4. Big Data Analytics

1. Explain:-

i. Mean (Average):- The mean is the sum of all values divided by the total number of values. It is calculated by summing up all the values in the dataset and dividing them by the number of values.

Formula: Mean (mu) = Σx_i / n
Example:
Data: [4, 8, 6, 5, 3, 7]
Mean = (4+8+6+5+3+7) / 6 = 33 / 6 = 5.5

ii. Median:-
The median is the middle value of a sorted dataset.If we have even number of values in the data set then median is sum of mid two numbers divided by 2.
Steps:
- Arrange the data in ascending order.
- If n is odd: take the middle value.
- If n is even: average of two middle values.

Example:
Data: [4, 8, 6, 5, 3, 7] → Sorted: [3, 4, 5, 6, 7, 8]
Median = (5 + 6) / 2 = 5.5

iii. Mode:-The mode is the value(s) that appears most frequently.The mode is the number that appears most frequently in the dataset.

Example:
Data: [4, 8, 6, 4, 3, 7]
Mode = 4

iv. **Variance:-**

Variance measures the dispersion of a dataset, indicating how much the values differ from the mean. It is the average of the squared differences from the mean.

Formula:

Population: sigma^2 = $\Sigma(x_i - mu)^2 / n$

Sample: s^2 = $\Sigma(x_i - \bar{x})^2 / (n - 1)$

Example:

Data: [4, 8, 6, 5, 3, 7], Mean = 5.5

Variance = [(4-5.5)^2 + ... + (7-5.5)^2] / 6 = 2.92

v. **Standard Deviation:-** Standard deviation is the square root of the variance, providing a measure of the spread of the dataset in the same units as the data.

Formula:

sigma = √Variance

Example:

Standard Deviation = √2.92 ≈ 1.71

2. Explain different data transformation techniques:-

- Data transformation in data mining refers to the process of converting raw data into a format that is suitable for analysis and modeling.
- It also ensures that data is free of errors and inconsistencies. The goal of data transformation is to prepare the data for data mining so that it can be used to extract useful insights and knowledge.

1. Smoothing

Smoothing is used to remove noise from a dataset using algorithms, highlighting important features and helping predict patterns. This technique simplifies data analysis by revealing trends and patterns not easily seen in raw data.

2. Aggregation

Aggregation involves collecting and presenting data in a summarized format. Data from various sources can be integrated for comprehensive analysis. For example, sales data can be aggregated monthly or yearly to identify trends and inform business decisions.

3. Discretization
Discretization converts continuous data into intervals or categories, making it easier for some data mining frameworks to process. For example, ages 1-10 can be categorized as 'young', and 11-20 as 'middle age'.

4. Attribute Construction
New attributes are created from existing ones to simplify data and enhance mining efficiency. This helps uncover hidden patterns that original attributes might not reveal directly.

5. Generalization
Generalization replaces low-level data with high-level concepts using a hierarchy. For example, converting numerical age data (22, 25) into categorical values (young, old) or generalizing address data to city or country level.

6. Normalization
Normalization scales data into a specific range. Common techniques include:
- Min-Max Normalization: v' = (v - min_A) / (max_A - min_A)
- Z-Score Normalization: Values are normalized using the attribute's mean and standard deviation to compute a z-score.

3.Explain different kinds of Big Data Analysis:-

1. Descriptive Analysis

Purpose: Understand what has happened over a specific period.

Description:
Descriptive analytics summarizes historical data to identify patterns and trends. It answers the question **"What happened?"**

Example:

- Monthly sales reports
- Website traffic analysis
- Customer demographics breakdown

Tools: Excel, Google Analytics, Tableau

2. Diagnostic Analysis

Purpose: Understand **why** something happened.

Description:
This analysis digs deeper into data to discover the causes of past outcomes. It often involves data mining, correlation, and drill-down techniques.

Example:

- Investigating a sudden drop in sales
- Finding the root cause of a system outage

Tools: SQL, SAS, R, Python

3. Predictive Analysis

Purpose: Forecast future outcomes based on historical data.

