

Viva Exam Q&A



Gropu B.pdf
PDF



Lab Manual SL3.pdf
PDF

You are a professional Viva examiner. For each of the two provided manuals, generate 10 distinct question-and-answer sets for each experiment suitable for a viva examination. Ensure the questions are comprehensive and test understanding of the experimental procedures and their underlying principles. Provide clear and concise answers for each question.

Below is a comprehensive set of viva question-answer pairs drawn from the content of the two provided manuals. The questions have been designed to probe both fundamental concepts and the practical steps of the experiments. In what follows, two separate sections are presented: one for “Group B” (which includes an experiment using Impala and another that uses Scala with Apache Spark) and one for “Lab Manual SL3” (which comprises several experiments covering data wrangling, descriptive statistics, regression – both linear and logistic). Each experiment is followed by 10 distinct Q&A sets.

Group B Manual

Experiment 1: Impala – Creating Databases, Tables, Inserting Data and Querying

1. **Question:** What is Impala and how does it differ from traditional Hive queries? **Answer:** Impala is an open-source SQL query engine designed for Hadoop that executes queries using an in-memory execution engine rather than MapReduce. This allows it to offer high performance and low latency compared to the more disk-oriented Hive queries.
2. **Question:** Explain the key advantages of using Impala in a Hadoop cluster. **Answer:** Impala’s advantages include fast query execution due to in-memory processing, low-latency responses, and seamless integration with the Hive Metastore for sharing

metadata. It is especially well-suited for medium-sized datasets where real-time query performance is expected.

3. **Question:** What are the basic steps to create a database in Impala? **Answer:** To create a database in Impala, you use the SQL command:

```
CREATE DATABASE my_database;
```

This command initializes a new database that can then be used for table creation and data analysis.

4. **Question:** How do you create a table in Impala, and what are some of the common data types used? **Answer:** You create a table using the `CREATE TABLE` statement. For example:

```
CREATE TABLE my_table (id INT, name STRING, age INT);
```

Common data types include INT, STRING, BIGINT, DECIMAL, BOOLEAN, DOUBLE, and FLOAT.

5. **Question:** Describe the process and command for inserting data into a table in Impala.

Answer: Data is inserted using the `INSERT INTO` statement. For example:

```
INSERT INTO my_table VALUES (1, 'John', 25), (2, 'Jane', 30), (3, 'Bob', 40);
```

This inserts three rows into the table, populating the columns with respective values.

6. **Question:** What is the role of the metadata store in Impala? **Answer:** The metadata store in Impala holds information regarding the databases and tables, including schema details and data location. It allows Impala to integrate seamlessly with Hive and speeds up the processing by caching metadata.

7. **Question:** How does Impala achieve faster query execution as compared to other tools using MapReduce? **Answer:** Impala executes queries in parallel across multiple nodes using its in-memory engine, minimizing disk I/O and eliminating the overhead associated with the MapReduce paradigm, which generally involves writing intermediate results to disk.

8. **Question:** In what scenarios would you prefer to use Impala? **Answer:** Impala is preferred when you require near real-time query performance, particularly on medium-sized datasets where quick SQL analytics are necessary. Its speed and low-latency response make it ideal for interactive analysis.

9. **Question:** Can Impala work with large-scale data on Hadoop? Explain any limitations.

Answer: While Impala is optimized for speed with moderate datasets and real-time queries, its in-memory processing model might become limiting for extremely large

datasets. In such cases, the system memory could become a bottleneck compared to batch-oriented systems.

10. **Question:** How would you verify that data has been correctly inserted into a table using Impala? **Answer:** You would run a simple SQL query such as `SELECT * FROM my_table;` to retrieve and inspect all rows of data, ensuring that the records appear as expected.
-

Experiment 2: Scala Using Apache Spark – Writing a Simple Program

1. **Question:** What are the two main paradigms that Scala supports and why is this beneficial? **Answer:** Scala supports both object-oriented and functional programming paradigms. This allows developers to write clear, concise, and expressive code, benefiting from the robustness of OO design as well as the elegant expressiveness and predictability of functional techniques.
2. **Question:** How does Scala's type system contribute to code safety? **Answer:** Scala is statically typed, which means that type-checking occurs at compile time. This helps detect errors early, ensures code reliability, and facilitates safe interactions especially when integrating with Java.
3. **Question:** Describe how Scala integrates with the Apache Spark framework. **Answer:** Scala is the native language for Apache Spark, making it convenient to write Spark applications. Spark's API in Scala allows the developer to write concise code that is compiled to JVM byte-code, ensuring high performance and smooth integration within Spark's cluster computing framework.
4. **Question:** What is the role of the "main" method in a Scala program and how is it typically defined? **Answer:** The `main` method is the entry point of a Scala application. It is defined inside an object (e.g., `object HelloWorld`) as follows:

```
scala

def main(args: Array[String]) {
  println("Hello, World!")
}
```

This method is executed when the program runs.

5. **Question:** Explain what implicit classes in Scala are and provide an example of their use. **Answer:** Implicit classes in Scala allow developers to add new methods to existing classes without modifying their source code. This enables a form of extension or enrichment of classes. For example, by defining an implicit class that adds a `toLowerCase` method to a `String`, you can call that method on any string instance.
6. **Question:** How do you compile and run a Scala program on a Unix-based system? **Answer:** To compile a Scala program, use the `scalac` compiler command (e.g., `scalac HelloWorld.scala`), which generates class files. Then run the program using the command `scala HelloWorld`.
7. **Question:** What makes Scala suitable for concurrent processing? **Answer:** Scala offers thread-based executors, support for functional programming constructs (which help avoid mutable state), and libraries such as Akka for actor-based concurrency. These features simplify the development of safe and scalable concurrent applications.
8. **Question:** In the context of Apache Spark, what is the significance of writing code in Scala? **Answer:** Writing Spark applications in Scala leverages Spark's native API, which is more concise and often more performant than alternatives. Scala's functional features also help in defining transformations and actions over resilient distributed datasets (RDDs) or DataFrames.
9. **Question:** What are some of the benefits of Scala's interoperability with Java? **Answer:** Scala's seamless interoperability with Java allows developers to use existing Java libraries and frameworks, share code across projects, and transition gradually from Java to Scala without a complete rewrite of the codebase.
10. **Question:** How does the concept of "immutability" in Scala contribute to writing robust Spark applications? **Answer:** Immutability helps prevent side effects and race conditions in distributed applications. When data structures are immutable, concurrent tasks can operate on them without risk of unintended modifications, thereby increasing reliability and easing debugging in Spark applications.
-

Lab Manual SL3

The following questions target the experiments described in Lab Manual SL3. Each experiment below includes 10 tailored Q&A sets.

