Dsbda Viva Ques

A1

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

👉 Data wrangling is cleaning and organizing raw data into a usable format. It's important for accurate analysis.

2. What are the steps involved in data wrangling?

👉 Loading data, checking missing values, formatting types, normalizing, and encoding categorical variables.

3. What is the Iris dataset about?

👉 It's about 150 iris flowers with features like sepal/petal length and width, classified into 3 species.

4. Where did you get the Iris dataset from?

👉 From UCI Machine Learning Repository / Seaborn GitHub.

5. What are the columns/features in the Iris dataset?

👉 Sepal length, sepal width, petal length, petal width, species.

6. How many rows and columns are there in the dataset?

👉 150 rows and 5 columns.

7. What type of variables are present in the dataset?

👉 Four numeric (float) variables, one categorical variable (species).

8. What does df.describe() show?

👉 Statistical summary like mean, min, max, and standard deviation for numeric columns.

9. How do you check for missing values in the dataset?

👉 Using df.isnull() and df.isnull().sum().

10. How do you drop a column? What does axis=1 mean?

👉 Use df.drop(['column_name'], axis=1). axis=1 means drop a column.

11. Why did you drop the petal.length column?

👉 Just for practicing the drop() method, not because it was necessary.

12. How do you check for duplicated rows?

👉 Using df.duplicated().

13. What is the use of df.info()?

👉 It shows the data types, non-null values, and memory usage.

14. Why do we need to convert categorical variables into numerical values?

👉 Because machine learning models and numerical analysis require numeric input.

15. How did you convert the variety column into numbers?

👉 Using df['variety'].replace() or astype('category').cat.codes.

16. What will happen if we don't encode all categories properly?

👉 The model or analysis will give errors or wrong results.

17. What is astype('category') used for?

👉 To convert text/categorical columns into efficient categorical data types.

18. What is the difference between isnull() and isnull().sum()?

👉 isnull() shows True/False for each cell; isnull().sum() gives the total missing values per column.

19. What is the role of pandas and NumPy in your code?

👉 Pandas for data manipulation; NumPy for numerical operations.

20. How would you handle missing data if there were missing values?

👉 Options: remove rows (dropna()), fill with mean/median (fillna()), or interpolate.

1. What happens if we have too many missing values in a column?

👉 If too many values are missing, it's better to **drop the column** because filling it would introduce bias.

2. What is normalization and why is it done?

👉 Normalization scales data (e.g., between 0 and 1) to make different features comparable, especially for machine learning models.

3. Difference between drop() and del in pandas?

👉 drop() can remove rows/columns flexibly without deleting the original DataFrame unless inplace=True.

👉 del directly deletes a column from the DataFrame.

4. Why do we check for duplicated data?

👉 Duplicates can bias the analysis and give wrong conclusions; we need to remove or handle them.

5. What does shape[0] and shape[1] represent?

👉 shape[0] gives the number of **rows**, shape[1] gives the number of **columns**.

A2

1. Q: What is Data Wrangling? Why is it important?

A:
Data wrangling is the process of cleaning, transforming, and organizing raw data into a usable format. It is important because real-world data is often messy, incomplete, or inconsistent, and wrangling prepares the data for analysis or machine learning models.

---

2. Q: How did you scan for missing values in your dataset?

A:
I used the isnull() and sum() functions in pandas.
Example:

python

CopyEdit

df.isnull().sum()

This shows how many missing values are there in each column.

---

3. Q: How did you handle missing values?

A:
Depending on the situation, I used:

- **Mean or median imputation** for numeric columns.
- **Mode imputation** for categorical columns.
- **Dropping rows** if too many values were missing. Example:

python

CopyEdit

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

---

4. Q: What techniques did you use to find outliers?

A:
I used the **Interquartile Range (IQR) method**.
I calculated Q1 and Q3, then found the lower and upper bounds:

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

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

IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR

upper_bound = Q3 + 1.5 * IQR

How to calculate IQR?

1. **Q1 (First Quartile)** → 25% of the data lies below this value.
   (25th percentile)

2. **Q3 (Third Quartile)** → 75% of the data lies below this value. (75th percentile)

3. **IQR = Q3 - Q1**

4. **Outlier detection:**
    - **Lower bound** = Q1 - 1.5 × IQR
    - **Upper bound** = Q3 + 1.5 × IQR
    - Any data point **below lower bound** or **above upper bound** is considered an **outlier**.

