Mahima Rabbi
Student IT Technician, SNOLAB

# Bridging Data, Infrastructure & Research at SNOLAB

From Healthcare Data to Research Applications

# AGENDA

01 Background

02 Experience - SNOLAB

03 Experience – Previous Projects

04 Bridge between IT and Science

05 Prospective Contribution

06 Conclusion

## Focus

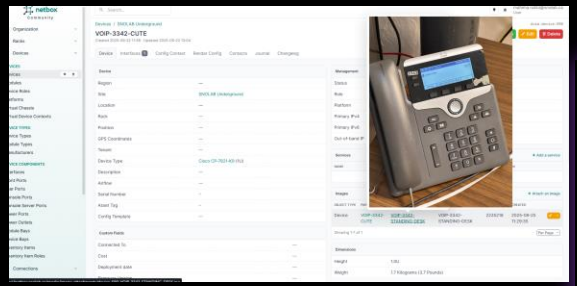
Data Analysis & Machine Learning
Complex Biomedical Data
AI-driven data processing
Building efficient data pipelines

## Core Strengths

Data cleaning & preprocessing at scale
Working with structured + unstructured datasets
Building automated pipelines
Applying ML models for prediction & analysis
Handling sensitive and distributed data
Report writing for Published Research work

# Experience @ SNOLAB

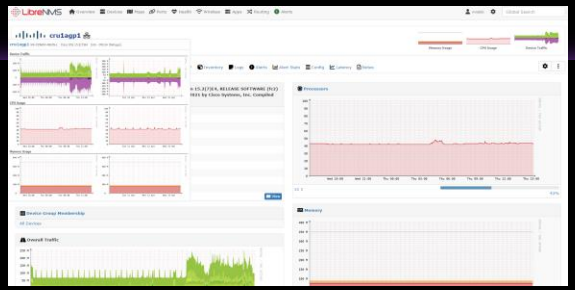
## 🖥️ Infrastructure Understanding




- Worked with **NetBox (IPAM)** to manage network data
- Maintained accurate mapping of **devices, subnets, and configurations**
- Learned how **structured infrastructure data** is critical at scale

👉 Key takeaway:
**Good data starts with well-organized systems**

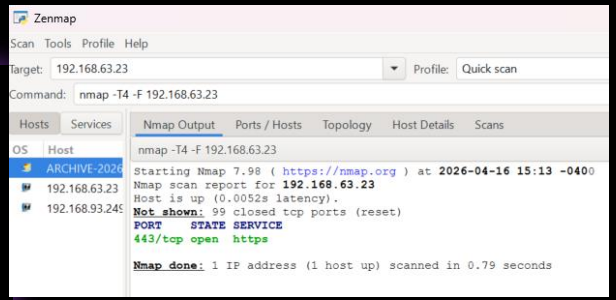
## 📊 Monitoring & System Visibility




- Used **LibreNMS** for real-time monitoring
- Tracked **performance, alerts, and uptime**
- Developed understanding of how systems behave over time