Assignment 1: Data Wrangling I – Preprocessing, Formatting, Normalization, and Handling Categorical Variables

- Question:** What is the purpose of importing libraries such as NumPy, Pandas, and Scikit-learn in data wrangling? **Answer:** These libraries provide efficient tools for numerical computation (NumPy), data manipulation and preprocessing (Pandas), and machine learning tasks (Scikit-learn), making them essential for handling, analyzing, and transforming datasets.
- Question:** How can missing values in a dataset be detected using Pandas? **Answer:** Missing values can be detected using methods like `isnull()`, `notnull()`, and functions like `sum()` applied on the DataFrame. For example, `df.isnull().sum()` shows the count of missing values per column.
- Question:** Describe the steps you would take to load a CSV file into a Pandas DataFrame. **Answer:** To load a CSV file, you would use the command `pd.read_csv('filename.csv')`. You can also specify parameters such as header presence and column names if not included in the file.
- Question:** What is the importance of checking the data types of variables during the preprocessing phase? **Answer:** Ensuring that variables have the correct data type (e.g., numeric, categorical) is crucial for proper data analysis and transformation. Incorrect types can lead to errors during computations or while applying functions specific to certain data types.
- Question:** Explain the concept of label encoding and how it is applied in data wrangling. **Answer:** Label encoding converts categorical variables, which are non-numeric, into a numerical form that can be processed by machine learning algorithms. This is often done using the `LabelEncoder` from Scikit-learn.
- Question:** What role does data normalization play in preprocessing data, and which common technique is mentioned? **Answer:** Data normalization scales numerical features to a standard range (commonly 0 to 1) to ensure that each feature contributes equally to the analysis. A common technique mentioned is Min-Max Scaling.
- Question:** How can you get an initial statistical summary of a DataFrame in Pandas? **Answer:** The `describe()` method of a DataFrame provides summary statistics such as mean, median, standard deviation, minimum, and maximum for numeric columns.
- Question:** What is the difference between data formatting and data normalization? **Answer:** Data formatting involves converting variables into the correct types and ensuring consistency (e.g., dates, strings, numbers), while normalization involves scaling

numeric data to a common range for improved comparability and performance in analysis.

9. **Question:** Describe how you would convert a non-numeric categorical feature into numeric values in Python. **Answer:** You can convert a categorical feature into numeric values using label encoding (via Scikit-learn's `LabelEncoder`) or one-hot encoding (using Pandas' `get_dummies()`), depending on the context and modeling requirements.
 10. **Question:** Why is it critical to examine the dimensions and structure of a DataFrame during preprocessing? **Answer:** Examining the dimensions (using `shape`) and structure (using `dtypes` and `columns`) helps validate that the data has been loaded correctly, ensures proper alignment of variables, and identifies any discrepancies or anomalies in the dataset.
-

Assignment 2: Data Wrangling II – Handling Missing Data, Outliers and Data Transformations

1. **Question:** How do you identify missing values in a dataset using Pandas? **Answer:** Missing values can be identified using `isnull()` or `isna()` functions. These functions create boolean masks that can be summed across columns or rows to count the missing values.
2. **Question:** What are common techniques to handle missing values in a dataset? **Answer:** Common techniques include dropping missing values using `dropna()`, filling them with a constant or statistical measure like mean or median using `fillna()`, or using interpolation methods.
3. **Question:** How can outliers affect your data analysis, and what is one method to detect them? **Answer:** Outliers can skew statistical measures like the mean and standard deviation. One common detection method is the use of boxplots, which visually highlight data points that lie beyond the whiskers.
4. **Question:** What is the Interquartile Range (IQR), and how is it used in outlier treatment? **Answer:** The IQR is the range between the first (25th percentile) and third quartile (75th percentile) values. Values falling outside a typical range (often $1.5 \times \text{IQR}$ below Q1 or above Q3) can be considered outliers and may be trimmed or capped.

5. **Question:** Explain the concept and steps involved in data transformation. **Answer:** Data transformation converts variables into forms that are more suitable for analysis. This may include rescaling (e.g., normalization), logarithmic transformation to reduce skewness, or converting a date into a duration.
 6. **Question:** Describe an algorithm to display the statistics of null values present in a DataFrame. **Answer:** One algorithm is:
 - Import Pandas.
 - Load the dataset into a DataFrame.
 - Use `df.isnull().sum()` to count nulls per column.
 - Optionally, sum these values to get a total count of missing entries.
 7. **Question:** What is the purpose of replacing an outlier value with the mean of that variable? **Answer:** Replacing outliers with the variable's mean can help reduce distortion in statistical calculations without entirely removing the record, thus preserving dataset size while mitigating the impact of extreme values.
 8. **Question:** How would you programmatically replace outliers using Pandas? **Answer:** You can calculate the mean for the variable and then use conditional indexing to replace values deemed as outliers. For example, using `df.loc[df['column'] > threshold, 'column'] = df['column'].mean()`.
 9. **Question:** What are some challenges you might face when dealing with missing data? **Answer:** Challenges include deciding whether to impute or delete missing values, avoiding bias in the dataset, and ensuring that imputation methods do not distort the underlying data distribution.
 10. **Question:** Why is it important to document your approach when handling outliers and missing values? **Answer:** Documentation ensures reproducibility and clarity. It explains the rationale behind the chosen techniques, helps in assessing the impact of preprocessing on results, and supports transparency in data analysis.
-

Assignment 3: Descriptive Statistics – Measures of Central Tendency and Variability

1. **Question:** Define mean, median, and mode in the context of descriptive statistics. **Answer:** The mean is the arithmetic average, the median is the middle value when data

are sorted, and the mode is the most frequently occurring value in a dataset.

