how to create a filter in Power BI using both visual-level and report-level filtering?

Visual-level filter: Select a visualization, go to the Filters pane, and drag the field to the Visual level filters area. Set the filter condition.

Report-level filter: In the Filters pane, drag the field to the Report level filters area. Set the filter condition.

2. Discuss a real-world scenario where row-level security would be implemented in Power BI and explain the steps to set it up.

Scenario: A company wants to restrict sales data access so that sales representatives can only see data for their assigned regions.

Steps:

- Define roles in Power BI Desktop using Manage Roles under the Modeling tab.

- Create DAX expressions to filter data for each role.

- Publish the report to Power BI Service.

- Assign users to the roles in the dataset settings in Power BI Service.

3. Can you provide an example of how you would create a data model in Power BI for a sales analysis dashboard?

Import sales data and related tables (e.g., customers, products).

Create relationships between tables (e.g., Sales[CustomerID] -> Customers[CustomerID]).

Add calculated columns and measures (e.g., Total Sales = SUM(Sales[Amount])).

Build visuals using the fields and measures from the model.

4. Show me how to create a slicer in Power BI and how it can be used to interactively filter data in visualizations.

In Power BI Desktop, go to the Visualizations pane and select the Slicer visual.

Drag the field you want to use for filtering (e.g., Date) to the slicer.

Adjust slicer settings (e.g., dropdown, list).

The slicer will filter data in other visuals on the same page interactively.

5. Write a DAX expression using the CALCULATE function to calculate the total sales for a specific region and time period in Power BI.

Total Sales for Region and Period = CALCULATE(SUM(Sales[Amount]),

Sales[Region] = "Specific Region",

Sales[Date] >= DATE(Year1, Month1, Day1),

Sales[Date] <= DATE(Year2, Month2, Day2))

**Tableau:**

1. Demonstrate how to connect Tableau to a CSV file and perform basic data exploration tasks.

Open Tableau and connect to the CSV file using Connect -> To a File -> Text File.

Load the CSV file and go to the Data Source tab.

Drag and drop fields to the Rows and Columns shelves to create basic visualizations.

Use Show Me to explore different chart types.

2. Explain how Tableau Data Server facilitates collaboration among team members in a data analysis project.

Tableau Data Server allows users to centrally manage and share data sources.

Users can publish data sources to Tableau Server, allowing others to connect to the same data.

This ensures consistency in data definitions and calculations across the team.

It also enables access control and version management.

3. How would you handle a large dataset in Tableau to optimize performance and ensure efficient data analysis?

Use data extracts instead of live connections to improve performance.

Aggregate data to a higher level of granularity before importing.

Use filters to limit the data being loaded.

Optimize calculations by using context filters and avoiding complex calculations.

4. Compare and contrast Tableau Desktop, Tableau Prep, and Tableau Server in terms of their functionalities and use cases.

Tableau Desktop: Used for data visualization and analysis. Allows users to connect to various data sources, create dashboards, and perform ad-hoc analysis.

Tableau Prep: Used for data preparation. Allows users to clean, shape, and combine data from different sources before analysis.

Tableau Server: Used for sharing and collaboration. Allows users to publish dashboards, set up scheduled refreshes, and manage permissions.

5. Connect Tableau to a relational database (e.g., MySQL) and create a basic visualization using the connected data.

Open Tableau and connect to the MySQL database using Connect -> To a Server -> MySQL.

Enter the server details and credentials to establish the connection.

Load the desired tables and go to the Data Source tab.

Drag and drop fields to the Rows and Columns shelves to create a basic visualization.

**DAY 13**

**Python:**

1. Write a Python function that takes a string as input and converts it into a list of characters.

def string\_to\_list(s):

return list(s)

2. Explain how the 'zip' function works in Python. Provide an example of how you would use it to combine two lists into a dictionary.

The zip function takes iterables (can be zero or more), aggregates them in a tuple, and returns it. Example:

list1 = ['a', 'b', 'c']

list2 = [1, 2, 3]

combined\_dict = dict(zip(list1, list2))

This would result in {'a': 1, 'b': 2, 'c': 3}

3. How would you unzip a list of tuples in Python?

Using the zip function with the asterisk (\*) operator to unpack the list:

zipped\_list = [('a', 1), ('b', 2), ('c', 3)]

list1, list2 = zip(\*zipped\_list)

This would result in list1 = ('a', 'b', 'c') and list2 = (1, 2, 3)

4. Discuss the difference between a module and a package in Python. Provide examples of each and explain their respective use cases.

A module is a single Python file that can be imported and contains Python code, such as functions and classes. Example: my\_module.py.

A package is a collection of modules in a directory that includes a special \_\_init\_\_.py file. Example: my\_package (directory) containing \_\_init\_\_.py, module1.py, module2.py.

Use cases:

- Modules are used to organize code logically into smaller, reusable parts.

- Packages are used to organize related modules into a larger application or library.

5. Explain the purpose of using backreferences in regular expressions. Provide an example of a regular expression pattern that utilizes backreferences.

Backreferences in regular expressions allow you to reuse part of the matched pattern. They are useful for ensuring the same text appears multiple times.

Example:

pattern = r'(\b\w+)\s+\1'

text = 'hello hello world'

match = re.search(pattern, text)

This pattern matches repeated words like "hello hello".

