Core Model – A Generalized LLM-Driven Analytical Engine for Multi-Domain Data Insights

DISSERTATION

Submitted in partial fulfillment of the requirements of the

Degree: M. Tech in Artificial Intelligence & Machine Learning

By

Anurag Sharma

2023aa05381

Under the supervision of

Janardan Jayaraman

(Vice President)



BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

Pilani (Rajasthan) INDIA

(July, 2025)

# Table of Contents

[Table of Contents 2](#_Toc204169350)

[Abstract 5](#_Toc204169351)

[List of Symbols & Abbreviations 7](#_Toc204169352)

[List of Tables 8](#_Toc204169353)

[List of Figures 11](#_Toc204169354)

[Chapter 1 – Introduction and Objectives 13](#_Toc204169355)

[Introduction 13](#_Toc204169356)

[Scope of Work 13](#_Toc204169357)

[Primary Objectives 14](#_Toc204169358)

[Objectives Met till Midterm 14](#_Toc204169359)

[Post Mid Term 14](#_Toc204169360)

[Chapter 2 - System Design and Architecture 16](#_Toc204169361)

[System Requirements 16](#_Toc204169362)

[Component Design 16](#_Toc204169363)

[Core Model Engine 16](#_Toc204169364)

[Dataset Manager 16](#_Toc204169365)

[Data Processing Pipeline 17](#_Toc204169366)

[Prompt Engineering 20](#_Toc204169367)

[Statistical Analysis Modules 20](#_Toc204169368)

[Core Model usage at BNY 25](#_Toc204169369)

[Purpose of the Core Engine 25](#_Toc204169370)

[Technical Validation Framework 26](#_Toc204169371)

[Mathematical Foundation: 26](#_Toc204169372)

[Statistical Tests Applied: 26](#_Toc204169373)

[Metric Categories Explained 27](#_Toc204169374)

[Statistical Accuracy (Weight: 25%) 27](#_Toc204169375)

[Completeness (Weight: 30%) 27](#_Toc204169376)

[Consistency (Weight: 25%) 27](#_Toc204169377)

[Efficiency (Weight: 20%) 27](#_Toc204169378)

[Use Cases and Applications at BNY 28](#_Toc204169379)

[Role in MRM Governance 28](#_Toc204169380)

[How core model works 29](#_Toc204169381)

[Directions for Future Work 29](#_Toc204169382)

[Bibliography / References 31](#_Toc204169383)

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI**

**SECOND SEMESTER 2024-25**

**AIMLCZG628T DISSERTATION**

Dissertation Title : Core Model – A Generalized LLM-Driven Analytical Engine for Multi-Domain Data Insights

Name of Supervisor : Janardan Jayaraman

Name of Student : Anurag Sharma

ID No. of Student : 2023aa05381

Courses Relevant for the Project & Corresponding Semester:

1. Conversational AI
2. NLP Applications
3. Information Retrieval
4. Machine Learning

## Abstract

This dissertation project focuses on building a flexible and scalable AI/ML framework called the Core Model. The goal is to create a centralized system that can analyze a wide variety of datasets—structured, unstructured, textual, numerical, or mixed—without requiring separate analytics pipelines for each team or domain. Instead of each team handling their own analysis manually, they can simply submit a data sample to the Core Model. From there, the system uses carefully crafted prompts to guide large language models (LLMs) in generating meaningful, structured insights.

What makes this framework innovative is its ability to interpret unfamiliar and diverse data formats with minimal human input. The Core Model uses LLMs to automatically carry out exploratory data analysis (EDA), statistical summaries, pattern recognition, anomaly detection, and correlation analysis. The insights it provides are clearly explained and categorized as either "known insights" (those that confirm domain knowledge) or "new discoveries" (previously hidden patterns found through advanced analysis).

A key feature of this system is its adaptability. The Core Model doesn’t rely on just one LLM. Instead, it benchmarks multiple models to determine which one performs best on a given dataset. It evaluates models based on accuracy, clarity of output, confidence in results, and overall relevance. Users will be informed about the expected quality of insights for their specific data type, helping them understand what level of performance they can expect from the system.

To keep things consistent and easy to use, the Core Model includes a standardized input/output framework. Teams can upload their datasets and receive analytical outputs in a reliable, repeatable format that doesn’t require technical expertise to interpret.

Overall, the Core Model aims to streamline data analysis across the organization. By using LLMs and strategic prompting, it reduces the need for custom pipelines, promotes consistency in insights, and enables smarter, faster decision-making. As LLM technology evolves, the Core Model will evolve with it—ensuring it remains accurate, interpretable, and highly useful for a wide range of data-driven tasks.

**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI**

**II SEMESTER 24-25**

**AIMLCZG628T DISSERTATION**

**Dissertation Outline**

**BITS ID No.**2023aa05381 **Name of Student:** Anurag Sharma

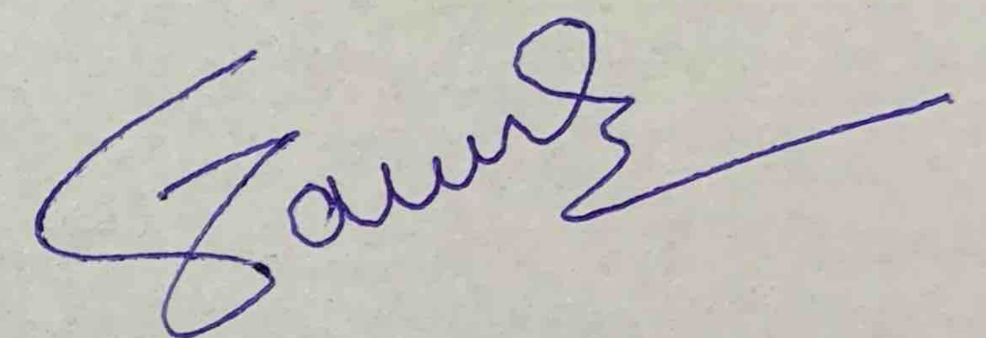
**Name of Supervisor:** Janardan Jayaraman

**Designation of Supervisor**: Vice President

**Qualification and Experience:** M. Tech and 17+ years

**Official E- mail ID of Supervisor:** Janardan.Jayaraman@bny.com

**Topic of Dissertation**: Core Model – A Generalized LLM-Driven Analytical Engine for Multi-Domain Data Insights

(Signature of Student) (Signature of Supervisor)

Date:.24/May/2025 Date:24/May/2025

# List of Symbols & Abbreviations

|  |  |
| --- | --- |
| **Symbol / Abbreviation** | **Description** |
| LLM | “Large Language Model” |
| EDA | “Exploratory Data Analysis” |
| API | “Application Programming Interface” |
| CSV | “Comma-Separated Values” |
| MRM | “Model review management” |
| JSON | “JavaScript Object Notation” |
| UI | “User Interface” |

# List of Tables

[Table 1: “Core Data Processing Features” 8](#_Toc204168033)

[Table 2: “Advanced Analytics & AI Features” 9](#_Toc204168034)

[Table 3: “System Integration & Performance Features” 10](#_Toc204168035)

[Table 4: “Tech Stack Used” 10](#_Toc204168036)

[Table 5: Detailed Plan of Work 15](#_Toc204168037)

Table 1: “Core Data Processing Features”

|  |  |  |  |
| --- | --- | --- | --- |
| **File Upload & Loading** | Multi-format Support | CSV, JSON, Excel (.xlsx, .xls) file upload | Valid data file |
| **Data Quality & Validation** | Data Quality Analysis | LLM-powered assessment of data types, missing values, constant columns | Uploaded dataset |
|  | Data Cleaning | Automatic removal of index columns, datetime conversion, numeric formatting | Any dataset |
|  | Missing Value Detection | Identifies and reports columns with missing data | Any dataset |
|  | Constant Column Detection | Finds columns with single values (retained for transparency) | Any dataset |
| **Basic Analysis** | Dataset Preview | Shows first few rows of data | Uploaded dataset |
|  | Basic Statistics | “Mean, median, mode, std dev, min, max” for all columns | Numeric columns |
|  | Data Types Summary | Column names, types, and missing value counts | Any dataset |
|  | Dataset Profile | Comprehensive JSON profile of the dataset | Any dataset |
| **Visualizations** | Distribution Plots | Histograms with skewness analysis | Numeric columns |
|  | Correlation Heatmap | Visual correlation matrix | ≥2 numeric columns |
|  | Scatter Plots | Interactive scatter plot generation | ≥2 selected features |
|  | Pairplot | Multi-variable relationship visualization | ≥2 numeric columns |