2. **Question:** How would you compute the standard deviation of a dataset by hand?

Answer: To compute standard deviation, calculate the mean, then the deviations from the mean for each data point, square each deviation, sum them, divide by the number of observations minus one (for a sample), and finally take the square root of that quotient.

3. **Question:** Why is the median preferred over the mean in the presence of outliers?

Answer: The median is more robust to outliers because it represents the middle value of a sorted dataset and is not skewed by extremely high or low values, unlike the mean.

4. **Question:** Explain how the `describe()` method in Pandas is useful for statistical analysis.

Answer: The `describe()` method automatically computes key summary statistics—including count, mean, standard deviation, minimum, quartiles, and maximum—which provide a quick overview of the distribution and spread of numeric data.

5. **Question:** How would you calculate and interpret the range of a dataset? **Answer:** The range is calculated by subtracting the minimum value from the maximum value. It indicates the spread or overall dispersion in the dataset.

6. **Question:** In a grouped analysis, how might you present summary statistics for different segments? **Answer:** You can use the Pandas `groupby()` method to segment the data based on a categorical variable and then apply `describe()` or aggregate functions (like mean, median) to summarize each group's statistics.

7. **Question:** What are the potential limitations of using mean as a measure of central tendency? **Answer:** The mean is sensitive to extreme values (outliers) which can distort the central value, making it less representative of the data's typical value.

8. **Question:** Describe one situation where the mode might be a more informative measure than the mean or median. **Answer:** In a dataset with categorical or non-numeric variables, or where data values repeat, the mode can be more informative by indicating the most common category or value.

9. **Question:** How does standard deviation assist in understanding the variability in data?

Answer: Standard deviation quantifies how spread out the data are around the mean. A high standard deviation indicates greater variability, while a low standard deviation suggests that the data points are closer to the mean.

10. **Question:** Provide a brief explanation of why calculating descriptive statistics is a critical step before further analysis. **Answer:** Descriptive statistics offer essential insights into the data's distribution, central tendencies, and variability, helping to identify anomalies,

guide data preprocessing, and choose the correct statistical or machine learning methods for further analysis.

Assignment 4: Linear Regression Model – Building and Evaluating a Model

- Question:** What is the fundamental difference between univariate and multivariate linear regression? **Answer:** Univariate linear regression involves one independent variable to predict the dependent variable, whereas multivariate regression incorporates two or more independent variables.
- Question:** How is the regression line estimated using the least-squares method? **Answer:** The least-squares method minimizes the sum of squared differences between observed values and the values predicted by the model, yielding estimates for the slope (β_1) and the intercept (β_0) of the regression line.
- Question:** What is the significance of the Mean Squared Error (MSE) in assessing a regression model? **Answer:** MSE quantifies the average squared difference between actual and predicted values. Lower MSE values indicate that the model's predictions are closer to the actual data points.
- Question:** Describe how R-squared (R^2) is interpreted in a regression context. **Answer:** R-squared represents the proportion of variance in the dependent variable that is predictable from the independent variable(s). An R^2 value closer to 1 suggests a better fit of the model to the data.
- Question:** Explain how you would compute the regression coefficients using a simple example. **Answer:** For a set of data points, you first compute the mean of X and Y, then use the formula:
$$\beta_1 = \frac{\sum((X - \text{mean}(X))(Y - \text{mean}(Y)))}{\sum((X - \text{mean}(X))^2)} \text{ and } \beta_0 = \text{mean}(Y) - \beta_1 \text{mean}(X).$$
- Question:** What does the intercept (β_0) represent in the regression equation? **Answer:** The intercept is the predicted value of the dependent variable when the independent variable is zero. It provides a starting point for the regression line.
- Question:** How would you evaluate the performance of a regression model on both training and test datasets? **Answer:** By calculating metrics such as MSE and R^2 on the training set and then applying the model to the test set to compare predicted values

with actual values. Consistency between training and test errors suggests good model generalizability.

8. **Question:** Provide a brief outline of the steps for building a linear regression model using Python on the Boston Housing dataset. **Answer:**
- Import libraries and load the dataset.
 - Split the data into training and test sets.
 - Fit a linear regression model using the training data.
 - Predict values on both sets and compute evaluation metrics such as MSE and R^2 .
 - Plot the predicted versus true values for visual assessment.
9. **Question:** What is the purpose of splitting data into training and test sets in the context of regression? **Answer:** Splitting data ensures that model evaluation is performed on unseen data, which helps prevent overfitting and provides a more realistic assessment of the model's predictive performance.
10. **Question:** How would you interpret a scenario where the training MSE is low but the test MSE is significantly higher? **Answer:** This scenario suggests that the model may be overfitting the training data, meaning it has learned the noise or specifics of the training set and does not generalize well to new data.
-

Assignment 5: Logistic Regression – Classification and Confusion Matrix Analysis

1. **Question:** What is the primary difference between logistic and linear regression? **Answer:** Logistic regression is used for classification problems where the response variable is categorical (often binary), while linear regression predicts continuous outcomes.
2. **Question:** Explain the role of the sigmoid function in logistic regression. **Answer:** The sigmoid function maps any real-valued number into a range between 0 and 1, effectively transforming the linear output into a probability that can be used for binary classification.
3. **Question:** What information does the confusion matrix provide in the evaluation of a classification model? **Answer:** The confusion matrix shows the counts of true positives

(TP), false positives (FP), true negatives (TN), and false negatives (FN), which are used to compute various performance metrics such as accuracy, precision, recall, and error rate.

