Building a Batch Processing System for Service Recommendations*

Joël Kazadi

MSc. student in Data Science – International University of Applied Sciences

joel.kazadi@iu-study.org

April 28, 2025

Project overview

The goal of this project is to design and implement a batch-processing-based data architecture that processes information on various establishments/organizations across different U.S. states. The system will ingest, store, preprocess, and aggregate large datasets to support a machine learning application that runs quarterly. The application will generate insights for recommending establishments to potential customers based on the services they offer (delivery, plumbing, air conditioning and heating) and their locations. These insights are generated from data on customer reviews, ratings, and other attributes. To enhance the interactivity of the project, a dashboard will be developed to visualize processed data, providing users with insights and recommendations in an accessible format. The system to build will ensure scalability, reliability, and maintainability while adhering to data governance and security standards.

The project follows a modular microservices architecture, specifically designed to handle the diverse data needs of the recommendation system to build. The architecture is built entirely on **local infrastructure**, using lightweight and reproducible tools. The batch-processing workflow will be executed on a weekly basis, optimizing the aggregation of data related to various services before delivering it to the machine learning application. The primary architectural components include:

(i) *Data Ingestion Microservice*: The input data placed into a local directory, originally available in `.csv` format[1], will be transformed into SQL database before ingestion into the system.
(ii) *Data Storage Microservice*: Ingested data will be stored in a MySQL relational database, managed via Docker Compose.
(iii) *Data Processing Microservice*: The data transformation and aggregation tasks will be handled using Apache Spark ("PySpark" batch jobs). This ensures efficient batch processing, allowing analysis of customer ratings and categorization of establishments based on service type to enhance recommendation accuracy.
(iv) *Data Delivery Microservice*: The final structured data will be stored in another SQL database, providing a reliable and scalable relational database for querying to support the recommendation engine.
(v) *Version Control*: All scripts and infrastructure code will be managed using GitHub, ensuring proper versioning and traceability.

To ensure reliability, local scheduling tools such as Task Scheduler (on Windows) will be used to automate the execution of batch-processing jobs. Logging and basic error-handling mechanisms are implemented directly within the PySpark scripts to ensure traceability and facilitate retries in case of failures. For scalability, the system leverages PySpark's parallel processing capabilities on the local machine and can easily scale vertically by increasing local hardware resources. Governance and security practices include ensuring appropriate file and database access permissions, regular backups of critical data, and the use of Docker container isolation to safeguard data integrity and maintain good operational standards.

*This document is produced as part of the conception phase for the exam of the Course **Project Data Engineering**.
[1]The data is available here: kaggle.com/datasets/abdulmajid115/yelp-dataset