👉 Key takeaway:
Data is not static — it reflects system behavior

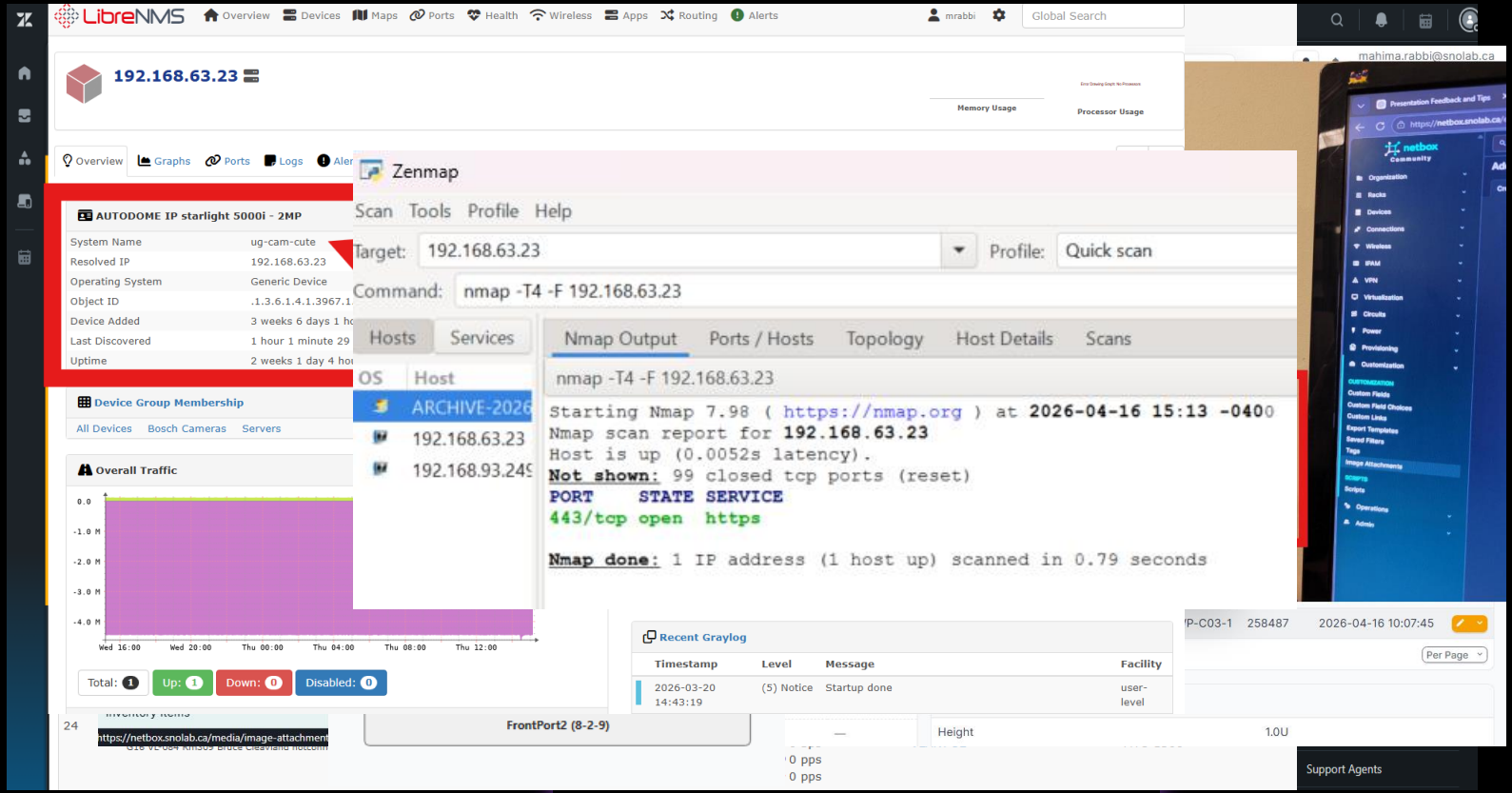
## 🔍 Data Integrity & Auditing




- Conducted **IP audits** to find inconsistencies
- Cleaned and validated infrastructure records
- Improved reliability of system data

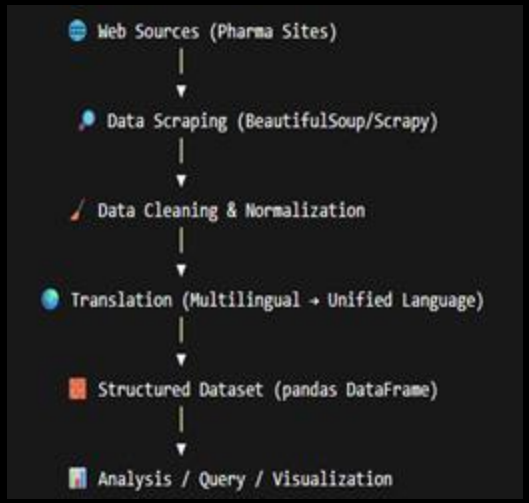
👉 Key takeaway:
**Small data errors → big system impact**

# Experience - SNOLAB



# Experience @ Laurentian University

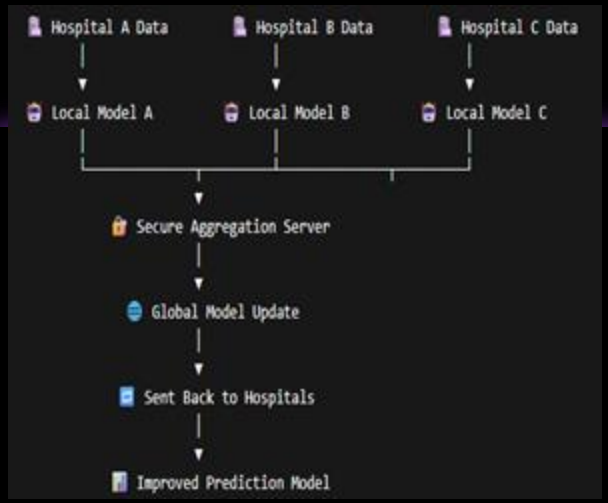
## Project 1:
## PharmaDB Data Pipeline




Make multilingual pharmaceutical data usable for analysis

Input: Raw, messy, multilingual data
Output: Clean, structured, analysis-ready dataset

