Assignment 1

Summary of the Experiment

The experiment involves performing operations on the Iris dataset using Python. The dataset contains 150 samples of iris flowers categorized into three species (Setosa, Versicolor, and Virginica). The objective is to load and explore the dataset, perform data preprocessing, visualize the data, and use machine learning techniques for classification.

Steps in the experiment:

1. **Dataset Loading**: The Iris dataset is loaded from an open-source URL using `pandas`.
2. **Data Exploration**: Basic statistical analysis, data visualization, and checking for missing values.
3. **Data Visualization**: Use of various plots like histograms, box plots, and pair plots to analyze the dataset.
4. **Classification**: Applying machine learning algorithms (e.g., Decision Tree, KNN) for classification and evaluating model performance.

Possible Viva Questions and Answers

1. **What is the Iris dataset, and what are its key features?**
   The Iris dataset contains 150 samples of iris flowers, classified into three species: Setosa, Versicolor, and Virginica. It has four features: Sepal Length, Sepal Width, Petal Length, and Petal Width. These features are numerical, and the target variable is the species of the flower.

2. **How do you handle missing data in a dataset?**
   Missing data can be handled in several ways: by removing rows or columns with missing values, filling in the missing values with the mean or median (imputation), or using more advanced techniques like regression or machine learning models to predict missing values. In this experiment, we check for missing values using `isnull()`.

3. **What are the common methods for visualizing data in Python?**
   Common methods for data visualization in Python include using libraries like `matplotlib`, `seaborn`, *and* `plotly`. *We can create histograms, scatter plots, pair plots, and box plots to explore the distribution of data, detect outliers, and examine relationships between features.*

- **Matplotlib**:
  Matplotlib is a basic plotting library in Python used to create static, animated, and interactive 2D plots like line graphs, bar charts, histograms, and scatter plots.

- **Seaborn**:
  Seaborn is built on top of Matplotlib and is used for creating more attractive, informative, and complex statistical graphics easily, like heatmaps, violin plots, and pair plots.

Ajit Reddy

- **Plotly**:
  Plotly is a library used to create highly interactive, web-based graphs and dashboards, including 2D and 3D plots, and is often used for real-time data visualization.

4. **What is the purpose of normalizing or scaling the data before applying machine learning algorithms?**
   Normalizing or scaling the data ensures that all features contribute equally to the model. This is particularly important for algorithms like KNN, SVM, and logistic regression, where the distance between data points matters. It ensures that larger numerical values do not dominate the model's learning process.

5. **Explain the difference between supervised and unsupervised learning.**
   Supervised learning involves training a model on labeled data, where the output (target variable) is known. Examples include classification and regression tasks. Unsupervised learning, on the other hand, deals with unlabeled data where the goal is to find patterns or structures, such as clustering or dimensionality reduction.

6. **How do you evaluate the performance of a classification model?**
   Performance evaluation for classification models can be done using metrics like accuracy, precision, recall, F1 score, and confusion matrix. For example, the confusion matrix provides insights into true positives, false positives, true negatives, and false negatives, helping assess the model's effectiveness.

7. **What machine learning algorithms can be used for classification in this experiment?**
   Common classification algorithms for this experiment include Decision Tree, K-Nearest Neighbors (KNN), Logistic Regression, and Support Vector Machines (SVM). These algorithms learn patterns from the training data and classify new, unseen instances based on those patterns.

8. **What is the role of cross-validation in machine learning?**
   Cross-validation is used to assess the performance of a model by dividing the dataset into multiple subsets (folds) and training/testing the model on different combinations of these folds. It helps in avoiding overfitting and provides a more reliable estimate of model performance.

9. **Why is the confusion matrix important in classification tasks?**
   The confusion matrix is crucial because it shows how well a classification model performs. It details the number of correct and incorrect predictions, distinguishing between the different classes. This allows for more detailed analysis of performance, such as calculating precision, recall, and F1 score.