4. **Question:** How would you define accuracy in the context of a logistic regression model's performance? **Answer:** Accuracy is the proportion of correct predictions (both true positives and true negatives) out of the total number of predictions made by the model.
5. **Question:** What are precision and recall, and why are they important? **Answer:** Precision is the ratio of true positives to all predicted positives, and recall is the ratio of true positives to all actual positives. These metrics are important for understanding the model's ability to correctly classify the minority class and for evaluating performance especially when classes are imbalanced.
6. **Question:** Describe the process for implementing logistic regression on the Social_Network_Ads dataset using Python. **Answer:** The process includes:
 - Importing necessary libraries and loading the dataset.
 - Preprocessing data (handling missing values and scaling features).
 - Splitting the data into training and test sets.
 - Training a logistic regression model using a library (e.g., scikit-learn's LogisticRegression).
 - Making predictions and evaluating performance using a confusion matrix and relevant metrics.
7. **Question:** What does an ROC curve illustrate in logistic regression, and how might it be used? **Answer:** An ROC (Receiver Operating Characteristic) curve plots the true positive rate against the false positive rate at various threshold settings, which helps in visualizing the trade-off between sensitivity and specificity. The area under the curve (AUC) is used as an overall measure of model performance.
8. **Question:** How can overfitting be detected in a logistic regression model? **Answer:** Overfitting can be detected if the model performs exceptionally well on the training data but poorly on the test data, often indicated by a significant drop in accuracy or a higher error rate on the test set.
9. **Question:** Explain how you would compute the error rate for a classification model. **Answer:** The error rate is computed as the proportion of incorrect predictions (FP + FN) divided by the total number of predictions made. It represents the fraction of misclassifications in the model.

10. **Question:** Why is it important to consider multiple evaluation metrics (such as precision, recall, and F1-score) rather than relying solely on accuracy in logistic regression?
- Answer:** Relying on a single metric like accuracy can be misleading, especially in imbalanced datasets where one class dominates. Precision, recall, and F1-score provide a more nuanced view of the model's performance regarding both false positives and false negatives, enabling more balanced evaluation.
-

These sets of question–answer pairs are designed to test a deep understanding of the practical procedures and underlying theoretical principles as outlined in the manuals. They should prove effective in a viva setting to assess both the technical and conceptual expertise of the candidate.

for all remaining practical please provide

Below is a complete set of viva-style question–answer sets covering all the remaining practical experiments from the two manuals. In the answer below, you'll find separate sections for:

- Group B experiments (which include:
 - Experiment on Impala (database/table creation, insertion, querying)
 - Experiment on Scala with Apache Spark)
- Lab Manual SL3 experiments (which include:
 - Assignment 1: Data Wrangling I
 - Assignment 2: Data Wrangling II
 - Assignment 3: Descriptive Statistics
 - Assignment 4: Linear Regression Model
 - Assignment 5: Logistic Regression and Confusion Matrix Analysis)

I have provided 10 distinct question–answer sets for each experiment. If additional topics (or further parts of an assignment that may have been truncated) are needed, please let me know.

Group B – Practical Experiments

Experiment 1: Using Impala for Database and Table Operations

1. **Question:** What is Impala and why is it preferred over Hive for real-time queries?

Answer: Impala is a distributed SQL query engine for Apache Hadoop. It uses an in-memory execution engine to avoid the disk I/O overhead of MapReduce, making it much faster for interactive, low-latency queries.

2. **Question:** How do you create a new database in Impala?

Answer: You create a new database by executing the SQL statement:

```
CREATE DATABASE my_database;
```

This initializes a container for tables and other objects.

3. **Question:** Describe how to create a table in Impala and state some common data types used in such definitions.

Answer: A table is created with the `CREATE TABLE` command, for example:

```
CREATE TABLE my_table ( id INT, name STRING, age INT );
```

Common data types include INT, STRING, BIGINT, BOOLEAN, DECIMAL, FLOAT, and DOUBLE.

4. **Question:** What is the purpose of the Hive Metastore in the context of Impala?

Answer: The Hive Metastore stores metadata about databases and tables. Impala accesses the metastore to share and manage schema information, which helps in speeding up query planning and execution.

5. **Question:** How is data inserted into an Impala table?

Answer: Data is inserted using the `INSERT INTO` statement. For example:

```
INSERT INTO my_table VALUES (1, 'John', 25), (2, 'Jane', 30), (3, 'Bob', 40);
```

6. **Question:** What command would you use to view all data from a table in Impala?

Answer: To view all data, you can run:

```
SELECT * FROM my_table;
```

7. **Question:** How does in-memory processing contribute to Impala's performance?

Answer: By storing intermediate results in memory (rather than writing them to disk as in MapReduce), Impala significantly reduces latency and speeds up the query execution process.

8. **Question:** When might Impala's performance be limited compared to other processing frameworks?

Answer: Impala is ideal for medium-sized datasets and real-time queries; however, for extremely large datasets where the required memory exceeds available resources, its in-memory design can become a limitation.

9. **Question:** What are some key benefits of using SQL with Impala in a Hadoop environment?

Answer: Using SQL with Impala provides familiarity for users, offers declarative data handling, and leverages the scalability and flexibility of Hadoop clusters for analyzing large datasets with interactive performance.

10. **Question:** How would you troubleshoot unexpected query performance issues in Impala?

Answer: Troubleshooting can involve checking cluster memory usage, reviewing query execution plans, verifying that the latest statistics are available in the metastore, and ensuring that the configuration settings (like memory limits) are optimized for the workload.

Experiment 2: Writing a Simple Scala Program Using Apache Spark

1. **Question:** What makes Scala a particularly good choice for writing Spark applications?

Answer: Scala is the native language for Spark, providing concise, functional, and object-oriented programming constructs. Its seamless integration with the JVM and Spark APIs offers performance efficiency and expressive code.

2. **Question:** How is the “main” method defined in a Scala application?

Answer: The main method is defined within an object. For example:

```
scala

object HelloWorld {
  def main(args: Array[String]) {
    println("Hello, World!")
  }
}
```

This serves as the entry point for the application.

3. **Question:** What are the benefits of Scala’s static type system in distributed computing contexts?

Answer: Scala’s static typing catches errors at compile time, leading to more robust code. This is essential in distributed settings where debugging runtime errors can be challenging.

4. **Question:** Explain the role of implicit classes in Scala.

Answer: Implicit classes enable developers to add new methods to existing types without modifying their original source code. They allow for cleaner, more modular extensions of functionality.

5. **Question:** Describe the steps to compile and run a Scala program on Ubuntu.

Answer: First, install the Scala compiler (using something like `sudo apt-get install scala`), then compile your code with `scalac HelloWorld.scala`, which generates class files. Finally, run the program with `scala HelloWorld`.