---

5. Q: How did you handle outliers?

A:
I **capped** the outliers — meaning if a value was above the upper bound, I replaced it with the upper bound, and if below lower bound, replaced it with the lower bound.

Example:

```
df['Marks'] = df['Marks'].apply(lambda x: upper_bound if x > upper_bound else (lower_bound if x < lower_bound else x))
```

---

6. Q: Which data transformation did you apply and why?

A:
I applied **Min-Max Scaling** on the 'Attendance' column to bring all values between 0 and 1.
Reason: To change the scale for better understanding and to prepare the data for any ML models that are sensitive to feature scales.

Example:

```
df['Attendance_Scaled'] = (df['Attendance'] - df['Attendance'].min()) / (df['Attendance'].max() - df['Attendance'].min())
```

---

7. Q: What are other common data transformations apart from Min-Max Scaling?

A:
Other common transformations include:

- **Standardization** (mean = 0, standard deviation = 1)
- **Log transformation** (to reduce skewness)
- **Box-Cox transformation** (to normalize distribution)
- **Square root transformation** (to reduce variance)

---

8. Q: Why is handling missing values important before analysis?

A:
Missing values can:

- Bias the analysis
- Cause errors in calculations
- Lead to wrong conclusions Therefore, it is important to handle them properly.

---

9. Q: What happens if you don't deal with outliers?

A:
Outliers can:

- Distort the mean and standard deviation
- Affect the performance of machine learning models
- Lead to misleading results in analysis.

---

10. Q: Can you explain the difference between normalization and standardization?

A:

- **Normalization (Min-Max Scaling):** Rescales the data between 0 and 1.
- **Standardization (Z-Score Scaling):** Rescales data to have a mean of 0 and a standard deviation of 1.

A3

1. What are descriptive statistics?

Answer:
Descriptive statistics are methods for summarizing and organizing data so it can be easily understood. They include **measures of central tendency** (like mean, median) and **measures of variability** (like range, variance, standard deviation).

---

2. What are measures of central tendency?

Answer:
Measures of central tendency describe the center point of a dataset. Gives general idea about the whole dataset. The main measures are:

- **Mean:** The average of the data values.
- **Median:** The middle value when data is sorted.
- **Mode:** The most frequent value.

---

3. What are measures of variability?

Answer:
Measures of variability describe how spread out the data values are. They include:

- **Range:** Difference between maximum and minimum.
- **Variance:** The average of the squared differences from the mean.
- **Standard Deviation:** The square root of the variance.

---

4. What is grouping by a categorical variable?

Answer:
Grouping by a categorical variable means splitting the data based on different categories (like age group, gender, species, etc.) and then analyzing each group separately. For example, finding the average income for different age groups.

---

5. What is a percentile?

Answer:
A percentile shows the position of a value in a dataset. For example, the 25th percentile means 25% of data values are below that point.

---

6. What is the Iris dataset?

Answer:
The Iris dataset is a famous dataset that contains measurements (sepal and petal lengths and widths) of 150 iris flowers, divided into 3 species: **Setosa, Versicolor, and Virginica**.

---

7. Why do we calculate mean, median, standard deviation separately for each species in Iris dataset?

Answer:
Because each species has different flower measurements. Grouping by species helps us understand how the characteristics differ between the species.

---

8. What Python libraries are used for this practical?

Answer:
We mainly use:

- **pandas** for data manipulation
- **numpy** for numerical operations (optional)
- **matplotlib / seaborn** for visualization (optional

A4

Data Analytics is the process of **examining, organizing, and interpreting raw data** to find useful information, draw conclusions, and support decision-making.
It involves various techniques like cleaning data, transforming it, finding patterns, making predictions, and visualizing the results.

What is Regression?

Regression is a type of supervised machine learning where the **output (target) is a continuous value**.
Example: Predicting house prices, predicting temperature, predicting sales.

What is Classification?

Classification is a type of supervised machine learning where the **output is a category or label**.
Example: Identifying emails as "spam" or "not spam," classifying animals as "dog" or "cat."

