AI-Powered Data Profiling & Compliance Assistant

- Sharanprasath S: CT (HLT-Originations)

1. Introduction

1.1 Overview

The AI-Powered Data Profiling & Compliance Assistant automates the extraction of validation rules from regulatory documents, validates datasets against these rules, detects anomalies using machine learning, and provides an interactive compliance assistant. This system is designed to streamline regulatory compliance and data accuracy in financial and enterprise datasets.

1.2 Problem Statement

Regulatory compliance in financial datasets is complex, requiring manual effort to extract rules, validate data, and detect inconsistencies. This leads to inefficiencies and increased risk of errors. Our system automates this process using AI and ML to improve accuracy, efficiency, and compliance.

1.3 Key Features

- **Automated Rule Extraction**: Extracts validation rules from regulatory PDFs using Generative AI.
- **Dataset Validation**: Applies extracted rules to detect inconsistencies in uploaded datasets.
- **Anomaly Detection**: Identifies outliers using Isolation Forest and other ML techniques.
- **AI-Powered Compliance Assistant**: Provides explanations for violations and suggests rule refinements.

---

2. Technical Approach

2.1 Rule Extraction (AI-Powered Regulatory Rule Extraction)

Process:

- **Upload regulatory PDF and sample dataset (CSV).**
- **Text & Table Extraction:** Extracts structured and unstructured data from the document.
- **Chunking & Vector Storage (RAG Approach):** Uses text chunking and ChromaDB for storage.
- **Rule Extraction using Google Gemini AI:** AI processes extracted content and generates structured validation rules.
- **JSON Rule Output:** Stores extracted rules in JSON format for dataset validation.

Detailed Technical Flow:

1. **PDF Processing:**
    - Uses **pdfplumber** or **PyPDFLoader** to extract both text and table data separately.

    - **Pre-processing:** Removes unnecessary content, identifies structured sections, and cleans extracted text.

2. **Chunking & Vector Storage (RAG Approach):**
    - Splits text into **semantic chunks** using **RecursiveCharacterTextSplitter**.
    - Uses **GoogleGenerativeAIEmbeddings** to generate vector embeddings.
    - Stores indexed chunks in **ChromaDB** for efficient retrieval.
3. **AI-Driven Rule Extraction:**
    - Sends extracted chunks to **Google Gemini API**.
    - AI generates **validation rules** based on dataset schema and extracted content.
    - Rules include **column constraints, expected formats, data types, and business rules**.
4. **JSON Output & Storage:**
    - Outputs structured rules in **JSON format**.
    - Users can **download extracted rules** for validation purposes.

2.2 Dataset Validation (Automated Compliance Checking)

Process:

- **Upload dataset (CSV) and validation rules (JSON).**
- **Batch Processing for Efficiency:** Splits dataset into smaller batches for validation.
- **AI-Powered Validation using Google Gemini:** Checks data integrity, constraints, and expected formats.
- **Validation Report Generation:** Generates structured JSON & CSV reports highlighting violations.

Detailed Technical Flow:

1. **Dataset & Rule Ingestion:**
    - Reads dataset using **Pandas**.
    - Loads extracted validation rules from JSON.
2. **Batch Processing for Efficiency:**
    - **Reduces API calls** by splitting dataset into **smaller batches**.
    - Sends batches to **Google Gemini AI** for rule-based validation.
3. **AI-Powered Validation:**
    - AI **checks each column** for **format, constraints, expected values, and missing data**.
    - Identifies **violations** and suggests fixes.

    - Ensures **logical consistency** (e.g., no negative loan amounts, proper date formats).

4. **Real-time UI & Reporting:**
    - **Progress bar updates dynamically** per batch.
    - AI-generated violations and suggestions are displayed in a structured format.
    - Users can **download the validation report (CSV)** for further analysis.

2.3 Compliance Chat Assistant (Interactive AI-Powered Queries)

Process:

- **User Query Processing:** Users ask about violations or request rule refinements.
- **Retrieval of Relevant Context:** Uses FAISS vector database for retrieving violations.
- **AI Response Generation using Google Gemini API:** Provides structured explanations and rule suggestions.
- **Interactive Chat UI:** Allows users to iteratively refine compliance queries.

Detailed Technical Flow:

1. **User Query Processing:**
    - Accepts **natural language queries** from users.
    - Determines intent (e.g., explanation request, rule refinement, compliance guidance).
2. **Retrieval-Augmented Search:**
    - Stores violations in **FAISS (vector database)**.
    - Retrieves **most relevant validation errors** for the query.
3. **AI Response Generation:**
    - AI processes retrieved violations and **generates structured explanations**.
    - If rule refinement is requested, AI **suggests modifications to the extracted rules**.
4. **Interactive Chat UI:**
    - Displays **user messages in chat bubbles**.
    - **Dynamically updates responses**, allowing **iterative compliance improvements**.

2.4 Anomaly Detection (Machine Learning-Based Outlier Detection)

Process:

- **Feature Selection & Preprocessing:** Selects numeric columns and standardizes features.
- **Model-Based Anomaly Detection:** Uses Isolation Forest for anomaly detection.

- **Auto-Calculated Contamination Rate:** AI estimates the proportion of anomalies dynamically.
- **Interactive UI & Visualization:** Scatter plots highlight anomalies; users can download flagged data.

Detailed Technical Flow:

1. **Feature Selection & Preprocessing:**
    - Selects **only numeric columns** for anomaly detection.
    - Standardizes features using **StandardScaler**.
2. **Model-Based Anomaly Detection:**
    - Uses **Isolation Forest** for anomaly detection (default model).
    - Allows switching to:
        - **Local Outlier Factor (LOF)** → Density-based anomaly detection.
        - **DBSCAN** → Clustering-based anomaly detection.
3. **Auto-Calculated Contamination Rate:**
    - Instead of manually setting the contamination rate, it is estimated using a hybrid approach:
    - Isolation Forest-based Scoring: Model initially assumes a higher contamination rate.
    - Z-Score Method: Identifies anomalies based on statistical deviations (2 standard deviations below mean).
    - Percentile-Based Approach: Flags the lowest 2% of anomaly scores as outliers.
    - Final Contamination Rate: The maximum value between the Z-Score and Percentile-Based Approach is chosen.
    - Ensures Stability: The contamination rate is clamped between 0.01 and 0.15 to avoid overfitting or under-detection.
4. **Interactive UI & Visualization:**
    - Generates **scatter plot with color-coded anomalies** (red for outliers).
    - Users can **download flagged anomalies (CSV) for further analysis**.

---

3. Implementation Details

3.1 Libraries & Technologies Used

- **Backend:** Python, LangChain, Pandas
- **Frontend:** Streamlit, Plotly, Altair
- **AI Models:** Google Gemini (LLM), Isolation Forest (ML)

- **Database:** ChromaDB (Vector Storage)
- **Deployment:** Streamlit Cloud, GitHub Actions

3.2 Optimization Strategies

- **Batch Processing:** Reduces API requests and improves validation efficiency.
- **Vector Search (FAISS):** Ensures fast and relevant retrieval of validation rules.
- **Dynamic Contamination Rate:** Improves anomaly detection accuracy.
- **Streamlit UI Enhancements:** Enhances user interaction with collapsible reports and real-time updates.