6. **Question:** How does Spark's distributed data processing leverage Scala's functional programming features?

Answer: Scala's functional programming constructs (such as higher-order functions, immutability, and concise lambda expressions) simplify the implementation of data transformations and actions, making it ideal for defining Spark RDD or DataFrame operations across a cluster.

7. **Question:** How does Scala support parallelism in Spark applications?

Answer: Scala supports parallelism through its immutable collections and concise syntax for transformations, which, when combined with Spark's execution engine, allow concurrent processing across multiple nodes.

8. **Question:** What is the significance of compiling Scala to JVM byte-code in the context of Spark?

Answer: Compiling Scala into JVM byte-code ensures compatibility with Java libraries and allows Spark to run efficiently on any JVM-based cluster, providing portability and performance.

9. **Question:** How might you integrate Java code in a Scala Spark application?

Answer: Scala is fully interoperable with Java; you can call Java methods or include entire Java libraries directly in your Scala code without additional wrappers, facilitating code reuse.

10. **Question:** Describe one practical advantage of using functional programming paradigms in your Scala code for Spark.

Answer: Functional programming minimizes mutable state. This leads to fewer side effects and easier reasoning about code execution, which is particularly beneficial in distributed environments to avoid concurrency issues.

Lab Manual SL3 – Practical Experiments

Assignment 1: Data Wrangling I

1. **Question:** Why is it important to import libraries like NumPy and Pandas before starting data wrangling?

Answer: These libraries provide essential functions for numerical operations (NumPy) and efficient data manipulation (Pandas), making it easier to load, clean, analyze, and transform datasets.

2. **Question:** What is the role of the `pd.read_csv()` function in data wrangling?

Answer: The `pd.read_csv()` function reads data from a CSV file and loads it into a Pandas DataFrame, providing a structured and easy-to-manipulate format for further analysis.

3. **Question:** How would you check for missing values in a dataset using Pandas?

Answer: You can use `df.isnull().sum()` to compute the number of missing (null) values for each column in the DataFrame.

4. **Question:** What is label encoding and how is it applied in a data wrangling context?

Answer: Label encoding transforms categorical string values into numerical labels, typically using scikit-learn's `LabelEncoder`, so that machine learning algorithms can process the data.

5. **Question:** How do you perform data normalization using min-max scaling in Python?

Answer: Data normalization can be performed using scikit-learn's `MinMaxScaler`. First, instantiate the scaler, fit it to your data, and then transform the data to scale each feature to the range 0–1.

6. **Question:** What is the purpose of using the `describe()` method on a DataFrame?

Answer: The `describe()` method quickly provides summary statistics such as mean, median, standard deviation, min, and max for numerical columns, offering an overview of the dataset's statistical properties.

7. **Question:** Explain why it is crucial to verify the data types of columns during preprocessing.

Answer: Verifying data types ensures that each variable is correctly interpreted (e.g., numeric, categorical, date), which is essential for applying correct transformations and for the performance of subsequent analyses or modeling tasks.

8. **Question:** Describe a scenario in which converting categorical variables to numeric values would be necessary.

Answer: When preparing data for a machine learning algorithm that requires numerical input (for example, regression or neural networks), categorical variables need to be encoded into numeric formats—either by label encoding or one-hot encoding.

9. **Question:** What steps would you take to check the dimensions and structure of your DataFrame?

Answer: Use `df.shape` to see the dimensions (number of rows and columns) and `df.info()` or `df.dtypes` to inspect the data types and overall structure of the DataFrame.

10. **Question:** Why is proper documentation of your data preprocessing steps important?

Answer: Documenting preprocessing steps ensures reproducibility, facilitates debugging, and provides clear rationale for the transformations and imputations applied, which is essential for transparency and effective troubleshooting.

Assignment 2: Data Wrangling II

1. **Question:** How can you use Pandas to count the total number of missing values in a DataFrame?

Answer: By applying `df.isnull().sum().sum()`, you can calculate the total number of missing values across all columns in the DataFrame.

2. **Question:** What are some common methods to fill missing data, and when might you use each?

Answer: Common methods include dropping missing values using `dropna()`, filling with a constant or statistical measure (like mean or median) using `fillna()`, or using interpolation. The choice depends on the extent and nature of missingness, and whether preserving the record is crucial.

3. **Question:** How are outliers typically identified in a dataset?

Answer: Outliers can be identified using visual tools like boxplots or mathematically by computing the Interquartile Range (IQR) and flagging values outside $1.5 \times \text{IQR}$ from the first or third quartile.

4. **Question:** Explain what the IQR method is and how it can be used to treat outliers.

Answer: The IQR is the range between the 25th and 75th percentiles. Values falling outside a typical bound (e.g., below $Q1 - 1.5 \times \text{IQR}$ or above $Q3 + 1.5 \times \text{IQR}$) can be flagged as outliers and then either removed or capped to reduce their impact.

5. **Question:** What is a common technique to reduce data skewness during transformation?

Answer: Applying a logarithmic transformation (e.g., log base 10) to right-skewed data is a common technique that compresses the range and helps achieve a more normal-like distribution.

6. **Question:** Outline the basic algorithm to display a summary of null values present in a dataset.

Answer:

- Import Pandas and load the dataset using `pd.read_csv()`.
- Use `df.isnull()` to generate a boolean DataFrame marking nulls.
- Sum the result by columns with `df.isnull().sum()` to get missing value counts for each column.

7. **Question:** How would you replace an outlier with the mean value of its variable in Pandas?

Answer: Identify the outlier(s) using conditional selection and use `df.loc[condition, 'column'] = df['column'].mean()` to replace them.

8. **Question:** Describe one challenge that might be encountered when dealing with missing data.

Answer: One challenge is determining whether to impute, interpolate, or remove missing values. Imputation can introduce bias if done improperly, while removing rows may reduce dataset size and statistical power.

9. **Question:** What does data transformation mean in the context of preparing data for analysis?

Answer: Data transformation involves modifying the raw data into a more suitable format for analysis—this can include scaling, converting variables, reducing skewness, or creating new features to enhance interpretability.