What is the difference between Regression and Classification?

Aspect	Regression	Classification
Output Type	Continuous (real numbers)	Discrete (categories or classes)
Example	Predicting salary	Predicting if a patient has a disease (Yes/No)
Algorithms	Linear Regression, Polynomial Regression	Logistic Regression, Decision Tree, Random Forest
Goal	Predict **how much** or **how many**	Predict **which class**

1. What is the objective of your practical?

Answer:
The objective is to create a Linear Regression model using Python or R to predict the prices of houses in Boston based on various features provided in the Boston Housing dataset.

---

2. What is Linear Regression?

Answer:
Linear Regression is a supervised machine learning algorithm that models the relationship between a dependent variable (target) and one or more independent variables (features) by fitting a linear equation to observed data.

---

3. What is the dependent variable in the Boston Housing dataset?

Answer:
The dependent variable is **MEDV** (Median value of owner-occupied homes in $1000s).

---

4. Name a few independent variables (features) from the dataset.

Answer:
Some independent variables are:

- **CRIM:** Crime rate per capita
- **RM:** Average number of rooms per dwelling
- **AGE:** Proportion of owner-occupied units built before 1940
- **LSTAT:** Percentage of lower status of the population
- **INDUS:** Proportion of non-retail business acres per town

---

5. How many samples and features are there in the Boston Housing dataset?

Answer:
There are **506 samples** and **14 feature variables** (including the target variable).

---

6. What Python libraries did you use for this practical?

Answer:
I used libraries like **pandas**, **numpy**, **matplotlib**, **seaborn**, and **scikit-learn** (sklearn).

---

7. How do you handle missing values in this dataset?

Answer:
The Boston Housing dataset is clean and does not have missing values. But if needed, missing values can be handled using methods like mean/median imputation or dropping missing rows.

---

8. What is the formula for simple linear regression?

Answer:
The formula is:

y=mx+cy = mx + cy=mx+c

where:

- yyy is the dependent variable
- xxx is the independent variable
- mmm is the slope of the line

- ccc is the intercept

---

9. How did you split the dataset?

Answer:
I split the dataset into **training** and **testing** sets, usually in a **80:20** or **70:30** ratio, using train_test_split from sklearn.model_selection.

---

10. Which model did you use to fit the data?

Answer:
I used **LinearRegression** from sklearn.linear_model.

---

11. What evaluation metrics did you use to measure your model's performance?

Answer:
I used metrics like:

- **Mean Squared Error (MSE)**
- **Root Mean Squared Error (RMSE)**
- **R^2 Score (Coefficient of Determination)**

---

12. What does the R^2 score tell us?

Answer:
The R^2 score measures how well the model's predictions match the actual data.

- An R^2 of **1** means perfect prediction.
- An R^2 closer to **0** means poor prediction.

---

13. What was your R^2 score on the test data?

Answer:
(*Answer according to your output. Typically, it is around 0.7–0.8 for simple models.*)

---

14. What are the assumptions of Linear Regression?

Answer:
The key assumptions are:

- Linearity
- Homoscedasticity (constant variance of errors)

- Independence of errors
- Normal distribution of errors
- No multicollinearity among independent variables

---

15. What is multicollinearity?

Answer:
Multicollinearity occurs when two or more independent variables are highly correlated, which can distort the importance of predictors in a regression model.

---

16. How can you detect multicollinearity?

Answer:
Multicollinearity can be detected using:

- Correlation Matrix
- Variance Inflation Factor (VIF)

---

17. Can you plot the actual vs predicted values?

Answer:
Yes, I can plot actual prices vs predicted prices using a scatter plot, which helps visualize how well the model is predicting.

---

18. Why do we standardize or normalize data before regression sometimes?

Answer:
Standardization is important when features have very different scales. It helps the model converge faster and improves interpretability of the coefficients.

---

19. What is the difference between simple and multiple linear regression?

Answer:

- **Simple Linear Regression**: One independent variable predicts the dependent variable.
- **Multiple Linear Regression**: Two or more independent variables predict the dependent variable.