Table 2: “Advanced Analytics & AI Features”

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Feature** | **Description** | **Input Requirements** |
| **Advanced Statistics** | Skewness & Kurtosis | Distribution shape analysis | Numeric columns |
|  | Advanced Stats | Custom statistical calculations | Numeric columns |
|  | PCA Analysis | Principal Component Analysis with explained variance | ≥2 numeric columns |
|  | Spearman Correlation | Non-parametric correlation analysis | ≥2 numeric columns |
|  | Outlier Detection | Z-score and IQR methods for anomaly detection | Numeric columns |
|  | Feature Entropy | Information theory-based feature analysis | Numeric columns |
|  | K-Means Clustering | Silhouette scores for different cluster numbers | ≥2 numeric columns |
| **LLM Integration** | Model Selection | Choose from “GPT-4, GPT-4o, GPT-4o Mini” | ‘OpenAI’ API key |
|  | Statistical Insights | AI-generated explanations of statistics | Any analysis results |
|  | Data Quality Feedback | LLM assessment of data issues and recommendations | Dataset statistics |
|  | Relationship Analysis | AI explanation of feature relationships | Selected features |
|  | Feature Importance | LLM-driven feature significance analysis | Any dataset |
|  | Analysis Flow Suggestions | Automated recommendation of analysis steps | Dataset schema |
| **Feature Engineering** | Feature Store | Extract and save engineered features | Any dataset |
|  | Feature Suggestions | LLM recommendations for categorical encoding | Categorical columns |
|  | Preprocessing Advice | AI-suggested data preprocessing steps | Any dataset |
| **Time Series Analysis** | Trend Detection | Decomposition into trend, seasonal, residual | Datetime columns |
|  | Seasonal Analysis | Seasonal pattern identification | Time series data |
|  | Temporal Insights | Time-based data analysis | Datetime columns |
| **Machine Learning Guidance** | ML Approach Recommendations | AI suggestions for modeling strategies | Any dataset |
|  | Algorithm Selection | LLM-guided choice of appropriate techniques | Dataset characteristics |

Table 3: “System Integration & Performance Features”

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Feature** | **Description** | **Input Requirements** |
| **Interactive Analysis** | Column Selection | User-selectable features for relationship analysis | Multiple columns |
|  | Dynamic Visualization | Real-time plot generation based on selections | Selected features |
|  | Correlation Matrix | Interactive correlation analysis | Numeric columns |
| **Export & Download** | Feature Store Export | Download engineered features as CSV | Generated features |
|  | Dataset Profile Export | Download comprehensive analysis as JSON | Analysis results |
|  | Statistics Export | Download advanced statistics as CSV | Statistical analysis |
| **MLflow Integration** | Experiment Tracking | Automatic logging of all analysis runs | Any analysis |
|  | Artifact Storage | Save plots, statistics, insights | Generated artifacts |
|  | Run Comparison | Compare multiple analysis sessions | Multiple runs |
|  | MLflow UI Launch | One-click launch of MLflow dashboard | MLflow setup |
| **Performance Optimization** | Parallel Processing | Multi-threaded analysis for large files | Large datasets (>100MB) |
|  | Memory Management | Efficient handling of large datasets | Large files |
|  | Processing Time Tracking | Monitor and display analysis duration | Any analysis |

Table 4: “Tech Stack Used”

|  |  |  |
| --- | --- | --- |
| **Layer** | **Technology** | **Purpose/Role** |
| Frontend | Streamlit | Web UI for data analysis |
| Visualization | Matplotlib, Seaborn, Plotly | Data visualization |
| Backend | Python 3.8+ | Core programming language |
| Data | Pandas, NumPy | Data manipulation & processing |
| ML/Stats | SciPy, scikit-learn | Statistics & machine learning |
| LLM/AI | OpenAI API | LLM-powered insights (GPT-4/4o) |
| Tracking | MLflow | Experiment tracking & artifact mgmt |
| Storage | Local File System | Stores datasets & results |

# List of Figures

[Figure 1: “Top View” 11](#_Toc204169384)

[Figure 2: “System Architecture” 12](#_Toc204169385)

[Figure 3: “Data flow diagram” 12](#_Toc204169386)

[Figure 4: File Upload 17](#_Toc204169387)

[Figure 5: Basis statistics for a dataset 17](#_Toc204169388)

[Figure 6: LLM selection and Question answering 18](#_Toc204169389)

[Figure 7 : Auto visualizations 18](#_Toc204169390)

[Figure 8: Correlation heatmap 19](#_Toc204169391)

[Figure 9: LLM generated Insights 19](#_Toc204169392)

[Figure 10: Feature Store 20](#_Toc204169393)

[Figure 11: Feature Store 21](#_Toc204169394)

[Figure 12 : Correlation Matrix 21](#_Toc204169395)

[Figure 13 : PCA 21](#_Toc204169396)

[Figure 14 : Spearman Correlation 22](#_Toc204169397)

[Figure 15 : Feature Entropy 22](#_Toc204169398)

[Figure 16 : Silhouette Scores 22](#_Toc204169399)

[Figure 17 : Skewness for a dataset 23](#_Toc204169400)

[Figure 18 : Metrics on a dataset for GPT 4.1 24](#_Toc204169401)

[Figure 19 : Metric rating legends 24](#_Toc204169402)

[Figure 20 : Core Model as Python Library 26](#_Toc204169403)

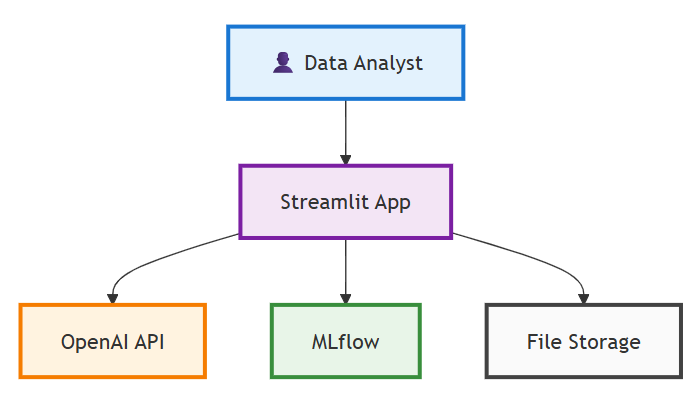


Figure : “Top View”

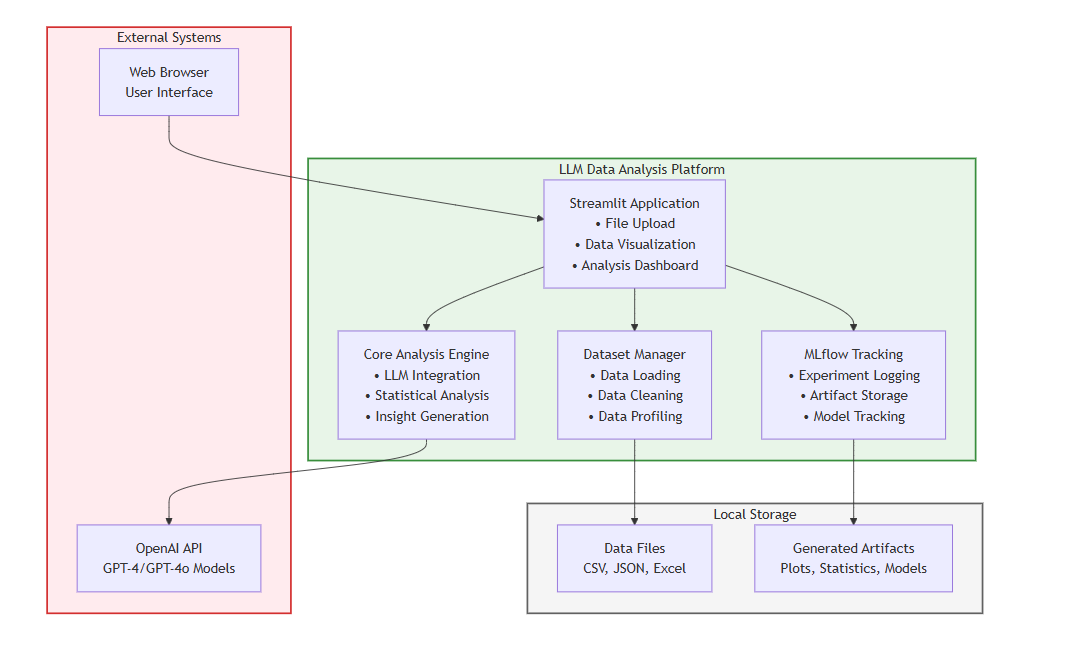


Figure : “System Architecture”