10. **Question:** Why is it important to document and justify your methods when handling missing values and outliers?

Answer: Clear documentation ensures that the modifications can be reviewed and reproduced later and helps validate that the methods chosen did not distort the underlying relationships in the data.

Assignment 3: Descriptive Statistics – Measures of Central Tendency and Variability

1. **Question:** Define the terms mean, median, and mode.

Answer: The mean is the arithmetic average, the median is the middle value in a sorted dataset, and the mode is the most frequently occurring value in the dataset.

2. **Question:** How would you calculate the mean of a dataset?

Answer: Sum all the numerical values and divide by the total number of observations. In Pandas, this can be done with `df['column'].mean()`.

3. **Question:** Why might the median be a more reliable measure of central tendency in the presence of outliers?

Answer: The median is less affected by extreme values than the mean, so it better represents the central value when data are skewed.

4. **Question:** What information does the `describe()` function in Pandas provide?

Answer: It returns summary statistics such as count, mean, standard deviation, minimum, quartiles (25%, 50%, 75%), and maximum for numeric columns in a DataFrame.

5. **Question:** How is the range of a dataset determined?

Answer: The range is calculated by subtracting the minimum value from the maximum value, indicating the spread of the data.

6. **Question:** What role does standard deviation play in understanding a dataset?

Answer: Standard deviation measures how much the data deviate from the mean on average; a higher standard deviation indicates more variability among the observations.

7. **Question:** In what situations might the mode be particularly useful?

Answer: The mode is especially useful for categorical data or when the data have repeating values, as it identifies the most common category or value.

8. **Question:** How can you calculate the median in Python using Pandas?

Answer: Use `df['column'].median()` to compute the median of a numeric column.

9. **Question:** Explain how grouping data by a categorical variable can enhance statistical analysis.

Answer: Grouping allows you to calculate summary statistics for each subgroup separately (using `groupby()`), revealing differences in central tendency or variability across segments of the data.

10. **Question:** Why is it essential to perform descriptive statistical analysis before applying advanced modeling techniques?

Answer: Descriptive analysis uncovers fundamental properties of the data, such as the distribution, variability, and potential anomalies, which guide further preprocessing and the selection of appropriate modeling methods.

Assignment 4: Linear Regression Model

1. **Question:** What is univariate linear regression and how does it differ from multivariate regression?

Answer: Univariate regression models a relationship between one independent variable and one dependent variable, whereas multivariate regression includes two or more independent variables to predict the dependent variable.

2. **Question:** How does the least-squares method determine the best-fit regression line?

Answer: It minimizes the sum of the squared differences between the observed values and the values predicted by the model, thus finding the slope and intercept that best fit the data.

3. **Question:** What is Mean Squared Error (MSE) and why is it important in regression analysis?

Answer: MSE measures the average of the squared differences between actual and predicted values. A lower MSE indicates that the model's predictions closely match the observed data.

4. **Question:** How is R-squared (R^2) used to assess a regression model's performance?

Answer: R-squared indicates the proportion of the variance in the dependent variable that is explained by the independent variable(s). An R^2 value closer to 1 means the model explains much of the variability.

5. **Question:** Describe the significance of the intercept (β_0) in the regression equation.

Answer: The intercept represents the predicted value of the dependent variable when the independent variable is zero. It serves as a baseline from which the effect of the independent variable is measured.

6. **Question:** Outline the steps to build a linear regression model using Python with the Boston Housing dataset.

Answer: Steps include:

- Loading the dataset and setting up a DataFrame
 - Splitting the data into training and test sets
 - Fitting a linear regression model using scikit-learn's `LinearRegression` on the training data
 - Predicting and evaluating model performance using metrics such as MSE and R^2
7. **Question:** What does it indicate if a model performs well on training data but poorly on test data?
- Answer:** This typically indicates overfitting, meaning the model has learned the noise in the training data rather than the underlying pattern and does not generalize well to unseen data.
8. **Question:** How can you compute regression coefficients from a given dataset?
- Answer:** Coefficients are computed using the formula:
- $$\beta_1 = \frac{\sum((X - \text{mean}(X)) \cdot (Y - \text{mean}(Y)))}{\sum((X - \text{mean}(X))^2)}$$
- and then $\beta_0 = \text{mean}(Y) - \beta_1 \cdot \text{mean}(X)$. Many libraries such as NumPy and scikit-learn automate this calculation.
9. **Question:** What is the purpose of plotting predicted values versus actual values in regression analysis?
- Answer:** Plotting these values helps visually assess how well the model fits the data; a close alignment indicates accurate predictions while systematic deviations reveal potential model inadequacies.
10. **Question:** In a practical scenario, how would you determine if your linear regression model is a “best fit”?
- Answer:** By comparing evaluation metrics (MSE, R^2) on training and test datasets, checking residual plots for randomness, and ensuring that the model accurately captures the underlying trends without overfitting.
-

Assignment 5: Logistic Regression and Confusion Matrix Analysis

1. **Question:** What is the main objective of logistic regression?
- Answer:** Logistic regression is used for classification tasks, typically binary classification, where the goal is to predict a categorical outcome based on input features.
2. **Question:** Why is the sigmoid (logistic) function important in logistic regression?
- Answer:** The sigmoid function maps the linear combination of inputs into a probability

between 0 and 1, making it ideal for interpreting outputs as class membership probabilities.

3. **Question:** What information does a confusion matrix provide in a classification problem?

Answer: A confusion matrix details the counts of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). These counts are then used to calculate metrics such as accuracy, precision, recall, and error rate.

4. **Question:** How do you calculate the accuracy of a logistic regression model?

Answer: Accuracy is calculated as $(TP + TN) / (TP + FP + TN + FN)$, representing the fraction of correct predictions over all predictions.

5. **Question:** Define precision and recall in the context of binary classification.

Answer: Precision is the ratio of true positives to all positive predictions $(TP/(TP+FP))$, and recall (sensitivity) is the ratio of true positives to all actual positives $(TP/(TP+FN))$, highlighting the model's performance in correctly identifying the positive class.

6. **Question:** What are the basic steps to implement logistic regression on a dataset like Social_Network_Ads using Python?