6. Create a function that receives a list of names and a specific name as input and returns a boolean indicating whether the specific name is present in the list.

def is\_name\_in\_list(names\_list, specific\_name):

return specific\_name in names\_list

**EDA:**

1. What are the formulae used to calculate the lower limit and upper limit in a box plot? Provide an example calculation based on a dataset.

Lower limit = Q1 - 1.5 \* IQR

Upper limit = Q3 + 1.5 \* IQR

Example:

Given Q1 = 25 and Q3 = 75, IQR = 50

Lower limit = 25 - 1.5 \* 50 = -50

Upper limit = 75 + 1.5 \* 50 = 150

2. In ACF (Autocorrelation Function) and PACF (Partial Autocorrelation Function) plots, what do the X-axis and Y-axis represent? Explain their interpretations.

The X-axis represents the lag (time interval).

The Y-axis represents the correlation coefficient.

Interpretation:

- ACF shows the correlation between the series and its lagged values.

- PACF shows the correlation between the series and its lagged values after removing the effects of intermediate lags.

3. What is the only calculation that can be performed on categorical data in EDA? Provide an example of how this calculation can be used to analyze categorical data.

The only calculation is the frequency count.

Example:

data = ['cat', 'dog', 'cat', 'bird', 'dog', 'dog']

from collections import Counter

frequency = Counter(data)

This would result in {'cat': 2, 'dog': 3, 'bird': 1}

4. Discuss the primary and secondary uses of a histogram in exploratory data analysis. Provide an example scenario where a histogram would be beneficial.

Primary use: To display the distribution of a dataset.

Secondary use: To identify the central tendency, spread, and skewness.

Example scenario:

A histogram would be beneficial to visualize the distribution of ages in a survey to understand the age demographics.

5. When visualizing a bar plot using Matplotlib and Seaborn, which Python functions are typically used? Provide code examples demonstrating the creation of bar plots using both libraries.

Matplotlib:

import matplotlib.pyplot as plt

categories = ['A', 'B', 'C']

values = [10, 20, 15]

plt.bar(categories, values)

plt.show()

Seaborn:

import seaborn as sns

import pandas as pd

data = pd.DataFrame({'Category': ['A', 'B', 'C'], 'Values': [10, 20, 15]})

sns.barplot(x='Category', y='Values', data=data)

plt.show()

6. Why is Seaborn known as an advanced visualization library? What are the various visualization libraries except Matplotlib and Seaborn?

Seaborn is known as an advanced visualization library because it provides high-level interfaces for drawing attractive and informative statistical graphics.

Other visualization libraries:

- Plotly

- Bokeh

- ggplot

- Altair

**Data preprocessing:**

1. Explain the purpose of feature selection in data preprocessing. Discuss different techniques for feature selection and their advantages/disadvantages.

Purpose: To select the most relevant features for model training, reducing dimensionality and improving model performance.

Techniques:

- Filter methods (e.g., correlation, chi-square): Simple and fast, but may miss interactions between features.

- Wrapper methods (e.g., forward selection, backward elimination): Consider feature interactions, but are computationally expensive.

- Embedded methods (e.g., Lasso, decision trees): Balance between performance and complexity, but may be biased by the model used.

2. How would you handle missing values in data preprocessing? Discuss different strategies and techniques such as imputation or deletion.

Strategies:

- Deletion: Remove rows or columns with missing values. Simple but can lead to loss of information.

- Imputation: Fill missing values using mean, median, mode, or predictive models. Retains data but can introduce bias.

Techniques:

- Mean/median imputation: df['column'].fillna(df['column'].mean(), inplace=True)

- Forward/backward fill: df['column'].fillna(method='ffill', inplace=True)

- Predictive modeling: Train a model to predict missing values.

3. Explain hypothesis testing and discuss what kind of testing is preferred to check the stationarity of time series data.

Hypothesis testing is a statistical method to make inferences or decisions based on sample data. It involves formulating null and alternative hypotheses, calculating a test statistic, and making a decision based on a significance level.

To check stationarity of time series data, the Augmented Dickey-Fuller (ADF) test is preferred.

4. Discuss the purpose of outlier treatment in data preprocessing and provide examples of commonly used techniques for outlier detection and handling.

Purpose: To improve model accuracy and performance by removing or correcting anomalous data points.

Techniques:

- Z-score method: Identifies outliers based on standard deviations from the mean.

- IQR method: Identifies outliers using the interquartile range.

- Clustering methods: Use clustering algorithms to identify outliers.

Example:

Z-score method:

from scipy import stats

df = df[(np.abs(stats.zscore(df)) < 3).all(axis=1)]

5. How do you deal with seasonality in time series data? Discuss different approaches and techniques for identifying and handling seasonality effects in time series analysis.

Approaches:

- Decomposition: Separate the time series into trend, seasonal, and residual components.

- Differencing: Remove seasonality by differencing the data at seasonal lags.

- Seasonal adjustment: Use methods like Seasonal and Trend decomposition using Loess (STL).

Techniques:

- Seasonal decomposition:

from statsmodels.tsa.seasonal import seasonal\_decompose

result = seasonal\_decompose(time\_series, model='additive', period=12)

result.plot()

plt.show()

- Seasonal differencing:

df['seasonal\_diff'] = df['data\_column'].diff(periods=12)