# 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)

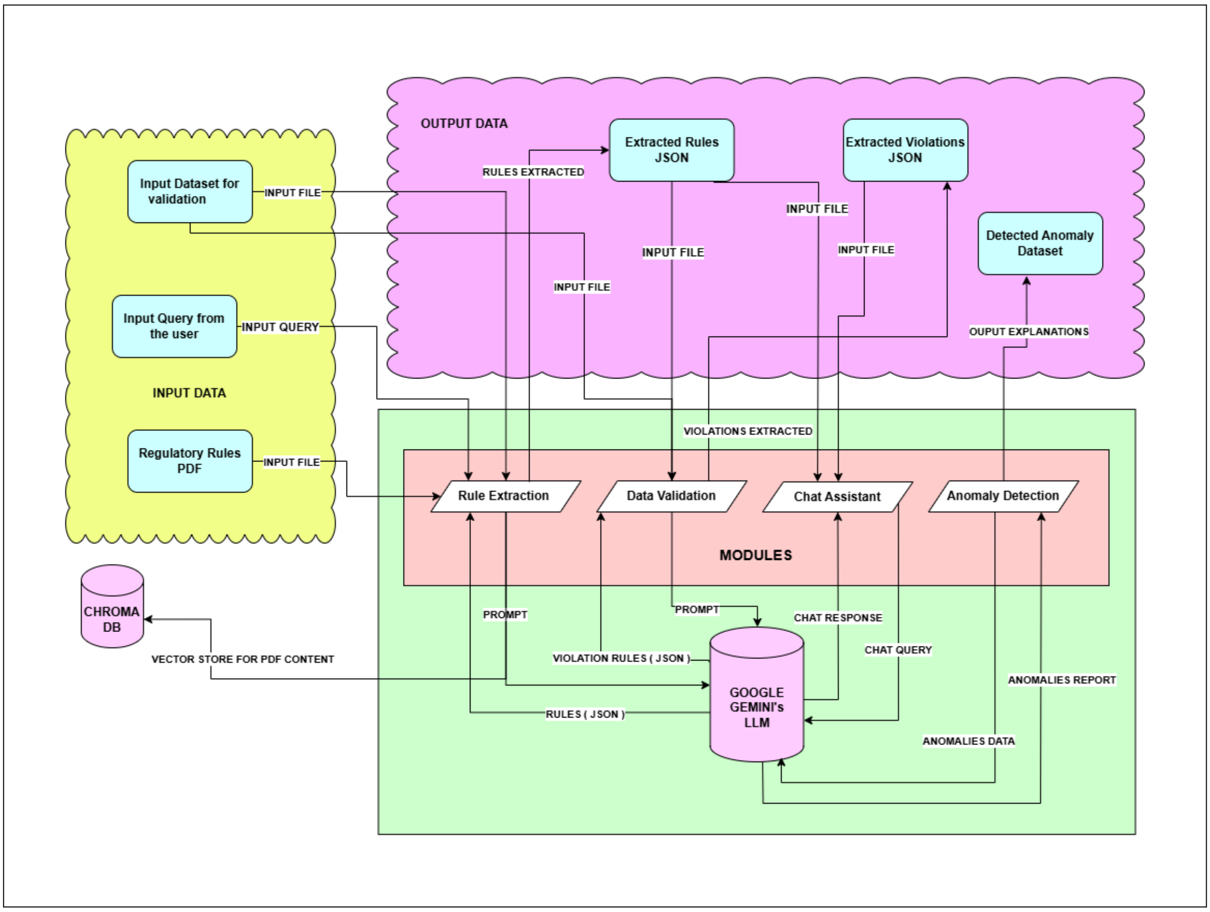
**Process:**

- **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**.

## 2.5 Architecture Diagram



---

## 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.

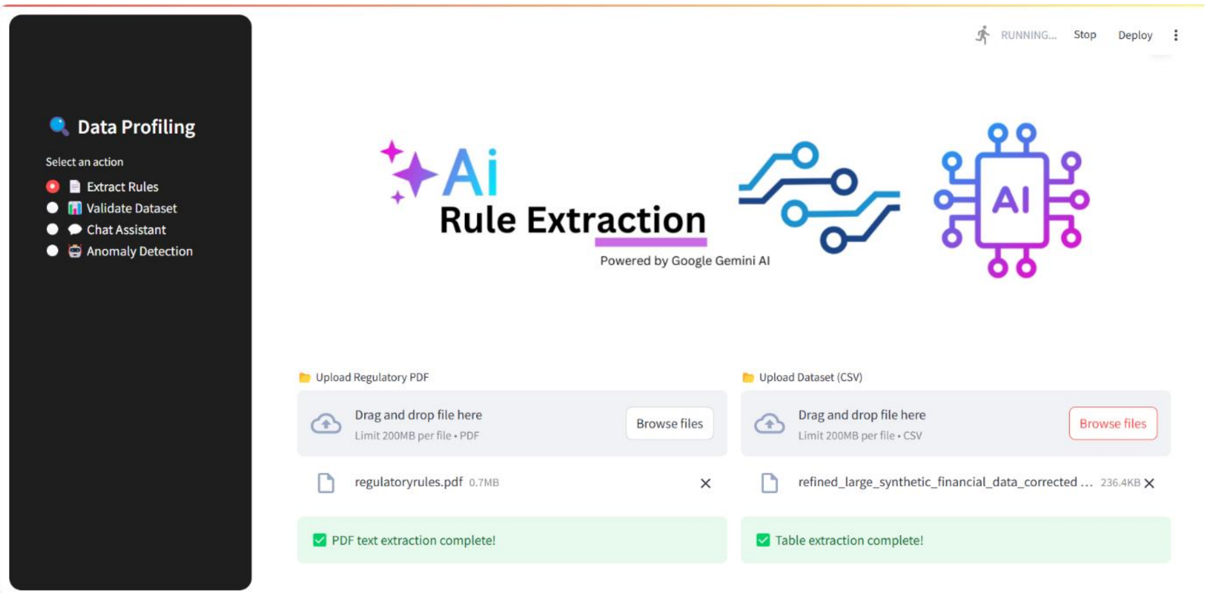
### 4.5 Making UI More Interactive

- **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)