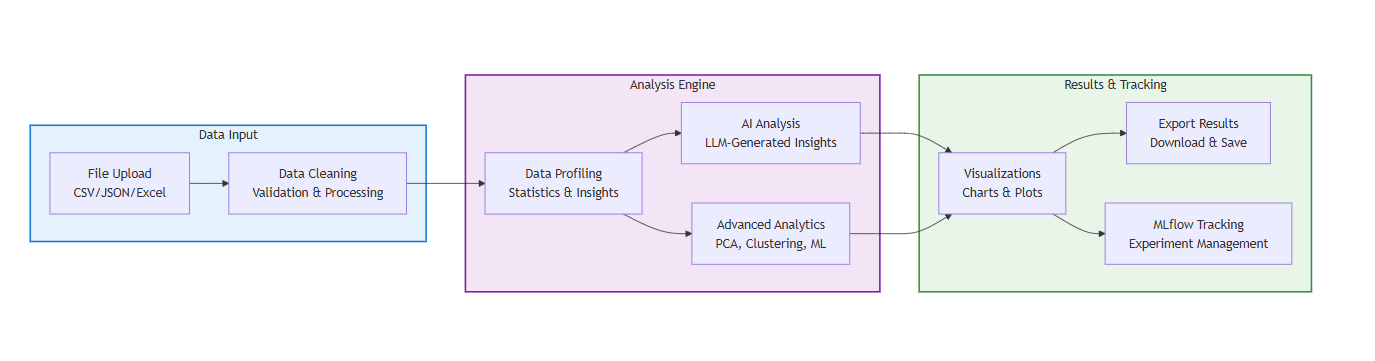


Figure : “Data flow diagram”

# Chapter 1 – Introduction and Objectives

## Introduction

The Core Model is a generalized analytical engine that uses LLMs to interpret diverse datasets and automate exploratory data analysis and insight generation.

The exponential growth of data in organizations has created a need for intelligent, automated data analysis tools that can provide access to data insights. Traditional data analysis approaches require specialized statistical knowledge and programming skills, creating barriers for domain experts who understand the business context but lack technical expertise.

Recent advancements in Large Language Models (LLMs) and computational power have opened new arenas for making data analysis more accessible through natural language interfaces and automated insight generation. These models can bridge the gap between complex statistical procedures and human-interpretable explanations, enabling a broader range of users to extract meaningful insights from their data.

# Scope of Work

The scope includes designing and implementing a system that:

* Utilize Large Language Models (LLMs) to semantically interpret input datasets of varying types, including structured (CSV, Excel), semi-structured (JSON, logs), and unstructured text.
* Combine LLMs with traditional machine learning techniques (e.g., clustering, statistical modelling) to perform exploratory data analysis (EDA), detect anomalies, and extract meaningful patterns without domain-specific customization.
* Support a modular input/output interface, where users provide sample datasets and receive automatically generated insights in a structured format—highlighting both expected trends and previously undiscovered correlations or patterns.
* Provide extendable support for model updates, allowing the pipeline to adapt to improvements in LLMs and machine learning models without requiring re-engineering.

# Primary Objectives

* Design and implement a web-based data analysis platform using Streamlit.
* Integrate OpenAI's LLM capabilities for generating intelligent data insights
* Develop comprehensive data processing and statistical analysis modules
* Implement MLflow for experiment tracking and artifact management
* Create an intuitive user interface for non-technical users
* Validate the system with multiple data formats (CSV, JSON, Excel)

## Objectives Met till Midterm

1. **COMPLETED** - Designed and implement a web-based data analysis platform using Streamlit framework. Migration to Angular is work in progress.
2. **COMPETED** - Requirement analysis completed.
3. **COMPLETED** - Integrated OpenAI's GPT (4.1 and 4-O) models for generating intelligent, contextual data insights.
4. **COMPLETED** - Developed comprehensive data processing modules supporting multiple formats (CSV, JSON, Excel)
5. **COMPLETED** - Implemented statistical analysis capabilities including descriptive statistics, correlation analysis, and advanced methods (PCA, clustering)
6. **COMPLETED** - Integrate MLflow for comprehensive experiment tracking and artifact management

## Post Mid Term

1. **IN PROGRESS** - Developing advanced feature engineering and recommendation capabilities
2. **PLANNED** - Conduct extensive user testing and performance evaluation
3. **PLANNED** – Benchmarking and risk register
4. **PLANNED** - Deploy the system for real-world validation

**Detailed Plan of Work** (for 16 weeks)

Table 5: Detailed Plan of Work

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Serial Number of Task/Phases | **Tasks or subtasks to be done** | **Start Date (NA)-End Date (16thAug 2025)** | Planned duration in weeks | Specific Deliverable in terms of the project |
|  | Requirement analysis, LLM selection & dataset curation | Week 1-2 | 2 Week | Problem statement, dataset repository, LLM options |
|  |
|  |
| 1 |
|  |
|  |
|  |
|  |
|  |
| 2 | Data ingestion pipeline & format handling | Week 3-4 | 2 Weeks | Unified parser for CSV, JSON, and text formats |
| 3 | LLM integration for semantic analysis & EDA | Week 5-7 | 3 Weeks | LLM-driven EDA, summary stats, outlier detection |
| 4 | Insight generation, classification & benchmarking | Week 8-11 | 4 Weeks | Categorized insights; benchmarking reports |
| 5 |  | Week 13-14 | 2 weeks | Validation across 3+ data domains |
|  |
| Cross-domain testing and optimization |
| 6 | UI/API pipeline developments | Week 14-15 | 2 Weeks |  |
|  |
| Interactive |
| interface |
| and/or API |
| access |
| 7 | Final evaluation and documentation | Week 16 | 1 Week | Final prototype, evaluation metrics, demo, report |

# Chapter 2 - System Design and Architecture

## System Requirements

Functional Requirements:

1. Data Import: Support for CSV, JSON, and Excel file formats
2. Data Processing: Automated cleaning, validation, and profiling
3. Statistical Analysis: Descriptive statistics, correlation analysis, advanced methods
4. LLM Integration: Automated insight generation and explanations
5. Visualization: Charts and plots
6. Export Capabilities: Download results and artifacts

Non-Functional Requirements:

1. Performance: Handle datasets up to 100Mb efficiently
2. Usability: Intuitive interface for non-technical users

## Component Design

1. Streamlit Interface Layer

* File upload handlers
* Interactive widgets for parameter selection
* Real-time visualization display
* Results export functionality

## Core Model Engine

* LLM integration and prompt management
* Statistical analysis orchestration
* Insight generation and interpretation
* Model selection and recommendation

## Dataset Manager

* Multi-format data loading
* Data cleaning and validation
* Statistical profiling
* Feature engineering

## Data Processing Pipeline

The data processing follows a standardized pipeline:

1. Data Ingestion: File upload and format detection
2. Validation: Schema validation and quality checks
3. Cleaning: Missing value handling, outlier detection
4. Profiling: Statistical summary generation
5. Analysis: Core statistical computations
6. Insight Generation: LLM-powered explanations
7. Visualization: Chart and plot creation
8. Tracking: MLflow experiment logging

