Lightspeed Asset Inventory Analysis

Jacob L. Porter

Western Governors University

Table of Contents

A. Project Highlights

The primary research question addressed in this capstone project was: Can Lightspeed accurately validate and predict which network assets are missing from its core management systems to improve data quality ahead of a security audit? This question reflects a critical organizational need to ensure accurate asset records across observability, inventory, and IP address management (IPAM) systems, reducing operational risk and improving audit readiness. This approach is consistent with prior research on large-scale network asset management, such as Facebook's Robotron system, which demonstrated the importance of automated data reconciliation and analytics-driven monitoring for improving data quality in production environments (Kim et al., 2020).

The scope of this project included the generation and analysis of simulated network asset data representing Observability, Inventory, and IPAM systems. The project covered data generation, preparation, analysis, scenario testing, and reporting, but intentionally excluded production data or direct integration with live systems to ensure security and compliance.

The implemented solution used a Python-based analytics pipeline built with Faker, Pandas, and Scikit-learn to generate and analyze synthetic data. Descriptive analytics and predictive modeling using a random forest classifier were applied to assess asset completeness and identify risk factors. The workflow was orchestrated and tracked with MLflow, and the results were documented in both structured reports and a Jupyter Notebook to ensure reproducibility and transparency.

B. Project Execution

The project was executed in alignment with the original plan and goals defined in Task 2, with minor refinements to improve flexibility.

Project plan: All major goals, objectives, and deliverables from Task 2 were completed:

- Validated data completeness: Generated synthetic datasets and performed completeness analysis, confirming that baseline completeness was 81.5% (statistically significant) and alternative scenario completeness was 74.6% (not statistically significant).
- Developed predictive models: Built and evaluated Random Forest models for Inventory and IPAM, both exceeding the success benchmark (Inventory: F1-score 0.87; IPAM: F1-score 0.90).
- Provided actionable insights: Identified “region” and “role” as key risk factors, enabling recommendations for targeted audits and monitoring.
- Ensured reproducibility: Executed the full MLflow pipeline and documented results with statistical tests, model metrics, and scenario comparisons in analysis.pdf.

Project Scope: The project scope remained unchanged from Task 2. Synthetic data simulation was used, the three target systems (Observability, Inventory, and IPAM) were incorporated, and all planned steps—including data generation, cleaning, labeling, analysis, training, and evaluation—were completed as proposed.

Project planning methodology: The CRISP-DM methodology remained effective for structuring the project. Each phase—data understanding, preparation, modeling, evaluation, and reporting—was executed as planned, requiring no methodological changes.

Project timeline and milestones: A minor refinement was made to the scenario-testing pipeline to improve flexibility, but this adjustment did not affect the overall schedule. The project was completed within the original **20-hour estimate**, and all milestones were achieved by the **7/31 deadline**.

Project Resources and Costs: The project was delivered on time and within the proposed budget. Total costs amounted to **$0.00** because the solution utilized open-source tools and a synthetic dataset.

C. Data Collection Process

How the data selection and collection process differed from the plan:

The data was generated using Python scripts and the Faker library to simulate realistic asset records for Observability, Inventory, and IPAM systems. This approach matched the plan outlined in Task 2, with no deviations required. Controlled missingness was introduced through configurable parameters to support scenario testing. This design provided flexibility for both baseline and alternative data-generation scenarios without requiring any changes to the original collection methodology.

How obstacles were handled during data collection:

Several obstacles were encountered and resolved during the data collection process:

- **MLflow parameter propagation bug:** When parameterizing the pipeline to accept alternative data generation configurations, an issue was discovered where MLflow did not consistently propagate custom config values across multi-step pipeline runs. To address this, pipeline steps were executed individually for scenario-specific configurations, ensuring that results matched the intended parameters.

- **Feature overlap between scenarios:** During notebook development, variable reuse between the default and alternative scenarios caused reporting confusion. This was resolved by isolating scenario outputs in separate data structures, maintaining clarity, and preventing cross-contamination of results.
- **Model output verification issue:** At one stage, model evaluation metrics displayed unrealistically perfect scores. Upon investigation, this was traced to a mismatch between feature sets and labels during data splitting. The data preparation process was audited and corrected, ensuring proper alignment between features and labels before re-running model training and evaluation.