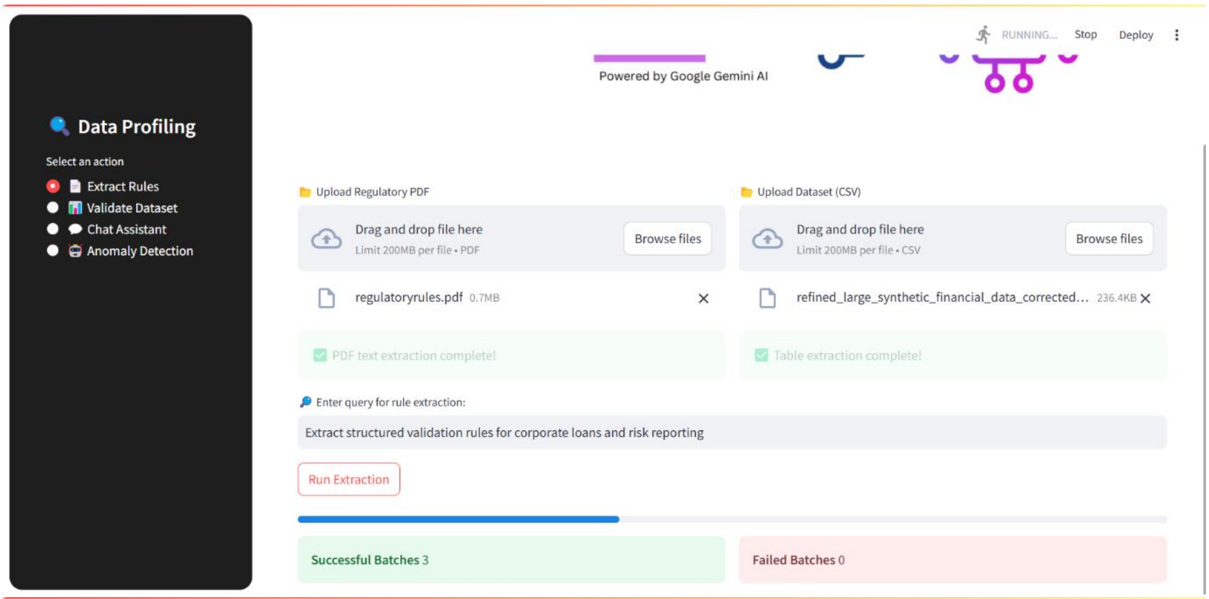


regulatoryrules.pdf 0.7MB

refined_large_synthetic_financial_data_corrected ... 236.4KB

✓ PDF text extraction complete!

✓ Table extraction complete!



regulatoryrules.pdf 0.7MB

refined_large_synthetic_financial_data_corrected… 236.4KB

✓ PDF text extraction complete!

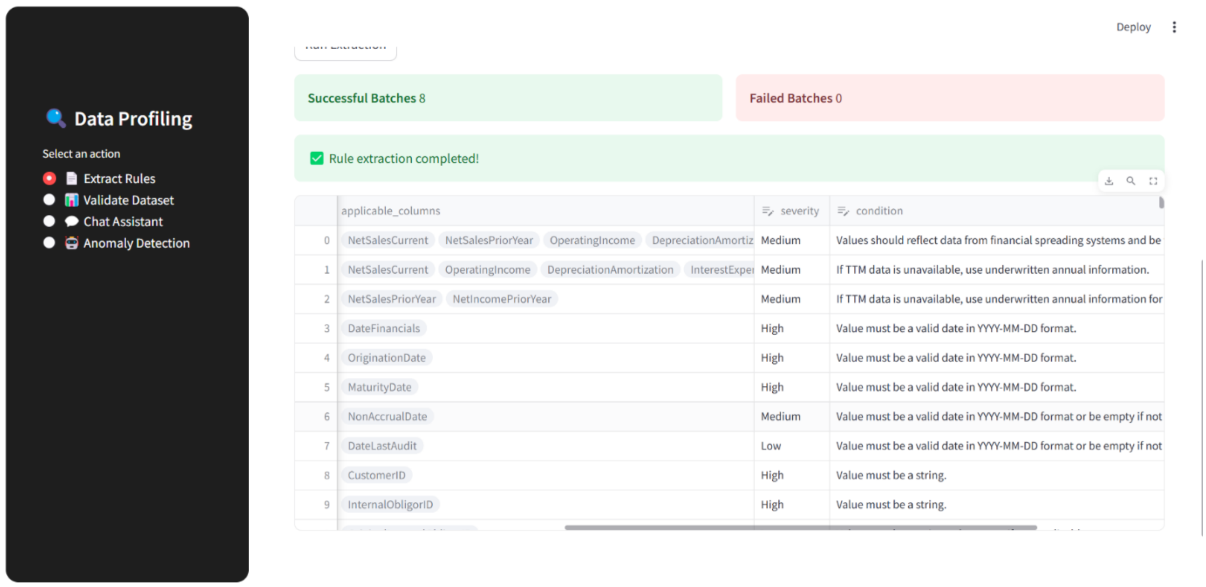
✓ Table extraction complete!

Enter query for rule extraction:

Extract structured validation rules for corporate loans and risk reporting

**Successful Batches** 3

**Failed Batches** 0

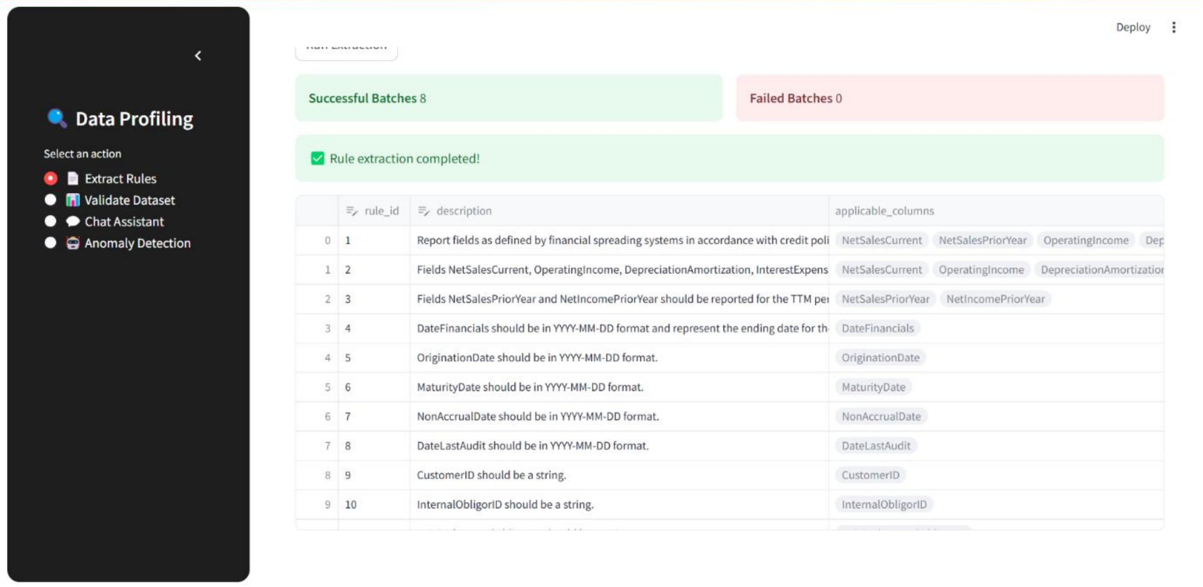


**Successful Batches** 8

**Failed Batches** 0

☑ Rule extraction completed!