Description:
Predictive analytics uses statistical models and machine learning techniques to predict future events or trends. It answers the question **"What is likely to happen?"**

Example:

- Predicting customer churn
- Forecasting sales for the next quarter

Tools: Python (Scikit-learn), R, RapidMiner, IBM SPSS

4. Prescriptive Analysis

Purpose: Recommend actions to take for optimal outcomes.

Description:
This advanced analysis suggests decision options based on predicted outcomes. It answers **"What should we do?"**

Example:

- Supply chain optimization
- Personalized marketing strategies

Tools: Apache Spark, MATLAB, decision management systems

5. Real-Time Analysis

Purpose: Provide immediate insights from streaming data.

Description:
Real-time analytics processes data as it's generated to allow instant decision-making. Useful in time-sensitive environments.

Example:

- Fraud detection in banking
- Monitoring traffic congestion or system performance

Tools: Apache Kafka, Apache Storm, Apache Flink

6. Text Analysis (Text Mining)

Purpose: Extract insights from unstructured text data.

Description:
Text analytics transforms unstructured data (emails, social media, reviews) into structured data for analysis.

Example:

- Sentiment analysis of product reviews
- Topic modeling from customer feedback

Tools: NLTK, SpaCy, IBM Watson, TextBlob

4. Data Visualization Role and Importance:-

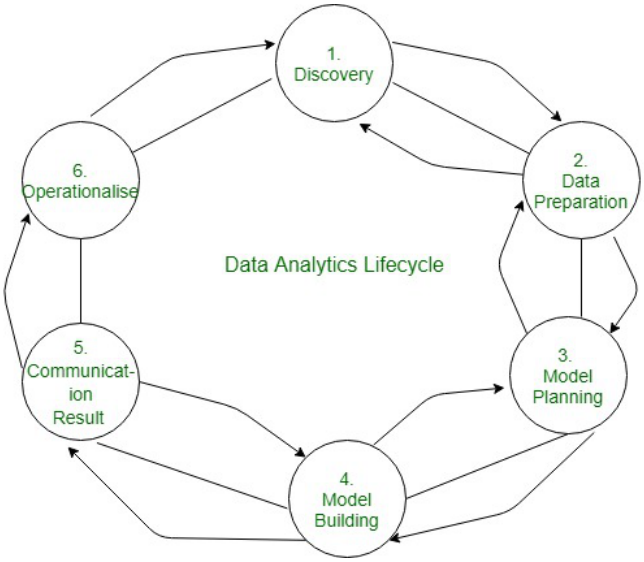
- Data visualization is the graphical representation of information..
- Data visualization translates complex data sets into visual formats that are easier for the human brain to understand.
- This can include a variety of visual tools such as:
    - **Charts**: Bar charts, line charts, pie charts, etc.
    - **Graphs**: Scatter plots, histograms, etc.
    - **Maps**: Geographic maps, heat maps, etc.
    - **Dashboards**: Interactive platforms that combine multiple visualizations.
- The primary goal of data visualization is to make data more accessible and easier to interpret allow users to identify patterns, trends, and outliers quickly. This is particularly important in big data where the large volume of information can be confusing without effective visualization techniques.

Role:-

1. **Data Exploration**
   Helps identify patterns, correlations, and outliers quickly in large datasets.
2. **Insight Discovery**
   Makes complex data understandable, helping uncover actionable insights.
3. **Real-Time Monitoring**
   Enables immediate decisions using live dashboards fed by streaming data.
4. **Enhanced Communication**
   Translates complex analytics into visuals easily understood by non-technical stakeholders.
5. **Strategic Decision Support**
   Helps executives and teams track KPIs and forecast trends.
6. **Simplifying Complexity**
   Converts multi-dimensional, unstructured, or high-volume data into digestible visuals.
7. **Interactive Analysis**
   Allows users to drill down, filter, and customize views for deeper insight.
8. **Supports Predictive Modeling**
   Visualizes machine learning outcomes, making them easier to interpret and apply.