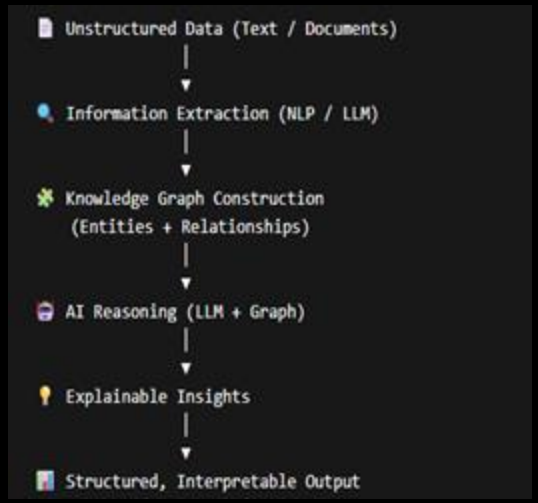
## Project 2:
## Federated Learning (Healthcare Data)




Predict patient readmission using distributed hospital data

Input: No raw data shared (Only model updates exchanged)
Output: Data-Privacy preserved trained model

## Project 3:
## AI + LLM + Knowledge Graphs




Improve interpretability of AI using structured data

Input: Unstructured data
Output: Explainable, structured knowledge

# Project 1: PharmaDB Data Pipeline

## What I Did

⟶ Enhanced a data scraping and processing pipeline

⟶ Extracted data from web sources

⟶ Cleaned and standardized inconsistent entries

⟶ Applied translation for multilingual datasets

## Tools & Technologies

⟶ Python

⟶ Scrapy

⟶ pandas, numpy

⟶ NLP / translation tools

⟶ Linux environment

## Data Transformation

⟶ Input

- Raw web data
- Multilingual, semi-structured text
- Inconsistent formatting

⟶ Processing

Parsing ⟶ Cleaning ⟶ Normalization ⟶ Translation

⟶ Output

- Structured dataset
- Consistent schema
- Analysis-ready format

## Analytical Impact

⟶ Enabled easier querying and filtering

⟶ Reduced manual preprocessing effort

⟶ Improved data reliability

Result: Faster and more accurate downstream analysis

# Project 2: Federated Learning (Healthcare Data)

## What I Did

⟶ Built a federated learning pipeline

⟶ Trained models across multiple datasets without centralizing data

⟶ Focused on privacy-preserving analysis

## Tools & Technologies

⟶ Python

⟶ TensorFlow / PyTorch

⟶ Flower

⟶ pandas, numpy

⟶ Classification models

## Data Transformation

⟶ Input

- Distributed patient datasets
- Sensitive healthcare records

⟶ Processing

- Local model training
- Aggregation of model updates
- Iterative model improvement

⟶ Output

- Predictive model for readmission risk
- No raw data sharing

## Analytical Impact

⟶ Identified patterns in patient readmission

⟶ Enabled analysis across multiple data sources

⟶ Maintained strict data privacy

Result: Scalable and privacy-aware data analysis

# Project 3: AI + +LLM + Knowledge Graphs

## What I Did

⟶ Transformed unstructured data into knowledge graphs

⟶ Integrated LLMs for reasoning

⟶ Designed explainable data workflows

## Tools & Technologies

⟶ Python

⟶ Graph-based data structures

⟶ NLP / LLM integration

⟶ Data modeling and querying

## Data Transformation

⟶ Input

- Unstructured datasets

⟶ Processing

- Structuring into graph format
- Applying AI models for reasoning

⟶ Output

- Interpretable insights
- Structured knowledge representation

## Analytical Impact

⟶ Turning raw data ⟶ structured, usable data

⟶ Building repeatable pipelines

⟶ Applying AI to reduce manual effort

⟶ Ensuring clarity and explainability in results

Result: Faster and more accurate downstream analysis

# Key Data Expertise Demonstrated

## Turning Raw Data → Structured, Usable Data

- → Handling unstructured / semi-structured inputs (HTML, text, logs)
- → Cleaning, normalizing, and transforming into tabular formats