| | applicable_columns | severity | condition |
|---|---|---|---|
| 0 | NetSalesCurrent<br>NetSalesPriorYear<br>OperatingIncome<br>DepreciationAmortiz | Medium | Values should reflect data from financial spreading systems and be |
| 1 | NetSalesCurrent<br>OperatingIncome<br>DepreciationAmortization<br>InterestExpe | Medium | If TTM data is unavailable, use underwritten annual information. |
| 2 | NetSalesPriorYear<br>NetIncomePriorYear | Medium | If TTM data is unavailable, use underwritten annual information for |
| 3 | DateFinancials | High | Value must be a valid date in YYYY-MM-DD format. |
| 4 | OriginationDate | High | Value must be a valid date in YYYY-MM-DD format. |
| 5 | MaturityDate | High | Value must be a valid date in YYYY-MM-DD format. |
| 6 | NonAccrualDate | Medium | Value must be a valid date in YYYY-MM-DD format or be empty if not |
| 7 | DateLastAudit | Low | Value must be a valid date in YYYY-MM-DD format or be empty if not |
| 8 | CustomerID | High | Value must be a string. |
| 9 | InternalObligorID | High | Value must be a string. |



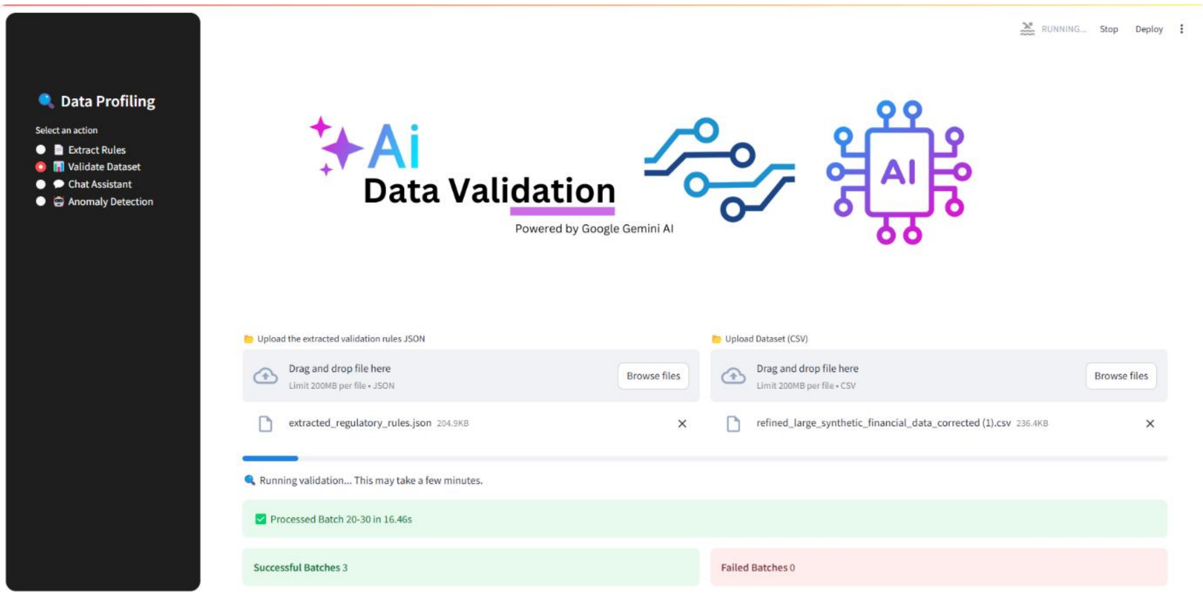
**Successful Batches** 8

**Failed Batches** 0

☑ Rule extraction completed!

| | rule_id | description | applicable_columns |
|---|---|---|---|
| 0 | 1 | Report fields as defined by financial spreading systems in accordance with credit poli | NetSalesCurrent<br>NetSalesPriorYear<br>OperatingIncome<br>Dep |
| 1 | 2 | Fields NetSalesCurrent, OperatingIncome, DepreciationAmortization, InterestExpens | NetSalesCurrent<br>OperatingIncome<br>DepreciationAmortizatio |
| 2 | 3 | Fields NetSalesPriorYear and NetIncomePriorYear should be reported for the TTM pe | NetSalesPriorYear<br>NetIncomePriorYear |
| 3 | 4 | DateFinancials should be in YYYY-MM-DD format and represent the ending date for th | DateFinancials |
| 4 | 5 | OriginationDate should be in YYYY-MM-DD format. | OriginationDate |
| 5 | 6 | MaturityDate should be in YYYY-MM-DD format. | MaturityDate |
| 6 | 7 | NonAccrualDate should be in YYYY-MM-DD format. | NonAccrualDate |
| 7 | 8 | DateLastAudit should be in YYYY-MM-DD format. | DateLastAudit |
| 8 | 9 | CustomerID should be a string. | CustomerID |
| 9 | 10 | InternalObligorID should be a string. | InternalObligorID |

## 5.2 Dataset Validation (Automated Compliance Checking)



Upload the extracted validation rules JSON

extracted_regulatory_rules.json 204.9KB

Upload Dataset (CSV)

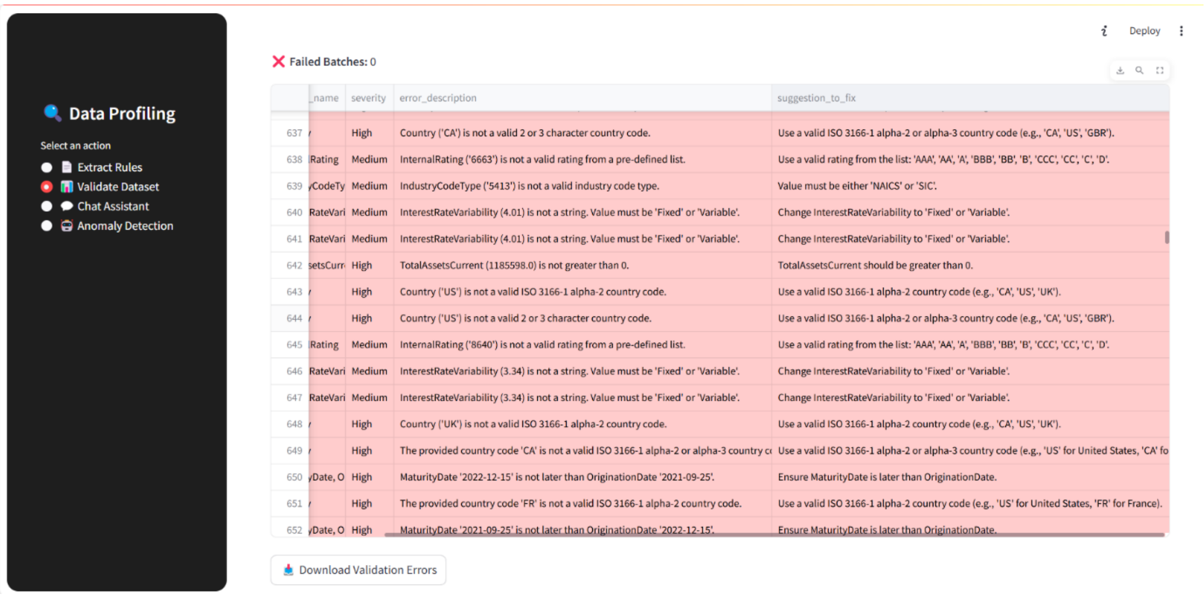
refined_large_synthetic_financial_data_corrected (1).csv 236.4KB