---

20. What improvements can you suggest to your model?

Answer:

- Use **Polynomial Regression** for non-linear relationships

- Use **Feature Selection** or **Regularization** (like Ridge/Lasso)
- Try other models like Decision Trees, Random Forests for better accuracy

A5

Q1. What is Logistic Regression?

Answer:
Logistic Regression is a supervised machine learning algorithm used for **binary classification problems**. It predicts the probability that a given input belongs to a particular class using the **logistic (sigmoid) function**, which outputs values between 0 and 1.

Q2. What is the equation of Logistic Regression?

Answer:
The logistic regression model predicts probability using the sigmoid function:

Q2. What is the equation of Logistic Regression?

Answer:

The logistic regression model predicts probability using the sigmoid function:

$$p = \frac{1}{1 + e^{-(b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_n x_n)}}$$

where b_0 is the intercept and $b_1, b_2, \ldots, b_n$ are the coefficients.

Q3. Why do we use the Sigmoid function in Logistic Regression?

Answer:

We use the sigmoid function because it converts any real-valued number into a value between **0 and 1**, which can be interpreted as a **probability**.

Q3. Why do we use the Sigmoid function in Logistic Regression?

Answer:
We use the sigmoid function because it converts any real-valued number into a value between **0 and 1**, which can be interpreted as a **probability**.

Q4. What is a Confusion Matrix?

Answer:
A confusion matrix is a table used to evaluate the performance of a classification model. It shows the number of **True Positives (TP)**, **True Negatives (TN)**, **False Positives (FP)**, and **False Negatives (FN)**.

Q5. What is TP, FP, TN, FN?

Answer:

- **TP (True Positive):** Correctly predicted positive cases.
- **FP (False Positive):** Incorrectly predicted as positive.
- **TN (True Negative):** Correctly predicted negative cases.
- **FN (False Negative):** Incorrectly predicted as negative.

Q6. How is Accuracy calculated?

Answer:

Accuracy is the proportion of correct predictions:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

Q7. What is Precision?

Answer:

Precision measures the correctness of positive predictions:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Q8. What is Recall?

Answer:
Recall measures how many actual positives were correctly predicted:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Q9. What is the Error Rate?

Answer:
Error Rate is the proportion of wrong predictions:

$$\text{Error Rate} = \frac{FP + FN}{TP + FP + TN + FN} \quad \text{or} \quad 1 - \text{Accuracy}$$

Q10. Why is Logistic Regression better than Linear Regression for classification tasks?

Answer:
Linear Regression predicts continuous values, which can be outside the range [0,1]. Logistic Regression outputs probabilities between 0 and 1 and is specifically designed for **classification problems**.

Q11. How would you evaluate if your logistic regression model is good?

Answer:
By checking metrics like **Accuracy, Precision, Recall, F1 Score**, and analyzing the **Confusion Matrix**.

Q12. Why do we split data into training and testing sets?

Answer:
To **train** the model on one part of the data and **test** its performance on unseen data, ensuring it generalizes well to new inputs.

Q13. What is overfitting?

Answer:
Overfitting occurs when the model performs very well on the training data but poorly on new, unseen data because it has memorized the training set instead of learning general patterns.

Q14. What are some assumptions of Logistic Regression?

Answer:

- No multicollinearity among independent variables.
- Linearity between independent variables and log odds.

- Large sample size is preferred.

Q15. What is the difference between Precision and Recall?

Answer:

- **Precision** focuses on how many predicted positives were actually correct.
- **Recall** focuses on how many actual positives were captured by the model.

A6

1. What is the Naïve Bayes algorithm?

Answer:
Naïve Bayes is a **supervised machine learning algorithm** based on **Bayes' Theorem**.
It assumes that the features are **independent** (hence "naïve") and uses probabilities to predict the class of a given data point.

---

2. Why is it called "Naïve"?

Answer:
It is called **"naïve"** because it **assumes all features are independent** of each other, which is rarely true in real data, but the model still works well in practice.

---

3. What dataset did you use in this practical?

Answer:
I used the **Iris dataset**, which contains data of 150 flowers of 3 species: **Setosa, Versicolor, and Virginica**, based on 4 features:

- Sepal length
- Sepal width
- Petal length
- Petal width

---

4. Which type of Naïve Bayes classifier did you use?

Answer:
I used **Gaussian Naïve Bayes** because the features (sepal and petal measurements) are **continuous numerical values**, and GaussianNB assumes normal distribution.