How unplanned data governance issues were handled:

No unplanned data governance issues were encountered. Because the project relied entirely on synthetic data, there were no privacy, security, or compliance concerns. This allowed full control over data generation and testing without restrictions on usage or sharing.

C.1 Advantages and Limitations of the Data Set

Advantages:

The dataset was entirely synthetic, which provided several significant benefits. Most notably, it was **fully reproducible**, ensuring that any analysis or modeling could be repeated with identical results. Additionally, because no production data was used, the dataset carried **no privacy, security, or compliance risks**, eliminating the need for additional governance controls. Finally, the dataset was **highly configurable**, allowing controlled adjustments to missingness parameters for scenario testing. This flexibility supported a robust evaluation of both baseline and alternative conditions without introducing unnecessary complexity.

Limitations:

The primary limitation of the dataset is that, as synthetic data, it cannot fully replicate the **unexpected variability and edge cases** often present in real-world production systems. For example, genuine network asset records may include irregularities such as inconsistent naming conventions or incomplete metadata, which were not modeled in the generated data. While substantial effort was made to create data that closely reflects real-world conditions, true user behavior is inherently difficult to reproduce in a synthetic environment. However, this approach was appropriate for the scope of this capstone, and the resulting pipeline and models are designed to be adapted and validated against production data in a real-world telecommunications environment in future phases.

D. Data Extraction and Preparation

All asset records were generated and labeled programmatically using Python scripts and the Faker library, then exported as CSV files for maximum portability and compatibility with downstream tools. Data extraction involved programmatically creating synthetic asset datasets that represented Observability, Inventory, and IPAM systems. These datasets were generated directly from configurable parameters, ensuring that controlled missingness rates and scenario-specific conditions could be reproduced on demand.

Data preparation included:

- Creating binary labels for asset presence in Inventory and IPAM based on configurable missingness probabilities.
- Standardize key attributes such as device role, region, and vendor to maintain data consistency.
- Performing data cleaning and feature engineering using Pandas, including deduplication, data type enforcement, and feature encoding for model training.

These processes were appropriate because they ensured full control, reproducibility, and transparency across the pipeline. By automating data generation and preparation with Python and Pandas, the project minimized human error, simplified scenario testing, and leveraged open-source tools that are widely adopted and scalable for future integration with real-world datasets.

E. Data Analysis Process

E.1 Data Analysis Methods

Descriptive statistics were used to measure asset presence across Observability, Inventory, and IPAM systems. This included calculating completeness metrics, such as the percentage of assets present in each system and the percentage present in both systems. These metrics provided the baseline needed to test the hypothesis that overall completeness exceeded the 75% target threshold. This method was appropriate because it provided an objective view of data quality and informed subsequent modeling and statistical testing. A one-proportion **z-test** was performed to evaluate whether asset completeness was statistically greater than 75%. This test was applied to both the baseline and alternative scenarios, providing formal evidence of whether completeness met the desired threshold. The z-test was appropriate because it is designed for binary outcomes, such as the presence or absence of an asset across systems, and provided a rigorous statistical foundation for validating the project's hypothesis.

Two **Random Forest classifiers** were developed to predict missing assets in Inventory and IPAM based on features such as region and vendor. Models were trained on synthetic labeled data and evaluated using **accuracy** and **F1-score**, ensuring balanced assessment of predictive performance. Additionally, **feature importance analysis** was performed to identify the most influential attributes, with "region" and "role" emerging as the top predictors. Random Forest was appropriate for this project because it handles categorical variables, resists overfitting, and provides interpretability through feature importance metrics. The pipeline was executed with an alternative configuration that altered missingness probabilities to simulate different operational conditions. Completeness metrics, z-test results, and model performance were compared between the baseline and alternative scenarios. Scenario analysis was appropriate because it validated the robustness of the pipeline and confirmed that the analytical methods could adapt to variations in data quality.