Assignment 2

Summary of the Experiment: Data Wrangling-II

In this experiment, a dataset named "StudentsPerformance" is created with 30 student performance records. The dataset includes features like `Math_Score`, `Reading_Score`, `Writing_Score`, and `Placement_Score`, along with a target variable, `Placement_Offer_Count`. The `Placement_Offer_Count` is categorized based on

the `Placement_Score` into 3 categories: 1 offer (if Placement_Score < 75), 2 offers (if Placement_Score between 75 and 85), and 3 offers (if Placement_Score > 85). Impurities (20%) are introduced into the dataset by modifying certain values outside the specified ranges and disrupting the relationship between `Placement_Score` and `Placement_Offer_Count`. The goal of this experiment is to apply various data wrangling techniques like handling missing values, detecting outliers, and transforming data to prepare it for analysis.

Possible Viva Questions and Answers

1. **What is the purpose of the dataset in this experiment, and what features does it include?**
    - The purpose of the dataset is to simulate student performance data and study the relationship between student scores and placement offers. It includes features like `Math_Score`, `Reading_Score`, `Writing_Score`, and `Placement_Score`, which are used to predict the number of placement offers (`Placement_Offer_Count`).
2. **How did you introduce impurities in the dataset, and why is it important?**
    - Impurities were added by modifying certain values to be outside the specified ranges and disrupting the relationship between the `Placement_Score` and the number of placement offers. This is important to simulate real-world data where imperfections and inconsistencies often occur, and handling such issues is a crucial part of data wrangling.
3. **What methods can be used to detect and handle missing values in a dataset?**
    - Missing values can be detected using functions like `isnull()` or `notnull()`. To handle missing values, we can either remove the rows/columns using `dropna()` or replace missing values using `fillna()` or `replace()` with appropriate values like the mean or median of the column.
4. **Explain how you can detect outliers in the dataset and how they can be handled.**
    - Outliers can be detected using visualizations like boxplots and scatterplots, or mathematically through methods like Z-score or Interquartile Range (IQR). Outliers can be handled by removing them, capping them based on percentile values, or replacing them with the median of the dataset.
5. **What is data transformation, and why is it important?**
    - Data transformation refers to the process of converting data into a format that is more suitable for analysis or modeling. It is important because it helps normalize skewed data, scales features to similar ranges, and can improve the performance of machine learning models. Techniques such as Z-score normalization or Min-Max scaling are commonly used.
6. **What is Z-score normalization, and when would you use it?**

- Z-score normalization is a technique where the data is transformed to have a mean of 0 and a standard deviation of 1. It is useful when the data follows a normal distribution, and it helps to eliminate differences in scale between variables, which can improve the performance of certain machine learning algorithms.

7. **How would you handle missing values in this experiment, and why would you choose those methods?**

   - In this experiment, missing values can be handled by using the `fillna()` method to replace missing values with the mean or median of the column, as this approach preserves the dataset's structure. Alternatively, rows with missing values could be dropped using `dropna()` if they are not significant to the analysis.

8. **What is the IQR method for detecting outliers, and how is it calculated?**

   - The Interquartile Range (IQR) method detects outliers by identifying values that fall below the first quartile (Q1) minus 1.5 times the IQR or above the third quartile (Q3) plus 1.5 times the IQR. Any data points outside this range are considered outliers.

9. **How do you visualize the distribution of data to detect outliers?**

   - Outliers can be visualized using boxplots, which show the spread of data and highlight any values that fall outside the typical range (the whiskers). Scatterplots can also be used to identify extreme values that deviate from the overall trend of the data.

10. **Why is data wrangling an essential step in the data analysis process?**

    - Data wrangling is crucial because it cleans the data by addressing issues like missing values, outliers, and inconsistencies, ensuring that the data is in a format suitable for analysis or machine learning. Without proper wrangling, inaccurate or biased results may be obtained.

Assignment 3

Title: Descriptive Statistics - Measures of Central Tendency and Variability

In this experiment, we performed descriptive statistical operations on open-source datasets.

- First, we grouped a numeric variable (like income) by a categorical variable (like age group) and calculated summary statistics (mean, median, minimum, maximum, standard deviation).

- Second, we analyzed the Iris dataset by extracting basic statistical details like percentiles, mean, and standard deviation for each Iris species.
  The experiment helped in understanding how to describe data using measures of central tendency (mean, median, mode) and measures of variability (standard deviation, range, variance).

