Project Title: Customer Data Analysis with Spark

Objective:

The goal of this project is to analyze customer data using Apache Spark to gain insights into customer behavior and trends. You will work with large dataset, perform data processing and analytics, and potentially apply machine learning techniques to uncover deeper insights.

Project Overview:

1. **Dataset Selection:**

Dataset is given;

2. Data Ingestion:

- **Task**: Load the chosen dataset for distributed processing.
    - Implement scripts to upload the data and ensure it is accessible for processing with Spark.

3. Data Processing and Cleaning:

- **Task**: Use Spark to clean and preprocess the data. Key steps include:
    - Handling missing values.
    - Removing duplicates.
    - Normalizing data formats (e.g., date formats, categorical variables).
    - Filtering irrelevant data.

4. Data Exploration and Analytics:

- **Task**: Perform exploratory data analysis (EDA) to uncover trends and patterns in customer behavior. This should include:
    - Descriptive statistics (mean, median, mode, counts).
    - Data visualizations (e.g., histograms, scatter plots, bar charts…) if needed
    - Key insights on customer demographics, purchasing patterns, and seasonal trends.

5. SQL Analytics with Spark:

- **Task**: Use Spark SQL to perform advanced queries on the cleaned dataset. Examples include:
    - Finding top customers by total spend.
    - Analyzing purchase frequency by customer segment.
    - Identifying trends in product purchases over time.

6. Optional Machine Learning Analysis:

- **Task**: Apply machine learning techniques using Spark MLlib to further analyze customer data. Potential ML tasks include:
    - Customer segmentation using clustering algorithms (e.g., K-means).
    - Predictive modeling to forecast future purchases or customer churn.

7. NoSQL Integration (Optional):

- **Task**: Store processed customer data in a NoSQL database like MongoDB or Cassandra for further analysis and quick access.
    - Implement scripts to insert, update, and query data.

8. Cloud Deployment:

- **Task**: Package the entire application and deploy it on a cloud platform (Google Cloud DataProc, AWS EMR, or Azure Databricks)

Language accepted : Python, Scala, Java

Annex

Project Repository Structure and Version Control Guidelines

1. Repository Setup:

- Each group creates **one Git repository** on GitHub (or any other Git hosting service).
- The repository should have a clear, descriptive name (e.g., big-data-pipeline-groupX).

2. Repository Structure:

Organize the project into clear folders for each major component.

Here is a SUGGESTION:

- /data_ingestion: Scripts for loading data into HDFS.
- /data_processing: Spark scripts and analytics code.
- /nosql_integration: Scripts for NoSQL database interactions.
- /ml_module (optional): Machine Learning module if MLlib is used.
- /cloud_deployment: Instructions and configuration files for cloud deployment.
- /docs: Documentation, reports, and screenshots.

3. Versioning and Branch Management:

- **Main Branch (main)**: The main branch should contain only stable, tested code. This branch represents the final, fully working version of the project.
- **Feature Branches**:
  - Each feature or component of the project should have its own branch, such as data_ingestion, data_processing, nosql_integration, ml_module, and cloud_deployment.
  - These branches should be regularly merged into main once they are fully developed and tested.
- **Workflow**:
  - Each group member can take responsibility for a specific feature or component, creating their own branch (e.g., data_ingestion-john).
  - Members should **commit frequently** with clear commit messages and **push** their changes to the shared repository.
- **Merging and Pull Requests**:
  - Each time a feature is ready, the group member should submit a **pull request** to merge their feature branch into the main feature branch (e.g., data_ingestion).
  - After team review and testing, feature branches are merged into main.

4. Submissions and Project Milestones:

- **Initial Submission** (in 3 weeks)
  - By the first milestone, groups should have an initial project structure in the repository with basic documentation and at least one working component (e.g., data ingestion).
- **Mid-Project Submission** (in 4 weeks)
  - At this stage, groups should have completed the main structure and be working on the core functionalities. Feature branches should be periodically merged into main.
- **Final Submission** (in 5-6 weeks)
  - The main branch should contain the final, polished version of the entire project, with all features fully integrated and tested.
  - Include a final README.md with clear instructions on running each component, dependencies, and any additional setup instructions.

5. Documentation and Code Review:

- **README and Documentation**:
  - Each feature branch should have a dedicated README or notes file to document specific instructions, key challenges, and testing procedures.
- **Code Review**:
  - Prior to merging a branch into main, group members should conduct a code review to ensure functionality, code quality, and adherence to Big Data best practices.