---

4. Challenges & Solutions

4.1 Handling Large PDFs

- **Challenge:** Extracting meaningful rules from complex regulatory documents.
- **Solution:** Used **ChromaDB for vector storage** and **chunk-based processing**.

4.2 Reducing API Usage & Cost

- **Challenge:** Generative AI models can be costly with frequent API calls.
- **Solution: Batch validation & caching** reduced unnecessary API calls.

4.3 Ensuring JSON Response Integrity

- **Challenge:** AI-generated responses sometimes contained incomplete JSON.
- **Solution:** Implemented **parsing checks & regex fixes** to ensure valid JSON.

4.4 Aligning Rule Extraction with Validation Logic

- **Challenge:** Extracted rules sometimes lacked dataset alignment.
- **Solution: Fine-tuned AI prompts** to improve rule relevance.

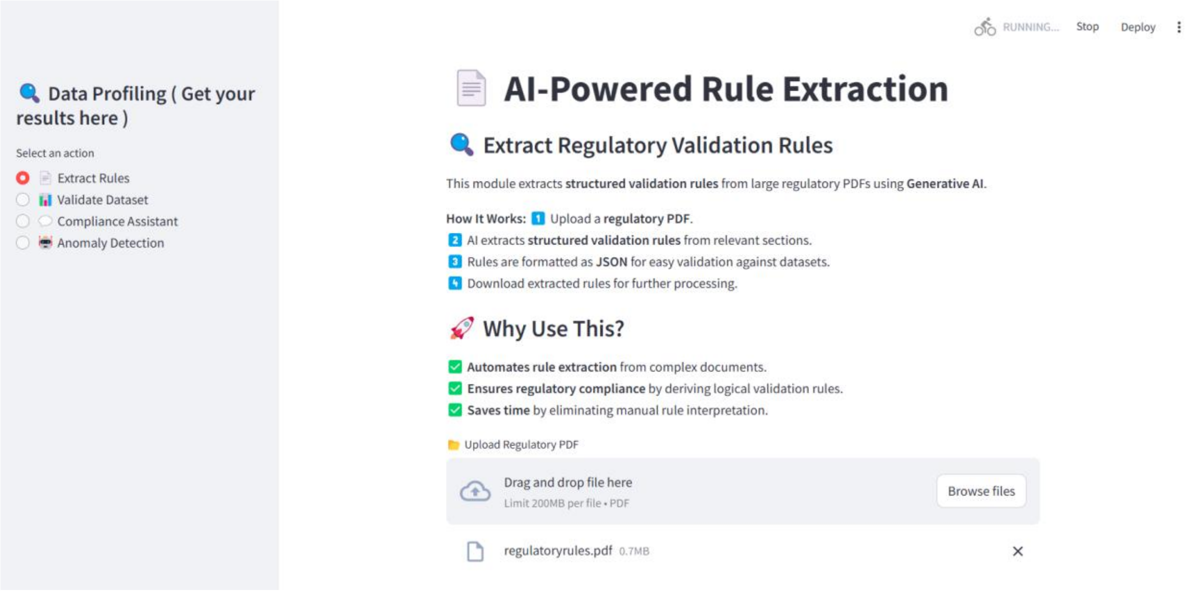
4.5 Making UI More Interactive

- **Challenge:** Users needed a clearer way to track validation & anomalies.
- **Solution:** Used **dynamic progress bars, collapsible reports, and color-coded chat bubbles**.

---

5. Demo

5.1 Rule Extraction (AI-Powered Regulatory Rule Extraction)



AI-Powered Rule Extraction

Extract Regulatory Validation Rules

This module extracts **structured validation rules** from large regulatory PDFs using **Generative AI**.

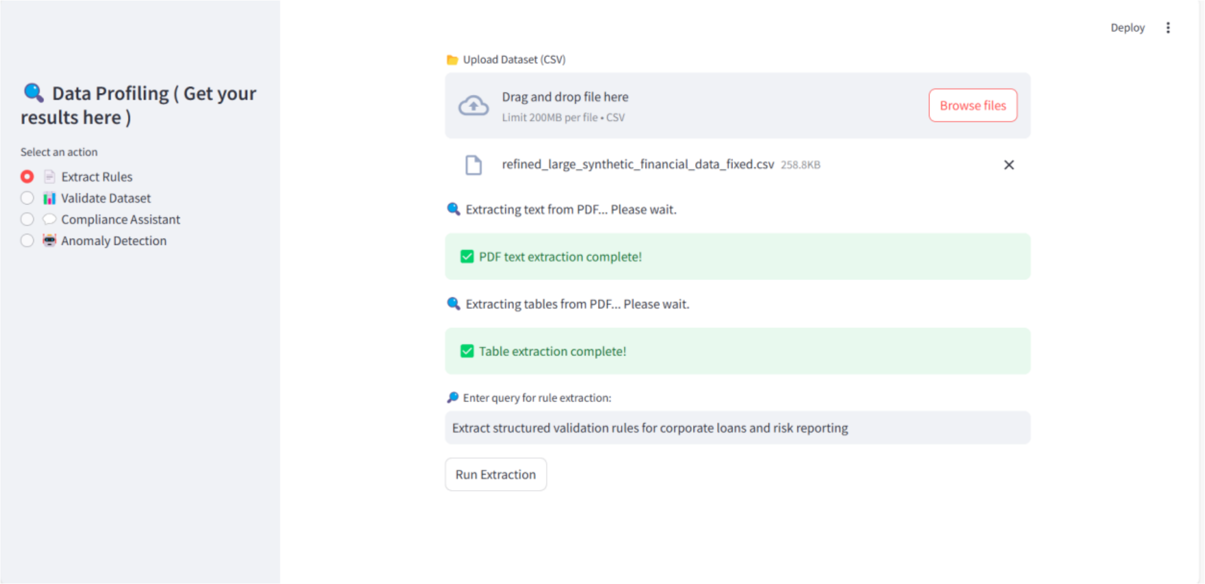
How It Works: 1 Upload a **regulatory PDF**.
2 AI extracts **structured validation rules** from relevant sections.
3 Rules are formatted as **JSON** for easy validation against datasets.
4 Download extracted rules for further processing.

Why Use This?

✓ **Automates rule extraction** from complex documents.
✓ **Ensures regulatory compliance** by deriving logical validation rules.
✓ **Saves time** by eliminating manual rule interpretation.

Upload Regulatory PDF

regulatoryrules.pdf 0.7MB



Upload Dataset (CSV)

refined_large_synthetic_financial_data_fixed.csv 258.8KB

Extracting text from PDF... Please wait.