Answer: The steps include:

- Preprocessing the data (handling missing values, scaling features, encoding categorical variables)
- Splitting the data into training and test sets
- Training a logistic regression model using a library such as scikit-learn's `LogisticRegression`
- Making predictions on the test set and evaluating the model using a confusion matrix and related metrics.

7. **Question:** How does the ROC curve help in evaluating a classifier's performance?

Answer: The ROC curve plots the true positive rate against the false positive rate at various threshold settings. The area under the ROC curve (AUC) provides an aggregate measure of performance across all classification thresholds.

8. **Question:** What does a high error rate on the confusion matrix indicate about the model?

Answer: A high error rate indicates that a significant proportion of predictions are incorrect, suggesting that the model may be underperforming or that the data may be imbalanced or improperly processed.

9. **Question:** How can you mitigate overfitting in a logistic regression model?

Answer: Overfitting can be reduced by applying regularization techniques (such as L1 or

L2 regularization), performing cross-validation, and ensuring that the model complexity is appropriate to the size and quality of the dataset.

10. **Question:** Why is it insufficient to only report accuracy for a classifier, especially in imbalanced datasets?

Answer: Accuracy alone may be misleading in imbalanced datasets; a high accuracy rate can mask poor performance on the minority class. Reporting additional metrics such as precision, recall, and the F1-score provides a more comprehensive view of the classifier's performance.

These comprehensive viva question–answer sets cover all the remaining practical assignments from both the Group B manual and Lab Manual SL3. They are designed to test both the theoretical understanding and the practical implementation skills required to successfully complete these experiments. If you need further details or additional sets for any other practical not covered here, please let me know.

from 6 to 9

Below you will find the viva-style question–answer sets for each practical experiment—but only for questions 6 through 9 from each experiment/assignment. These Q&A pairs are drawn from the previously provided comprehensive sets and focus on the later four questions in each section.

Group B – Practical Experiments

Experiment 1: Using Impala for Database and Table Operations

6. **Question:** What command would you use to view all data from a table in Impala?

Answer: To view all data, you run a SQL query such as:

```
SELECT * FROM my_table;
```

This returns all rows from the specified table.

7. **Question:** How does in-memory processing contribute to Impala's performance?

Answer: In-memory processing enables Impala to store intermediate results in RAM,

bypassing the slower disk writes that occur in MapReduce. This leads to significantly lower latency and faster query execution.

8. Question: When might Impala's performance be limited compared to other processing frameworks?

Answer: Impala is designed for medium-sized datasets and real-time queries. For extremely large datasets or when system memory becomes a bottleneck, its in-memory approach may be less effective than batch-oriented frameworks designed to handle very large data volumes.

9. Question: What are some key benefits of using SQL with Impala in a Hadoop environment?

Answer: SQL with Impala provides ease of use through a familiar query language, efficient and low-latency access to data across a Hadoop cluster, and seamless integration with the Hive Metastore for shared metadata.

Experiment 2: Writing a Simple Scala Program Using Apache Spark

6. Question: How does Spark's distributed data processing leverage Scala's functional programming features?

Answer: Scala's functional features—such as higher-order functions, immutability, and concise lambda expressions—allow Spark to define transformations and actions over distributed datasets (RDDs or DataFrames) in a clear and efficient way, facilitating parallel computation.

7. Question: How does Scala support parallelism in Spark applications?

Answer: Scala's design promotes immutability and functional constructs, which reduce side effects and make it easier for Spark to execute operations in parallel across multiple nodes without concurrency issues.

8. Question: What is the significance of compiling Scala to JVM byte-code in the context of Spark?

Answer: Compiling to JVM byte-code ensures compatibility with Java libraries and guarantees that Spark applications run efficiently on any standard JVM-based cluster, maximizing performance and portability.

9. Question: How might you integrate Java code into a Scala Spark application?

Answer: Scala's interoperability with Java allows you to call Java methods or classes directly

within Scala code. This means you can leverage existing Java libraries or incorporate legacy Java code without requiring significant modifications.

Lab Manual SL3 – Practical Experiments

Assignment 1: Data Wrangling I

6. Question: What is the purpose of using the `describe()` method on a DataFrame?

Answer: The `describe()` method quickly computes and displays key summary statistics (such as count, mean, standard deviation, min, and quartiles) for each numeric column, helping to understand the overall distribution of the data.

7. Question: Explain why it is crucial to verify the data types of columns during preprocessing.

Answer: Verifying data types ensures that the variables are interpreted correctly (for instance, numeric versus categorical), which is vital for accurate data analysis, transformation, and when applying machine learning algorithms that expect certain formats.

8. Question: Describe a scenario in which converting categorical variables to numeric values would be necessary.

Answer: When preparing data for machine learning algorithms (such as regression or classification models) that can only process numeric inputs, converting categorical variables (e.g., via label encoding) is necessary

6. Implement Simple Naïve Bayes classification algorithm using Python/R on iris.csv dataset.. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.

7. Extract Sample document and apply following document preprocessing methods: Tokenization, POS Tagging, stop words removal, Stemming and Lemmatization. 2. Create representation of document by calculating Term Frequency and Inverse Document Frequency.

give me viva Q & A for this aswell

Below are viva-style question–answer sets addressing the two additional practical exercises. Each section includes questions that probe both conceptual understanding and practical implementation details.

Assignment 6: Simple Naïve Bayes Classification on the iris.csv Dataset

1. **Question:** What is the core idea behind the Naïve Bayes classification algorithm, and what “naïve” assumption does it make?

Answer: Naïve Bayes is a probabilistic classifier based on Bayes’ theorem. It assumes that all features are conditionally independent given the class label—which is the “naïve” assumption—even though in real data, features may exhibit some correlation.

2. **Question:** Why is the iris dataset considered appropriate for a Naïve Bayes classification model?

Answer: The iris dataset is small, well-balanced, and its features (such as sepal and petal measurements) are reasonably independent. This simplicity and clear class separability make it an ideal candidate for experimenting with Naïve Bayes.

3. **Question:** What pre-processing steps might be required before applying the Naïve Bayes algorithm to the iris dataset?