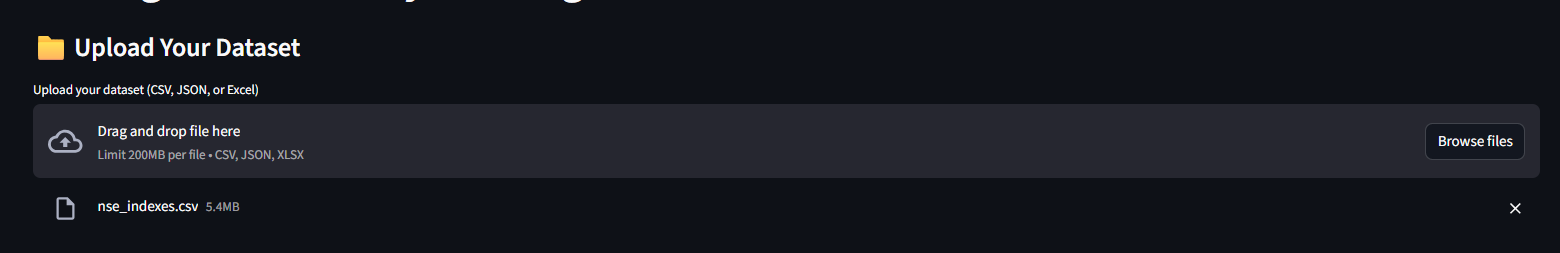


Figure : File Upload

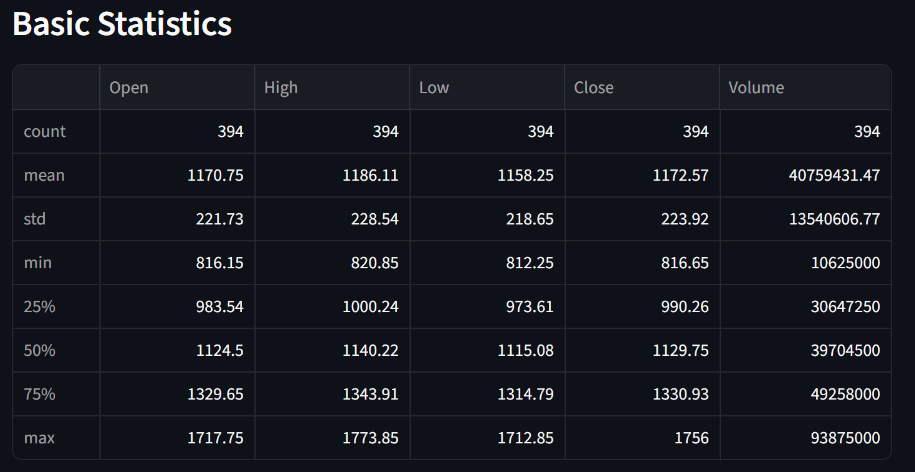


Figure : Basis statistics for a dataset

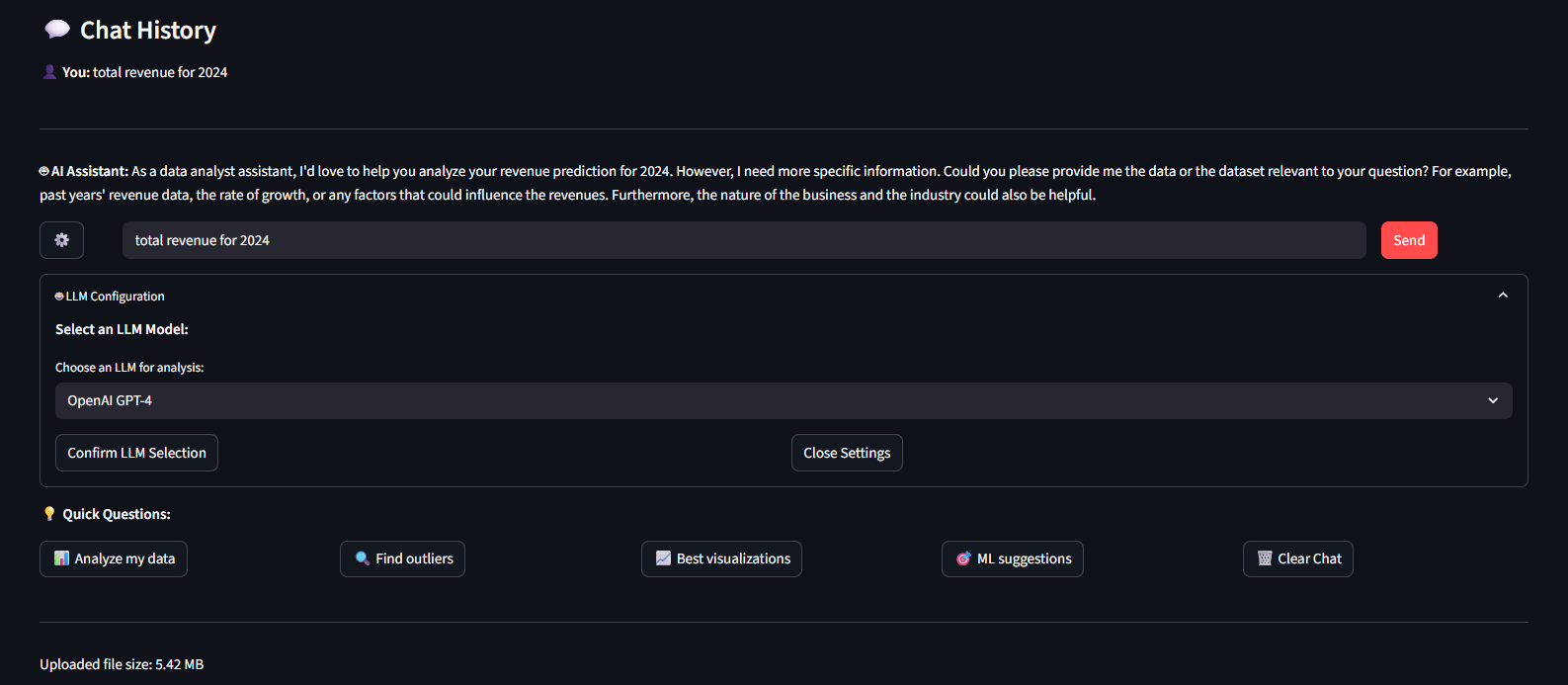


Figure : LLM selection and Question answering

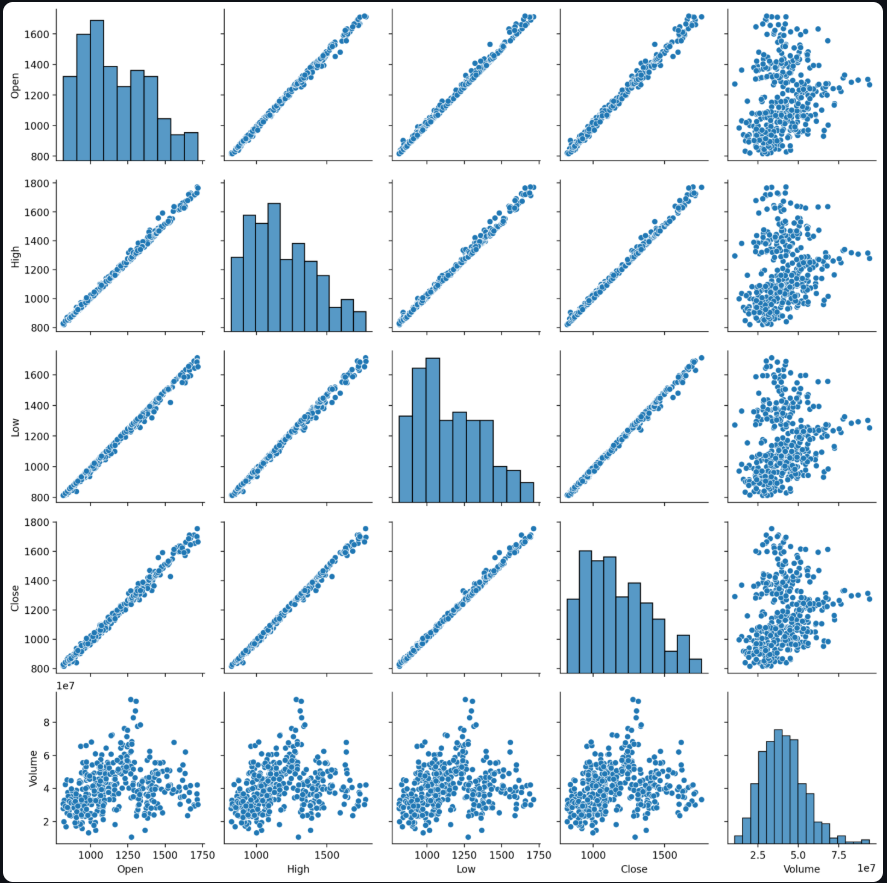


Figure : Auto visualizations

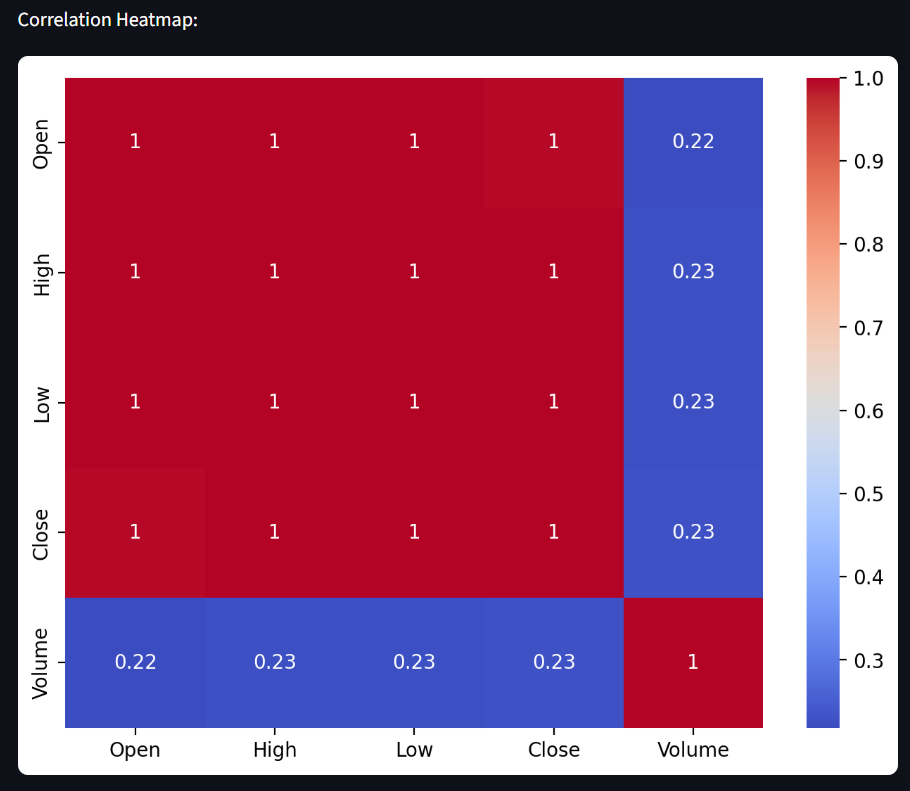


Figure : Correlation heatmap

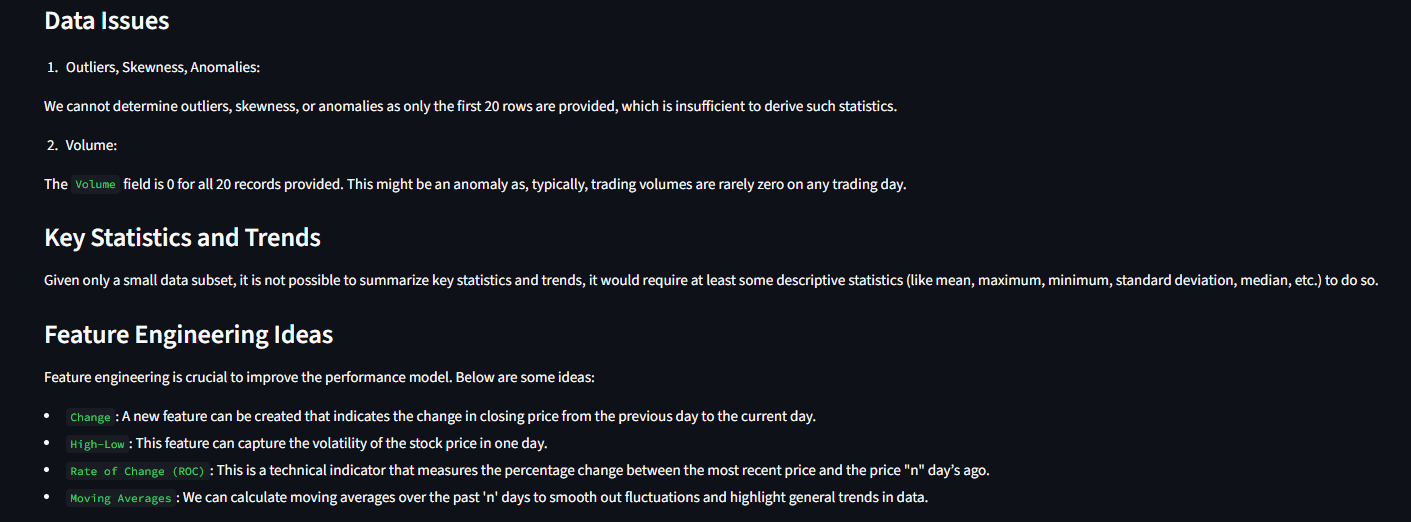


Figure : LLM generated Insights

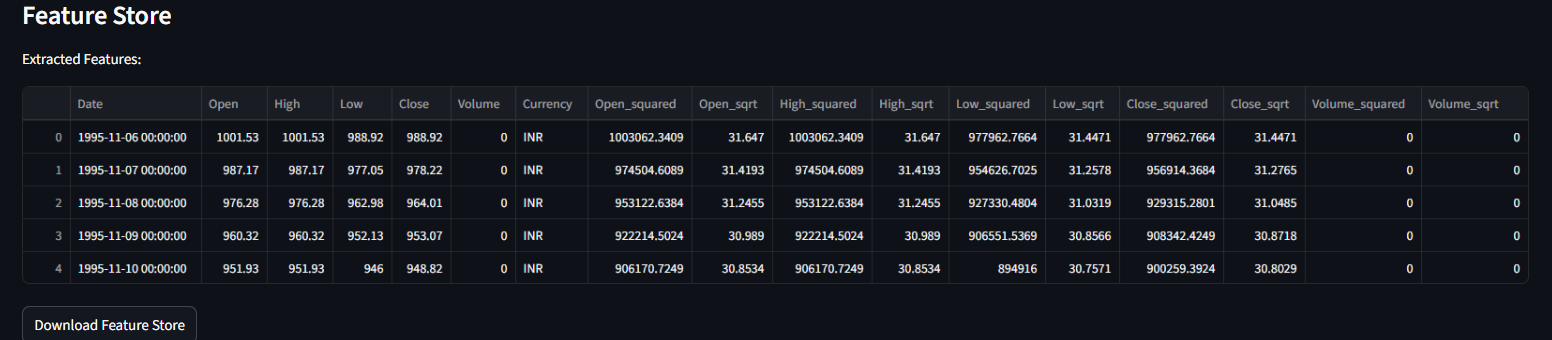


Figure : Feature Store

## Prompt Engineering

Developed specialized prompts for:

* Data quality assessment
* Statistical result interpretation
* Feature relationship analysis
* Recommendation generation

## Statistical Analysis Modules

1. Descriptive Statistics

* Basic statistics (“mean, median, mode, std dev”)
* Distribution analysis (skewness, kurtosis)
* Missing value analysis
* Data type profiling

1. Advanced Analytics

* “Principal Component Analysis” (PCA)
* “K-means” clustering along with ‘silhouette analysis’
* Correlation analysis (Pearson, Spearman)
* Outlier detection (Z-score, IQR methods)