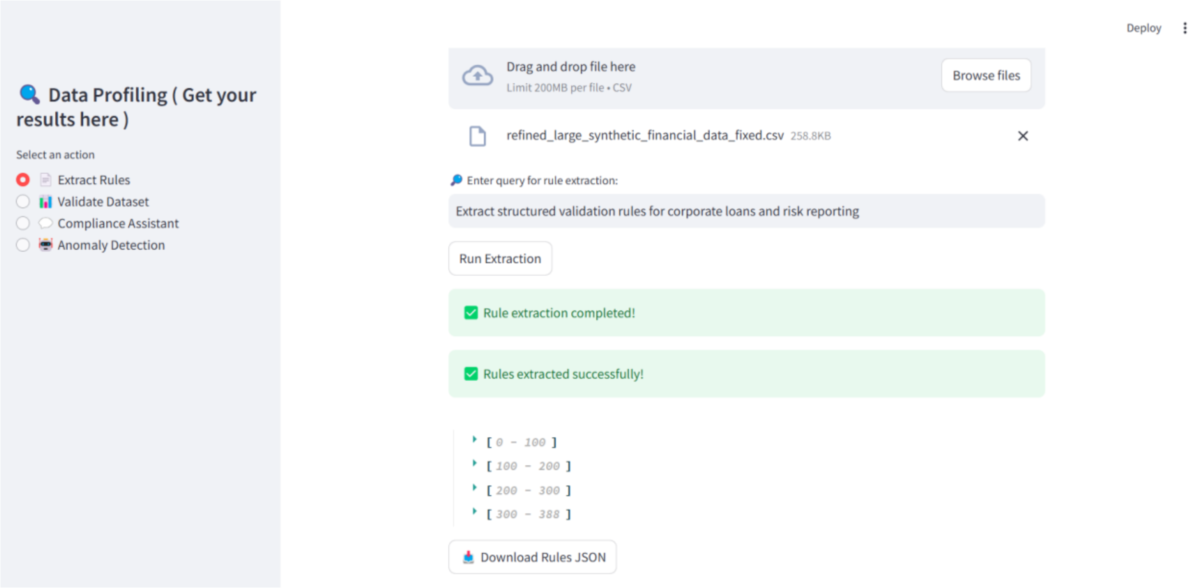
✓ PDF text extraction complete!

Extracting tables from PDF... Please wait.

✓ Table extraction complete!

Enter query for rule extraction:

Extract structured validation rules for corporate loans and risk reporting



refined_large_synthetic_financial_data_fixed.csv 258.8KB

Enter query for rule extraction:

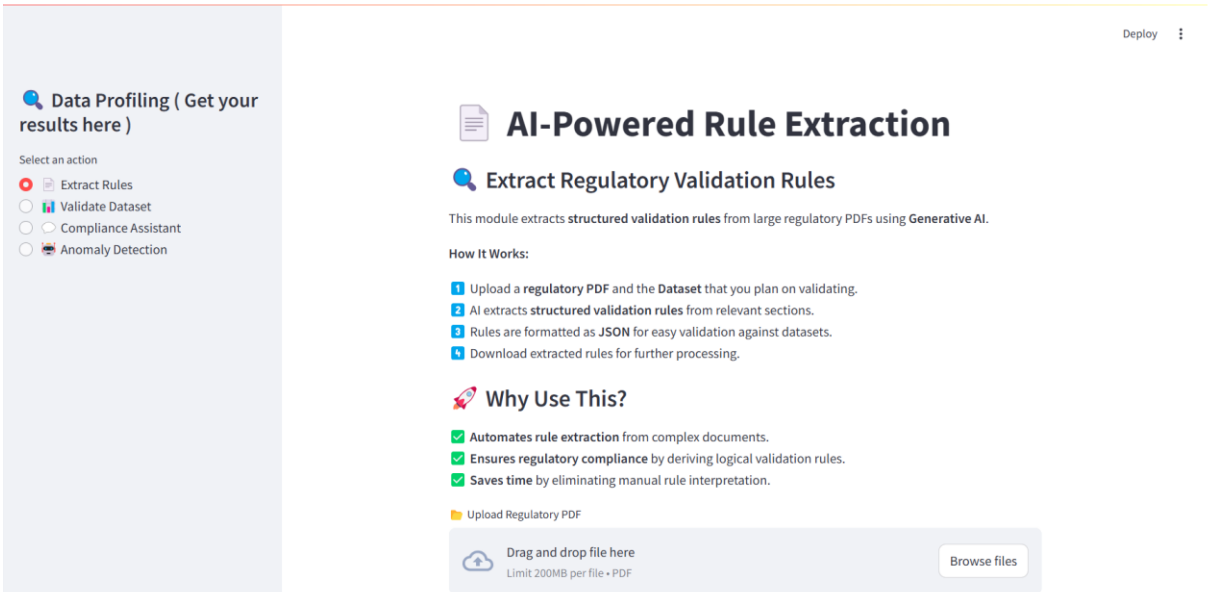
Extract structured validation rules for corporate loans and risk reporting

✓ Rule extraction completed!

✓ Rules extracted successfully!

```
▸ [ 0 - 100 ]
▸ [ 100 - 200 ]
▸ [ 200 - 300 ]
▸ [ 300 - 388 ]
```

5.2 Dataset Validation (Automated Compliance Checking)



AI-Powered Rule Extraction

🔍 Extract Regulatory Validation Rules

This module extracts **structured validation rules** from large regulatory PDFs using **Generative AI**.

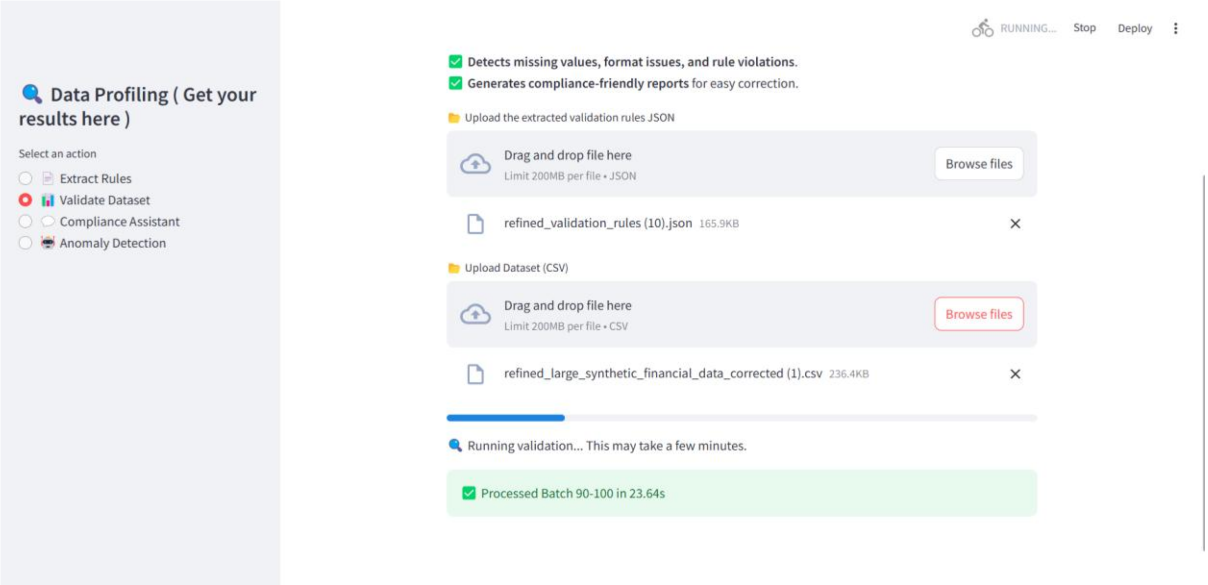
How It Works:

1 Upload a **regulatory PDF** and the **Dataset** that you plan on validating.
2 AI extracts **structured validation rules** from relevant sections.
3 Rules are formatted as **JSON** for easy validation against datasets.
4 Download extracted rules for further processing.

🚀 Why Use This?

- ✓ **Automates rule extraction** from complex documents.
- ✓ **Ensures regulatory compliance** by deriving logical validation rules.
- ✓ **Saves time** by eliminating manual rule interpretation.

📁 Upload Regulatory PDF



- ✓ **Detects missing values, format issues, and rule violations.**
- ✓ **Generates compliance-friendly reports** for easy correction.

📁 Upload the extracted validation rules JSON

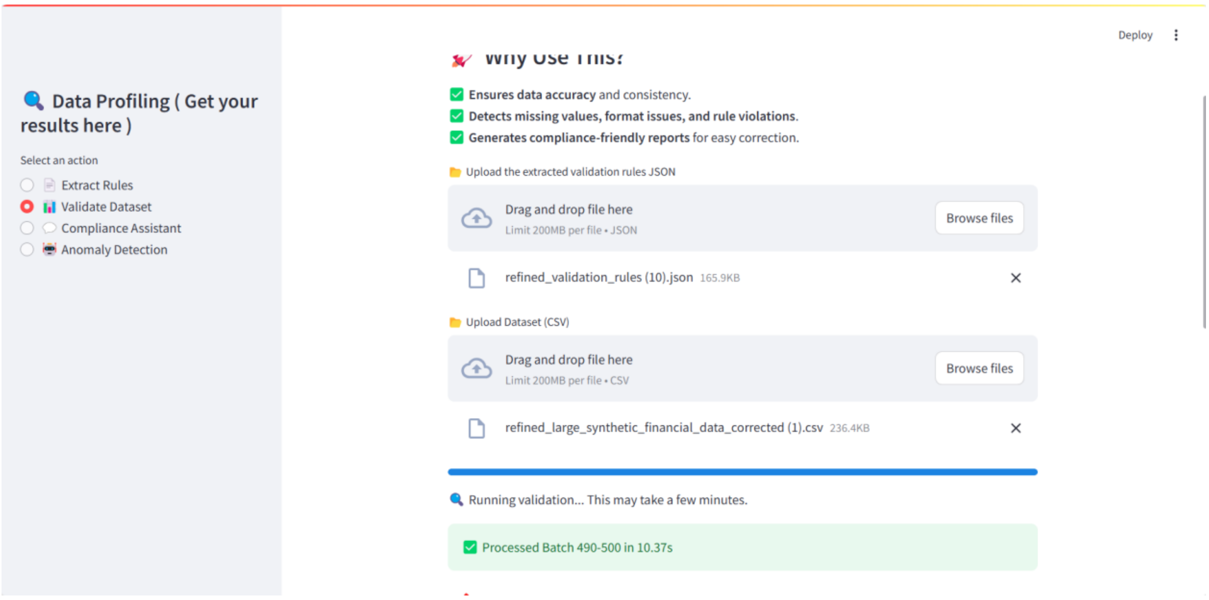
refined_validation_rules (10).json 165.9KB

📁 Upload Dataset (CSV)

refined_large_synthetic_financial_data_corrected (1).csv 236.4KB

🔍 Running validation... This may take a few minutes.

✓ Processed Batch 90-100 in 23.64s



🔍 Data Profiling (Get your results here)

Select an action

○ Extract Rules
● Validate Dataset
○ Compliance Assistant
○ Anomaly Detection

Why Use This?

- ✓ **Ensures data accuracy** and consistency.
- ✓ **Detects missing values, format issues, and rule violations.**
- ✓ **Generates compliance-friendly reports** for easy correction.

Upload the extracted validation rules JSON

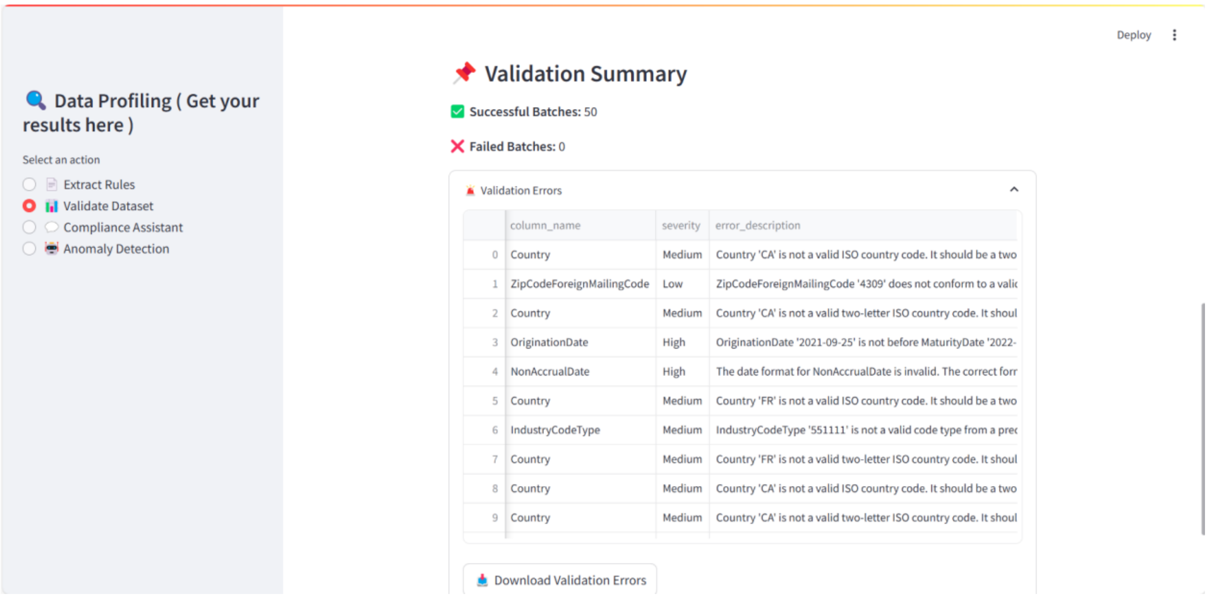
refined_validation_rules (10).json 165.9KB

Upload Dataset (CSV)

refined_large_synthetic_financial_data_corrected (1).csv 236.4KB

🔍 Running validation... This may take a few minutes.