5.Life Cycle Phases of Data Analytics:-

- The Data analytic lifecycle is designed for Big Data problems and data science projects.
- The cycle is iterative to represent real project.
- To address the distinct requirements for performing analysis on Big Data, step–by–step methodology is needed to organize the activities and tasks involved with acquiring, processing, analyzing, and repurposing data.



Phase 1: Discovery

In this phase, the data science team investigates the problem, understands the project context, and identifies what data is needed and available. They also create an initial hypothesis to test later with data.
Example: A company wants to reduce employee turnover. The team reviews HR reports, interview records, and employee feedback to define what data should be collected and what patterns to look for.

Phase 2: Data Preparation

Data is collected, cleaned, and organized in an analytic environment (sandbox) where it can be safely explored and transformed. This step may be repeated several times depending on the data quality and needs.
Example: The team merges attendance logs, salary records, and employee surveys, removes inconsistencies, and fills missing values to prepare the dataset for analysis.

Phase 3: Model Planning

The team explores relationships within the data and selects the most suitable modeling techniques and variables. They also prepare separate data sets for training and testing.
Example: For predicting employee resignation, the team finds that work experience, promotion history, and job satisfaction are key factors, and chooses logistic regression and decision tree models.

Phase 4: Model Building

Here, the selected models are built, tested, and fine-tuned using the prepared datasets. The team checks whether existing tools are sufficient or if new tools are needed.
Example: The team tests different models using training data, evaluates their performance, and decides to use a random forest model that gives the best accuracy.

Phase 5: Communication of Results

The team evaluates the model results and presents them to stakeholders with visualizations and narratives. They highlight the key findings, limitations, and potential business impact.
Example: The team shows that long overtime hours are a strong predictor of turnover and suggests changes in workload policies, supported by charts and cost analysis.

Phase 6: Operationalize

The model is tested in a real-world setting through a pilot project. This allows the team to monitor performance and make adjustments before full deployment. Final reports, code, and recommendations are delivered.
Example: The company runs the model in one department to identify at-risk employees and provide career growth options. After success, the model is rolled out across the company.