Possible Viva Questions and Answers

1. What is Descriptive Statistics?
Descriptive statistics involves summarizing and organizing data to make it easily understandable. It uses measures like mean, median, mode, standard deviation, and range without making any predictions beyond the data.

2. What are Measures of Central Tendency?
Measures of central tendency describe the center or average of a data set. The three main measures are mean (average), median (middle value), and mode (most frequent value).

3. What is Standard Deviation?
Standard deviation measures the amount of variation or dispersion in a dataset. A low standard deviation indicates that data points are close to the mean; a high standard deviation shows they are spread out.

4. What is the difference between Mean and Median?
Mean is the average of all data points, while median is the middle value when data is sorted. Mean is affected by extreme values (outliers), but median is not.

5. What is Variance? How is it related to Standard Deviation?
Variance measures how far each data point is from the mean. Standard deviation is the square root of variance and provides dispersion in the same unit as the data.

6. What is Skewness in a dataset?
Skewness measures the asymmetry of the data distribution. Positive skewness means a longer right tail; negative skewness means a longer left tail.

7. Why do we use group-wise summary statistics?
Group-wise summary statistics help to compare how a numeric variable behaves across different categories, providing better insights into the relationship between variables.

8. What libraries can be used for statistical operations in Python?
Libraries like Pandas, NumPy, SciPy, Matplotlib, Seaborn, and Plotly are commonly used for statistical calculations and visualizations.

1. Pandas

Used for data manipulation and analysis. It provides data structures like **DataFrame** and **Series** for handling structured data.

2. NumPy

Used for numerical computing. It supports **multidimensional arrays**, **mathematical functions**, and **linear algebra** operations efficiently.

3. SciPy

Built on top of NumPy, it's used for **scientific computing**, such as **statistics**, **signal processing**, **integration**, and **optimization**.

4. Matplotlib

Used for creating **static, animated, and interactive visualizations** in Python. It is ideal for **line plots, bar charts, scatter plots**, etc.

5. Seaborn

Built on top of Matplotlib, it's used for **statistical data visualization**. It provides a high-level interface for **drawing attractive and informative plots** (e.g., heatmaps, boxplots).

6. Plotly

Used for **interactive plots and dashboards**. It supports **3D charts, animations**, and can be used in web-based data apps.

Assignment 4

In this experiment, we implement a **trigger in Oracle SQL** to **prevent unauthorized updates** on a specific column (`DeptID`) of the `Department` table. Triggers in Oracle are powerful tools used to **automate actions** or **enforce rules** when a DML event occurs (`INSERT`, `UPDATE`, `DELETE`).

The trigger is defined as a **BEFORE UPDATE** trigger, which means it executes **before** the actual update is performed on the table. The logic inside the trigger checks whether the new value of `DeptID` is different from the old one. If it is, the trigger **raises an application error** using `RAISE_APPLICATION_ERROR`, which stops the update and displays a custom error message.

This kind of mechanism is useful in cases where certain columns should remain **immutable**, such as **primary identifiers, audit fields**, or **critical configuration values**. It helps in maintaining **data consistency, integrity, and control** over modifications.

Key concepts demonstrated in this experiment:

- Creation of a trigger using `CREATE OR REPLACE TRIGGER`
- Use of `BEFORE UPDATE` timing
- Accessing old and new values using `:OLD` and `:NEW` qualifiers
- Throwing a custom error using `RAISE_APPLICATION_ERROR`
- Testing the trigger by trying to perform an update

❓ Viva Questions and Answers (4–5 lines each)

1. What is a trigger in Oracle SQL?
A trigger is a stored block of code that **automatically executes** in response to specific events like `INSERT`, `UPDATE`, or `DELETE`. Triggers help in maintaining **data integrity**, implementing

business rules, or **auditing** actions without modifying application code. Unlike procedures, triggers are **event-driven** and not called manually.

2. Why is the `BEFORE UPDATE` clause used in the experiment?
The `BEFORE UPDATE` clause ensures that the trigger fires **before any changes** are applied to the database. This allows us to **validate or restrict** the update based on conditions. If the condition fails, the update is **prevented** using `RAISE_APPLICATION_ERROR`, protecting critical data.