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

Processed Batch 20-30 in 16.46s

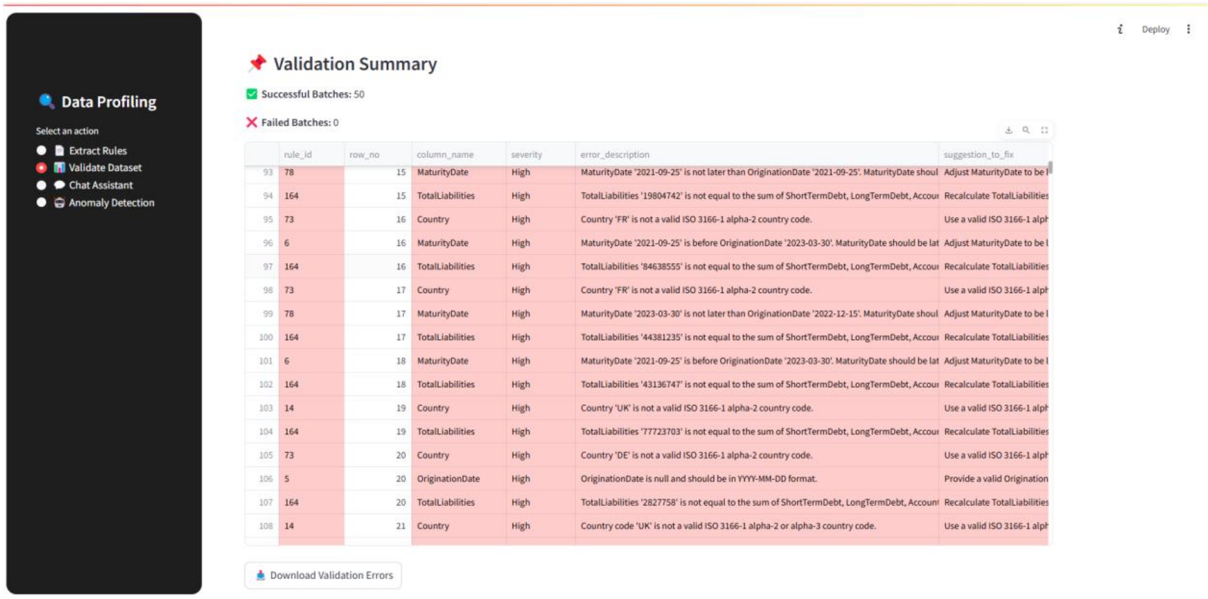
Successful Batches 3

Failed Batches 0



Failed Batches: 0

| | _name | severity | error_description | suggestion_to_fix |
|---|---|---|---|---|
| 637 | | High | Country ('CA') is not a valid 2 or 3 character country code. | Use a valid ISO 3166-1 alpha-2 or alpha-3 country code (e.g., 'CA', 'US', 'GBR'). |
| 638 | Rating | Medium | InternalRating ('6663') is not a valid rating from a pre-defined list. | Use a valid rating from the list: 'AAA', 'AA', 'A', 'BBB', 'BB', 'B', 'CCC', 'CC', 'C', 'D'. |
| 639 | CodeTy | Medium | IndustryCodeType ('5413') is not a valid industry code type. | Value must be either 'NAICS' or 'SIC'. |
| 640 | RateVari | Medium | InterestRateVariability (4.01) is not a string. Value must be 'Fixed' or 'Variable'. | Change InterestRateVariability to 'Fixed' or 'Variable'. |
| 641 | RateVari | Medium | InterestRateVariability (4.01) is not a string. Value must be 'Fixed' or 'Variable'. | Change InterestRateVariability to 'Fixed' or 'Variable'. |
| 642 | setsCurr | High | TotalAssetsCurrent (1185598.0) is not greater than 0. | TotalAssetsCurrent should be greater than 0. |
| 643 | | High | Country ('US') is not a valid ISO 3166-1 alpha-2 country code. | Use a valid ISO 3166-1 alpha-2 country code (e.g., 'CA', 'US', 'UK'). |
| 644 | | High | Country ('US') is not a valid 2 or 3 character country code. | Use a valid ISO 3166-1 alpha-2 or alpha-3 country code (e.g., 'CA', 'US', 'GBR'). |
| 645 | Rating | Medium | InternalRating ('8640') is not a valid rating from a pre-defined list. | Use a valid rating from the list: 'AAA', 'AA', 'A', 'BBB', 'BB', 'B', 'CCC', 'CC', 'C', 'D'. |
| 646 | RateVari | Medium | InterestRateVariability (3.34) is not a string. Value must be 'Fixed' or 'Variable'. | Change InterestRateVariability to 'Fixed' or 'Variable'. |
| 647 | RateVari | Medium | InterestRateVariability (3.34) is not a string. Value must be 'Fixed' or 'Variable'. | Change InterestRateVariability to 'Fixed' or 'Variable'. |
| 648 | | High | Country ('UK') is not a valid ISO 3166-1 alpha-2 country code. | Use a valid ISO 3166-1 alpha-2 country code (e.g., 'CA', 'US', 'UK'). |
| 649 | | High | The provided country code 'CA' is not a valid ISO 3166-1 alpha-2 or alpha-3 country c | Use a valid ISO 3166-1 alpha-2 or alpha-3 country code (e.g., 'US' for United States, 'CA' fo |
| 650 | Date, O | High | MaturityDate '2022-12-15' is not later than OriginationDate '2021-09-25'. | Ensure MaturityDate is later than OriginationDate. |
| 651 | | High | The provided country code 'FR' is not a valid ISO 3166-1 alpha-2 country code. | Use a valid ISO 3166-1 alpha-2 country code (e.g., 'US' for United States, 'FR' for France). |
| 652 | Date, O | High | MaturityDate '2021-09-25' is not later than OriginationDate '2022-12-15'. | Ensure MaturityDate is later than OriginationDate. |



