AI Developer Assignment – SR Repeat Pattern Analysis Report

1. Executive Summary

This report presents the analysis and AI-driven approach to detect and isolate repeat patterns in Service Request (SR) data. The study leverages historical resolution details, issue classification, and clustering using HDBSCAN to uncover recurring service issues, customer-level patterns, and insights for proactive maintenance.

2. Objective

To identify and group repeated or similar issues in the SR dataset using open-source AI/ML tools, providing actionable insights that can enhance automation, reduce resolution time, and improve customer experience.

3. Methodology

Data Preprocessing

- Performed EDA to understand data structure and column-level imbalance.
- Standardized and cleaned text data from SRSUMMARY, X_RESOLUTION_COMMENT, and AUTOMATION_RCA_CONCLUSION_TEXT.
- Created combined text embeddings using Sentence Transformers.
- Derived new features such as Duration_hours, Month, and SLA breach flags.

Clustering Approach

Implemented HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) to group similar SRs based on text embeddings.

Why **HDBSCAN**?
- Does not require a pre-defining number of clusters (K).
- Automatically identifies noise/outliers (-1).
- Works well for dense text embeddings.
- Provides well-separated clusters.

Implementation Example:

```
import hdbscan
clusterer = hdbscan.HDBSCAN(min_cluster_size=3, metric='euclidean')
df['cluster'] = clusterer.fit_predict(embeddings)
```

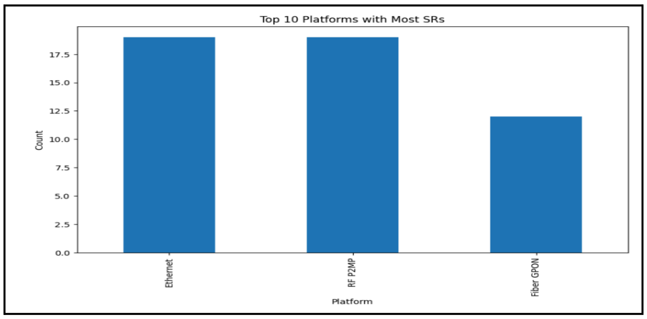
4. Key Observations

4.1 Dataset Imbalance

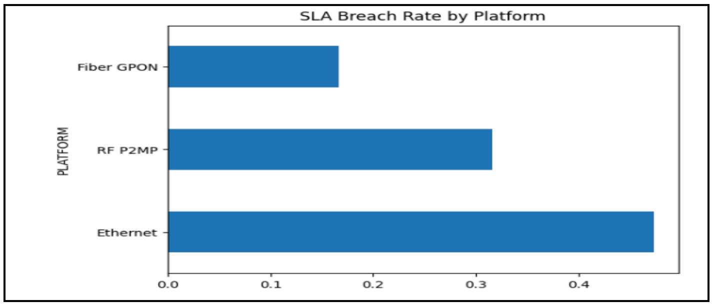
Columns such as ATTRIBUTEDTO, IMPACT, SEVERITY, and CASETYPE contained only one category each — providing **no analytical variance**.

4.2 Platform-Level Insights

Ethernet and RFP2Mp platforms recorded the highest SR volume.
45% of SLA breaches occurred on Ethernet-related platforms.
Fiber GPON contributed to ~15% of SLA breaches.
[SR Count by Platform]

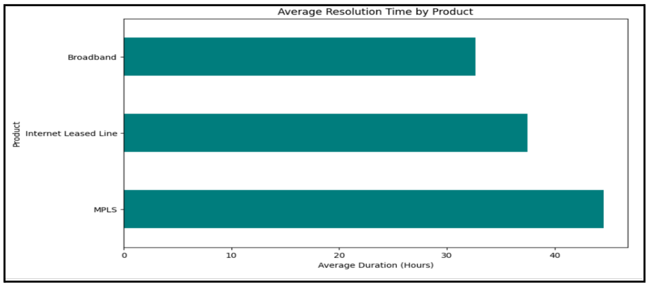


[SLA Breach % by Platform]

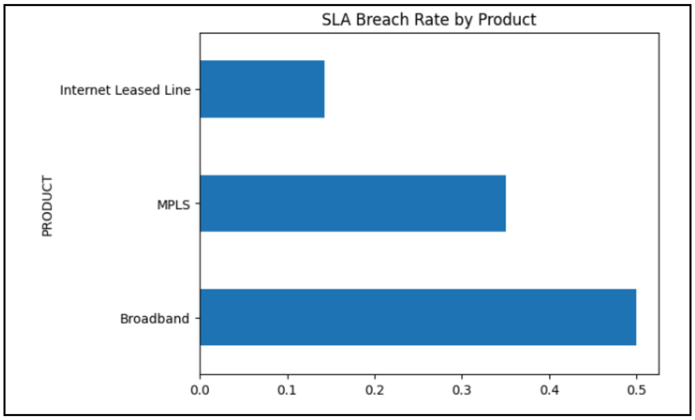


4.3 Product-Level Insights

MPLS (Multiprotocol Label Switching) and Broadband products showed the highest SR recurrence.
MPLS had an average resolution time of 40+ hours, higher than Leased Lines and Broadband.
SLA Breach Distribution:
- Broadband: ~45% **<-->** MPLS: ~35% & Leased Lines: ~15%
[Avg Resolution Time by Product]