✓ Processed Batch 490-500 in 10.37s



🔍 Data Profiling (Get your results here)

Select an action

○ Extract Rules
● Validate Dataset
○ Compliance Assistant
○ Anomaly Detection

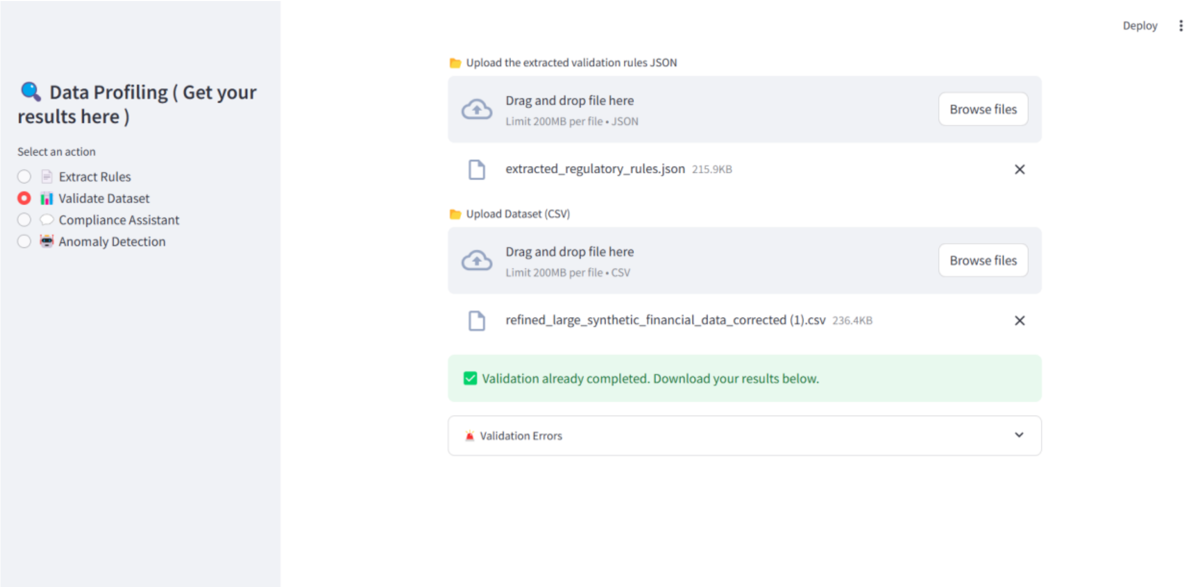
📌 Validation Summary

✓ **Successful Batches:** 50

✗ **Failed Batches:** 0

Validation Errors

	column_name	severity	error_description
0	Country	Medium	Country 'CA' is not a valid ISO country code. It should be a two
1	ZipCodeForeignMailingCode	Low	ZipCodeForeignMailingCode '4309' does not conform to a vali
2	Country	Medium	Country 'CA' is not a valid two-letter ISO country code. It shoul
3	OriginationDate	High	OriginationDate '2021-09-25' is not before MaturityDate '2022-
4	NonAccrualDate	High	The date format for NonAccrualDate is invalid. The correct for
5	Country	Medium	Country 'FR' is not a valid ISO country code. It should be a two
6	IndustryCodeType	Medium	IndustryCodeType '551111' is not a valid code type from a pre
7	Country	Medium	Country 'FR' is not a valid two-letter ISO country code. It shoul
8	Country	Medium	Country 'CA' is not a valid ISO country code. It should be a two
9	Country	Medium	Country 'CA' is not a valid two-letter ISO country code. It shoul



🔍 Data Profiling (Get your results here)

Select an action

○ Extract Rules
● Validate Dataset
○ Compliance Assistant
○ Anomaly Detection

Upload the extracted validation rules JSON

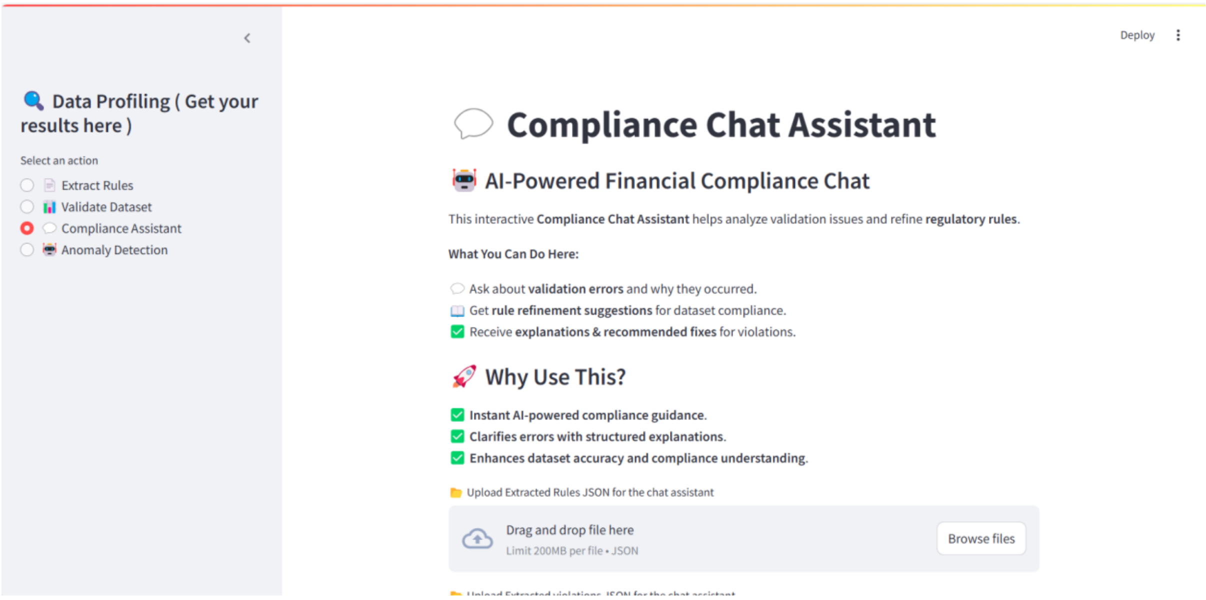
extracted_regulatory_rules.json 215.9KB

Upload Dataset (CSV)

refined_large_synthetic_financial_data_corrected (1).csv 236.4KB

✓ Validation already completed. Download your results below.