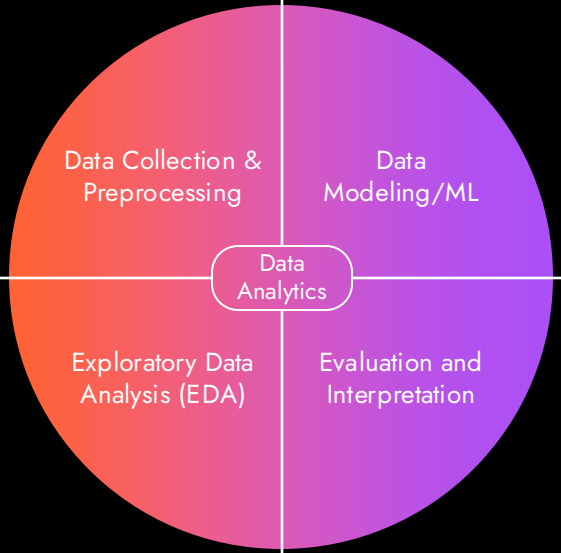
```
df = df.dropna()
df['drug_name'] = df['drug_name'].str.lower().str.strip()
df = df.drop_duplicates()
```

## Applying AI to Reduce Manual Effort

- → Using ML/NLP to automate complex tasks (classification, translation)
- → Reducing human intervention in large datasets

```
from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier()
model.fit(X_train, y_train)
predictions = model.predict(X_test)
```




```
def data_pipeline(url):
    raw = scrape_data(url)
    clean = clean_data(raw)
    structured = transform(clean)
    return structured
```

## Building Repeatable Data Pipelines

- → Designing step-by-step workflows: ingestion → processing → output
- → Automating pipelines for consistency and scalability

```
import networkx as nx

G = nx.Graph()
G.add_edge("Drug", "Disease")
```

## Ensuring Clarity & Explainability

- → Structuring outputs for interpretability
- → Using graphs / structured formats for better understanding

# Relevance to Scientific Data at SNOLAB

## Similar Data Characteristics

⟶ Large-scale, high-volume datasets

⟶ Noisy, incomplete, or unstructured raw data

⟶ Data requires preprocessing before meaningful analysis

## Transferable Data Techniques

⟶ Data cleaning & normalization (handling inconsistencies, missing values)

⟶ Pipeline automation (reducing manual preprocessing steps)

⟶ Pattern detection using ML models

⟶ Structuring data for efficient querying and analysis

## Practical Application in Research Workflows

⟶ Preparing raw detector data for analysis

⟶ Filtering noise and irrelevant signals

⟶ Automating repetitive preprocessing tasks

⟶ Supporting faster and more reliable downstream analysis

# Bridging IT Infrastructure & Scientific Data Workflows at SNOLAB

## IT Capabilities

🖥️ Servers | Storage | Network

⟶ Systems, storage, and networking

⟶ Data access & infrastructure support

⟶ Automation of operational processes

## Data Bridge

⚙️ Data Pipelines | 🤖 ML | 🧹 Cleaning

⟶ Build data pipelines (ingestion → cleaning → structured output)

⟶ Automate repetitive preprocessing tasks

⟶ Apply ML for pattern detection & filtering

⟶ Enable efficient, analysis-ready datasets

## Research Needs

🔬 Raw Data → 📊 Insights

⟶ Large-scale experimental datasets

⟶ Data cleaning & preprocessing

⟶ Noise filtering & structuring

⟶ Analysis-ready data preparation

# Prospective Contribution: Enabling Data Support Through IT

## What I Propose

→ Researchers can raise simple **data support requests (tickets)**

→ **Focus on:**
- Data preprocessing
- Data cleaning & formatting
- Repetitive data tasks

→ IT can support by:
- Automating small workflows
- Creating reusable scripts/tools
- Standardizing data formats

## How This Helps

→ Reduces manual, repetitive work for researchers
→ Speeds up data preparation before analysis
→ Improves consistency and data quality
→ Enables more efficient research workflows

## IT Role

→ Act as a bridge between:
- IT systems
- Data workflows
- Research needs

→ Identify common repetitive tasks
→ Build simple, practical data solutions
→ Support researchers with structured, usable data


https://github.com/Mahima-Rabbi-Mohi/Parkinson-Detection-Web-App

# THANK YOU

Any questions?