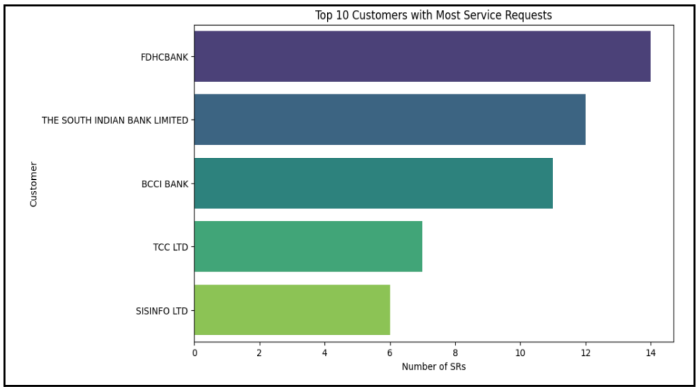


[SLA Breach % by Product]

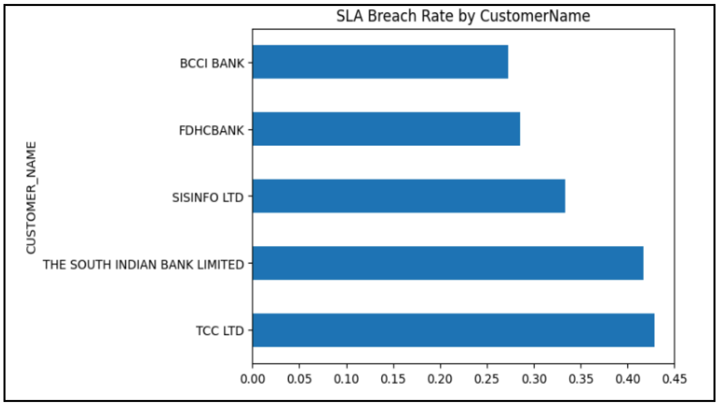


4.4 Customer-Level Insights

FDHC, SIB, and BCCI customers reported 10+ SRs each, signaling repeat issue patterns. TCC and SIB customers had ~40–45% SLA breaches, while BCCI Bank was lower at ~27%.
[SR Count by Customer]

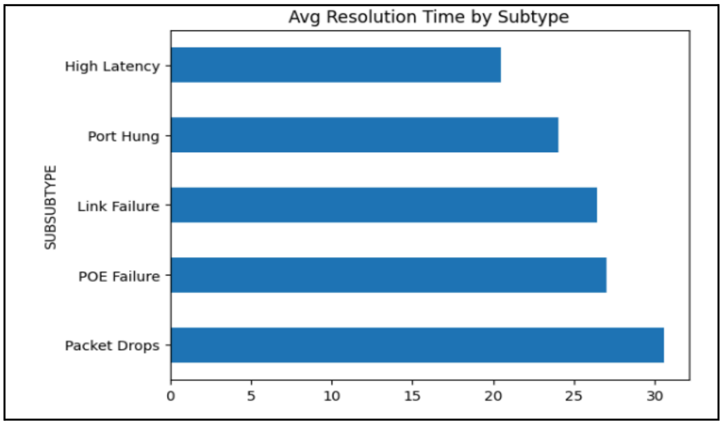


[SLA Breach % by Customer]



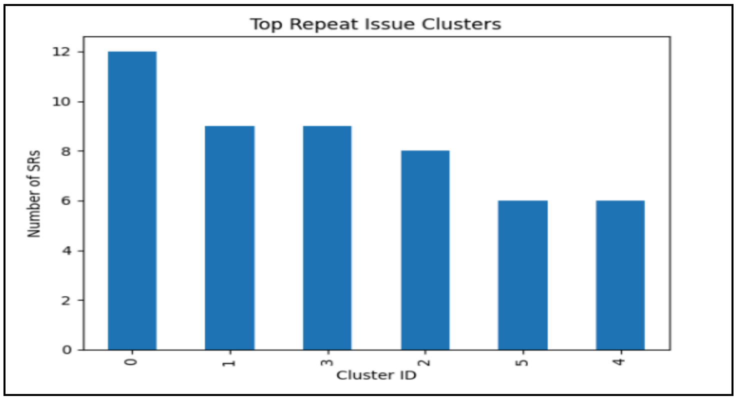
4.5 Subtype Insights

Packet Drops issues took >30 hours on average to resolve.
High Latency SRs were resolved faster.
[Avg Resolution Time by SR Subtype]

