import pandas as pd

import numpy as np

\_

df=pd.read\_csv('iris.csv')

df.describe() # Print description of DataFrame

\_

print("First 5 values:\n", df.head())

print ("Last 5 values:\n", df.tail())

\_

df.duplicated()

\_

df.isnull()

\_

df.info()

\_

df.shape

\_

df["sepal.length"].isnull()

\_

y = df.drop(["petal.length"], axis=1) # axis=1 column. For row, axis=0

print(y)

\_

df['variety'].replace(['Setosa', 'Virginica'], [0,1], inplace=True)

print(df)

\_

df.isnull().sum()

\_

Great questions, Vishwas! These are core topics in **Data Preprocessing** for DSBDA. Let’s break each one down clearly with real-world examples where possible.

**1️⃣ Explain Data Preprocessing in Detail**

**Data Preprocessing** is the step in data analysis and machine learning where raw data is cleaned and transformed into a usable format. Real-world data is often messy — it can have missing values, inconsistent formats, outliers, or irrelevant features.

**Key Steps in Data Preprocessing:**

| **Step** | **Description** |
| --- | --- |
| **1. Data Cleaning** | Remove or fix incorrect, incomplete, or inconsistent data (e.g., nulls, typos). |
| **2. Data Integration** | Combine data from multiple sources (e.g., joining multiple CSVs). |
| **3. Data Transformation** | Normalize, scale, or encode data so ML models can use it. |
| **4. Data Reduction** | Reduce the volume of data (e.g., PCA, feature selection). |
| **5. Data Discretization** | Convert continuous data into categorical (e.g., age groups). |

**Example:**

If a dataset has a column Age with some values missing and Gender as text ("Male", "Female"), preprocessing would:

* Fill missing ages (mean/median)
* Convert gender to numerical format (e.g., 0 = Male, 1 = Female)
* Normalize age values between 0 and 1

**2️⃣ Explain DataFrame with Suitable Example**

A **DataFrame** is a 2D table-like data structure (rows and columns), mainly used in **Pandas** (Python library). Think of it as an Excel spreadsheet in Python.

**Features:**

* Labeled rows and columns
* Can hold multiple data types
* Easy manipulation and analysis

**Example (Python Code):**

import pandas as pd

data = {

'Name': ['Vishwas', 'Aarav', 'Sana'],

'Age': [21, 22, 20],

'Score': [85, 90, 88]

}

df = pd.DataFrame(data)

print(df)

**Output:**

Name Age Score

0 Vishwas 21 85

1 Aarav 22 90

2 Sana 20 88

**3️⃣ What is the Limitation of the Label Encoding Method?**

**Label Encoding** converts categorical values into integers. For example, "Red", "Green", "Blue" → 0, 1, 2.

**Limitation:**

It **introduces a false sense of order** or hierarchy.

Example:

Color: Red = 0, Green = 1, Blue = 2

A model might think Blue > Green > Red — but colors don't have such order.

**Solution:**

Use **One-Hot Encoding** instead, which treats all categories equally.

**4️⃣ What is the Need for Data Normalization?**

**Normalization** adjusts the scale of numeric features to a common range (usually 0 to 1 or -1 to 1).

**Why Needed:**

* Algorithms like KNN, SVM, and Gradient Descent-based models are sensitive to feature scale.
* Prevents features with large values from dominating the model.

**Example:**

If you have:

* Income: 50,000 to 1,00,000
* Age: 18 to 60

Without normalization, Income dominates. After normalization, all features contribute equally.

**5️⃣ What are the Different Techniques for Handling Missing Data?**

**Techniques:**

| **Method** | **Description** |
| --- | --- |
| **Remove Rows** | Delete rows with missing values (only if very few). |
| **Mean/Median/Mode Imputation** | Replace missing values with mean (for continuous), or mode (for categorical). |
| **Forward/Backward Fill** | Use previous/next value in the column. |
| **Interpolation** | Estimate missing values using linear or polynomial methods. |
| **Use Algorithms (KNN Imputer, MICE)** | Advanced methods to predict and fill missing data. |
| **Flag Missing Data** | Create a new column indicating which rows had missing values (for transparency). |

Would you like the code implementation for any of these techniques in Python (using Pandas or Scikit-learn)?

Perfect, Vishwas!  
Here’s your full list of **important viva questions with detailed answers** based exactly on your practical ("Data Wrangling-I") and **your provided code**.

# 📚 ****Detailed Viva Questions and Answers****

## 1. ****What is data wrangling? Why is it important?****

**Answer:**  
Data wrangling is the process of cleaning, structuring, and enriching raw data into a desired format for better decision-making. It is important because real-world data is messy — it can have missing values, inconsistent formats, duplicate data, etc. Cleaned and properly formatted data improves the accuracy and performance of data analysis or machine learning models.

## 2. ****What are the steps involved in data wrangling?****

**Answer:**  
The main steps are:

* Importing data
* Understanding the data
* Cleaning missing or incorrect values
* Formatting and normalizing data types
* Removing duplicates
* Transforming categorical data
* Saving the cleaned data for analysis

## 3. ****What are missing values? How can you handle them?****

**Answer:**  
Missing values are data entries that are empty or not available (like NaN).  
Handling methods include:

* Removing rows/columns with too many missing values
* Filling missing values using methods like mean, median, or mode
* Predicting missing values based on other data
* Leaving them if they don’t affect analysis

## 4. ****What are categorical variables? How do you convert them into numerical variables?****

**Answer:**  
Categorical variables represent types or categories (like 'Setosa', 'Versicolor' in Iris dataset).  
They can be converted into numbers using:

* **Label Encoding** (e.g., 'Setosa' → 0, 'Versicolor' → 1)
* **One-hot Encoding** (creates a separate column for each category)

In your code, you used **label encoding** with replace().

