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

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).

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.

4. Visualizations

- Missingness bar chart (Figure 1).

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

- 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). Time parsing used epoch-unit inference to avoid invalid datetime coercion. 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. Figure 2 (plus the validation tables) shows heavy-tailed behavior in value (high skewness and kurtosis), with value_log1p improving but not fully normalizing distribution shape. The PCA and t-SNE projections both show patterned structure in the telemetry records, with partial source overlap rather than perfect source separation. These outputs support a source-aware feature engineering strategy 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. Proxy-based fairness screening indicated material differences across source-group rates, signaling a risk that downstream models may inherit source imbalance effects. To reduce risk, future iterations should report group-wise model performance, test sensitivity to imputation choices, and enforce source-aware validation design. The IBM AIF360 was used to help identify risks in the data.

Reproducibility

This project supports reruns through a dependency file (requirements.txt) and deterministic controls (subset size and random state). The notebook is intended to run top-to-bottom without errors on the provided environment. Version control is used to track progress with multiple commits and at least one additional branch for development work.

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