3. What is the role of `:OLD` and `:NEW` in a trigger?
In row-level triggers, `:OLD` holds the column value **before the event**, while `:NEW` holds the value **after the event**. In this experiment, they are used to **compare changes** in the `DeptID` field. If a change is detected, the trigger prevents it from being saved.

4. What does `RAISE_APPLICATION_ERROR` do in a trigger?
`RAISE_APPLICATION_ERROR` is used in PL/SQL to **raise a custom error** from a trigger or procedure. It takes an error number and message as arguments. In this experiment, it is used to throw an error when an update is attempted on `DeptID`, stopping the transaction with a meaningful message.

5. Can a trigger block updates to a specific column only? How?
Yes, triggers can block changes to specific columns by **comparing `:OLD.column` and `:NEW.column`**. If they differ, it means the column is being updated. Based on this condition, you can **cancel the operation** using a custom error. This allows **fine-grained control** over updates.

6. What would happen if someone tries to update `DeptID` after the trigger is active?
The trigger will detect the change in `DeptID` by comparing `:OLD` and `:NEW` values. If a difference is found, it will raise a custom error using `RAISE_APPLICATION_ERROR`, and the update operation will be **terminated automatically**, ensuring data protection.

7. What are different types of triggers in Oracle SQL?
Oracle supports:

- **BEFORE** and **AFTER** triggers (based on timing)
- **ROW-level** and **STATEMENT-level** triggers (based on scope)
- Triggers for **INSERT**, **UPDATE**, **DELETE**, and **INSTEAD OF** events
  Each type serves a specific purpose in controlling or auditing data changes.

8. Is it possible to create multiple triggers on a single table?
Yes, multiple triggers can be created on the same table, even for the same event type. However, their **execution order is not guaranteed**, so logic should be designed carefully. Oracle processes all applicable triggers for a given event, unless prevented by errors.

9. How do triggers help in data integrity and business rule enforcement?
Triggers automatically check and enforce rules **at the database level**, independent of application code. They prevent invalid transactions, restrict unauthorized changes, and ensure that **critical business rules** (like not changing a primary ID) are strictly followed.

10. What are the advantages and limitations of using triggers?
Advantages:

- Automatic enforcement of rules
- Centralized validation logic
- Useful for logging and auditing

Limitations:

- Hard to debug
- Can affect performance
- Overuse can lead to complex, unpredictable behavior

Assignment 5

Detailed Summary:

Title:

Logistic Regression & Confusion Matrix Analysis on Social_Network_Ads Dataset

Objective:

To understand and implement **logistic regression** for binary classification, and evaluate the model using a **confusion matrix** with metrics like Accuracy, Precision, Recall, etc.

Theory:

1. **Logistic Regression:**
    - A statistical method used for **binary classification** problems.
    - Predicts the probability of a **binary outcome** using a **logit function**.
    - Commonly used for applications like spam detection, disease prediction, etc.
    - Uses the **sigmoid function** to map predicted values to a range between 0 and 1.
2. **Sigmoid Function:**

- Formula: `1 / (1 + e^-x)`
- Output lies between 0 and 1, creating an **S-shaped curve**.
- If output > 0.5, classify as 1 (Yes); else 0 (No).
- Example: 0.75 implies a 75% chance of a condition being true.

3. **Types of Logistic Regression:**
   - **Binary Logistic Regression** – Two categories (e.g., Yes/No).
   - **Multinomial Logistic Regression** – Three or more **nominal** classes.
   - **Ordinal Logistic Regression** – Three or more **ordered** categories.
4. **Confusion Matrix & Metrics:**
   - Helps evaluate classification performance.
   - Key terms: TP, TN, FP, FN.
   - From these we derive:
     - **Accuracy** = (TP + TN) / Total
     - **Error Rate** = (FP + FN) / Total
     - **Precision** = TP / (TP + FP)
     - **Recall** = TP / (TP + FN)

Conclusion:

Successfully implemented logistic regression for classification on the **Social_Network_Ads** dataset and computed **performance metrics** using a confusion matrix. Also visualized results using a **heatmap**.