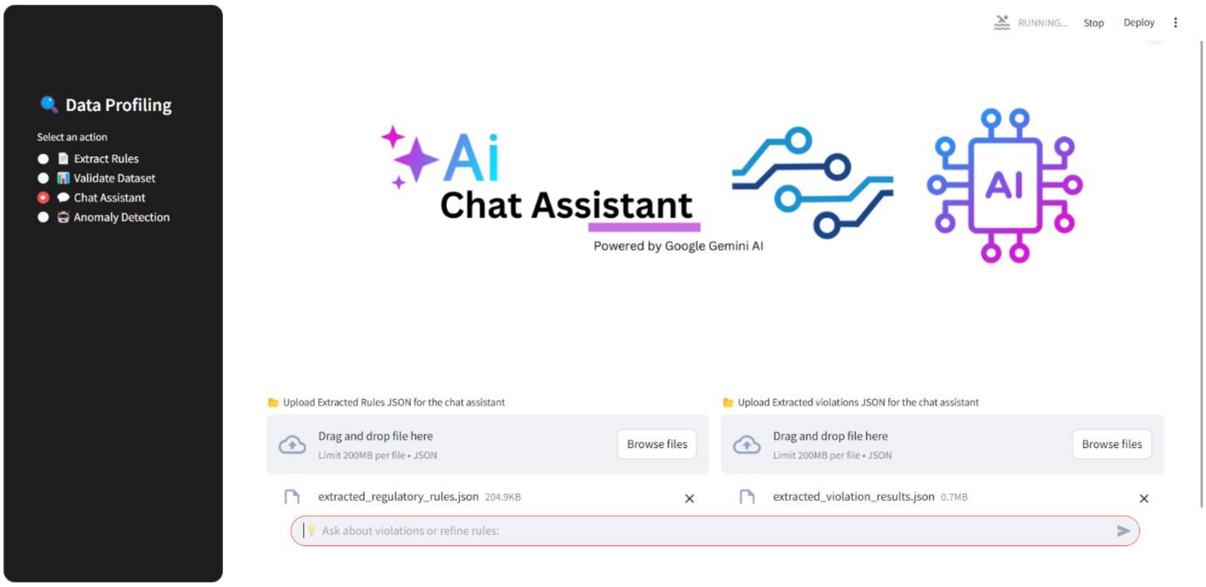
Validation Summary

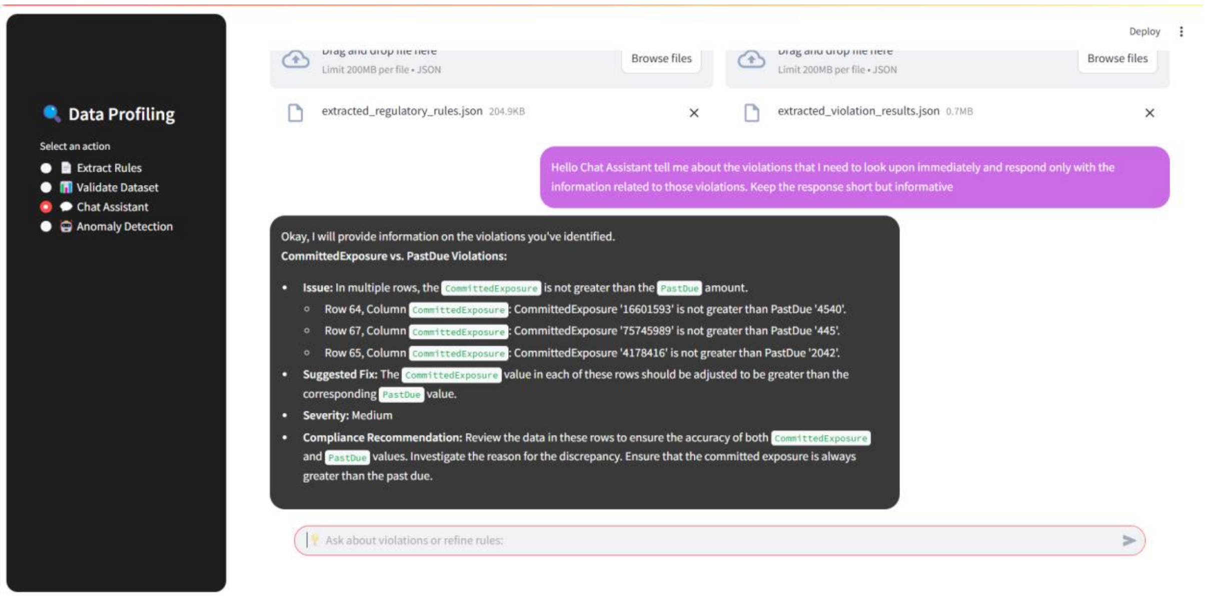
Successful Batches: 50

Failed Batches: 0

| | rule_id | row_no | column_name | severity | error_description | suggestion_to_fix |
|---|---|---|---|---|---|---|
| 93 | 78 | 15 | MaturityDate | High | MaturityDate '2021-09-25' is not later than OriginationDate '2021-09-25'. MaturityDate shoul | Adjust MaturityDate to be l |
| 94 | 164 | 15 | TotalLiabilities | High | TotalLiabilities '19804742' is not equal to the sum of ShortTermDebt, LongTermDebt, Accou | Recalculate TotalLiabilities |
| 95 | 73 | 16 | Country | High | Country 'FR' is not a valid ISO 3166-1 alpha-2 country code. | Use a valid ISO 3166-1 alph |
| 96 | 6 | 16 | MaturityDate | High | MaturityDate '2021-09-25' is before OriginationDate '2023-03-30'. MaturityDate should be lat | Adjust MaturityDate to be l |
| 97 | 164 | 16 | TotalLiabilities | High | TotalLiabilities '84638555' is not equal to the sum of ShortTermDebt, LongTermDebt, Accou | Recalculate TotalLiabilities |
| 98 | 73 | 17 | Country | High | Country 'FR' is not a valid ISO 3166-1 alpha-2 country code. | Use a valid ISO 3166-1 alph |
| 99 | 78 | 17 | MaturityDate | High | MaturityDate '2023-03-30' is not later than OriginationDate '2022-12-15'. MaturityDate shoul | Adjust MaturityDate to be l |
| 100 | 164 | 17 | TotalLiabilities | High | TotalLiabilities '44381235' is not equal to the sum of ShortTermDebt, LongTermDebt, Accou | Recalculate TotalLiabilities |
| 101 | 6 | 18 | MaturityDate | High | MaturityDate '2021-09-25' is before OriginationDate '2023-03-30'. MaturityDate should be lat | Adjust MaturityDate to be l |
| 102 | 164 | 18 | TotalLiabilities | High | TotalLiabilities '43136747' is not equal to the sum of ShortTermDebt, LongTermDebt, Accou | Recalculate TotalLiabilities |
| 103 | 14 | 19 | Country | High | Country 'UK' is not a valid ISO 3166-1 alpha-2 country code. | Use a valid ISO 3166-1 alph |
| 104 | 164 | 19 | TotalLiabilities | High | TotalLiabilities '77723703' is not equal to the sum of ShortTermDebt, LongTermDebt, Accou | Recalculate TotalLiabilities |
| 105 | 73 | 20 | Country | High | Country 'DE' is not a valid ISO 3166-1 alpha-2 country code. | Use a valid ISO 3166-1 alph |
| 106 | 5 | 20 | OriginationDate | High | OriginationDate is null and should be in YYYY-MM-DD format. | Provide a valid Origination |
| 107 | 164 | 20 | TotalLiabilities | High | TotalLiabilities '2827758' is not equal to the sum of ShortTermDebt, LongTermDebt, Account | Recalculate TotalLiabilities |
| 108 | 14 | 21 | Country | High | Country code 'UK' is not a valid ISO 3166-1 alpha-2 or alpha-3 country code. | Use a valid ISO 3166-1 alph |

