Data Glacier – Virtual Internship

Week 9: Data Cleansing and Transformation Report

Team Name: DataVision

Team Members:

- **Raj Pawar**
  Email: rajpawar32646@gmail.com
  Country: UK
  College: University of Liverpool
  Specialization: Data Analyst

- **Naga Pavithra Jajala**
  Email: pavithrajajala8naga@gmail.com
  Country: UK
  College: Birmingham City University
  Specialization: Data Analyst

Problem Description

XYZ Credit Union is facing challenges in cross-selling banking products. Customers are typically associated with only one product despite multiple offerings. The organization seeks analytical insights from their customer dataset to uncover patterns, cleanse inconsistencies, and ready the data for deeper analysis (EDA, modelling, and dashboarding).

Dataset Overview

- **Original Files Used:**
  - train.csv (~2.13 GB, 48 columns)
- **Data Source:** Provided dataset covering demographic, behavioural, and product subscription data for customers over time
- **Objective:** Prepare a clean, complete, and analysis-ready version of the train.csv for downstream tasks such as EDA and hypothesis testing

---

Data Cleansing Workflow

Phase 1: Initial Cleaning by Raj Pawar

Raj focused on essential transformations required to make the dataset usable:

- **File Handled:** Raw - train.csv
- **Framework Used:** Dask (for efficient large-file processing)
- **Operations Performed:**
    - Converted problematic fields like age, antiguedad to numeric by stripping and replacing string "NA"
    - Filled missing values:
        - Used **median** for numerical fields (renta, age, antiguedad)
        - Used **mode** for categorical fields (segmento, indrel_1mes, canal_entrada, etc.)
    - Invalid ages capped to [18, 100]
    - Saved intermediate file: **final_cleaned_raj_dask.csv**

Review by Naga Pavithra Jajala:

Raj's cleaning efficiently handled NA values using global strategies. His clean file maintained structure and data integrity and was ideal for my second-stage enhancements.

Phase 2: Advanced Cleansing by Naga Pavithra Jajala

Building on Raj's intermediate file, Pavithra implemented more nuanced, business-informed logic:

- **File Handled:** final_cleaned_raj_dask.csv (renamed by Pavithra for ease of use to Train.csv)
- **Framework Used:** Dask
- **Operations Performed:**
    - **Segment-wise imputation** for renta and age
    - Outlier handling via **IQR-based clipping** for renta
    - Cleaned and converted date and numeric types with accurate typing
    - Final file saved as: **Final_cleaned.csv**

Review by Raj Pawar:

Pavithra's logic added segment-awareness and business insight to the cleansing process. Her use of map_partitions and proper outlier handling made the dataset robust for EDA and modelling.

Final Output

- **File Name: Final_cleaned.csv**
- **Size:** ~3 GB (post-cleaning)
- **Usability:**
    - Cleaned and consistent
    - Suitable for visual analytics, hypothesis testing, and model training
    - No major missing values or dtype inconsistencies

Errors Encountered and Solutions

During the data cleansing phase, we encountered several challenges working with a large-scale, real-world dataset. Below is a summary of key issues and how we resolved them:

1. Mixed Data Types on Load
Issue: Dask raised ValueError for inconsistent column data types (e.g., age, indrel_1mes, conyuemp) when reading CSVs.
Cause: Some columns had strings like "NA" or unexpected object values in numeric columns.
Solution: Explicitly declared column dtypes during dd.read_csv() and treated problematic numeric fields like age and antiguedad as object initially. Converted after cleansing.

2. Cannot Reindex on Axis with Duplicate Labels
Issue: Raised while using .transform() on a Dask DataFrame.
Cause: Missing metadata or structure issue in grouped operations.
Solution: Explicitly specified the meta parameter in .transform() calls.

3. Sampling Not Supported with n in Dask
Issue: .sample(n=5) threw an error.
Cause: Dask only supports sampling by frac.
Solution: Used .sample(frac=0.0001) instead.

4. Segment Median Imputation Failures
Issue: Some segments had all missing values for columns like renta, breaking .transform().
Solution: Added chained .fillna() fallback to use overall median.

5. DtypeWarnings and Performance Bottlenecks
Issue: Repeated warnings and memory usage spikes.
Solution: Used low_memory=False, fine-tuned dtype handling, used .persist() and .compute() only when needed.

These fixes stabilized our Dask workflow and allowed robust processing of a dataset over 2 GB, enabling successful downstream analytics.

Conclusion

This two-stage collaborative approach allowed us to work on a real-world scale dataset using distributed computing (Dask), while aligning with business goals and data analyst best practices. The final dataset is now ready for EDA in Week 10 and further statistical and predictive analysis in the weeks to follow.