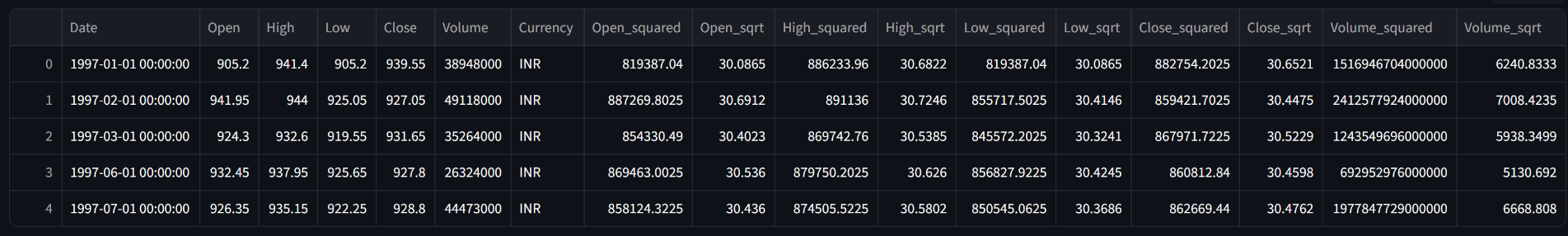


Figure : Feature Store

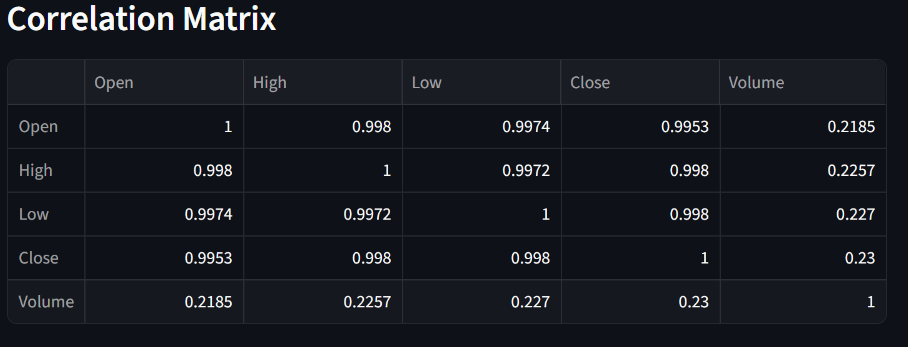


Figure : Correlation Matrix

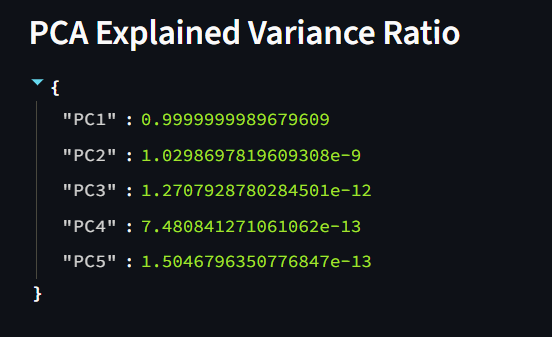


Figure : PCA

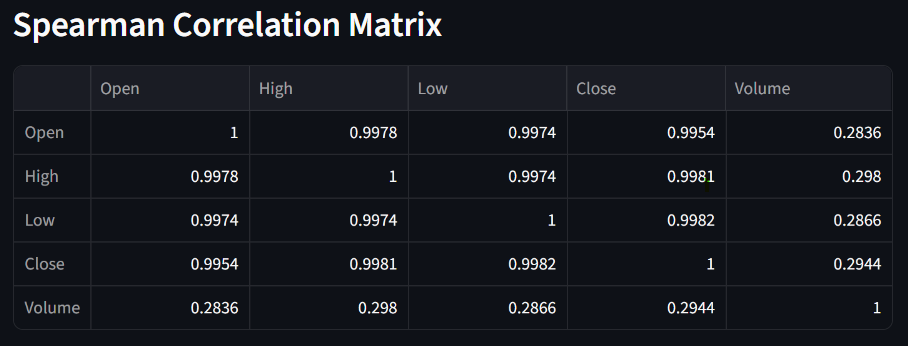


Figure : Spearman Correlation

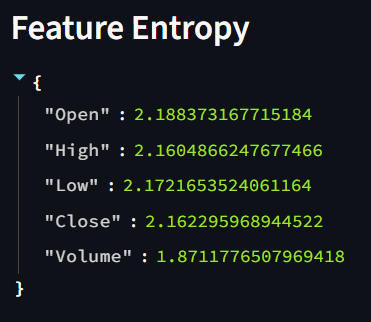


Figure : Feature Entropy

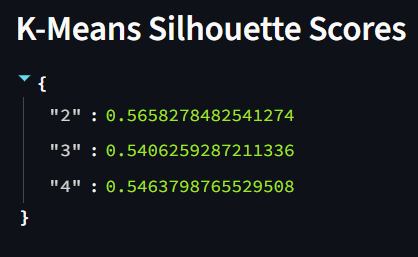
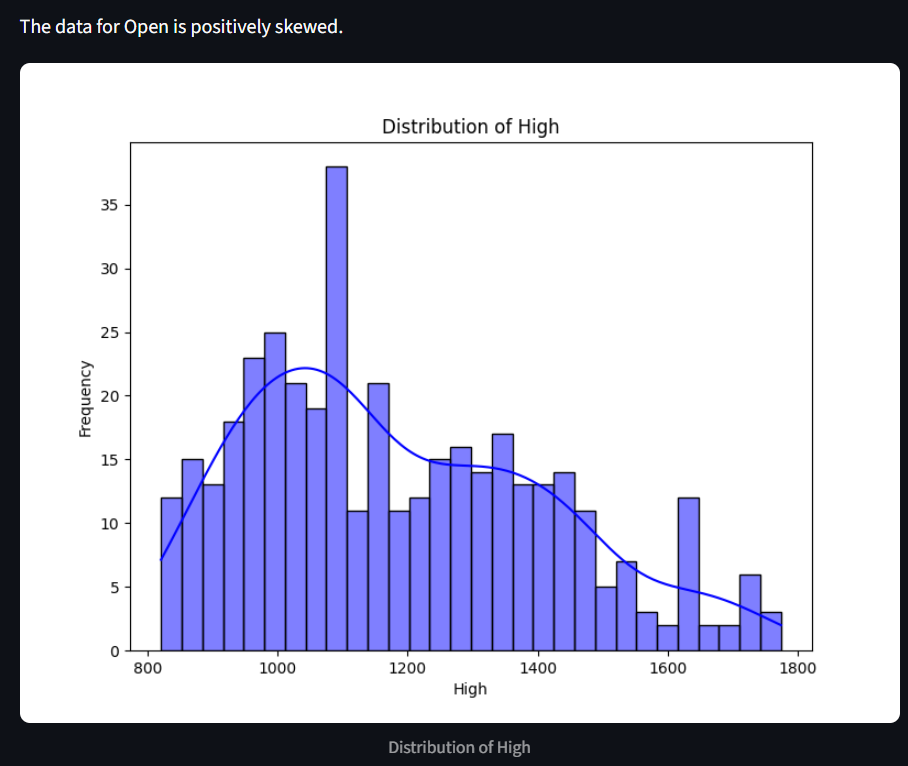
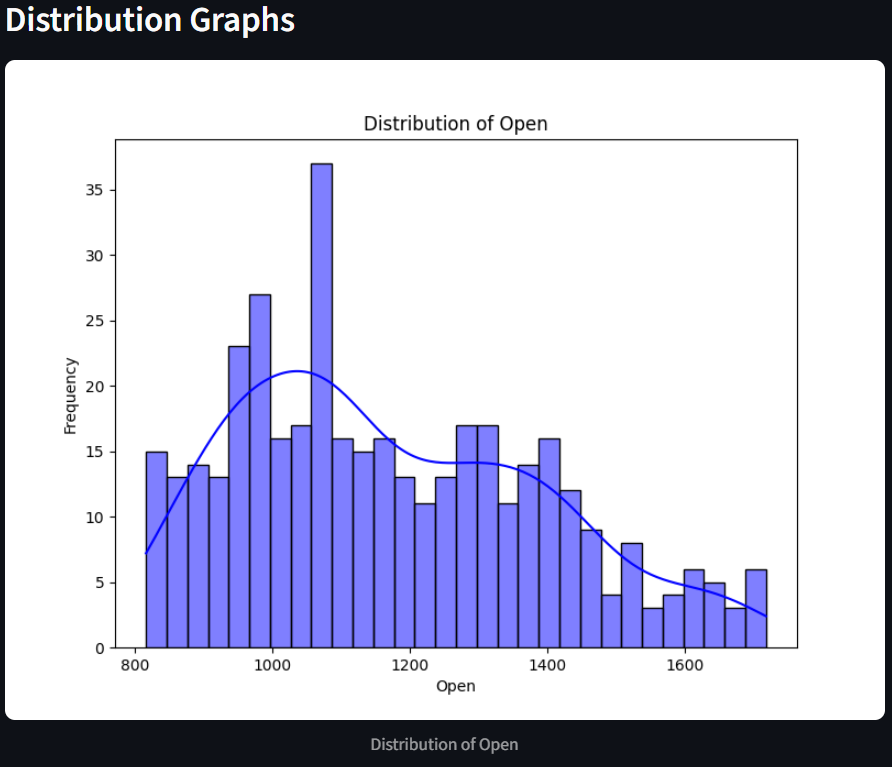


