**🔹 1. How to Identify Null Values**

In **Python (Pandas)**:

python

CopyEdit

import pandas as pd

# Sample DataFrame

df = pd.DataFrame({

'A': [1, 2, None, 4],

'B': [None, 2, 3, 4]

})

# Identify null values

df.isnull() # Shows True/False

df.isnull().sum() # Count of nulls per column

df.isnull().any() # Check if any null exists per column

df.isnull().values.any() # Check if there are any nulls in the entire DataFrame

**🔹 2. How to Impute Null Values**

**Imputation** means filling missing values using strategies like:

python

CopyEdit

# Fill with a specific value

df['A'].fillna(0, inplace=True)

# Fill with mean/median/mode

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

df['A'].fillna(df['A'].median(), inplace=True)

df['A'].fillna(df['A'].mode()[0], inplace=True)

**🔹 3. Various Methods of Null Value Imputation and Removal**

**✅ Imputation Techniques**

* **Mean/Median/Mode Imputation**
* **Forward Fill (ffill) or Backward Fill (bfill)**
* **K-Nearest Neighbors (KNN) Imputation**
* **Multivariate Imputation (MICE)**
* **Interpolation (linear, time-based)**

**🔁 Example:**

python

CopyEdit

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

**❌ Removal Techniques**

* **Drop Rows with Nulls:**

python

CopyEdit

df.dropna(inplace=True)

* **Drop Columns with Nulls:**

python

CopyEdit

df.dropna(axis=1, inplace=True)

**🔹 4. What are Outliers?**

Outliers are **data points that differ significantly** from other observations.

Example: In a dataset of human height (average 5.5 to 6.5 ft), a value of **10 ft** or **0.5 ft** is an outlier.

**🔹 5. Ways to Identify Outliers**

**📊 Statistical Methods:**

1. **Z-score** (standard deviation from mean):

python

CopyEdit

from scipy import stats

z = np.abs(stats.zscore(df['A']))

df[z > 3] # Outliers if Z > 3

1. **IQR (Interquartile Range)**:

python

CopyEdit

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

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

IQR = Q3 - Q1

outliers = df[(df['A'] < (Q1 - 1.5 \* IQR)) | (df['A'] > (Q3 + 1.5 \* IQR))]

**📈 Visualization Methods:**

* **Boxplot**
* **Histogram**
* **Scatter Plot**

**🔹 6. How to Remove Outliers**

**Using IQR:**

python

CopyEdit

df = df[~((df['A'] < (Q1 - 1.5 \* IQR)) | (df['A'] > (Q3 + 1.5 \* IQR)))]

**Using Z-score:**

python

CopyEdit

df = df[z < 3]

**Optional: Cap or Transform Outliers**

* Replace with boundaries (winsorization)
* Use log or square root transformation

Would you like a small dataset example with code to see this in action?

4o

**You said:**

What are independent and dependent variables? What is standard deviation, covariance? Explain linear, logistic, and Naïve Bayes with their advantages and disadvantages. What is supervised and unsupervised learning? Why perform scaling and normalization? How to perform scaling and normalization? What is overfitting and underfitting of data? How and why to convert categorical values into numerical (all methods of conversion)? What is hyperparameter tuning? What are evaluation parameters for classification and regression algorithms? What is Precision, Recall, F-Score, MSE, MAE, R² score? What does the describe() method specify?

**ChatGPT said:**

Great set of questions! These are essential for understanding machine learning and data analysis. Below are clear, concise explanations for each.

**🔹 1. What are Independent and Dependent Variables?**

* **Independent Variables (Features):** Input variables used to predict outcomes (e.g., age, income).
* **Dependent Variable (Target):** Output variable you're trying to predict (e.g., house price).

Example: In predicting house prices:

* Independent: size, location, age of house
* Dependent: price

**🔹 2. What is Standard Deviation and Covariance?**

* **Standard Deviation (σ):** Measures how spread out the data is from the mean.
* **Covariance:** Measures how two variables vary together (positive: same direction, negative: opposite).

