Mission Integration and Ops Data Solutioning

PURDUE UNIVERSITY | The Data Mine | ASU Angelo State University | BARRIOS

INTRODUCTION

Barrios Technology is modernizing NASA's search engine to streamline critical information access during International Space Station (ISS) operations. The project aims to efficiently retrieve and display key data, enabling Flight Directors to quickly locate essential information during time-sensitive and high-stakes decision-making scenarios.

Definitions:

- **Flight rules** - A set of predefined guidelines that govern nominal and off-nominal operations
- **Anomaly reports** - An operational database that documents occurrences of off-nominal events
- **Flight Director** - Leader of the Flight Control Team with operational responsibility for crew safety, vehicle safety, and mission execution
- **MERFIT** -A collection of detailed engineering documentation related to failure investigations of ISS anomalies

4 TEAMS, 1 MISSION

Each team, split into four sub-teams, contributes to a connected pipeline that transforms raw data into actionable insights supporting NASA decision-makers during live ISS operations (see Figure 1):

1. **Data Architects:** Designed the overall data & solution architecture.
2. **Data Engineers**: Built pipelines to process raw data from multiple sources.
3. **Data Scientists:** Applied NLP to optimize search & retrieval of key information.
4. **Front-End Engineers:** Developed the user interface for real-time, querying.

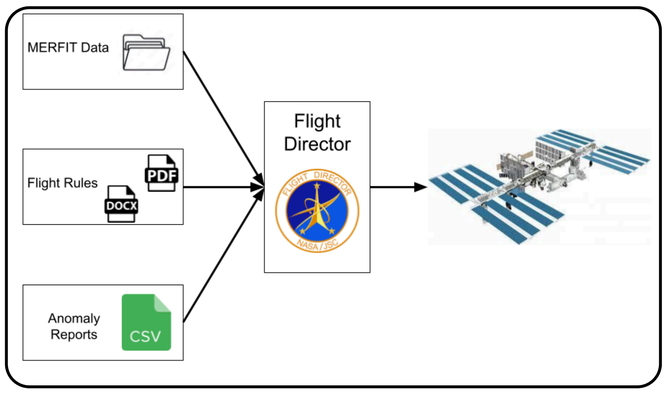

Figure 1: NASA's Data Sprawl in Action

THE DATA ARCHITECTS

How (Data & Solution Architecture Steps - Figure 2):

1. Python crawler continuously checks for updated source datasets and performs Extract, Transform, & Load (ETL) process, transferring data from each unique dataset into its respective table in the target database.
2. The Full-Text Search features of PostgreSQL will be leveraged on cleaned and regularly updated data to power the core search functionality of the web application.
3. The UI will allow users to submit search queries for specific anomaly data, which will be returned by the API. Data science models may be used to enhance search functionality or provide statistical summaries.

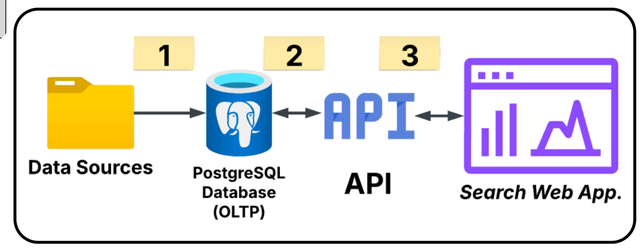

Figure 2: Data Architecture diagram - Pipeline Process

THE DATA SCIENTISTS

Why: Match search queries made by Flight Directors to relevant flight rules, making it easier to find the right guidelines for handling anomalies

How:

1. We used BERT (a language model developed by google) to convert text descriptions of anomalies and flight rules into numerical representations
2. These numerical representations can then be compared, which lets us map a 'description' to its relevant subsystem, and find it's relevant rules. As seen in Figure 4, we can then visualize these clusters made from the subsystem descriptions and flight rules.
3. As an additional feature, we built a machine learning pipeline that cleans and combines text and category data using TF-IDF and one-hot encoding, then trains a tuned logistic regression model to predict if an anomaly is of High, Medium, or Low severity.

Next Steps:

With the foundations cemented, the next steps would be creating our own model to create more accurate custom embeddings that can better capture the meanings of the aerospace jargon in our dataset

THE DATA ENGINEERS

Database: PostgreSQL database server to centralize the data into a single standardized format able to be accessed by multiple teams. Selected because it is free and open source, and is hosted on the Gilbreth cluster, sourced from datamine-help. Figure 3 showcases this database and the variables within each of the datasets.