Figure : Silhouette Scores

1. Visualization Engine

* Distribution plots with statistical annotations
* Correlation heatmaps
* Scatter plots and pair plots



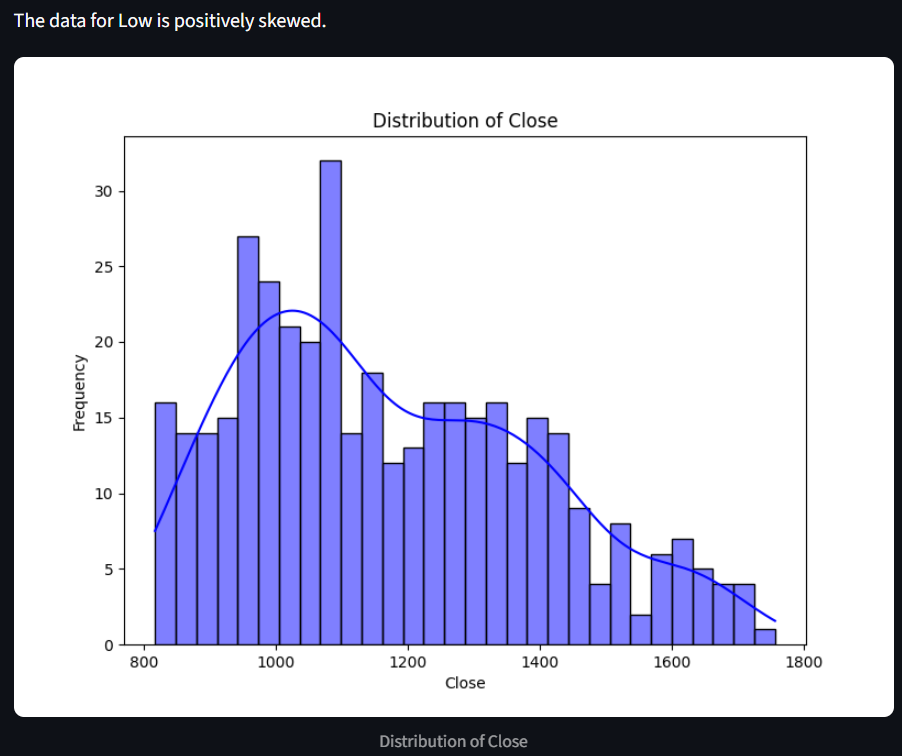
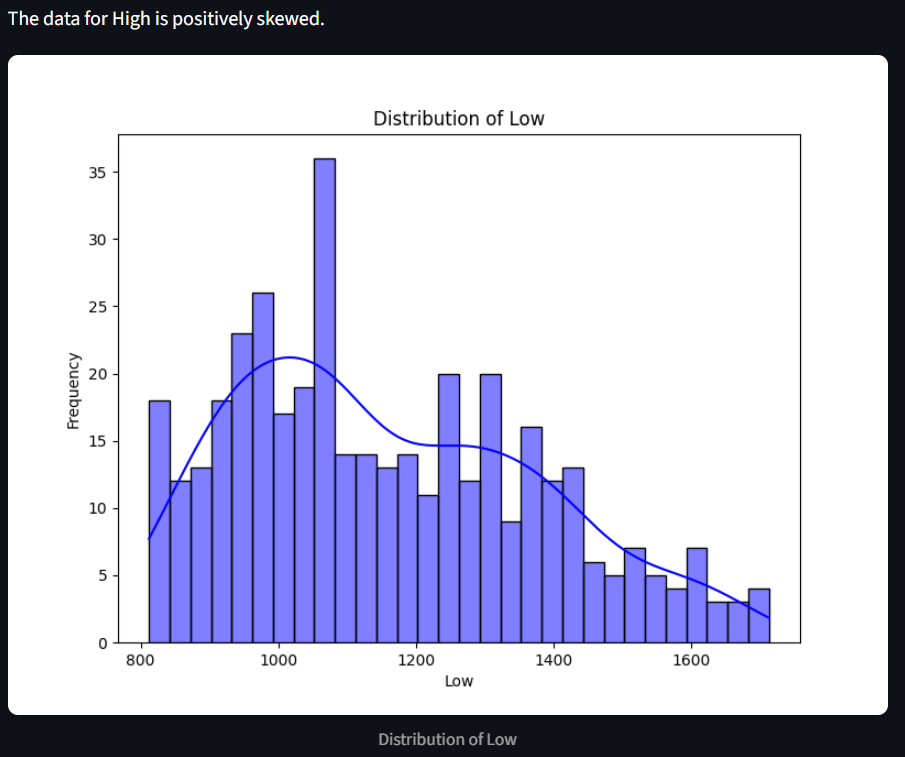