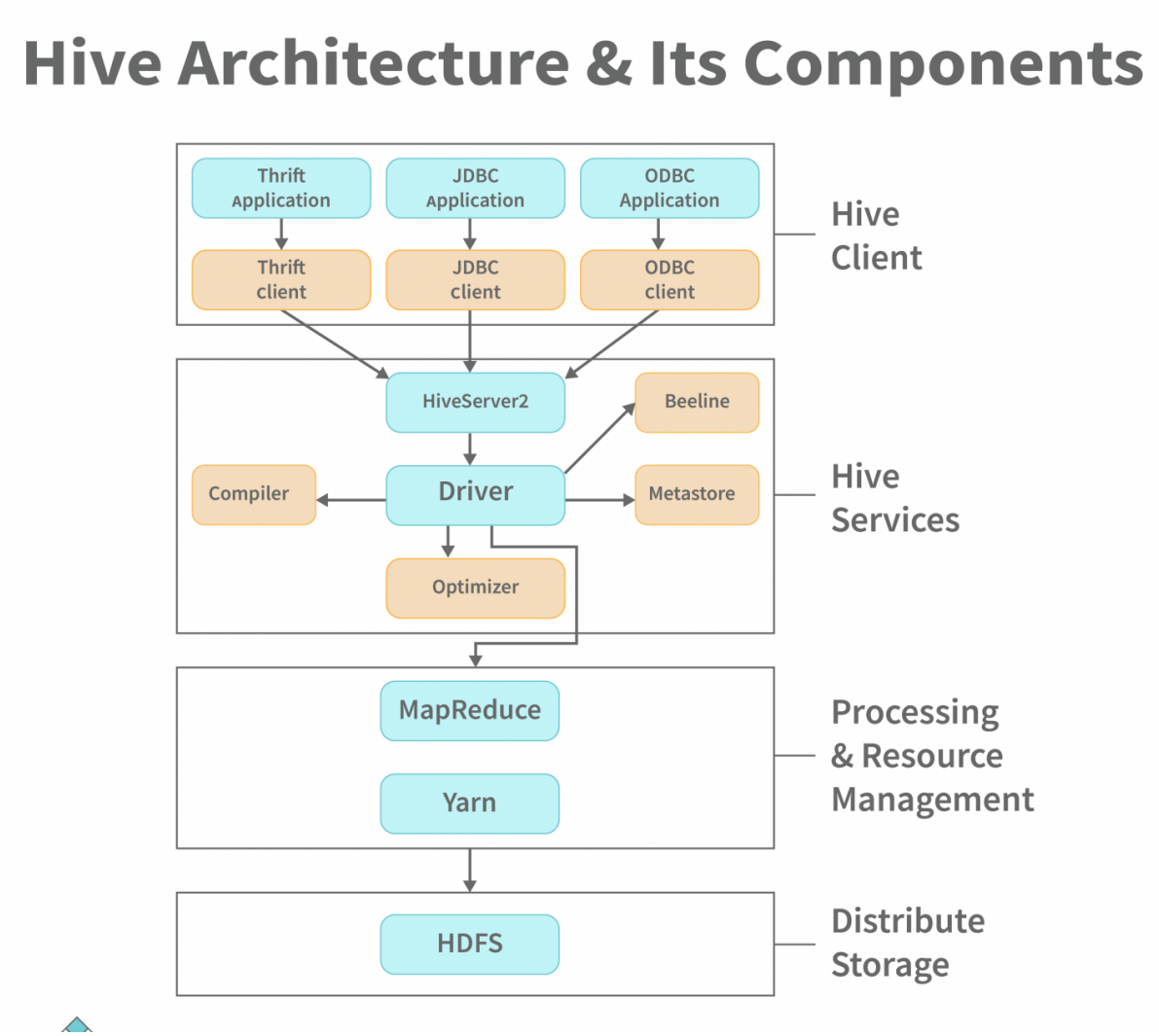
6. Draw and explain Architecture of HIVE:-

- The Facebook open-source data warehousing tool Apache Hive was designed to eliminate the job of writing the MapReduce Java program. Facebook developed it to decrease the amount of code it requires.
- Apache Hive is a data warehouse infrastructure built on top of Hadoop. It allows users to query and analyze large datasets stored in Hadoop Distributed File System (HDFS) using a SQL-like language called **HiveQL**.

1. Hive Client Layer

This is where users interact with Hive.

- **Thrift, JDBC, ODBC Applications**: These are interfaces that allow external applications to connect to Hive.
- **Thrift, JDBC, ODBC Clients**: These clients act as communication channels between Hive and the external tools.
- These connect to **HiveServer2**, which acts as the central control unit for handling Hive queries.



2. Hive Services Layer

This layer handles query compilation, optimization, and execution planning.

- **HiveServer2**: Receives queries from clients, manages sessions, and returns results.
- **Beeline**: A command-line shell that connects to HiveServer2 for executing HiveQL queries.
- **Driver**: Manages the execution lifecycle of a HiveQL query:
  - **Compiler**: Translates HiveQL into an execution plan (usually MapReduce).
  - **Optimizer**: Refines the execution plan for better performance.
- **Metastore**: Stores metadata like table schema, partitions, and HDFS paths.

3. Processing & Resource Management Layer

This is the execution engine layer that actually processes the data.

- **MapReduce**: The default execution engine that processes the data in a distributed manner.
- **YARN (Yet Another Resource Negotiator)**: Manages and schedules system resources for jobs.

4. Distributed Storage Layer

Where the actual data is stored.

- **HDFS (Hadoop Distributed File System)**: This is the storage system used by Hive to store large amounts of data across multiple nodes.

How it works:

1. A query is submitted using a client (e.g., Beeline, JDBC).
2. HiveServer2 receives it and sends it to the **Driver**.
3. The **Compiler** checks metadata from the **Metastore** and creates an execution plan.
4. The **Optimizer** improves the plan.
5. The plan is executed using **MapReduce**, and resources are managed by **YARN**.
6. Data is read from or written to **HDFS**.
7. The result is returned to the user.

