Christopher Aaron O'Hara

Udacity MSc Capstone – Data Workflow

February 15, 2026

Overview

This project implements a complete, reproducible data workflow using OpenRCA Telecom telemetry data. The notebook includes ingestion, cleaning, exploratory analysis, visualization, and an AIF360-based screening check for potential group-wise imbalance. The workflow is designed as a base layer for future ML, deep learning, and agentic RCA modules.

Dataset: OpenRCA (Telecom)

Link: https://github.com/microsoft/OpenRCA

Project Repository: https://github.com/Ohara124c41/ai-programming-foundations-project

Dataset Description

The dataset used in this notebook is a combined Telecom telemetry slice from OpenRCA, constructed from five metric files (metric_app, metric_container, metric_middleware, metric_node, metric_service) for day 2020_04_20. The original combined pool contains 592,921 rows and multiple schema-specific columns; a reproducible subset of 50,000 rows is used for EDA runtime control. The analysis focuses on key variables such as metric_source, timestamp, value, and selected service-level fields (avg_time, num, succee_rate) where available. The merged-table structure creates structural sparsity because not every metric source emits every field.

Workflow Description (High Level)

1. Ingestion

- Load OpenRCA Telecom metric CSV files with Pandas.

- Merge metric files into one analysis frame and preserve source provenance via metric_source.

- Apply deterministic subsetting for reproducible EDA scale.

2. Cleaning

- Standardize column names to snake_case.

- Remove duplicate rows.

- Parse timestamp-like columns using epoch-unit inference.

- Apply selective imputation for non-structural missingness.

- Add derived features (event_hour, event_dayofweek, value_log1p).

- Keep highly sparse source-specific columns for schema transparency instead of globally imputing them.

3. Exploratory Analysis

- Generate summary statistics, missingness profile, cardinality, correlation matrix, and moment statistics (mean/std/skewness/kurtosis/IQR/CV).

- Produce source-level breakdown tables and time coverage summaries.

- Add explicit statistical validation tables and covariance analysis for high-coverage numeric features.

4. Visualizations

- Missingness bar chart (Figure 1).

- Distribution-shape chart for skewness/kurtosis (Figure 2).

- Supplementary value distribution plots (histogram and boxplot).

- Supplementary PCA projection (linear structure view).

- t-SNE projection (Figure 3, nonlinear local structure view).

5. Summary

- Interpret data quality, distribution behavior, structural heterogeneity, and fairness-risk signals.

Key Decisions and Assumptions

The key design decision was to preserve metric_source and treat much of the sparsity as structural, not random. This aligns with reproducible workflow principles: transparent transformations, explicit assumptions, and deterministic configuration choices (Danchev et al., 2022). In the cleaning stage, the raw merged data showed mixed naming conventions, epoch-like timestamps stored as numeric values, and schema-driven nulls. Cleaning evidence from the executed notebook shows: 0 duplicate rows removed, 1 of 3 time-like columns parsed to datetime, 325 values imputed across 7 eligible columns, and 6 structural sparse columns retained (servicename, starttime, avg_time, num, succee_num, succee_rate), each with 99.942% missingness. Imputation was constrained to lower-missingness columns to avoid fabricating values in highly source-specific fields. For fairness-risk screening, a proxy label was used when explicit anomaly labels were unavailable, which is useful for diagnostics but not a causal fairness claim.

Results and Interpretation

Figure 1 shows structural missingness concentrated in source-specific service fields, while shared telemetry fields are largely complete. Supplementary distribution plots make the heavy-tailed value behavior visually explicit, and Figure 2 plus the validation tables quantify that pattern: value has strong right-tail behavior (skewness about 10.82; kurtosis about 130.77), while value_log1p improves but does not fully normalize shape (skewness about 2.84; kurtosis about 8.90). The supplementary PCA view and Figure 3 (t-SNE) both show patterned structure with partial source overlap rather than clean source separation. Source imbalance is substantial (metric_node = 25,703 rows versus metric_app = 29 rows), which reinforces the need for source-aware feature engineering and evaluation before model training.

Responsible Practice (Bias and Data Quality)

Bias can be introduced if structural missingness is treated as random or if one global cleaning rule is applied across incompatible metric schemas. A concrete example in this dataset is the six service-level columns with 99.942% nulls; globally imputing those fields would fabricate values for 49,971 rows that never had those attributes. This can cause downstream models to learn source identity artifacts instead of true anomaly or failure patterns. Source imbalance (for example metric_node versus metric_app volume) increases this risk. The AIF360 proxy screening indicated material source-group differences (statistical_parity_difference about 0.0959; disparate_impact about 29.34), so future iterations should report group-wise performance, test sensitivity to imputation choices, and enforce source-aware validation design.

Reproducibility

This project supports reruns through a dependency file (requirements.txt) and deterministic controls (subset size and random state). The notebook runs top-to-bottom without execution errors in the recorded environment. Version control is used to track progress with multiple commits and at least one additional branch for development work. The repository is publicly linked for reviewer verification.

Sources and Citations

- Reproducibility design decision: (Danchev et al., 2022)

- OpenRCA benchmarking context: (Xu et al., 2025)

- RCA benchmark framing for telemetry RCA: (Pham et al., 2025)

References

1. Danchev, V., Sood, H., Rodriguez, M., Fadadu, R. P., Baca, C. N., Lendvay, T. S., Jackson, G. P., Hu, Y., & Kung, H. (2022). Reproducible Data Science with Python: An Open Learning Resource. **Journal of Open Source Education, 5**(50), 137., https://jose.theoj.org/papers/10.21105/jose.00156
2. Xu, J., Zhang, Q., Zhong, Z., He, S., Zhang, C., Lin, Q., Pei, D., He, P., Zhang, D., & Zhang, Q. (2025). **OpenRCA: Can Large Language Models Locate the Root Cause of Software Failures?** International Conference on Learning Representations (ICLR 2025). https://openreview.net/forum?id=M4qNIzQYpd
3. Pham, L., Zhang, H., Ha, H., Salim, F., & Zhang, X. (2025). **RCAEval: A Benchmark for Root Cause Analysis of Microservice Systems with Telemetry Data.** Companion Proceedings of The Web Conference 2025, 777-780. https://arxiv.org/abs/2412.17015