E.2 Advantages and Limitations of Tools and Techniques

Python/Pandas:

- **Advantage:** Python, combined with Pandas, provides a fast and flexible environment for data generation, preparation, and analysis. Its extensive libraries and community support allowed for efficient feature engineering and cleaning operations.

- **Limitation:** Pandas operates in memory, which can limit scalability when dealing with extremely large datasets. Although this was not a constraint for the synthetic datasets in this project, it could become a factor in future production deployments.

Scikit-learn:

- **Advantage:** Scikit-learn offered a robust suite of machine learning algorithms and evaluation metrics, making it straightforward to train, evaluate, and interpret Random Forest models. Its built-in feature importance tools add interpretability to the modeling process.
- **Limitation:** Scikit-learn models require data to be fully prepared in advance (e.g., encoded categorical variables), which can add preprocessing complexity for production pipelines.

MLflow:

- **Advantage:** MLflow provided a clear framework for orchestrating pipeline steps and ensured reproducibility across different runs. It enabled parameterized scenario testing and streamlined the tracking of configurations and results.
- **Limitation:** A known issue was encountered with parameter propagation when running multi-step pipelines, requiring the execution of steps individually for certain scenarios. While this workaround was effective, it added minor operational overhead.

Jupyter Notebook:

- **Advantage:** Jupyter Notebook allowed for interactive development, clear documentation, and seamless integration of code, visualizations, and narrative explanations, which supported both analysis and presentation.

- **Limitation:** While excellent for exploration, Jupyter notebooks can be less suitable for productionized workflows, requiring eventual migration of logic into standalone Python scripts for long-term maintainability.

E.3 Application of Analytical Methods

The analytical methods described in Section E.1 were applied systematically using the MLflow pipeline and Python-based tooling. Each step was implemented with explicit validation of assumptions to ensure accuracy and reproducibility.

1. **Data generation and preparation:**

   Synthetic asset data was generated using Python and the **Faker** library to simulate Observability, Inventory, and IPAM systems. Scenario-driven missingness probabilities were applied to create both a baseline dataset and an alternative scenario for robustness testing. The data was loaded into **Pandas**, where presence flags for Inventory and IPAM were assigned, and all features were standardized (e.g., device role, region, vendor). This ensured that the data set reflected the expected structure and supported both descriptive analysis and predictive modeling.

2. **Descriptive analysis:**

   Descriptive analytics was performed to calculate asset presence rates across systems. Using Pandas, metrics such as completeness percentages were computed and cross-validated against the expected values defined by the scenario configuration. This step confirmed the integrity of the data and served as the foundation for statistical testing.

3. **Statistical testing (z-test for proportions):**

   A one-proportion z-test was applied to determine whether asset completeness exceeded the predefined 75% threshold. The test was performed using the counts of assets present in both systems relative to the total population. Assumptions for the z-test, including a sufficiently large sample size and independent observations, were verified. The test was executed for both baseline and alternative scenarios, producing statistical evidence to confirm or reject the hypothesis regarding completeness.

4. **Predictive modeling:**

   The dataset was split into features (e.g., region, vendor) and labels representing asset presence or absence. Prior to modeling, class balance was evaluated to ensure that neither class (present or missing) was disproportionately represented. A **Random Forest classifier** was then trained using **Scikit-learn** with an 80/20 train-test split. This algorithm was selected because it is robust to overfitting, handles categorical feature encoding effectively, and provides interpretability through feature importance scores.

5. **Model evaluation:**

   Model performance was evaluated using accuracy, precision, recall, and F1-score. Feature importance plots were also generated to identify the most influential predictors of asset missingness, with "region" and "role" emerging as key drivers. Assumptions for the model—including correct feature encoding and acceptable class balance—were validated before training to ensure reliable results.

6. **Scenario analysis:**

   To assess the robustness of the pipeline, the analysis was repeated with an alternative data generation configuration that altered missingness probabilities. Completeness metrics, z-test results, and model performance were compared between scenarios. This validated that the methodology could adapt to different data conditions and reinforced the reliability of the analytical process.