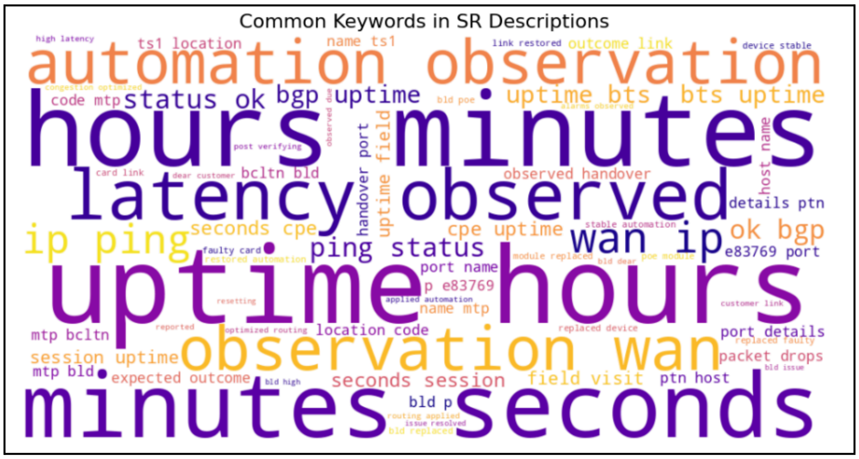


4.6 Cluster Insights

Cluster quality showed good separation.
Cluster 0 dominated with POE-related automation and stability issues.
[Cluster Distribution (HDBSCAN)]



[Cluster Word Cloud / Common Keywords]



5. Business Impact & Recommendations

Operational Value

- Clear identification of repetitive patterns enables automation.
- High recurrence in Ethernet and MPLS indicates hardware stability challenges.
- Enables predictive maintenance and automated RCA tagging.

Strategic Recommendations

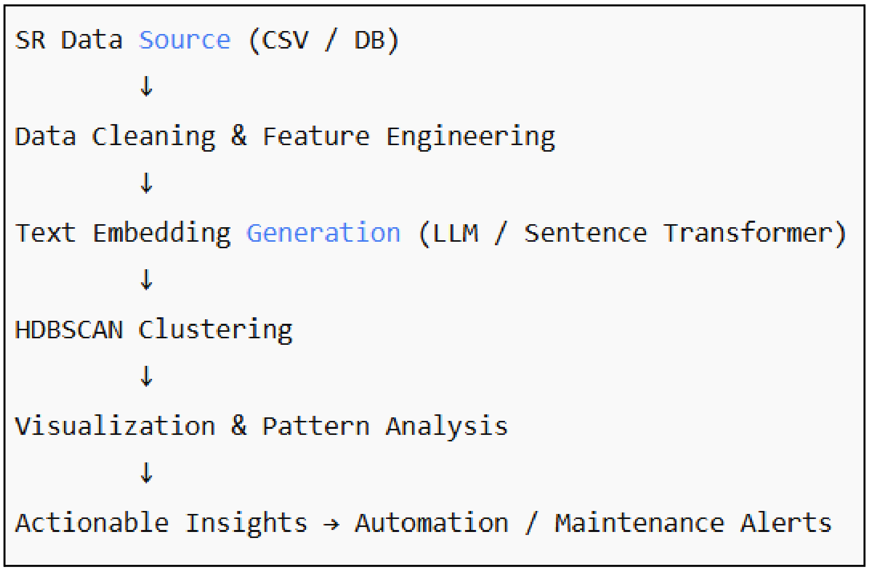
- Integrate this pipeline with live SR systems for real-time similarity detection.
- Automate ticket tagging and routing using top cluster descriptions.
- Deploy LLM summarization models to link new SRs to historical clusters dynamically.
- Establish preventive maintenance triggers for top recurring platforms.

6. Tool & Technology Stack

Component	Tool / Library Used
Data Processing	Python (Pandas, NumPy)
Text Embeddings	Sentence Transformers
Clustering	HDBSCAN
Visualization	Matplotlib, Seaborn
Automation / Reporting	Jupyter Notebook, Google Docs
Extension	Streamlit Dashboard

7. Solution Architecture

[Logical Architecture Diagram]



```
SR Data Source (CSV / DB)
        ↓
Data Cleaning & Feature Engineering
        ↓
Text Embedding Generation (LLM / Sentence Transformer)
        ↓
HDBSCAN Clustering
        ↓
Visualization & Pattern Analysis
        ↓
Actionable Insights → Automation / Maintenance Alerts
```

SR Data Source (CSV / DB) → Data Cleaning → Embedding Generation → HDBSCAN Clustering → Visualization → Actionable Insights

8. Conclusion

The pipeline successfully detects and isolates repeat SR patterns using AI/ML clustering. Findings reveal recurring problem categories across platforms, customers, and products — forming the foundation for proactive maintenance, automated RCA, and reduced resolution time.