Answer: Preprocessing includes loading the CSV into a DataFrame, verifying that all features are in numeric form, checking for missing values, and splitting the dataset into training and testing sets to validate the model’s performance.

4. **Question:** How does Laplace (additive) smoothing benefit a Naïve Bayes classifier?

Answer: Laplace smoothing helps by preventing zero probabilities for unseen feature-class combinations in the training data. This ensures that the classifier can still calculate a non-zero posterior probability even if a particular attribute value is missing in the training set.

5. **Question:** Describe the components of a confusion matrix, and explain what TP, FP, TN, and FN represent.

Answer: A confusion matrix tabulates the model’s predictions against actual labels.

- **TP (True Positives):** Correct positive predictions.
- **FP (False Positives):** Incorrect positive predictions (Type I error).

- **TN (True Negatives):** Correct negative predictions.
- **FN (False Negatives):** Incorrect negative predictions (Type II error).

6. **Question:** How are performance metrics such as Accuracy, Error Rate, Precision, and Recall derived from the confusion matrix?

Answer:

- **Accuracy:** $(TP + TN) / \text{Total Predictions}$, measuring overall correctness.
- **Error Rate:** $1 - \text{Accuracy}$, the proportion of incorrect predictions.
- **Precision:** $TP / (TP + FP)$, the proportion of correct positive predictions among all positive predictions.
- **Recall:** $TP / (TP + FN)$, the proportion of actual positives that are correctly identified.

7. **Question:** Outline a typical Python pipeline to implement Naïve Bayes classification on the iris dataset.

Answer: A typical pipeline would include:

- Loading the dataset using Pandas.
- Splitting data into training and testing sets.
- Importing and initializing the Naïve Bayes model from scikit-learn (e.g., using GaussianNB).
- Fitting the model on the training data.
- Predicting on the test set.
- Computing the confusion matrix and derived metrics with scikit-learn's metrics functions.

8. **Question:** What advantages does the Naïve Bayes classifier provide in terms of computational efficiency?

Answer: Due to its simplicity and the conditional independence assumption, Naïve Bayes requires fewer computational resources. It is very fast both during training and prediction even with a large number of features.

9. **Question:** How would you validate the robustness of your Naïve Bayes model on the iris dataset?

Answer: You can validate the model using techniques such as k-fold cross-validation or by holding out a test set. Evaluating metrics (accuracy, precision, recall) on unseen data helps ensure that the model generalizes well.

10. **Question:** What are some inherent limitations of the Naïve Bayes classifier when applied to real-world data?

Answer: Its strong independence assumption can be unrealistic for many real-world datasets where features are correlated. Additionally, if the actual conditional probabilities are far from the assumed ones, the classifier's performance may suffer, particularly with highly imbalanced or noisy data.

Assignment 7: Document Preprocessing and TF-IDF Representation

1. **Question:** What are the essential steps involved in preprocessing a document for text analysis?

Answer: Key steps include tokenization (splitting text into words or tokens), part-of-speech (POS) tagging, removal of stop words, and normalization procedures such as stemming and lemmatization. These steps help to standardize the text and remove noise.

2. **Question:** How does tokenization contribute to the process of document preprocessing?

Answer: Tokenization breaks text into individual units (tokens), such as words or phrases, which form the basic elements for further text processing and analysis. This allows algorithms to work with discrete items rather than the entire continuous text.

3. **Question:** What is the purpose of part-of-speech (POS) tagging in natural language processing?

Answer: POS tagging assigns grammatical labels (like noun, verb, adjective) to each token. This information is useful for understanding the syntactic structure of the text and can help in more advanced tasks such as named entity recognition and context-aware filtering.

4. **Question:** Why is stop word removal important, and how does it affect text analysis?

Answer: Stop words are common words (e.g., "and", "the", "is") that may not carry significant meaning for analysis. Removing them reduces noise and dimensionality, allowing algorithms to focus on the more informative parts of the text.

5. **Question:** What are the differences between stemming and lemmatization, and why might you choose one over the other?

Answer: Stemming crudely cuts off word endings to reduce words to their base or stem,

which may not be an actual word. Lemmatization, on the other hand, uses linguistic knowledge to convert a word to its meaningful base form (lemma). Lemmatization typically produces more accurate results but may be computationally more intensive.

6. **Question:** How is Term Frequency (TF) calculated for a given document?

Answer: TF is calculated by counting the number of times a term appears in a document and then optionally normalizing this count by the total number of terms in that document. It reflects the importance of the term in that particular document.

7. **Question:** Explain the concept of Inverse Document Frequency (IDF) and its role in document representation.

Answer: IDF measures how unique or rare a term is across a collection of documents. It is calculated by taking the logarithm of the total number of documents divided by the number of documents that contain the term. A higher IDF score indicates that the term is more discriminative, reducing the weight of common terms.

8. **Question:** How does the TF-IDF representation combine TF and IDF, and why is this beneficial?

Answer: TF-IDF multiplies the term frequency by the inverse document frequency to balance the importance of a term within a document against its prevalence in the entire corpus. This weighting helps to highlight terms that are significant in a document while diminishing those that are common across all documents.

9. **Question:** Describe some challenges you might encounter during document preprocessing.

Answer: Challenges include handling ambiguous tokens (homonyms), dealing with slang or informal language, managing a high level of noise from irrelevant words, and choosing the right balance in preprocessing steps so as not to remove meaningful information inadvertently.

10. **Question:** What Python libraries or tools can be used to implement these preprocessing steps and compute TF-IDF, and how do they integrate together?

Answer: Libraries like NLTK and spaCy are used for tokenization, POS tagging, stop word removal, stemming, and lemmatization. Scikit-learn provides the `TfidfVectorizer` that can automatically calculate TF-IDF scores from a collection of documents. These tools can be used in a pipeline to first preprocess the text and then convert it into a numerical feature representation suitable for text analysis or machine learning models.

These question–answer sets for Assignment 6 and Assignment 7 address both the algorithmic details and theoretical considerations required in a viva examination setting. They are designed to evaluate your understanding of implementing Naïve Bayes classification, computing evaluation metrics, and applying advanced document preprocessing techniques along with TF-IDF-based document representation.