By following these structured steps and validating assumptions at each stage, the project ensured that its findings were statistically sound, reproducible, and aligned with the intended research question.

F Data Analysis Results

F.1 Statistical Significance

This project used both statistical testing and predictive modeling to evaluate asset completeness and support the hypothesis that asset completeness across Inventory and IPAM exceeds 75% and can be accurately predicted using asset attributes such as region and role.

A one-proportion z-test was used to measure whether completeness exceeded the 75% threshold. In the baseline scenario, completeness was 81.6%, resulting in a z-statistic of 17.90 and a p-value of 5.47×10^{-72}. At an alpha level of 0.05, the null hypothesis (completeness $\leq$ 75%) was rejected, confirming that completeness was statistically significant. In contrast, the alternative scenario showed a completeness rate of 74.6%, a z-statistic of –0.86, and a p-value of 0.804, leading to a failure to reject the null hypothesis. These results demonstrate that while baseline conditions meet the threshold, completeness can degrade under adverse conditions, reinforcing the value of predictive modeling. The extremely small p-value in the baseline scenario is expected, given the large sample size (11,246 assets), where even small deviations from the threshold yield high statistical power.

Predictive modeling further validated the hypothesis. Two Random Forest classifiers—one for Inventory and one for IPAM—were trained on the dataset. In the baseline scenario, the Inventory model achieved 88.6% accuracy and an F1-score of 0.87, while the IPAM model achieved 93.2% accuracy and an F1-score of 0.90, both exceeding the success benchmark of an F1-score ≥ 0.80. Under the alternative scenario, the Inventory model achieved 95.6% accuracy with an F1-score of 0.96, while the IPAM model achieved 91.2% accuracy with an F1-score of 0.89, demonstrating robustness across data conditions. Feature importance analysis identified "region" and "role" as the most influential predictors of missing assets, aligning with expectations and confirming the validity of the modeling approach.

These findings confirm that the baseline scenario meets statistical completeness requirements and that predictive modeling provides a reliable means of identifying and mitigating missing asset risk, supporting the project's hypothesis.

F.2 Practical Significance

The results of this project have strong practical significance because they provide a clear, actionable framework for improving asset completeness and audit readiness. The baseline scenario confirmed that asset completeness across Inventory and IPAM was 81.6% and statistically significant, which validates that the organization's asset data is reliable under normal operating conditions. More importantly, the predictive models—both exceeding the F1-score benchmark of 0.80—offer a practical way to identify missing or high-risk assets before they affect compliance, enabling proactive data governance rather than reactive audits.

For example, the Random Forest models can be integrated into the organization’s data quality pipeline to automatically flag assets predicted to be missing from Inventory or IPAM. Operations teams could then prioritize investigations into these flagged assets, reducing audit preparation time and minimizing the risk of failed compliance reviews. Additionally, feature importance analysis identified “region” and “role” as the strongest predictors of asset missingness, which provides specific insights that the organization can act on—such as focusing remediation efforts on certain regions or operational processes where missing data is more likely to occur.

Even in the alternative scenario, where completeness dropped to 74.6% and failed to meet statistical significance, the predictive models maintained high performance, demonstrating their value in detecting risks early. This capability supports continuous improvement in asset management, targeted data quality interventions, and a measurable reduction in audit-related risk. As a result, the project outcomes not only validate the effectiveness of the analytics solution but also provide a scalable path to operationalize it within a real-world telecommunications environment.

F.3 Overall Success

This project was highly successful in meeting the objectives defined in Task 2 and delivering a data-driven framework for improving asset completeness and audit readiness. The statistical analysis confirmed that baseline asset completeness was 81.6% and statistically significant, validating that the organization’s asset records are largely reliable under normal conditions. Furthermore, the predictive models for Inventory and IPAM both exceeded the success benchmark, with F1-scores of 0.87 and 0.90, respectively, in the baseline scenario, demonstrating that machine learning can accurately identify assets at risk of being missing.

The alternative scenario reinforced the value of the solution by showing that even when completeness dropped to 74.6%—failing to meet statistical significance—the predictive models continued to perform effectively, with F1-scores of 0.96 for Inventory and 0.89 for IPAM. This adaptability highlights the robustness of the analytical approach and its ability to deliver actionable insights even in degraded conditions.