## 5.3 Compliance Chat Assistant (Interactive AI-Powered Queries)

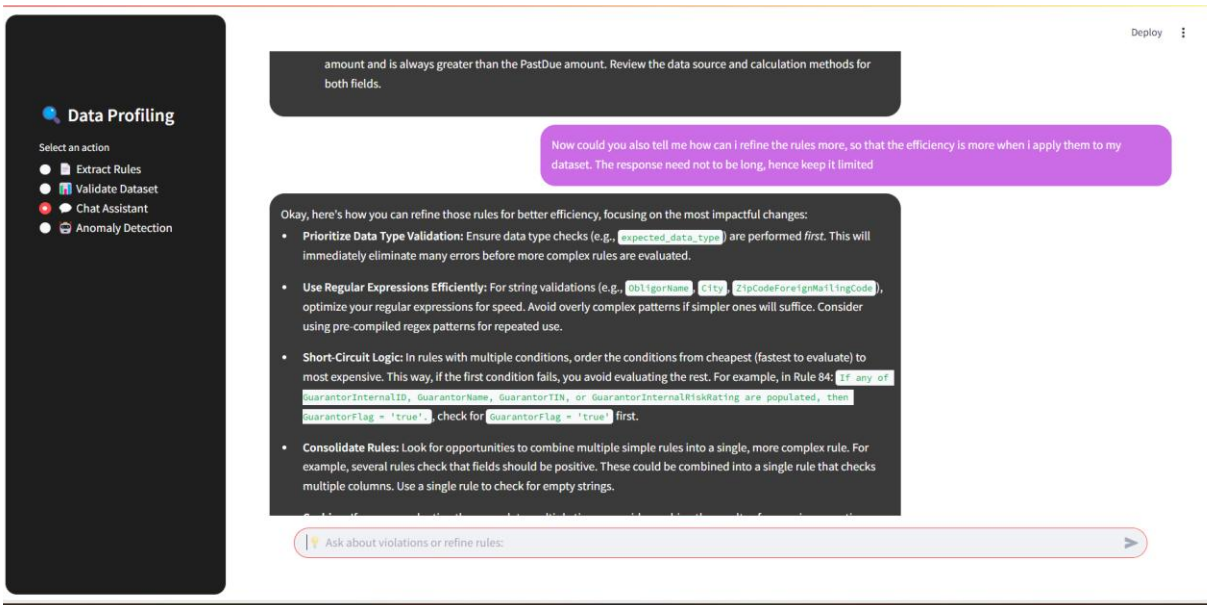

Data Profiling
Select an action
Extract Rules
Validate Dataset
Chat Assistant
Anomaly Detection
RUNNING...
Stop
Deploy
Ai
Chat Assistant
Powered by Google Gemini AI
AI
Upload Extracted Rules JSON for the chat assistant
Drag and drop file here
Limit 200MB per file • JSON
Browse files
extracted_regulatory_rules.json 204.9KB
Upload Extracted violations JSON for the chat assistant
Drag and drop file here
Limit 200MB per file • JSON
Browse files
extracted_violation_results.json 0.7MB
Ask about violations or refine rules:


Data Profiling
Select an action
Extract Rules
Validate Dataset
Chat Assistant
Anomaly Detection
Deploy
Limit 200MB per file • JSON
Browse files
extracted_regulatory_rules.json 204.9KB
Limit 200MB per file • JSON
Browse files
extracted_violation_results.json 0.7MB
Ask about violations or refine rules:

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, I will provide information on the violations you've identified.
**CommittedExposure vs. PastDue Violations:**

- **Issue:** In multiple rows, the `CommittedExposure` is not greater than the `PastDue` amount.
  - Row 64, Column `CommittedExposure`: CommittedExposure '16601593' is not greater than PastDue '4540'.
  - Row 67, Column `CommittedExposure`: CommittedExposure '75745989' is not greater than PastDue '445'.
  - Row 65, Column `CommittedExposure`: CommittedExposure '4178416' is not greater than PastDue '2042'.
- **Suggested Fix:** The `CommittedExposure` value in each of these rows should be adjusted to be greater than the corresponding `PastDue` value.
- **Severity:** Medium
- **Compliance Recommendation:** Review the data in these rows to ensure the accuracy of both `CommittedExposure` and `PastDue` values. Investigate the reason for the discrepancy. Ensure that the committed exposure is always greater than the past due.



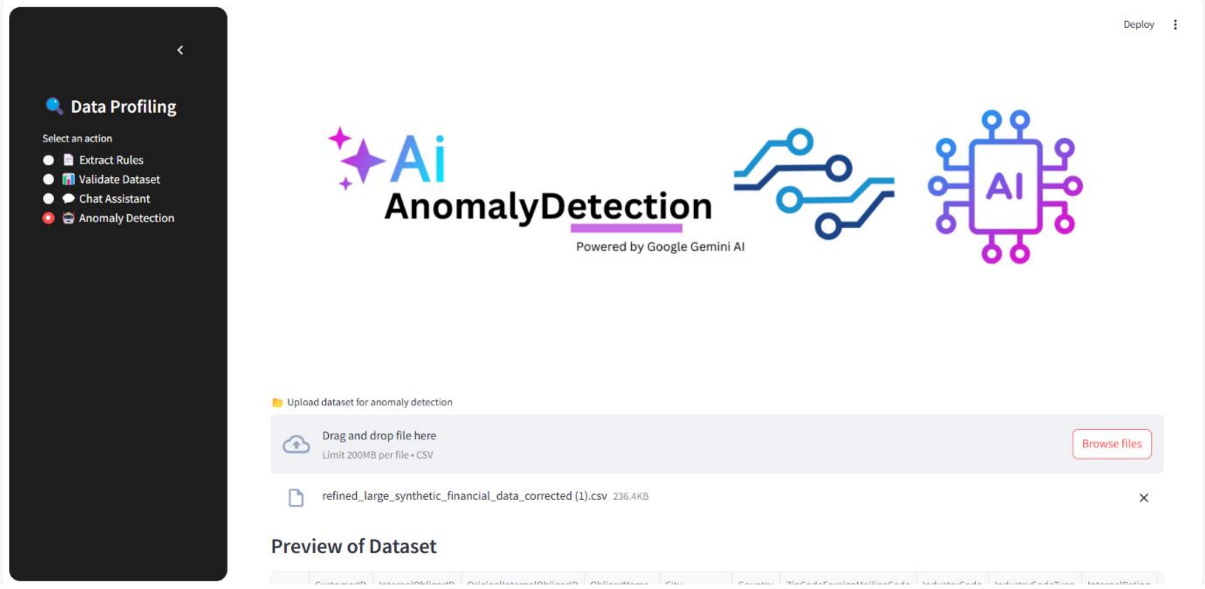
amount and is always greater than the PastDue amount. Review the data source and calculation methods for both fields.