Data Parsing: Processing raw data into structured formats using Python libraries:

- Database interaction: psycopg2
- Document parsing: python-docx (for .docx), pypdf (for .pdf)
- Data manipulation: pandas (for .csv)

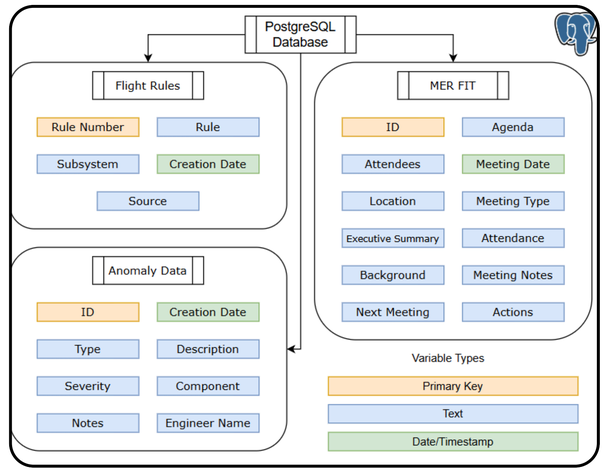

Figure 3: Our Database on PostgreSQL

THE FRONT-END ENGINEERS

Why: Need a way to display info to Flight Director for faster information lookup and decision making process.

How:

1. Use an open-source application (streamlit.io) to help us quickly create our application.
2. Used Python to create a search-based user interface that outputs the wanted data.

Next Steps:

Seamlessly integrate the API into the application. Make the user interface (See Figure 5) as simple and easily readable as possible, for the flight directors to quickly find what they're looking for.

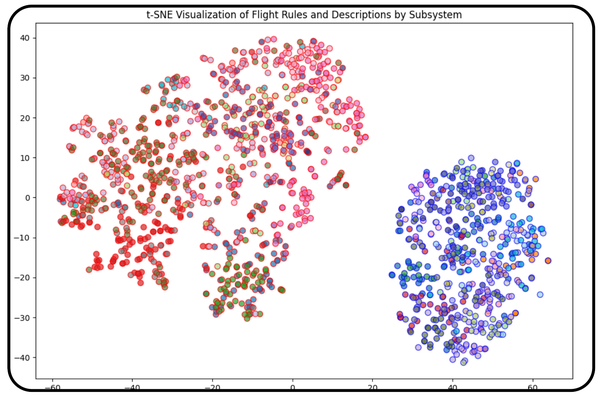

Figure 4: Subsystem & Flight Rules Clusters

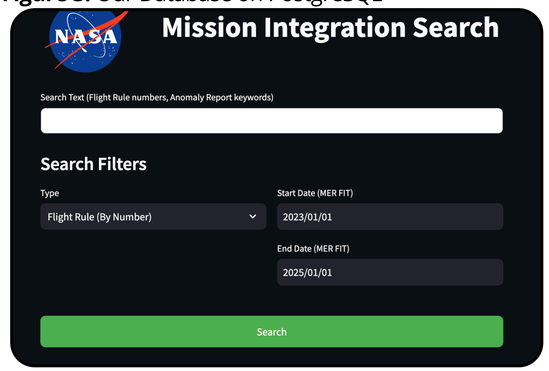

Figure 5: Sample UI for Search Engine

CONCLUSION

Conclusion:

Model built by data scientists using TF-IDF and logistic regression matches subsystems to components with an accuracy of 76%. Building this model will link the flight rules data to the anomalies data, to help flight firectors reach the right solutions. For now, the search engine is can keyword match the anomalies, a number match to flight rules, and match the dates of the MERFIT documents.

Acknowledgements:

Thanks to Data Mine staff and our mentor Elaine Ciaccio for their support and guidance throughout the semester.

Yazan Mahmoud (Teaching Assistant), Trisha Arun, Abhilasha Jain, Sreepranav Chetlapalli, Daniel Garcia, Conner Glass, Darshan Joshi, Kevin Liu, Yikai Liu, Soham Pradhan, Dominic Ramirez, Logan Searle, Harshil Shah, Ruhani Sharma, Javonna Stephens, Rishi Velma, Mary Voorhees, Logan Wongbangchuad