5.3 Compliance Chat Assistant (Interactive AI-Powered Queries)



🔍 Data Profiling (Get your results here)

Select an action

○ Extract Rules
○ Validate Dataset
● Compliance Assistant
○ Anomaly Detection

💬 Compliance Chat Assistant

🤖 AI-Powered Financial Compliance Chat

This interactive **Compliance Chat Assistant** helps analyze validation issues and refine **regulatory rules**.

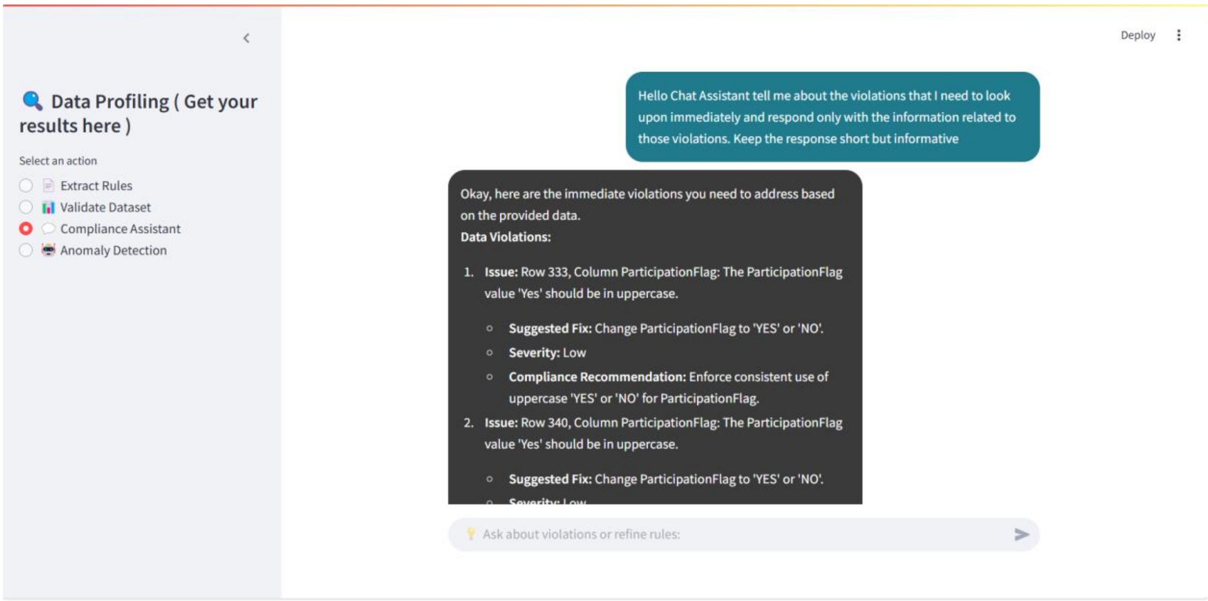
What You Can Do Here:

💬 Ask about **validation errors** and why they occurred.
📖 Get **rule refinement suggestions** for dataset compliance.
✅ Receive **explanations & recommended fixes** for violations.

🚀 Why Use This?

✅ **Instant AI-powered compliance guidance.**
✅ **Clarifies errors with structured explanations.**
✅ **Enhances dataset accuracy and compliance understanding.**

📁 Upload Extracted Rules JSON for the chat assistant



🔍 Data Profiling (Get your results here)

Select an action

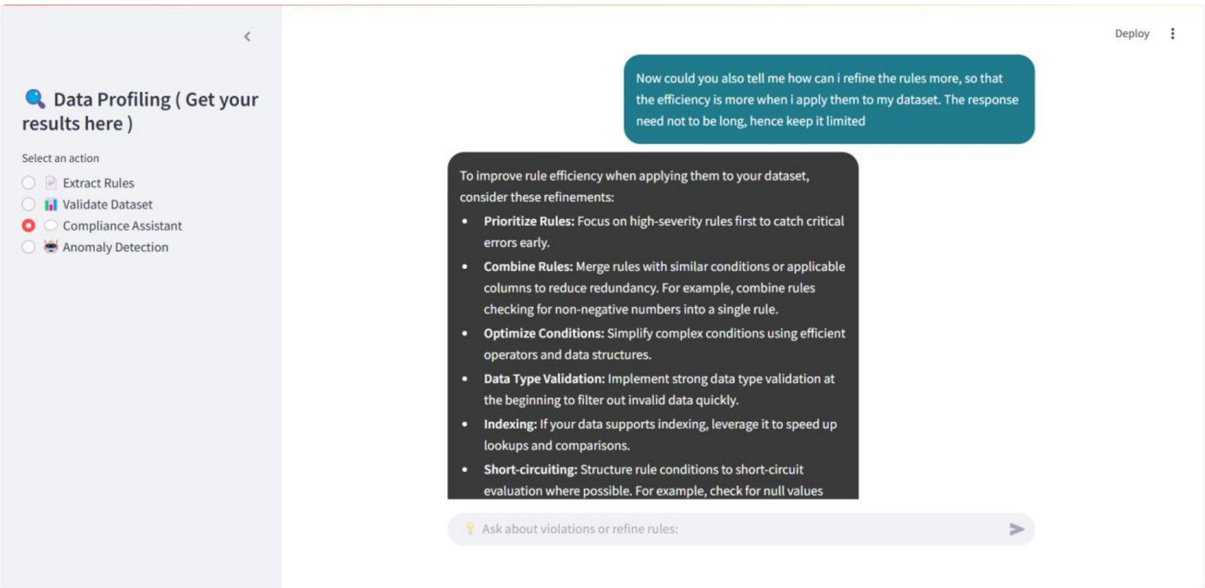
○ Extract Rules
○ Validate Dataset
● Compliance Assistant
○ Anomaly Detection

Hello Chat Assistant tell me about the violations that I need to look upon immediately and respond only with the information related to those violations. Keep the response short but informative

Okay, here are the immediate violations you need to address based on the provided data.

Data Violations:

1. **Issue:** Row 333, Column ParticipationFlag: The ParticipationFlag value 'Yes' should be in uppercase.
   - **Suggested Fix:** Change ParticipationFlag to 'YES' or 'NO'.
   - **Severity:** Low
   - **Compliance Recommendation:** Enforce consistent use of uppercase 'YES' or 'NO' for ParticipationFlag.
2. **Issue:** Row 340, Column ParticipationFlag: The ParticipationFlag value 'Yes' should be in uppercase.
   - **Suggested Fix:** Change ParticipationFlag to 'YES' or 'NO'.