## 5. ****What is the difference between**** describe()****,**** info()****, and**** shape ****in pandas?****

**Answer:**

* describe(): Shows statistical summary (mean, std, min, max) for numerical columns.
* info(): Gives a concise summary of the DataFrame (data types, non-null counts).
* shape: Shows dimensions of the DataFrame (rows, columns) as a tuple.

## 6. ****What is normalization? Why do we need it?****

**Answer:**  
Normalization is scaling numerical data between a certain range (like 0 and 1).  
It is important because it prevents large numbers from dominating smaller ones and helps machine learning models perform better.

## 7. ****What is the difference between**** isnull() ****and**** isnull().sum()****?****

**Answer:**

* isnull(): Returns a DataFrame of **True/False** indicating where values are missing.
* isnull().sum(): Gives the **total count** of missing values in each column.

## 8. ****What does**** duplicated() ****function do in pandas?****

**Answer:**  
duplicated() returns a Boolean Series indicating duplicate rows (True if the row is a duplicate of a previous one).

## 9. ****How do you load a dataset in Python using pandas?****

**Answer:**  
By using pd.read\_csv('filename.csv'), you can load a CSV file into a pandas DataFrame.

## 10. ****What are different types of data types available in pandas?****

**Answer:**  
Common data types:

* **int64**: Integer numbers
* **float64**: Decimal numbers
* **object**: Text or mixed types (for strings)
* **bool**: Boolean values (True/False)
* **category**: Categorical data

## 11. ****What does**** drop() ****function do? What is axis=0 and axis=1?****

**Answer:**

* drop() removes rows or columns.
* axis=0: Drop a row.
* axis=1: Drop a column.

In your code:

y = df.drop(["petal.length"], axis=1)

drops the **petal.length** column.

## 12. ****How do you replace values in a column?****

**Answer:**  
Using replace() function:

df['variety'].replace(['Setosa', 'Virginica'], [0,1], inplace=True)

This replaces 'Setosa' with 0 and 'Virginica' with 1 **inside** the "variety" column.

## 13. ****How do you find dimensions of a dataset?****

**Answer:**  
Using df.shape, which returns (rows, columns).

## 14. ****What is label encoding and one-hot encoding?****

**Answer:**

* **Label Encoding**: Converts each category into a single number (e.g., 'A'→0, 'B'→1).
* **One-hot Encoding**: Converts categories into separate columns with 0/1 values.

## 15. ****What do you mean by open-source dataset?****

**Answer:**  
An open-source dataset is freely available for public use without any restrictions.  
Example sources: Kaggle, UCI Machine Learning Repository, GitHub.

# ⚡ ****Code-Specific Questions with Answers****

## 16. ****Which dataset did you use?****

**Answer:**  
I used the **Iris dataset** (iris.csv), commonly available from Kaggle or UCI ML repository.

## 17. ****What does**** df.describe() ****show?****

**Answer:**  
It shows statistics like **count**, **mean**, **std deviation**, **minimum**, **maximum**, and **quartiles** for numeric columns.

## 18. ****What information does**** df.info() ****provide?****

**Answer:**  
It provides:

* Column names
* Non-null counts
* Data types
* Memory usage

It helps understand the structure of the dataset quickly.

## 19. ****What is the output of**** df.shape****?****

**Answer:**  
It returns a tuple (rows, columns), like (150, 5) for the Iris dataset.

## 20. ****What does**** df.duplicated() ****check?****

**Answer:**  
It checks if there are any duplicate rows in the dataset.

## 21. ****What does**** df.isnull() ****return?****

**Answer:**  
It returns True where there are missing (null) values and False elsewhere.

## 22. ****Why did you use**** drop(["petal.length"], axis=1)****?****

**Answer:**  
I dropped the "petal.length" column to create a new DataFrame without it, maybe for analysis or model preparation.

## 23. ****What does**** df['variety'].replace(['Setosa', 'Virginica'], [0,1], inplace=True) ****do?****

**Answer:**  
It replaces the text categories 'Setosa' and 'Virginica' with 0 and 1 **inside** the "variety" column.

## 24. ****After replacing categorical values, what happens to the "variety" column?****

**Answer:**  
It becomes a numerical column instead of categorical (object type).

## 25. ****Why did you check**** df["sepal.length"].isnull() ****separately?****

**Answer:**  
To specifically check if there are any missing values in the "sepal.length" column.

## 26. ****What is the importance of handling missing values before analysis?****

**Answer:**  
Missing values can lead to incorrect statistical results and errors in machine learning models, so handling them properly improves model accuracy.

## 27. ****After replacing values with 0 and 1, is "variety" still categorical or numeric?****

**Answer:**  
It becomes numeric (integer type).

## 28. ****What is inplace=True doing in the**** replace() ****function?****

**Answer:**  
inplace=True updates the original DataFrame without creating a new one.

## 29. ****Why is it necessary to check for duplicated data?****

**Answer:**  
Duplicates can mislead analysis results, so they should be removed or handled properly.

## 30. ****If there were missing values, what methods could you use to handle them?****

**Answer:**

* Drop rows/columns with missing data
* Fill missing values with mean, median, mode
* Predict missing values based on other data