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

Okay, here's how you can refine those rules for better efficiency, focusing on the most impactful changes:

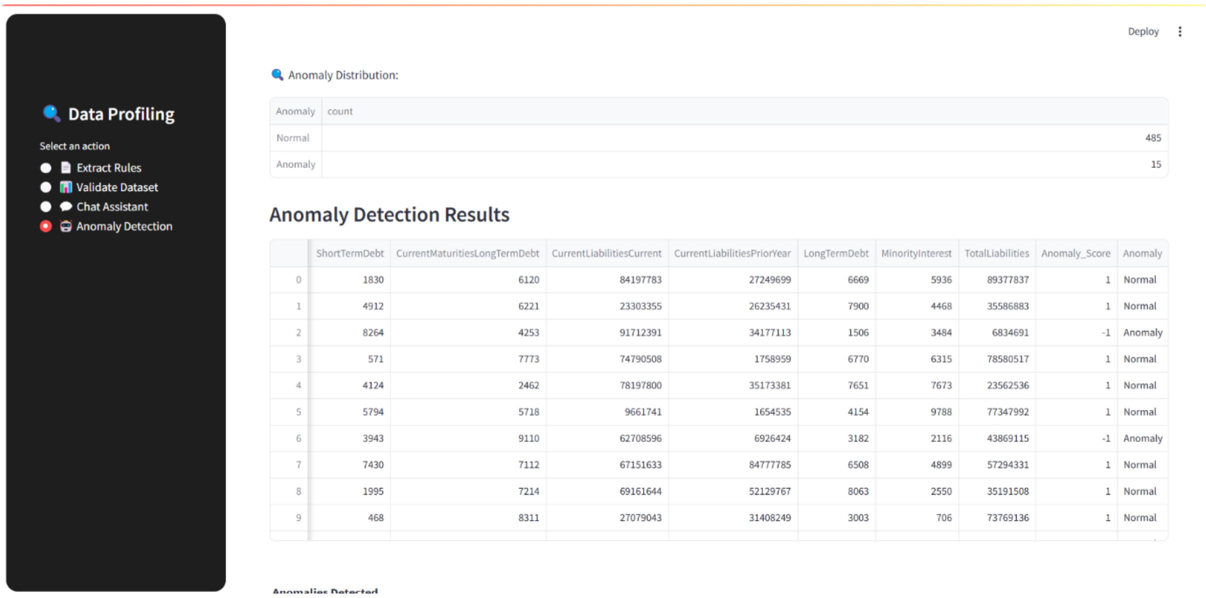
- **Prioritize Data Type Validation:** Ensure data type checks (e.g., `expected_data_type`) are performed *first*. This will immediately eliminate many errors before more complex rules are evaluated.
- **Use Regular Expressions Efficiently:** For string validations (e.g., `ObligorName`, `City`, `ZipCodeForeignMailingCode`), optimize your regular expressions for speed. Avoid overly complex patterns if simpler ones will suffice. Consider using pre-compiled regex patterns for repeated use.
- **Short-Circuit Logic:** In rules with multiple conditions, order the conditions from cheapest (fastest to evaluate) to most expensive. This way, if the first condition fails, you avoid evaluating the rest. For example, in Rule 84: `If any of GuarantorInternalID, GuarantorName, GuarantorTIN, or GuarantorInternalRiskRating are populated, then GuarantorFlag = 'true'.`, check for `GuarantorFlag = 'true'` first.
- **Consolidate Rules:** Look for opportunities to combine multiple simple rules into a single, more complex rule. For example, several rules check that fields should be positive. These could be combined into a single rule that checks multiple columns. Use a single rule to check for empty strings.

## 5.4 Anomaly Detection (Machine Learning-Based Outlier Detection)



refined_large_synthetic_financial_data_corrected (1).csv 236.4KB

Preview of Dataset

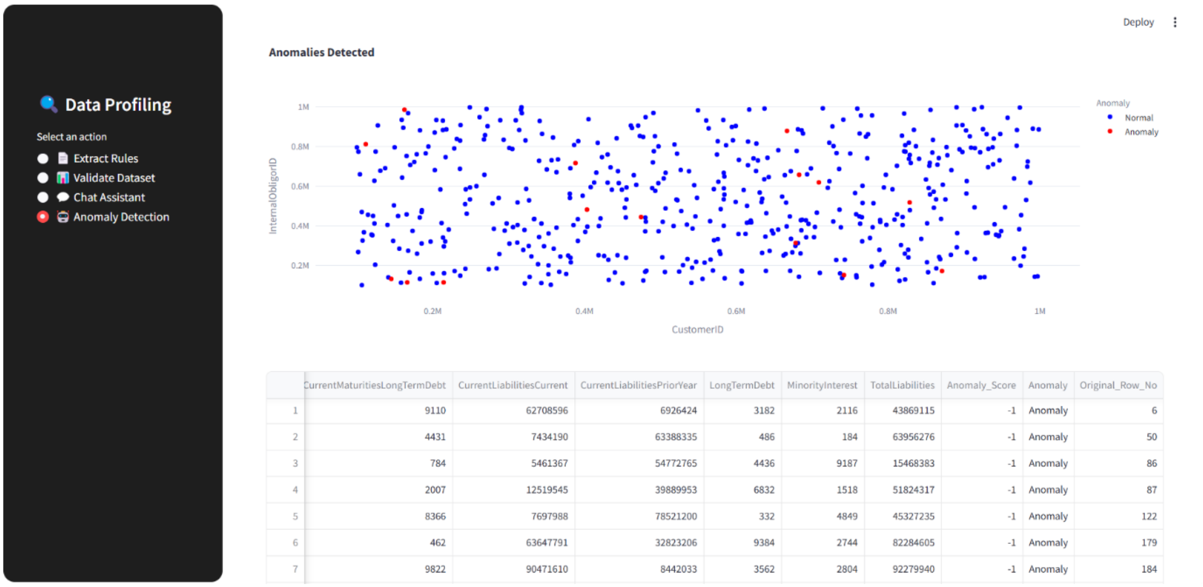


Anomaly Distribution:

| Anomaly | count |
|---|---|
| Normal | 485 |
| Anomaly | 15 |

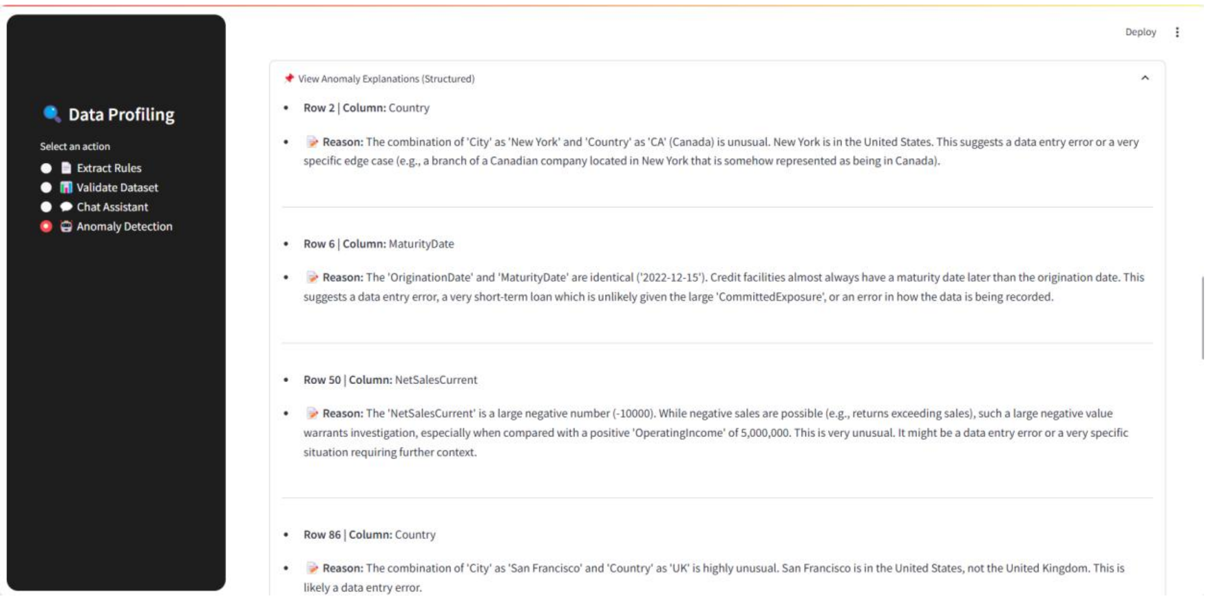
## Anomaly Detection Results

| | ShortTermDebt | CurrentMaturitiesLongTermDebt | CurrentLiabilitiesCurrent | CurrentLiabilitiesPriorYear | LongTermDebt | MinorityInterest | TotalLiabilities | Anomaly_Score | Anomaly |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1830 | 6120 | 84197783 | 27249699 | 6669 | 5936 | 89377837 | 1 | Normal |
| 1 | 4912 | 6221 | 23303355 | 26235431 | 7900 | 4468 | 35586883 | 1 | Normal |
| 2 | 8264 | 4253 | 91712391 | 34177113 | 1506 | 3484 | 6834691 | -1 | Anomaly |
| 3 | 571 | 7773 | 74790508 | 1758959 | 6770 | 6315 | 78580517 | 1 | Normal |
| 4 | 4124 | 2462 | 78197800 | 35173381 | 7651 | 7673 | 23562536 | 1 | Normal |
| 5 | 5794 | 5718 | 9661741 | 1654535 | 4154 | 9788 | 77347992 | 1 | Normal |
| 6 | 3943 | 9110 | 62708596 | 6926424 | 3182 | 2116 | 43869115 | -1 | Anomaly |
| 7 | 7430 | 7112 | 67151633 | 84777785 | 6508 | 4899 | 57294331 | 1 | Normal |
| 8 | 1995 | 7214 | 69161644 | 52129767 | 8063 | 2550 | 35191508 | 1 | Normal |
| 9 | 468 | 8311 | 27079043 | 31408249 | 3003 | 706 | 73769136 | 1 | Normal |



**Anomalies Detected**

| | CurrentMaturitiesLongTermDebt | CurrentLiabilitiesCurrent | CurrentLiabilitiesPriorYear | LongTermDebt | MinorityInterest | TotalLiabilities | Anomaly_Score | Anomaly | Original_Row_No |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 9110 | 62708596 | 6926424 | 3182 | 2116 | 43869115 | -1 | Anomaly | 6 |
| 2 | 4431 | 7434190 | 63388335 | 486 | 184 | 63956276 | -1 | Anomaly | 50 |
| 3 | 784 | 5461367 | 54772765 | 4436 | 9187 | 15468383 | -1 | Anomaly | 86 |
| 4 | 2007 | 12519545 | 39889953 | 6832 | 1518 | 51824317 | -1 | Anomaly | 87 |
| 5 | 8366 | 7697988 | 78521200 | 332 | 4849 | 45327235 | -1 | Anomaly | 122 |
| 6 | 462 | 63647791 | 32823206 | 9384 | 2744 | 82284605 | -1 | Anomaly | 179 |
| 7 | 9822 | 90471610 | 8442033 | 3562 | 2804 | 92279940 | -1 | Anomaly | 184 |



📌 View Anomaly Explanations (Structured)

- **Row 2 | Column:** Country
- 📝 **Reason:** The combination of 'City' as 'New York' and 'Country' as 'CA' (Canada) is unusual. New York is in the United States. This suggests a data entry error or a very specific edge case (e.g., a branch of a Canadian company located in New York that is somehow represented as being in Canada).

---

- **Row 6 | Column:** MaturityDate
- 📝 **Reason:** The 'OriginationDate' and 'MaturityDate' are identical ('2022-12-15'). Credit facilities almost always have a maturity date later than the origination date. This suggests a data entry error, a very short-term loan which is unlikely given the large 'CommittedExposure', or an error in how the data is being recorded.

---

- **Row 50 | Column:** NetSalesCurrent
- 📝 **Reason:** The 'NetSalesCurrent' is a large negative number (-10000). While negative sales are possible (e.g., returns exceeding sales), such a large negative value warrants investigation, especially when compared with a positive 'OperatingIncome' of 5,000,000. This is very unusual. It might be a data entry error or a very specific situation requiring further context.

---

- **Row 86 | Column:** Country
- 📝 **Reason:** The combination of 'City' as 'San Francisco' and 'Country' as 'UK' is highly unusual. San Francisco is in the United States, not the United Kingdom. This is likely a data entry error.

## 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**.