Possible Questions with Answers (4-5 lines each):

Q1. What is logistic regression and where is it used?
Ans: Logistic regression is a statistical method used for **binary classification** problems. It predicts the probability of a **categorical dependent variable**, typically having two outcomes. It is widely used in spam detection, disease prediction, and click-through rate estimation.

Q2. Explain the sigmoid function in logistic regression.
Ans: The sigmoid function maps any real-valued number into a value between **0 and 1**, forming an **S-shaped curve**. It is used to calculate the probability of the outcome. If the output is greater than 0.5, the result is classified as 1; otherwise, it is classified as 0.

Q3. Differentiate between linear and logistic regression.
Ans: Linear regression predicts a **continuous output** using OLS, while logistic regression predicts a **categorical output** using MLE. Linear regression is used for forecasting values like stock prices, whereas logistic regression is used for binary outcomes like Yes/No.

Q4. What are the types of logistic regression?
Ans: Logistic regression can be of three types:

1. **Binary** – for two outcomes (e.g., Yes/No).
2. **Multinomial** – for multiple **nominal** categories.
3. **Ordinal** – for multiple **ordered** categories like product ratings.

Q5. What is a confusion matrix and how is it useful?
Ans: A confusion matrix is a **performance measurement tool** for classification models. It shows **actual vs predicted** values and helps derive metrics like **Accuracy, Precision, Recall**, etc. It includes TP, TN, FP, and FN to evaluate prediction correctness.

Q6. Define Accuracy, Precision, and Recall with formulae.
Ans:

- **Accuracy** = (TP + TN) / Total
- **Precision** = TP / (TP + FP) → confidence of correct positive prediction
- **Recall** = TP / (TP + FN) → ability to find all positives
  These help understand the model's reliability.

Q7. What is the significance of TP, FP, TN, FN in classification?
Ans:

- **TP (True Positive):** Correctly predicted positive cases.
- **TN (True Negative):** Correctly predicted negative cases.
- **FP (False Positive):** Incorrectly predicted positive.
- **FN (False Negative):** Missed positive cases.
  These form the basis for evaluating classification performance.

Q8. What kind of dataset is used in this assignment and why?
Ans: The dataset used is **Social_Network_Ads.csv**, which is suitable for **binary classification**. It contains user data such as age, salary, and whether they clicked on an ad or not, making it ideal for logistic regression-based prediction tasks.

Q9. How can we visualize the performance of a logistic regression model?
Ans: We can use a **heatmap of the confusion matrix** to visually inspect correct vs incorrect

classifications. It helps highlight areas where the model is making errors and aids in understanding model behavior quickly.

Assignment 6

Summary:

In this assignment, the Naïve Bayes Classification algorithm is applied to the `iris.csv` dataset using Python. The goal is to train the model, predict the outcomes, and evaluate the performance using metrics like accuracy, precision, recall, and the confusion matrix. The Gaussian Naïve Bayes model from scikit-learn is used, assuming feature independence. The evaluation is done using a heatmap to visualize the confusion matrix. This helps understand how Naïve Bayes can classify data based on probabilistic principles.

Viva Questions with Answers:

1. What is the Naïve Bayes algorithm and how does it work?
Naïve Bayes is a classification algorithm based on Bayes' Theorem with an assumption that all features are independent. It calculates the probability of each class given the input features and selects the class with the highest probability as the prediction.

2. What kind of data is suitable for Naïve Bayes?
Naïve Bayes works best with categorical data but can also handle continuous data using Gaussian distribution. It is effective for text classification, spam detection, and simple datasets like the Iris dataset.

3. What does the confusion matrix tell us?
A confusion matrix shows the count of actual vs predicted values, giving True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). It helps calculate accuracy, precision, recall, and error rate of a model.

4. Why do we assume feature independence in Naïve Bayes?
This assumption simplifies computation. It allows the model to multiply individual feature probabilities instead of calculating joint probabilities, which can be complex and computationally expensive.

5. What is GaussianNB and why was it used in this assignment?
`GaussianNB` is a Naïve Bayes classifier for continuous features which assumes they follow a Gaussian (normal) distribution. It was used here since the Iris dataset contains continuous numerical features.