Additionally, the feature importance analysis provided clear guidance for operational improvement by identifying “region” and “role” as the strongest predictors of missing assets. This finding enables targeted interventions, such as improving processes in specific regions or refining role-specific workflows, to directly reduce data gaps.

By successfully validating data completeness, delivering accurate predictive models, and generating actionable insights, this project has met all major goals while establishing a reproducible pipeline through MLflow. These outcomes not only address the immediate research question but also create a scalable foundation for continuous data quality monitoring, proactive audit preparation, and long-term operational improvements.

G. Conclusion

G.1 Summary of Conclusions

This project confirmed that asset completeness across Inventory and IPAM can be effectively measured and predicted. The baseline scenario showed an 81.6% completeness rate, which was statistically significant, while the alternative scenario dropped to 74.6% and failed to meet the threshold, emphasizing the value of predictive modeling for risk detection. The Random Forest models for both Inventory and IPAM exceeded the success benchmark, achieving F1-scores of 0.87 and 0.90, respectively, in the baseline scenario and remaining robust under alternative conditions. Feature importance analysis identified “region” and “role” as the key drivers of missing assets, providing actionable insights to focus remediation efforts where they will have the greatest impact. These results validate the project’s hypothesis, deliver a reproducible analytics pipeline, and establish a practical foundation for ongoing data quality monitoring and proactive audit preparation.

G.2 Effective Storytelling

The visualizations created during this project were essential for translating technical results into clear, actionable insights for stakeholders. Completeness summaries provided simple percentage-based metrics and statistical test results that demonstrated whether data met the 75% threshold, making it easy to communicate the state of asset quality. Feature importance charts for the Random Forest models highlighted “region” and “role” as the primary drivers of missing assets. These visuals allowed non-technical stakeholders to quickly understand why certain assets were at higher risk and where to focus remediation efforts. By combining these visual elements with clear reporting, the analysis effectively told a data-driven story that supports decision-making, from validating data quality to prioritizing operational improvements.

G.3 Recommended Courses of Action

1. **Implement predictive monitoring for asset completeness:** Integrate the Random Forest models into the organization's data quality workflows to automatically flag assets likely to be missing from Inventory or IPAM. This will allow teams to proactively address data gaps, reduce manual audit preparation, and ensure ongoing compliance.
2. **Target remediation efforts by region and role:** Use the feature importance insights to focus audit and remediation efforts on the regions and asset roles most associated with missing records. Prioritizing these high-risk areas will improve data quality faster and with fewer resources, creating measurable operational efficiency gains.

H Panopto Presentation

WGU Panopto Presentation -

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=e92c9c25-246d-4bf4-8881-b32b004ea556

Recording Name: lightspeed-capstone

References

Kim, J., Narayanan, R., Agneeswaran, V. S., Ma, K., Puthenpurayil, J., & Agrawal, D. (2020). Robotron: Top-down network management at Facebook scale. *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*, 19–35. https://www.usenix.org/conference/nsdi20/presentation/kim

Appendix A

Evidence of Project Completion

The following artifacts provide evidence of the completed project, including all code, documentation, data, and analysis outputs:

GitHub Repository (Code and Pipeline): https://github.com/Baelfur/lightspeed/tree/main

This repository contains the full MLflow pipeline implementation, including data generation, preparation, training, and reporting scripts, along with configuration files for scenario testing.

Project Presentation (PowerPoint): D502 – Analytics Capstone Presentation

A slide presentation summarizing the project goals, methodology, results, and conclusions, suitable for stakeholder review.

Analysis Workbook (PDF): Analysis Notebook PDF

A fully executed version of the analysis notebook, including descriptive statistics, statistical tests, predictive model training results, feature importance visualizations, and scenario comparisons.

Final Dataset: Labeled Asset Dataset (CSV)

The enriched dataset generated during the project was used for completeness analysis and model training.

These artifacts collectively demonstrate the successful completion of the project and provide all materials required for review or future reproduction.