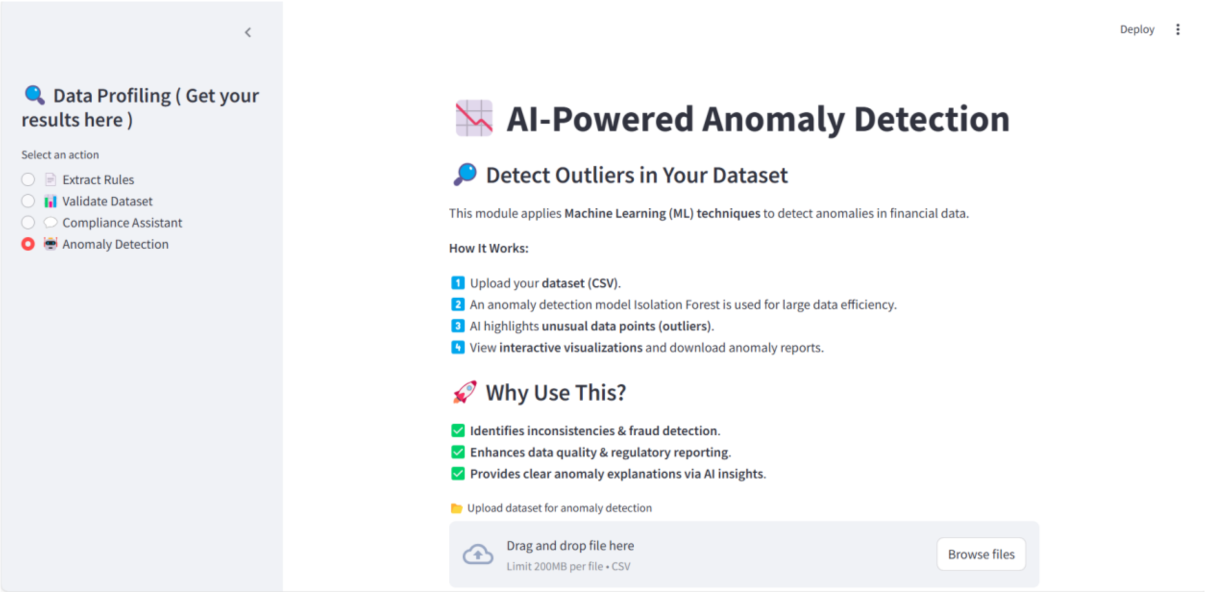


Now could you also tell me how can i refine the rules more, so that the efficiency is more when i apply them to my dataset. The response need not to be long, hence keep it limited

To improve rule efficiency when applying them to your dataset, consider these refinements:

- **Prioritize Rules:** Focus on high-severity rules first to catch critical errors early.
- **Combine Rules:** Merge rules with similar conditions or applicable columns to reduce redundancy. For example, combine rules checking for non-negative numbers into a single rule.
- **Optimize Conditions:** Simplify complex conditions using efficient operators and data structures.
- **Data Type Validation:** Implement strong data type validation at the beginning to filter out invalid data quickly.
- **Indexing:** If your data supports indexing, leverage it to speed up lookups and comparisons.
- **Short-circuiting:** Structure rule conditions to short-circuit evaluation where possible. For example, check for null values

5.4 Anomaly Detection (Machine Learning-Based Outlier Detection)



AI-Powered Anomaly Detection

Detect Outliers in Your Dataset

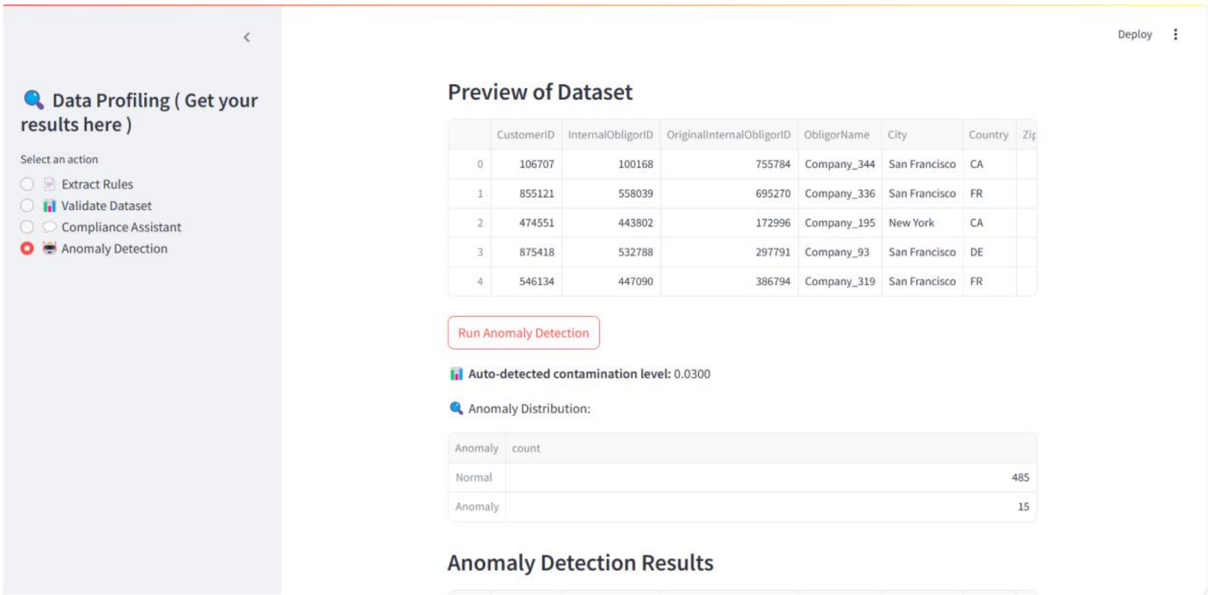
This module applies **Machine Learning (ML) techniques** to detect anomalies in financial data.

How It Works:

1 Upload your **dataset (CSV)**.
2 An anomaly detection model Isolation Forest is used for large data efficiency.
3 AI highlights **unusual data points (outliers)**.
4 View **interactive visualizations** and download anomaly reports.

Why Use This?

✓ **Identifies inconsistencies & fraud detection.**
✓ **Enhances data quality & regulatory reporting.**
✓ **Provides clear anomaly explanations via AI insights.**