7.What is data Wrangling? Why do you need it? explain data Wrangling methods?

- Data wrangling, or data munging, is a crucial process in the data analytics workflow that involves cleaning, structuring, and enriching raw data to transform it into a more suitable format for analysis.
- This process includes cleaning the data by removing or correcting inaccuracies, inconsistencies, and duplicates.
- It also involves structuring the data, often converting it into a tabular form that is easier to work with in analytical applications.
- The goal is to make data more usable, reliable, and ready for tasks like analytics, machine learning, or reporting.

Need Data Wrangling:-

- Missing values
- Duplicates
- Inconsistent formats

- Irrelevant or redundant information
- Outliers

Data Wrangling Methods:-

1. Data Collection

- Gathering data from various sources (e.g., CSV files, databases, APIs, web scraping).

2. Data Cleaning

Fixing or removing bad data.

- **Removing duplicates**
- **Handling missing values** (filling, dropping, or flagging)
- **Correcting errors** (e.g., typos, wrong formats)
- **Filtering out irrelevant records**

3. Data Structuring

Transforming data into a desired shape or model.

- **Pivoting/unpivoting tables**
- **Splitting/merging columns**
- **Normalizing nested data (e.g., JSON into flat tables)**

4. Data Enrichment

Adding new information to improve the dataset.

- **Merging with external datasets**
- **Deriving new columns** (e.g., age from date of birth)

5. Data Validation

Ensuring the data conforms to rules and expectations.

- Check for valid ranges, formats, and data types.
- Example: "Age should be between 0 and 120".

6. Data Transformation

Applying changes to standardize and prepare data.

- **Scaling numeric values**
- **Encoding categorical variables**
- **Standardizing text (e.g., lowercasing, trimming whitespace)**

8. Missing values are filled in Pandas Data Frame:-

In Pandas, missing data occurs when some values are missing or not collected properly and these missing values are represented as:

- **None**: A Python object used to represent missing values in object-type arrays.
- **NaN**: A special floating-point value from NumPy which is recognized by all systems that use IEEE floating-point standards.

Checking Missing Values in Pandas

1. Using isnull()

isnull() returns a DataFrame of Boolean value where **True** represents missing data (**NaN**). This is simple if we want to find and fill missing data in a dataset.

2. Using notnull()

notnull() function returns a DataFrame with Boolean values where **True** indicates non-missing (valid) data. This function is useful when we want to focus only on the rows that have valid, non-missing values.

```
import pandas as pd

import numpy as np

d = {'First Score': [100, 90, np.nan, 95],

     'Second Score': [30, 45, 56, np.nan],

     'Third Score': [np.nan, 40, 80, 98]}

df = pd.DataFrame(d)

mv = df.isnull()

nmv = df.notnull()

print(mv)
```

	First Score	Second Score	Third Score
0	False	False	True
1	False	False	False
2	True	False	False
3	False	True	False

Filling Missing Values in Pandas

Following functions allow us to replace missing values with a specified value or use interpolation methods to find the missing data.

1. Using fillna():-

```
import pandas as pd

import numpy as np

d = {'First Score': [100, 90, np.nan, 95],

     'Second Score': [30, 45, 56, np.nan],

     'Third Score': [np.nan, 40, 80, 98]}

df = pd.DataFrame(d)

df.fillna(0)
```

	First Score	Second Score	Third Score
0	100.0	30.0	0.0
1	90.0	45.0	40.0
2	0.0	56.0	80.0
3	95.0	0.0	98.0

2. Using replace()

Use **replace()** function to replace NaN values with a specific value.2. Using replace()]

Example

```
import pandas as pd
import numpy as np

data = pd.read_csv("/content/employees.csv")
data[10:25]
data.replace(to_replace=np.nan, value=-99)
```

Dropping Missing Values in Pandas:-

The **dropna()** function used to removes rows or columns with NaN values. It can be used to drop data based on different conditions.

1. Dropping Rows with At Least One Null Value

Remove rows that contain at least one missing value.