6. How do you calculate accuracy and precision using Python?
Using `sklearn.metrics`, we use `accuracy_score()` for accuracy and `precision_score()` for precision. These functions compare predicted labels with actual test labels and return evaluation scores.

7. What preprocessing steps are necessary before applying Naïve Bayes?
Preprocessing includes handling missing values, converting categorical variables to numerical form, splitting the dataset into training and testing sets, and scaling if required.

Assignment 7

Summary:

In this assignment, basic **text preprocessing** techniques such as **tokenization, POS tagging, stop word removal, stemming, and lemmatization** were applied on a sample document using Python's NLTK library. Then, the **TF-IDF algorithm** was used to compute numerical values representing the importance of words in documents. This helped convert textual data into vectors for further analysis. The assignment demonstrates how text data can be prepared and represented for machine learning tasks using **text analytics**.

Viva Questions & Answers:

1. **Q: What is tokenization?**
   A: Tokenization is the process of splitting a text into smaller units like words or sentences. It helps in breaking the raw text into individual tokens for further processing.

2. **Q: Why do we remove stop words?**
   A: Stop words like "is", "the", "and" add little meaning and are often removed to reduce noise and improve the efficiency of text analysis.

3. **Q: What is the difference between stemming and lemmatization?**
   A: Stemming cuts off prefixes/suffixes to get the root form, which may not be a real word. Lemmatization uses a dictionary to find the actual base form of a word based on context.

4. **Q: What is POS tagging?**
   A: Part-of-Speech tagging assigns grammatical categories (like noun, verb) to words in a sentence, which helps understand sentence structure and meaning.

5. **Q: What is TF-IDF?**
   A: TF-IDF (Term Frequency-Inverse Document Frequency) calculates how important a word is in a document relative to a collection of documents. It reduces the weight of commonly occurring words and highlights significant ones.

6. **Q: What is the disadvantage of TF-IDF?**
   A: TF-IDF does not consider word meaning or synonyms, and it may become computationally expensive with a large vocabulary.

7. **Q: How is Bag of Words different from TF-IDF?**
   A: Bag of Words counts word frequency without considering word importance or meaning,

whereas TF-IDF adjusts the frequency based on how common the word is across all documents.

Assignment 8

Summary:

In this assignment, basic **text preprocessing** techniques such as **tokenization, POS tagging, stop word removal, stemming, and lemmatization** were applied on a sample document using Python's NLTK library. Then, the **TF-IDF algorithm** was used to compute numerical values representing the importance of words in documents. This helped convert textual data into vectors for further analysis. The assignment demonstrates how text data can be prepared and represented for machine learning tasks using **text analytics**.

Viva Questions & Answers:

1. **Q: What is tokenization?**
   A: Tokenization is the process of splitting a text into smaller units like words or sentences. It helps in breaking the raw text into individual tokens for further processing.

2. **Q: Why do we remove stop words?**
   A: Stop words like "is", "the", "and" add little meaning and are often removed to reduce noise and improve the efficiency of text analysis.

3. **Q: What is the difference between stemming and lemmatization?**
   A: Stemming cuts off prefixes/suffixes to get the root form, which may not be a real word. Lemmatization uses a dictionary to find the actual base form of a word based on context.

4. **Q: What is POS tagging?**
   A: Part-of-Speech tagging assigns grammatical categories (like noun, verb) to words in a sentence, which helps understand sentence structure and meaning.

5. **Q: What is TF-IDF?**
   A: TF-IDF (Term Frequency-Inverse Document Frequency) calculates how important a word is in a document relative to a collection of documents. It reduces the weight of commonly occurring words and highlights significant ones.

6. **Q: What is the disadvantage of TF-IDF?**
   A: TF-IDF does not consider word meaning or synonyms, and it may become computationally expensive with a large vocabulary.

7. **Q: How is Bag of Words different from TF-IDF?**
   A: Bag of Words counts word frequency without considering word importance or meaning, whereas TF-IDF adjusts the frequency based on how common the word is across all documents.

Assignment 9

Summary:

In this assignment, a box plot is created using the inbuilt Titanic dataset to visualize the distribution of passenger ages based on gender (`sex`) and survival status (`survived`). Using Seaborn's `boxplot()` function, the data is grouped and compared. The plot helps identify age patterns among males and females and shows survival trends based on age groups. This enhances understanding of how demographic factors relate to survival.