---

5. What is a confusion matrix?

Answer:
A **confusion matrix** is a table that shows the performance of a classification model.
It compares the **actual labels** with the **predicted labels** and helps calculate metrics like **accuracy, precision, recall**, etc.

---

6. What are TP, FP, FN, and TN?

Term	Full Form	Meaning
TP	True Positive	Correctly predicted as positive
FP	False Positive	Incorrectly predicted as positive
FN	False Negative	Incorrectly predicted as negative
TN	True Negative	Correctly predicted as negative

---

7. How is Accuracy calculated?

Answer:

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

Accuracy measures how often the classifier was correct.

---

8. How is Error Rate calculated?

Answer:

Error Rate=1−Accuracy\text{Error Rate} = 1 - \text{Accuracy}Error Rate=1−Accuracy

It measures the proportion of wrong predictions.

---

9. How is Precision calculated?

Answer:

Precision=TPTP+FP\text{Precision} = \frac{TP}{TP + FP}Precision=TP+FPTP

Precision measures how many selected items are relevant.

---

10. How is Recall calculated?

Answer:

Recall=TPTP+FN\text{Recall} = \frac{TP}{TP + FN}Recall=TP+FNTP

Recall measures how many relevant items were selected.

---

11. Why did you split the dataset into training and testing?

Answer:
We split the data so that we can **train** the model on one part and **test** how well it performs on unseen data.
This helps in checking if the model can generalize to new data.

---

12. Which libraries did you use in Python?

Answer:
I used the following libraries:

- **pandas** (for data handling)
- **sklearn** (for Naïve Bayes and confusion matrix)
- **train_test_split** (to split data into training and testing sets)

---

13. What was your model's accuracy?

Answer:
(⭐ *You should answer based on your output. Example:* ⭐)
"My model achieved around **0.95** or **95% accuracy** on the test set."

---

14. What are some applications of Naïve Bayes?

Answer:

- Email Spam Detection
- Sentiment Analysis
- Document Categorization
- Medical Diagnosis

---

15. Why is Naïve Bayes good for small datasets?

Answer:
Because it is a **simple, fast** algorithm that works well even with **limited data** and does not require much computational power.

A7

Part 1: Document Preprocessing

Q1. What is tokenization?
A1. Tokenization is the process of splitting text into individual units, called tokens, which can be words, phrases, or symbols. It is the first step in text preprocessing.

Q2. What is POS tagging?
A2. POS (Part of Speech) tagging is the process of labeling each token (word) with its corresponding part of speech, like noun, verb, adjective, etc., based on its context in the sentence.

Q3. Why do we remove stop words?
A3. Stop words like "is", "the", "and", etc., are very common words that do not carry much meaningful information for analysis, so we remove them to focus on important words.

Q4. What is stemming?
A4. Stemming is the process of reducing a word to its base or root form by chopping off prefixes or suffixes. For example, "running", "runner" become "run".

Q5. What is lemmatization? How is it different from stemming?
A5. Lemmatization also reduces words to their base form (lemma), but it uses dictionary meaning and returns actual words, unlike stemming which can produce non-words.
Example:

- Stemming of "better" → "bett"
- Lemmatization of "better" → "good"

Q6. Which libraries are commonly used in Python for these preprocessing tasks?
A6. Common libraries are:

- nltk (Natural Language Toolkit)
- spacy
- re (for regular expressions)

Part 2: Term Frequency and Inverse Document Frequency (TF-IDF)

Q7. What is Term Frequency (TF)?
A7. Term Frequency measures how frequently a term occurs in a document. It is usually calculated as:

$$TF = \frac{\text{Number of times term appears in a document}}{\text{Total number of terms in the document}}$$

Q8. What is Inverse Document Frequency (IDF)?
A8. IDF measures how important a term is by reducing the weight of terms that occur very frequently across all documents. It is calculated as:

$$IDF = \log\left(\frac{\text{Total number of documents}}{\text{Number of documents containing the term}}\right)$$

Q9. What is TF-IDF and why is it important?
A9. TF-IDF is the product of TF and IDF. It highlights terms that are important in a document but not too common across all documents. It helps in improving text mining tasks like search and classification.