**🔹 3. Linear, Logistic Regression & Naïve Bayes**

| **Algorithm** | **Use Case** | **Advantages** | **Disadvantages** |
| --- | --- | --- | --- |
| **Linear Regression** | Predicting continuous values | Simple, interpretable, fast | Assumes linearity, sensitive to outliers |
| **Logistic Regression** | Binary classification (Yes/No) | Good baseline, probabilistic output | Doesn't work well with non-linear data |
| **Naïve Bayes** | Classification (spam detection) | Fast, handles high dimensional data | Assumes feature independence (often unrealistic) |

**🔹 4. Supervised vs. Unsupervised Learning**

| **Type** | **Labeled Data?** | **Examples** |
| --- | --- | --- |
| **Supervised** | ✅ Yes | Regression, Classification |
| **Unsupervised** | ❌ No | Clustering, Dimensionality Reduction |

**🔹 5. Why Perform Scaling and Normalization?**

* Many ML models (like KNN, SVM) are distance-based and get biased if features are on different scales.
* To bring all features to the same scale.

**🔹 6. How to Perform Scaling and Normalization?**

python

CopyEdit

from sklearn.preprocessing import StandardScaler, MinMaxScaler

# Standard Scaling (mean = 0, std = 1)

scaler = StandardScaler()

df\_scaled = scaler.fit\_transform(df)

# Normalization (0 to 1 range)

scaler = MinMaxScaler()

df\_normalized = scaler.fit\_transform(df)

**🔹 7. What is Overfitting and Underfitting?**

| **Concept** | **Meaning** | **Problem** | **Solution** |
| --- | --- | --- | --- |
| **Overfitting** | Model learns noise | High accuracy on training but poor test performance | Simplify model, regularization |
| **Underfitting** | Model too simple | Poor on both training and test data | Add complexity or features |

**🔹 8. Convert Categorical to Numerical**

**Methods:**

1. **Label Encoding** (ordinal data):

python

CopyEdit

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

df['gender'] = le.fit\_transform(df['gender'])

1. **One-Hot Encoding** (nominal data):

python

CopyEdit

pd.get\_dummies(df['color'])

1. **Ordinal Encoding** (manually define order):

python

CopyEdit

size\_map = {'Small': 0, 'Medium': 1, 'Large': 2}

df['size'] = df['size'].map(size\_map)

**🔹 9. What is Hyperparameter Tuning?**

* Process of finding the best parameters that control the learning process (like learning rate, number of neighbors in KNN).

**Methods:**

* Grid Search
* Random Search
* Bayesian Optimization

python

CopyEdit

from sklearn.model\_selection import GridSearchCV

**🔹 10. Evaluation Parameters**

**For Classification:**

* Accuracy
* Precision
* Recall
* F1 Score
* Confusion Matrix
* ROC-AUC

**For Regression:**

* **MSE** (Mean Squared Error)
* **MAE** (Mean Absolute Error)
* **R² Score** (Coefficient of Determination)

**🔹 11. Precision, Recall, F1, MSE, MAE, R²**

| **Metric** | **Description** |
| --- | --- |
| **Precision** | TP / (TP + FP) – Out of predicted positives, how many are correct? |
| **Recall** | TP / (TP + FN) – Out of actual positives, how many did we catch? |
| **F1 Score** | 2 \* (Precision \* Recall) / (Precision + Recall) – Harmonic mean of precision and recall |
| **MSE** | Mean of squared errors – penalizes large errors |
| **MAE** | Mean of absolute errors – more interpretable |
| **R² Score** | How well the model explains variance (1 = perfect, 0 = mean prediction) |

**🔹 12. What does describe() Method Specify?**

python

CopyEdit

df.describe()

Returns summary statistics for numeric columns:

* Count
* Mean
* Standard Deviation
* Min, Max
* 25%, 50%, 75% (Quartiles)

Would you like this compiled as a one-page PDF or cheat sheet?