Possible Viva Questions & Answers:

Q1. What does a box plot show in data visualization?
A box plot shows the distribution of numerical data through quartiles. It highlights the median, interquartile range, and potential outliers in the dataset.

Q2. Why did we use the 'sex', 'age', and 'survived' columns in the Titanic dataset?
We used these columns to analyze how age distribution varies by gender and how it may influence survival chances during the Titanic disaster.

Q3. What is the use of the `hue` parameter in Seaborn's boxplot?
The `hue` parameter adds a second categorical grouping. In this case, it differentiates passengers based on whether they survived or not within each gender.

Q4. What can be inferred from the box plot generated from this Titanic data?
The box plot may show that younger females had a higher survival rate, and male passengers had more variation in age distribution but lower survival chances.

Q5. How is a violin plot different from a box plot?
While both show distribution, a violin plot includes a KDE (kernel density estimate) to display the full distribution shape, giving more visual information than a box plot.

Assignment 10

Summary:

In this assignment, the Iris flower dataset (or a similar dataset) is used for data visualization. The features are first identified and classified by type (numerical/nominal). Histograms are plotted for each feature to understand distributions, and box plots are created to visualize data spread and identify outliers. Tools like Seaborn and Matplotlib are used. The visualization helps in understanding data patterns and detecting anomalies.

Viva Questions and Answers:

Q1. What is the Iris dataset and what are its features?
The Iris dataset contains data about three species of Iris flowers: Setosa, Versicolor, and Virginica. It includes four numeric features: Sepal Length, Sepal Width, Petal Length, and Petal Width, and one nominal feature: Species.

Q2. What is the purpose of using a histogram in data visualization?
A histogram shows the frequency distribution of a numeric variable. It helps visualize how data points are spread across different intervals and is useful to detect skewness, modality, or spread.

Q3. What is a box plot and what does it indicate?
A box plot displays the distribution of data based on five key values: minimum, Q1, median, Q3, and maximum. It also helps in identifying outliers using whiskers and gives a summary of data spread.

Q4. How do you detect outliers using a box plot?
Outliers are detected as points outside the whiskers in a box plot. Any data value beyond 1.5 * IQR (interquartile range) from Q1 or Q3 is considered an outlier and usually marked as individual points.

Q5. What libraries are required for this assignment?
The assignment uses Python's **Seaborn** and **Matplotlib** libraries for creating histograms and box plots. Pandas is used to load and manipulate the dataset.

Assignment 11

Summary

This assignment involves creating a simple Word Count application in Java using the Hadoop MapReduce framework in a local-standalone setup. It introduces Hadoop's architecture including HDFS, YARN, MapReduce, and Hadoop Common. The program reads input text, splits it, maps each word to a key-value pair, and reduces them to count occurrences of each word. The assignment demonstrates the fundamental concept of distributed processing using Hadoop.

Possible Viva Questions and Answers

1. What is Hadoop and why is it used?
Hadoop is an open-source framework designed for distributed storage and processing of big data. It allows clustering multiple systems to work together for fast, fault-tolerant, and scalable data analysis across large datasets.

2. What are the core components of Hadoop?
The four main components are:

- HDFS (Hadoop Distributed File System)
- YARN (Yet Another Resource Negotiator)
- MapReduce (data processing model)
- Hadoop Common (supporting Java libraries and utilities)

3. Explain the MapReduce programming model.
MapReduce works in two phases:

- *Map Phase:* Converts input into key-value pairs (e.g., word → 1).
- *Reduce Phase:* Aggregates values by key (e.g., word → count).
  It enables parallel processing across multiple nodes.

4. How does the Word Count program work in Hadoop?
The input text is split line by line. The mapper emits each word with a count of 1. The reducer sums up the counts for each unique word, giving total occurrences.

5. What are key-value pairs in MapReduce?
MapReduce processes data as `<key, value>` pairs. The key helps group data, and the value carries the associated information. For Word Count, it's like `<word, 1>` in the Map phase and `<word, total count>` in the Reduce phase.