Preview of Dataset

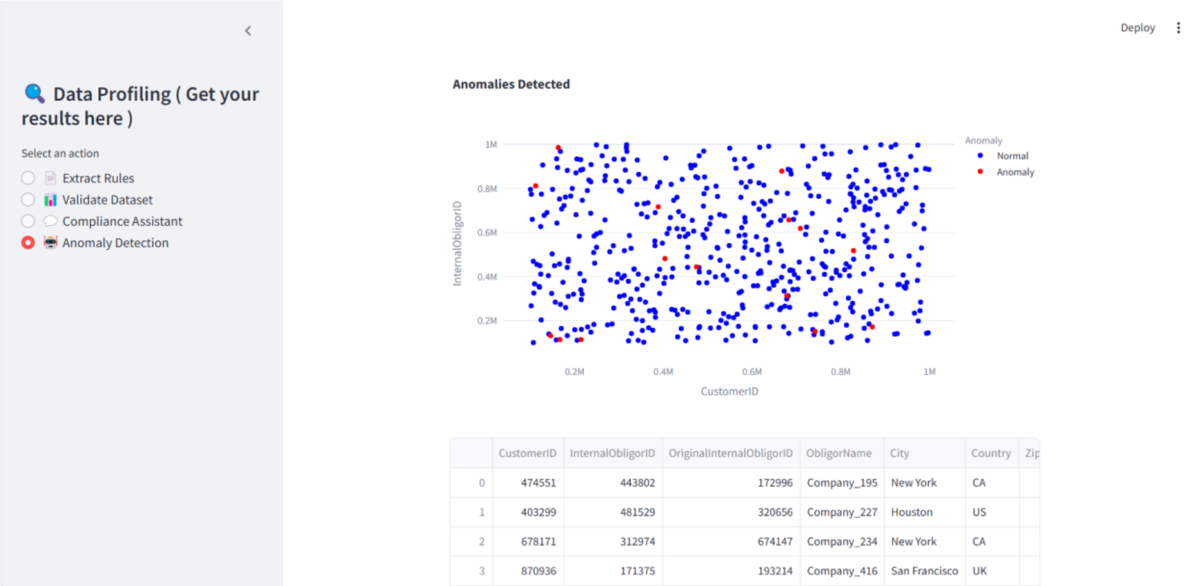
	CustomerID	InternalObligorID	OriginalInternalObligorID	ObligorName	City	Country	Zip
0	106707	100168	755784	Company_344	San Francisco	CA	
1	855121	558039	695270	Company_336	San Francisco	FR	
2	474551	443802	172996	Company_195	New York	CA	
3	875418	532788	297791	Company_93	San Francisco	DE	
4	546134	447090	386794	Company_319	San Francisco	FR	

📊 **Auto-detected contamination level:** 0.0300

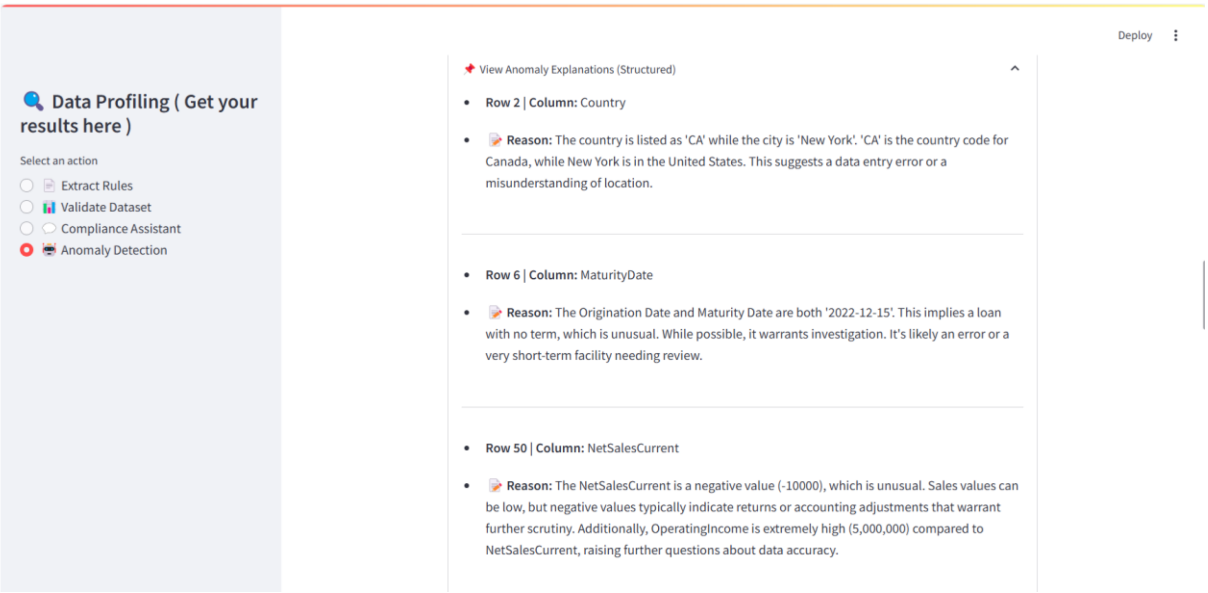
🔍 Anomaly Distribution:

Anomaly	count
Normal	485
Anomaly	15

Anomaly Detection Results



	CustomerID	InternalObligorID	OriginalInternalObligorID	ObligorName	City	Country	Zip
0	474551	443802	172996	Company_195	New York	CA	
1	403299	481529	320656	Company_227	Houston	US	
2	678171	312974	674147	Company_234	New York	CA	
3	870936	171375	193214	Company_416	San Francisco	UK	



- **Row 2 | Column:** Country
- 📝 **Reason:** The country is listed as 'CA' while the city is 'New York'. 'CA' is the country code for Canada, while New York is in the United States. This suggests a data entry error or a misunderstanding of location.

- **Row 6 | Column:** MaturityDate
- 📝 **Reason:** The Origination Date and Maturity Date are both '2022-12-15'. This implies a loan with no term, which is unusual. While possible, it warrants investigation. It's likely an error or a very short-term facility needing review.

- **Row 50 | Column:** NetSalesCurrent
- 📝 **Reason:** The NetSalesCurrent is a negative value (-10000), which is unusual. Sales values can be low, but negative values typically indicate returns or accounting adjustments that warrant further scrutiny. Additionally, OperatingIncome is extremely high (5,000,000) compared to NetSalesCurrent, raising further questions about data accuracy.

5. Future Scope

- **Advanced Machine Learning for Anomaly Detection:** Incorporate deep learning-based anomaly detection (e.g., Autoencoders, GANs). Implement ensemble models that combine multiple anomaly detection techniques for better accuracy
- **Real-Time Data Validation & Streaming Support:** Integrate with real-time data sources (Kafka, Spark Streaming). Perform continuous validation instead of batch processing.
- **Automated Data Correction & Compliance Reports:** Instead of just detecting violations, AI can suggest & auto-correct issues in datasets. Generate automated compliance reports for auditors & regulators.
- **Adaptive AI-Powered Rule Learning:** Enable AI to dynamically learn new validation rules over time based on past violations. Use reinforcement learning to improve compliance recommendations
- **API Integration & Enterprise Deployment:** Develop REST APIs for easy integration with existing banking & enterprise systems.Deploy as a cloud-based SaaS solution for scalability & security..

---

6. Conclusion

The AI-Powered Data Profiling & Compliance Assistant **automates regulatory validation**, **improves dataset accuracy**, and **reduces compliance risks**.

Key Takeaways:

- **AI-Driven Rule Extraction** saves time & ensures compliance.
- **Batch-Based Validation** optimizes performance & reduces API costs.
- **Unsupervised Anomaly Detection** identifies potential data inconsistencies.
- **Interactive AI Chat Assistant** enhances compliance understanding.
- **Scalable & Future-Ready** with potential for real-time integration.

This system provides a **powerful, automated approach** to data validation, making regulatory compliance **faster, smarter, and more efficient**.