Example

```
import pandas as pd
import numpy as np

dict = {'First Score': [100, 90, np.nan, 95],
        'Second Score': [30, np.nan, 45, 56],
        'Third Score': [52, 40, 80, 98],
        'Fourth Score': [np.nan, np.nan, np.nan, 65]}
df = pd.DataFrame(dict)

df.dropna()
```

Output

	First Score	Second Score	Third Score	Fourth Score
3	95.0	56.0	98	65.0

2. Dropping Rows with All Null Values

We can drop rows where all values are missing using **dropna(how='all')**.

Example

```
dict = {'First Score': [100, np.nan, np.nan, 95],
        'Second Score': [30, np.nan, 45, 56],
        'Third Score': [52, np.nan, 80, 98],
        'Fourth Score': [np.nan, np.nan, np.nan, 65]}
df = pd.DataFrame(dict)

df.dropna(how='all')
```

Output

	First Score	Second Score	Third Score	Fourth Score
0	100.0	30.0	52.0	NaN
2	NaN	45.0	80.0	NaN
3	95.0	56.0	98.0	65.0

9. What is categorical variable? Why do you need categorical variable encoding? With an example, explain one-hot encoding:-

Categorical Variable:-

A **categorical variable** is a variable that can take on **limited and usually fixed** values, representing categories or labels. These are **non-numeric** values like:

- **Colors**: Red, Green, Blue
- **Genders**: Male, Female
- **Cities**: Delhi, Mumbai, Bangalore

They don't have a mathematical meaning—**you can't calculate average or sum** of categories.

Most **machine learning algorithms** work only with **numerical input**. Since categorical variables are non-numeric, we must **convert them to numbers** so that the model can understand and learn from them.

🔢 One-Hot Encoding: With Example

One-Hot Encoding is a method where **each category is converted into a new binary column (0 or 1)**.

🧾 Example:

Suppose we have a column `City`:

Person	City
A	Delhi
B	Mumbai
C	Bangalore

Using **one-hot encoding**, we convert this into:

Person	City_Bangalore	City_Delhi	City_Mumbai
A	0	1	0
B	0	0	1
C	1	0	0

Each city gets its own column, and a **1 indicates** the person belongs to that city, **0 otherwise**.

10. What is a Dataset:-

A **Dataset is a set of data grouped into a collection** with which developers can work to meet their goals.

In a dataset, the rows represent the number of data points and the columns represent the features of the Dataset.

They are mostly used in fields like machine learning, business, and government to gain insights, make informed decisions, or train algorithms.

Datasets may vary in size and complexity and they mostly require cleaning and preprocessing to ensure data quality and suitability for analysis or modeling.

Types of Datasets:-

1. **Numerical Dataset:** They include numerical data points that can be solved with equations. These include temperature, humidity, marks and so on.

   Example of a Numerical Dataset in Python (using Pandas)

```
import pandas as pd

# Creating a numerical dataset
data = {
    'Student': ['Alice', 'Bob', 'Charlie'],
    'Math_Score': [88, 92, 79],
    'Science_Score': [85, 90, 95],
    'Attendance': [28, 30, 27]
}

df = pd.DataFrame(data)
print(df)
```

   🔽 **Output:-**

```
  Student  Math_Score  Science_Score  Attendance
0   Alice          88             85          28
1     Bob          92             90          30
2 Charlie          79             95          27
```

2. **Categorical Dataset:** These include categories such as colour, gender, occupation, games, sports and so on.

Example: Using pandas DataFrame

```
import pandas as pd

# Structured dataset as a table
data = {
   'Name': ['Alice', 'Bob', 'Charlie'],
   'Age': [25, 30, 35],
   'City': ['Delhi', 'Mumbai', 'Bangalore']
}

df = pd.DataFrame(data)
print(df)
```

🔽 **Output:**

```
      Name  Age      City
0    Alice   25     Delhi
1      Bob   30    Mumbai
2  Charlie   35  Bangalore
```

3. **Web Dataset:** These include datasets created by calling APIs using HTTP requests and populating them with values for data analysis. These are mostly stored in JSON (JavaScript Object Notation) formats.