4o

**You said:**

What is data visualization? What is the difference between univariate, bivariate, and multivariate analysis? What is the RMSE? How can you select K for K-Means? Explain normal distribution. What is a confusion matrix? What do you understand about the true-positive rate and false-positive rate? What is the difference between a Type I and Type II error? How can you calculate accuracy using a confusion matrix? Explain what tokenization is in NLP and why it is important. Explain the difference between stemming and lemmatization. Explain tokenization, POS tagging, and stop words. Explain the difference between a bar chart and a histogram. How do you choose the right type of chart for your data? What libraries or tools do you prefer for data visualization in Python? Why? How do you handle missing data when creating visualizations? Explain the difference between qualitative and quantitative data visualization. How do you evaluate the effectiveness of a data visualization? Additional Tools What is SCALA and Impala? How to install and execute programs in Scala and Impala?

**ChatGPT said:**

Here's a comprehensive answer to each of your questions, categorized into themes for clarity:

**🔹 DATA VISUALIZATION & ANALYSIS**

**1. What is Data Visualization?**

It is the graphical representation of information and data using charts, graphs, and maps to help understand trends, patterns, and outliers.

**2. Difference Between Univariate, Bivariate, and Multivariate Analysis**

| **Type** | **Variables** | **Purpose** | **Example Chart** |
| --- | --- | --- | --- |
| Univariate | 1 | Describe single variable | Histogram, Boxplot |
| Bivariate | 2 | Analyze relationship | Scatter plot |
| Multivariate | 3+ | Understand interactions | Heatmap, 3D plot |

**3. What is RMSE (Root Mean Squared Error)?**

It measures the average squared difference between predicted and actual values:

RMSE=1n∑(yi−y^i)2\text{RMSE} = \sqrt{\frac{1}{n} \sum (y\_i - \hat{y}\_i)^2}RMSE=n1​∑(yi​−y^​i​)2​

* Lower RMSE = better model performance.

**4. How to Select K in K-Means Clustering?**

* **Elbow Method**: Plot K vs inertia (WCSS) and look for the “elbow” point.
* **Silhouette Score**: Measures how well points fit within their cluster.
* **Gap Statistic**: Compares performance against a reference null model.

**5. Explain Normal Distribution**

* A symmetric, bell-shaped distribution.
* Mean = Median = Mode.
* 68% of data within 1σ, 95% within 2σ, 99.7% within 3σ.

**🔹 CONFUSION MATRIX & CLASSIFICATION METRICS**

**6. What is a Confusion Matrix?**

A 2x2 table used for binary classification:

|  | **Predicted Yes** | **Predicted No** |
| --- | --- | --- |
| **Actual Yes** | TP (True Positive) | FN (False Negative) |
| **Actual No** | FP (False Positive) | TN (True Negative) |

**7. True-Positive Rate (Recall) & False-Positive Rate**

* **TPR (Recall)** = TP / (TP + FN): Correctly predicted positives.
* **FPR** = FP / (FP + TN): Incorrectly predicted negatives as positives.

**8. Type I vs. Type II Error**

| **Error Type** | **Meaning** |
| --- | --- |
| **Type I** | False Positive – Rejecting true null |
| **Type II** | False Negative – Accepting false null |

**9. Calculate Accuracy from Confusion Matrix**

Accuracy=TP+TNTP+TN+FP+FN\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}Accuracy=TP+TN+FP+FNTP+TN​

**🔹 NLP BASICS**

**10. What is Tokenization in NLP?**

Breaking text into words, phrases, or sentences.

* Important for transforming raw text into analyzable units.

**11. Stemming vs. Lemmatization**

| **Method** | **Description** | **Example (from "running")** |
| --- | --- | --- |
| **Stemming** | Cuts suffix (may be incorrect) | "run" |
| **Lemmatization** | Converts to root using dictionary | "run" (accurate) |

**12. Tokenization, POS Tagging, Stop Words**

* **Tokenization**: Splitting text into words/tokens.
* **POS Tagging**: Identifying part of speech (noun, verb, etc.)
* **Stop Words**: Common words like "the", "is" that are usually removed.