6. What is the role of Writable and WritableComparable interfaces?
These interfaces are required for Hadoop to serialize and sort data. Writable handles data serialization, and WritableComparable allows sorting of keys in the MapReduce process.

7. What are the steps in the MapReduce workflow?
The five steps are:

1. Splitting
2. Mapping
3. Intermediate Splitting
4. Reducing
5. Combining final output

8. How is Hadoop different from traditional databases?
Traditional databases are optimized for transactional operations on limited data sizes, while Hadoop is designed for large-scale, distributed data processing using a cluster of commodity hardware.

Assignment 12

Summary (4-5 lines):

This assignment focuses on designing a distributed application using MapReduce to process a system log file. Hadoop's HDFS is used for data storage and processing, with a setup involving NameNode, DataNode, Resource Manager, and Job History Server. Java-based Mapper and Reducer classes are compiled and packaged into a JAR to run the job. The log data is uploaded to HDFS and processed using the MapReduce framework to produce meaningful output.

Possible Viva Questions & Answers:

1. What is the role of HDFS in this assignment?
HDFS stores the large input log file in a distributed manner across the cluster. It enables parallel access to the data by splitting it into blocks stored on different DataNodes, which is essential for MapReduce processing.

2. What is MapReduce and how is it used here?
MapReduce is a programming model used for processing large data sets. In this assignment, the `Mapper` processes log lines to generate key-value pairs and the `Reducer` aggregates them to produce the final result.

3. What are NameNode and DataNode in Hadoop?
The NameNode is the master that manages the file system and metadata, while DataNodes store actual data blocks. They work together to handle and retrieve data in HDFS efficiently.

4. Why is a JAR file created in this project?
The JAR file packages the compiled `Mapper`, `Reducer`, and `Driver` Java classes. It allows Hadoop to execute the MapReduce job by referencing the entry point defined in the `Manifest.txt`.

5. How do you run a MapReduce job in this setup?
After uploading the input file to HDFS, the job is executed using the `hadoop jar` command, specifying the JAR file, input path, and output path. The results are then retrieved from the output directory in HDFS.

Assignment 13

Summary:

This assignment involves reading a weather dataset (like `sample_weather.txt`) using Hadoop's MapReduce framework to calculate the average **temperature**, **dew point**, and **wind speed**. It helps students understand how to apply **data analytics** and **parallel processing** concepts using Hadoop. Hadoop modules like **HDFS**, **MapReduce**, and **YARN** are used to process large volumes of weather data efficiently. The weather dataset typically includes parameters such as temperature, humidity, wind speed, and dew point, and is useful for forecasting and environmental analysis.

Viva Questions with Answers (4–5 lines each):

1. What is the objective of this assignment?
The main objective is to read weather data from a text file and calculate the average temperature, dew point, and wind speed using Hadoop MapReduce. This helps understand distributed data processing.

2. What is Hadoop and why is it used here?
Hadoop is an open-source framework that enables processing of large datasets using distributed

computing. It is used here to handle and analyze the large weather dataset efficiently using parallel tasks.

3. What are the main modules of Hadoop?
Hadoop has four main modules:

1. HDFS (Storage),
2. YARN (Resource Management),
3. MapReduce (Data Processing), and
4. Hadoop Common (Libraries and utilities).

4. What is MapReduce and how does it work?
MapReduce is a programming model in Hadoop that processes data in parallel. The **Map** function converts input into key-value pairs, and the **Reduce** function aggregates these pairs to produce the final output.

5. What type of data does a weather dataset contain?
A weather dataset typically contains values like temperature, dew point, humidity, wind speed, pressure, and visibility. These can be used to derive meaningful trends and perform analysis.

6. What is the role of data cleaning in analytics?
Data cleaning removes duplicate, incomplete, or incorrect data entries. It ensures accuracy and quality in the dataset before analysis, especially in large-scale systems like Hadoop.

7. What is the difference between real-time and historical weather data?
Real-time data is updated live and used for current forecasts. Historical data spans over years or decades and is used for long-term trend analysis and climate modeling.

8. Why is weather data analysis important?
Weather data helps in agriculture, transportation, disaster management, and infrastructure planning. It provides insights for decision-making based on environmental conditions.