4. **Time series Dataset:** These include datasets between a period, for example, changes in geographical terrain over time.

5. **Image Dataset:** It includes a dataset consisting of images. This is mostly used to differentiate the types of diseases, heart conditions and so on.

6. **Ordered Dataset:** These datasets contain data that are ordered in ranks, for example, customer reviews, movie ratings and so on

11. Explain Min-max scaling. For the following dataset carry out min max Scaling, X = 24,28,53,30,40,18,15,21.

Min-Max Scaling (also known as **Normalization**) is a method used to scale numerical data into a fixed range, usually **0 to 1**.

📌 Min-Max Scaling Formula:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Where:

- X = original value
- X_{min} = minimum value in the dataset
- X_{max} = maximum value in the dataset

🔢 Given Dataset:

```ini
X = [24, 28, 53, 30, 40, 18, 15, 21]
```

- Minimum value $X_{min} = 15$
- Maximum value $X_{max} = 53$

✅ Apply Min-Max Scaling to each value:

$$\text{Scale}(x) = \frac{x - 15}{53 - 15} = \frac{x - 15}{38}$$

Original X	Calculation	Scaled X
24	(24 - 15) / 38 = 9 / 38	**0.2368**
28	(28 - 15) / 38 = 13 / 38	**0.3421**
53	(53 - 15) / 38 = 38 / 38	**1.0000**
30	(30 - 15) / 38 = 15 / 38	**0.3947**
40	(40 - 15) / 38 = 25 / 38	**0.6579**
18	(18 - 15) / 38 = 3 / 38	**0.0789**
15	(15 - 15) / 38 = 0 / 38	**0.0000**
21	(21 - 15) / 38 = 6 / 38	**0.1579**

📄 Scaled Dataset:

```csharp
[0.2368, 0.3421, 1.0, 0.3947, 0.6579, 0.0789, 0.0, 0.1579]
```

12. Explain z-score normalization. following dataset carry out z-score normalization(standardization), X = 23, 29, 52, 31, 45, 19, 18, 27.

Z-score normalization, also called **standardization**, is a method used to rescale data such that it has a **mean (μ) of 0** and a **standard deviation (σ) of 1**.

The formula for calculating the **z-score** for a value x is:

$$z = \frac{x - \mu}{\sigma}$$

Where:

- x is the original value,
- μ is the mean of the dataset,
- σ is the standard deviation of the dataset.

Step-by-Step for Dataset

Given data:

$$X = [23, 29, 52, 31, 45, 19, 18, 27]$$

Step 1: Calculate the Mean (μ)

$$\mu = \frac{23 + 29 + 52 + 31 + 45 + 19 + 18 + 27}{8} = \frac{244}{8} = 30.5$$

Step 2: Calculate the Standard Deviation (σ)

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}$$

$$= \sqrt{\frac{(23 - 30.5)^2 + (29 - 30.5)^2 + \ldots + (27 - 30.5)^2}{8}}$$

$$= \sqrt{\frac{(56.25 + 2.25 + 462.25 + 0.25 + 210.25 + 132.25 + 156.25 + 12.25)}{8}}$$

$$= \sqrt{\frac{1032}{8}} = \sqrt{129} \approx 11.36$$

Step 3: Apply Z-score Formula

Now we calculate z-scores:

$$z = \frac{x - 30.5}{11.36}$$

X	z-score
23	$\frac{23-30.5}{11.36} \approx -0.66$
29	$\frac{29-30.5}{11.36} \approx -0.13$
52	$\frac{52-30.5}{11.36} \approx 1.89$
31	$\frac{31-30.5}{11.36} \approx 0.04$
45	$\frac{45-30.5}{11.36} \approx 1.28$
19	$\frac{19-30.5}{11.36} \approx -1.01$
18	$\frac{18-30.5}{11.36} \approx -1.10$
27	$\frac{27-30.5}{11.36} \approx -0.31$