Q10. Which libraries/tools are used to compute TF-IDF in Python?
A10.

- TfidfVectorizer from sklearn.feature_extraction.text
- TfidfTransformer with CountVectorizer

Q11. What are some applications of TF-IDF?
A11. TF-IDF is used in:

- Information retrieval (like search engines)
- Document clustering
- Text classification
- Keyword extraction

Q12. Why is normalization often applied after TF-IDF?
A12. Normalization ensures that documents of different lengths are comparable by scaling the vectors to have unit norm.

A8

1. Which dataset are you using?

I am using the built-in **'titanic'** dataset provided by Seaborn. It contains information about passengers aboard the Titanic ship, such as their age, class, fare, survival status, etc.

---

2. What is the purpose of using df.head()?

df.head() shows the **first five rows** of the dataset. It helps to quickly preview the data and understand the columns and sample values.

---

3. What does df.info() tell you?

df.info() provides **a summary of the dataset**, showing:

- Number of entries (rows)
- Column names
- Data types (int, float, object, etc.)
- Number of non-null (non-missing) values in each column.

---

4. What is df.describe() used for?

df.describe() gives **statistical summary** for numerical columns, like:

- Mean, Standard Deviation
- Minimum and Maximum values
- 25%, 50%, 75% Percentiles (quartiles)

---

5. What is the shape of the Titanic dataset?

The shape is **(891, 15)**, meaning it has **891 rows** and **15 columns**.

---

6. Explain the use of sns.histplot(x='fare', data=df).

It plots a **histogram** showing the distribution of the ticket **fare** among passengers.
The x-axis is the fare amount, and the y-axis shows the number of passengers.

---

7. What does sns.displot(x='age', data=df, bins=70) do?

It plots a **distribution plot** (histogram) of passenger **ages** with **70 bins** for finer granularity, showing how ages are spread among passengers.

---

8. Explain sns.catplot(x='survived', data=df, kind='count', hue='pclass').

It creates a **count plot** showing how many passengers survived (1) or did not survive (0), split by **passenger class** (pclass) using different colors.

---

9. What is a Violin plot, and what does sns.violinplot(x='class', y='age', data=df) show?

A **violin plot** shows the **distribution and density** of age for each passenger class.
It combines a box plot and a density curve together for better visual understanding.

---

10. What is the use of a Strip plot?

A **strip plot** (sns.stripplot) shows **individual data points** of age against passenger class, slightly spread out to prevent overlap.

---

11. What is a Swarm plot?

A **swarm plot** (sns.swarmplot) is similar to a strip plot but **adjusts points** so they don't overlap, making each data point visible clearly.

---

12. What does sns.scatterplot(x='age', y='fare', data=df, hue='survived') show?

It plots a **scatter plot** between **age** and **fare**, with different colors based on **survival status** (0 = not survived, 1 = survived).

---

13. What is the use of sns.countplot(x='class', data=df) and ad.bar_label(i)?

countplot shows the **count of passengers** in each class (First, Second, Third).

ad.bar_label(i) adds **numbers** (labels) on top of the bars for better readability.

---

14. Why do we use warnings.filterwarnings('ignore')?

It is used to **suppress warning messages** from libraries, making the output cleaner during execution.

---

15. What does %matplotlib inline do?

%matplotlib inline is a **magic command** used in Jupyter Notebook to **display plots inside the notebook** itself.

A9

1. Which dataset are you using?

I am using the built-in **'titanic'** dataset provided by the Seaborn library.

---

2. What type of graph did you plot first?

I plotted a **box plot** to show the **distribution of age** with respect to **gender** and **survival status**.

---

3. What does a box plot represent?

A **box plot** shows the **median, quartiles, minimum, maximum, and outliers** in the data.
It summarizes the distribution of a numerical variable across different categories.

---

4. What is the role of hue='survived' in the box plot?

hue='survived' separates the data based on whether the passenger **survived (1)** or **did not survive (0)**, using different colors.

---

5. What inference can you make from the box plot?

- Females had a higher survival rate than males.
- Among males, **younger** passengers had a better survival rate.
- The age distribution for survivors is slightly **lower** than for non-survivors.

