PROJECT OVERVIEW – ecommerce_seller_recommendation

Course Context: Data Engineering, ETL, Spark, Hudi, Recommendation Systems

Student ID: 2025em1100102 | Name: Shaik Khaja Nayab Rasool

INTRODUCTION

The ecommerce_seller_recommendation project implements an end-to-end ETL pipeline that ingests messy sales catalog, company sales and competitor sales data, enforces data quality, produces cleaned transactional tables, and generates ranked seller recommendations for downstream use. It ingests raw data, cleans and validates it, enforces data quality rules, stores transformed data in Apache Hudi tables, and generates category-wise recommendations for sellers. The pipeline follows a layered lakehouse design similar to modern enterprise systems: raw to bronze to silver to gold to recommendations.

Functionally, the pipeline automates the flow from raw CSV snapshots (bronze) through validated, de-duplicated seller/sales records (silver) to business-ready aggregates and recommendation lists (gold). It captures and quarantines invalid rows for audit, maintains run-state metadata for idempotent and incremental processing, and produces outputs suitable for dashboards, reporting, or an API that serves seller recommendations by category and region.

The solution is built on PySpark with Apache Hudi as the transactional store (Hudi helper modules and a hudi-defaults.conf supply table settings), enabling efficient upserts, incremental reads, and compaction control—ideal for evolving seller records and frequent feature updates. Configuration and environment parameters live in the repo (config files, Hudi options and libs/ utilities), and runs are orchestrated via simple wrapper scripts that invoke spark-submit with the required JARs. Data-quality rules, normalization, and transformation logic are implemented as reusable Python modules so the pipeline is config-driven, testable, and portable between local and cloud (S3) deployments; gold outputs contain engineered features (RFM, growth, velocity) and ranked candidate lists used by the recommendation scoring stage.

PROJECT STRUCTURE

This structure separates ingestion, data quality, storage layers, helpers, and testing modules.

```
DSP_GA_2025EM1100102_20112025

└── 2025em1100102

    └── ecommerce_seller_recommendation

        ├── s3

        │   ├── conf

        │   ├── configs
        │   └── data
        │       ├── dqcheck
        │       └── local
        │           ├── clean
        │           └── dirty
        ├── helper
        ├── libs
        ├── scripts
        ├── src
        └── tests
```

OUTPUT LAYER STRUCTURE

```
2025em1100102
└── dsp_ga_2025em1100102_20112025
input
 |-raw
output
    ├── bronze
    ├── quarantine
    └── silver
processed
    ├── gold
    └── recommendations
```

PROJECT DESCRIPTION

The ecommerce_seller_recommendation system processes company sales, competitor sales, and seller catalog datasets. It cleans, validates, and standardizes the incoming data using Spark. These datasets are transformed into a unified model stored in Apache Hudi. The project then computes sales metrics and generates item-level recommendations that each seller should consider onboarding. This reflects a real-world ecommerce analytics workflow.

FUNCTIONAL SUMMARY

Data Ingestion

- Reads raw CSV files for company sales, competitor sales, and seller catalog.
- Supports category and non-category based ingestion patterns.

Bronze Layer

- Stores raw snapshots without modification.
- Maintains run-date partitions for auditing and reproducibility.

Silver Layer

- Performs cleaning, standardization, and schema enforcement.
- Applies data quality rules and sends invalid rows to quarantine.
- Deduplicates data using business keys.
- Writes clean versioned data to Apache Hudi tables.

Gold Layer

- Generates business aggregates such as 7-day, 30-day, and 90-day KPIs.
- Computes sales growth, velocity, demand spikes, and RFM indicators.

Recommendation Layer

- Generates category-level ranked recommendations for sellers.
- Uses metrics such as velocity, recency, growth, and stability scores.
- Outputs placed under processed/recommendations.

TECHNICAL ARCHITECTURE

Layer and Purpose
Raw: Input CSV files.
Bronze: Immutable raw snapshots.
Silver (Hudi): Cleaned and deduplicated tables.
Gold: Aggregations and business KPIs.
Recommendations: Final recommended items per seller.

EXECUTION WORKFLOW

1. Run ETL via spark-submit using wrapper scripts.
2. Initialize Spark session with Hudi configurations.
3. Read raw data and write bronze snapshots.
4. Apply data quality checks and send failures to quarantine.

5. Standardize and transform data.
6. Write cleaned data as Hudi silver tables.
7. Compute gold-level aggregates.
8. Generate recommendations.
9. Store logs and execution metadata.

TECHNOLOGIES USED

PySpark for distributed ETL.

Apache Hudi for storage, incremental updates, and schema management.

Local filesystem or S3 paths for data.

Python helpers for data quality, logging, and utilities.

Shell scripts to orchestrate execution.

HUDI FEATURES USED

- Copy on Write mode.
- Compaction scheduling.
- Incremental pulls for faster reprocessing.
- Schema reconciliation and metadata table.

These ensure reliable updates, auditability, and performance.

RECOMMENDATION ENGINE LOGIC

- Feature inputs include sales volume, velocity, growth, recency, competitor influence, demand changes, and RFM signals.
- Final ranking score is derived using weighted velocity, recency, growth, and stability factors.
- Results are stored in recommendation folders partitioned by run date.

TESTING AND VALIDATION

- Transformation logic validations.
- Numeric consistency tests.
- Recommendation accuracy tests.
- Cross-checks to ensure excluded seller-item pairs are not recommended again.

SUBMISSION CHECKLIST

- ETL spark-submit scripts.
- Hudi configuration file.
- Sample logs and run metadata.
- Bronze, silver, gold output samples.
- Quarantine examples.
- Recommendation outputs.