**🔹 CHARTS & VISUALIZATION**

**13. Difference Between Bar Chart and Histogram**

| **Feature** | **Bar Chart** | **Histogram** |
| --- | --- | --- |
| Data Type | Categorical | Continuous numerical |
| Gaps between bars | Yes | No |

**14. How to Choose the Right Chart**

* **Categorical** → Bar, Pie
* **Continuous** → Histogram, Line
* **Relationships** → Scatter, Heatmap
* **Time Series** → Line Chart
* **Distribution** → Boxplot, Violin

**15. Preferred Python Libraries for Data Visualization**

| **Library** | **Why Use It** |
| --- | --- |
| **Matplotlib** | Foundation, fully customizable |
| **Seaborn** | Built on matplotlib, statistical visuals |
| **Plotly** | Interactive visualizations |
| **Altair** | Declarative, great for quick exploration |

**16. Handling Missing Data in Visualizations**

* Drop or impute missing values before plotting.
* Use color/annotations to indicate missing points.
* Avoid misleading plots due to gaps in data.

**17. Qualitative vs. Quantitative Data Visualization**

| **Type** | **Data Type** | **Visualization Example** |
| --- | --- | --- |
| **Qualitative** | Categorical | Pie chart, bar chart |
| **Quantitative** | Numerical | Histogram, line plot |

**18. How to Evaluate Effectiveness of Visualization**

* **Clarity**: Is it easy to understand?
* **Accuracy**: Does it reflect the true data?
* **Insight**: Does it convey a story or pattern?
* **Minimalism**: Avoid clutter or overdesign.

**🔹 TOOLS & LANGUAGES**

**19. What is Scala?**

* A **functional and object-oriented programming language** that runs on the JVM.
* Often used with Apache Spark for big data processing.

**20. What is Impala?**

* A **SQL engine for Hadoop**, used for querying large datasets stored in HDFS or Hive using low latency.

**21. Installing and Running Scala and Impala**

**✅ Scala:**

* **Install**: Download from <https://scala-lang.org>
* **Run**:

bash

CopyEdit

scala # for REPL

scalac Hello.scala # compile

**✅ Impala:**

* Part of **Cloudera** distribution.
* **Run via shell**:

bash

CopyEdit

impala-shell

Scala 12

# Step 1: Install Java (Scala needs it)

sudo apt update

sudo apt install default-jdk -y

# Step 2: Install Scala

sudo apt install scala -y

# Step 4: Compile the Scala file

scalac HelloWorld.scala

# Step 5: Run the program

scala HelloWorld

code1:

object SumTwoNumbers {

def main(args: Array[String]): Unit = {

val a = 10

val b = 20

val sum = a + b

println(s"The sum of $a and $b is $sum")

}

}

Code2:

object PrintNumbers {

def main(args: Array[String]): Unit = {

for (i <- 1 to 5) {

println(s"Number: $i")

}

}

}

11.Impala

# Step 1: Install Docker

sudo apt update

sudo apt install docker.io -y

sudo systemctl start docker

sudo systemctl enable docker

# Step 2: Run Impala + Hive + Hadoop using Docker

docker run -d --name impala \

-p 25000:25000 -p 21000:21000 -p 21050:21050 \

bigdataeu/impala:latest

# Step 3: Wait for services to be ready (optional sleep)

sleep 30

# Step 4: Enter the Impala container and run SQL

docker exec -i impala impala-shell <<EOF

SHOW DATABASES;

CREATE DATABASE sl3\_impala;

USE sl3\_impala;

CREATE TABLE employees (

id INT,

name STRING,

department STRING,

salary DOUBLE

)

STORED AS PARQUET;

INSERT INTO employees VALUES

(1, 'Aarav', 'Engineering', 95000),

(2, 'Priya', 'Marketing', 72000),

(3, 'Rohan', 'Finance', 85000),

(4, 'Saanvi', 'Engineering', 98000),

(5, 'Karthik', 'Engineering', 87000),

(6, 'Neha', 'Marketing', 74000);

SELECT \* FROM employees WHERE department = 'Engineering';

SELECT \* FROM employees WHERE salary > 90000;

SELECT department, AVG(salary) AS avg\_salary FROM employees GROUP BY department;

SELECT \* FROM employees ORDER BY salary DESC LIMIT 3;

SELECT e.name, e.department, e.salary FROM employees e

JOIN (

SELECT department, AVG(salary) AS avg\_salary FROM employees

GROUP BY department

) dept\_avg

ON e.department = dept\_avg.department

WHERE e.salary < dept\_avg.avg\_salary;

EOF