---

6. What does sns.barplot(x='sex', y='age', hue='survived') show?

It shows the **average age** of males and females, further divided based on **survival status** using different colors.

---

7. What is the purpose of a violin plot?

A **violin plot** (sns.violinplot) shows the **density distribution** of the data along with a **box plot** inside it.
It helps to understand both **spread** and **concentration** of values.

---

8. What does sns.stripplot(x='sex', y='age') do?

A **strip plot** plots **individual points** of passengers' ages for each gender.
It shows raw data points, helping us spot patterns and clustering.

---

9. What libraries are you using?

I am using **Seaborn** for plotting and **Matplotlib** for figure sizing and showing plots.

---

10. Why do we use plt.figure(figsize=(8,4))?

It sets the **size** of the plot window to **8 inches wide and 4 inches tall**, making it clearer and easier to read.

---

11. What are the types of variables involved in this analysis?

- **Sex** → Categorical variable

- **Survived** → Categorical variable
- **Age** → Numerical variable

---

12. What is an outlier in the box plot?

Outliers are **individual points** that lie **far outside** the typical range of the data.
In the box plot, they appear as **dots** beyond the whiskers.

---

13. How is violin plot different from box plot?

A **box plot** summarizes distribution with basic statistics (median, quartiles),
but a **violin plot** shows the **full probability distribution** and is more detailed visually.

🧠 **Extra Short Final Inference (if they ask):**

Conclusion:
Younger passengers, especially females, had a **better survival rate**.
Older male passengers had a **lower survival chance**.

A10

1. Which dataset did you use?

I used the **Iris flower dataset**, available from Seaborn or the UCI Machine Learning Repository.

---

2. What are the features (columns) in the Iris dataset?

- sepal_length (numeric)
- sepal_width (numeric)
- petal_length (numeric)
- petal_width (numeric)
- species (nominal / categorical)

---

3. What type of data is present in the Iris dataset?

- **Numeric Data:** sepal_length, sepal_width, petal_length, petal_width
- **Nominal (Categorical) Data:** species (Setosa, Versicolor, Virginica)

---

4. How did you create histograms for the features?

I used the DataFrame.hist() function from pandas to create histograms for each numeric feature to observe their distributions.

Example Code:

python

CopyEdit

iris.hist(figsize=(10,8))

plt.show()

---

5. What does a histogram show?

A histogram shows the **frequency distribution** of a feature — how often different values occur, grouped into ranges (bins).

---

6. How did you create boxplots for the features?

I used the seaborn.boxplot() function for each numeric feature to visualize **median, quartiles, spread**, and **outliers**.

Example Code:

python

CopyEdit

sns.boxplot(y=iris['sepal_length'])

---

7. What information does a boxplot provide?

A boxplot shows:

- **Median** (middle value)
- **Quartiles** (25th and 75th percentile)
- **Minimum and Maximum** values (excluding outliers)
- **Outliers** (individual points outside normal range)

---

8. Did you find any outliers in the Iris dataset?

- Yes, there are some outliers, especially in the **sepal width** feature.
- Other features like **sepal length, petal length, and petal width** had fewer or no major outliers.

---

9. How are the feature distributions in the Iris dataset?

- **Sepal Length:** Approximately normal distribution.
- **Sepal Width:** Slightly skewed with a few outliers.
- **Petal Length and Width:** Clustered into three groups (based on species).

---

10. Why do we check for outliers?

Outliers can **distort statistical analysis** and **affect machine learning models**.
Identifying them helps in **better data cleaning** and **model performance**.

---

11. What libraries did you use?

I used:

- **Pandas** for data handling
- **Seaborn** for visualization
- **Matplotlib** for plotting

---

12. What is the role of 'species' in the dataset?

species is a **categorical variable** that identifies the class of Iris flowers — **setosa**, **versicolor**, or **virginica**.

---

13. What is the purpose of plotting both histograms and boxplots?

- **Histograms** help understand the **overall distribution** of a feature.
- **Boxplots** help detect **spread**, **symmetry**, and **outliers** in the feature values.

Conclusion:
The Iris dataset shows clear differences in petal measurements among species.
Sepal width has some outliers.
Understanding distribution and outliers is important for preparing data for machine learning tasks.