Figure : Skewness for a dataset

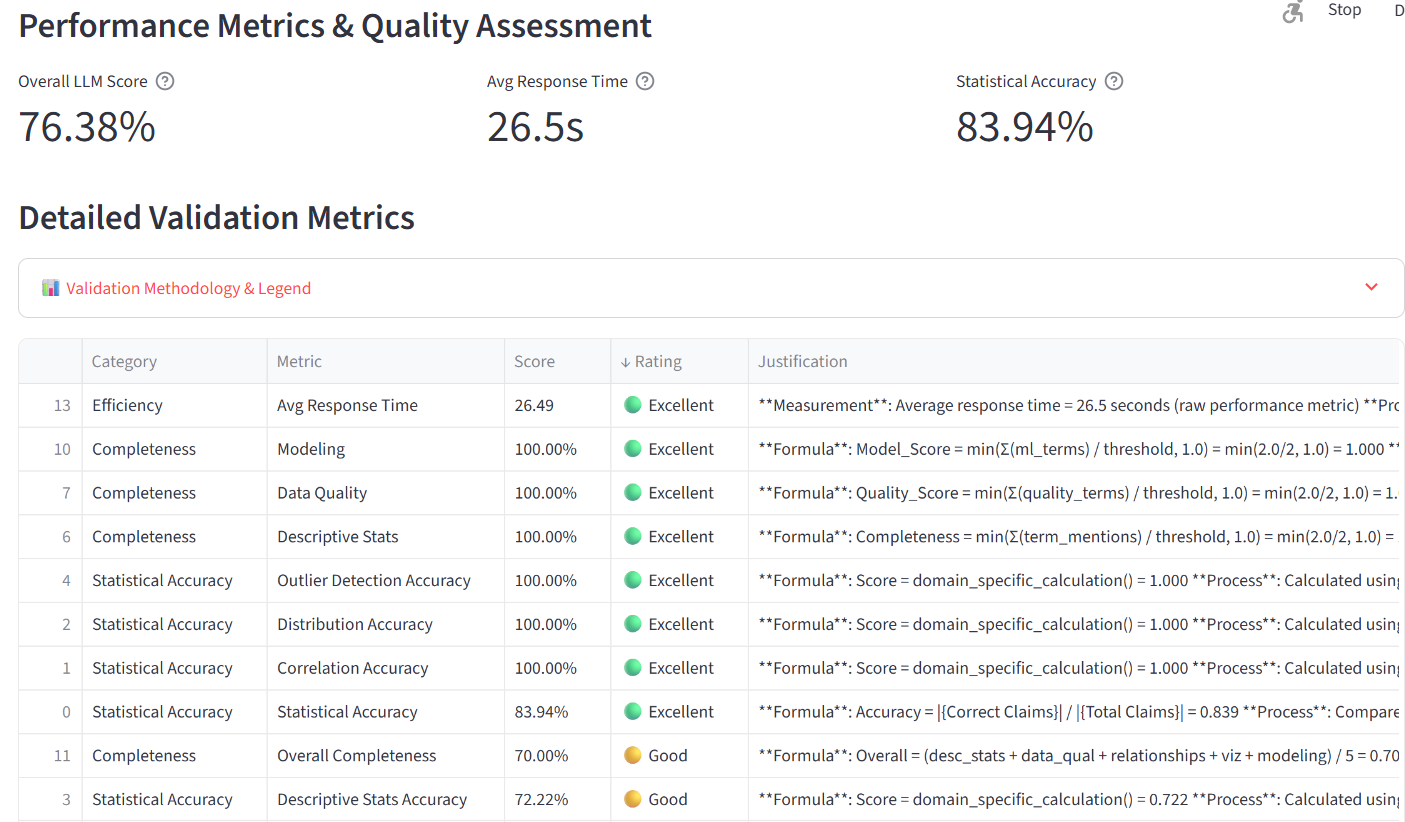


Figure : Metrics on a dataset for GPT 4.1

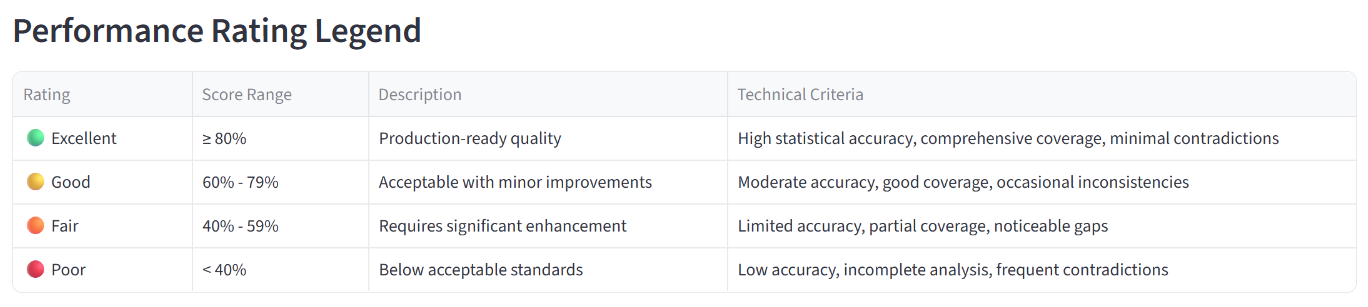


Figure : Metric rating legends

# Core Model usage at BNY

At BNY, the **Model Review Management (MRM)** process plays a pivotal role in ensuring the integrity, transparency, and reproducibility of machine learning solutions. The MRM team's primary responsibility is to monitor, review, and validate any data-driven models or analytical solutions developed across the organization. A key principle of this process is that any approved model must be fully reproducible and auditable.

Some of the things that MRM team validates for each model are: -

* 1. Data field relevance
  2. Feature analysis
  3. Data field transformations
  4. Range of data types
  5. Class Imbalance Checks
  6. Choice of technique in model/solution design
  7. Feature engineering performed
  8. PCA
  9. Feature Correlation matrix
  10. Data distribution trends

The core model is capable of providing and storing all such details instantly.

# Purpose of the Core Engine

To streamline this review process and enhance analytical reliability, a centralized Core Model Engine has been developed. This engine acts as a validation and analytical support system for all teams performing exploratory data analysis (EDA) as part of their model development lifecycle. The core engine:

* Verifies and replicates statistical analysis conducted by the teams.
* Supports validation of data insights using Large Language Models (LLMs).
* Enables inspection of data statistics, visualizations, Principal Component Analysis (PCA), feature correlations, and temporal data changes.
* Logs and stores all analyses in MLflow, ensuring historical traceability and aiding in MRM audits.
* This LLM-powered engine reduces manual effort for the MRM team by automatically validating key aspects like feature correlation, data drift, and insight consistency.

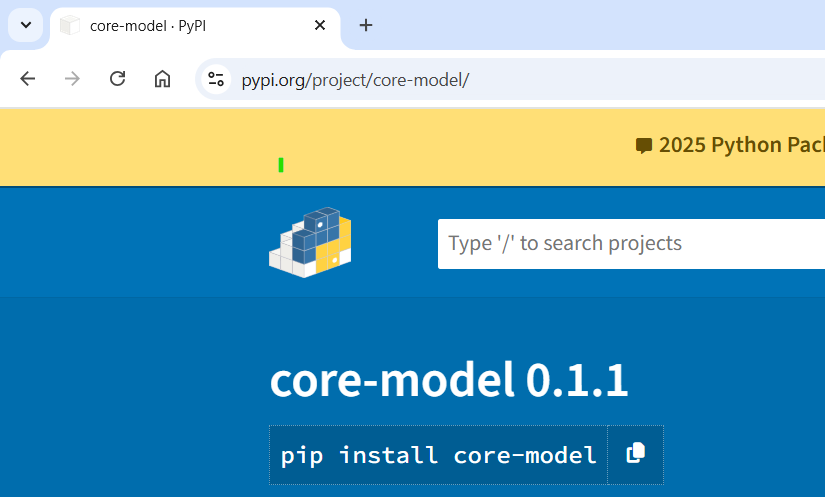


Figure : Core Model as Python Library

## Technical Validation Framework

Our LLM validation employs multi-dimensional assessment using statistical ground truth comparison and domain-specific evaluation metrics.

## Mathematical Foundation:

* Overall Score: Σ(normalized\_metric\_i) / n where n = total metrics, each metric ∈ [0,1]
* Statistical Accuracy: |{correct\_claims}| / |{total\_claims}| using Pearson correlation analysis
* Completeness: Σ(component\_coverage\_i) / |components| across 5 analysis domains
* Consistency: 1 - (contradictions\_detected / total\_contradiction\_checks)
* Efficiency: Tier-based scoring using response time quantiles

## Statistical Tests Applied:

* Shapiro-Wilk Test: H₀: data follows normal distribution (α = 0.05)
* Pearson Correlation: r = Σ((xᵢ-x̄)(yᵢ-ȳ)) / √(Σ(xᵢ-x̄)²Σ(yᵢ-ȳ)²)
* Fisher's Skewness: γ₁ = E[(X-μ)³]/σ³ for distribution asymmetry detection
* Z-Score Outlier Detection: |z| = |(x-μ)/σ| > 3 threshold

## Metric Categories Explained

### Statistical Accuracy (Weight: 25%)

* Validates LLM claims against computed dataset statistics
* Uses tolerance-based matching for numerical assertions
* Employs fuzzy string matching for categorical claims

### Completeness (Weight: 30%)

* Measures coverage across 5 domains: descriptive stats, data quality, relationships, visualization, modeling
* Binary presence scoring: min(mentioned\_terms / threshold, 1.0)
* Holistic assessment ensuring comprehensive analysis

### Consistency (Weight: 25%)

* Detects logical contradictions using opposing term pairs
* Implements semantic contradiction detection algorithms
* Ensures coherent analytical narrative

### Efficiency (Weight: 20%)

* Tier-based response time evaluation
* Balances quality with computational performance
* Considers user experience and system scalability

## Use Cases and Applications at BNY

1. **Volume Prediction in SWIFT Transactions**

One major implementation of the core model is in the domain of volume prediction for SWIFT MT and MX messages. BNY processes approximately 5 to 6 million SWIFT messages daily, across various transaction types such as payments, reimbursements, and confirmations.

The core engine validates the EDA results and ensures model readiness before deployment. Supporting MRM team in their model review process.

1. **NACK Message Analysis and Revenue Impact**

Another critical use case is the detection of anomalies in NACK (Negative Acknowledgment) messages. These messages indicate transaction failures due to formatting errors or rule violations during MT/MX validations on the SWIFT platform. NACK messages represent missed revenue opportunities, prompting the need for robust analysis.

The core model assists in identifying outliers or unusual spikes in NACK volumes.

It helped the team to do the analysis for building the solution for NACK anomalies with their EDA.

# Role in MRM Governance

* This solution is heavily integrated with the MRM governance process, offering:
* Reproducibility: All model artifacts, parameters, and analysis outputs are logged in MLflow.
* Auditability: Any model reviewed by MRM can be recreated from scratch using stored runs.
* LLM Augmentation: LLMs assist in explaining statistical outputs, generating natural language summaries of data behavior, and detecting inconsistencies.
* Programmatic Access: The platform currently offers Python-based APIs and is transitioning to a full-stack implementation using Angular (frontend) and Spring Boot with Spring AI (backend), making it scalable and enterprise-ready.

# How core model works

**The Core Model operates in two primary modes: as a Python library and as a user-friendly web-based interface.**

Users can input their datasets directly into the analyze () function of the Core Model by specifying the type of analysis required through a prompt. Additionally, they can provide an example output format to guide the structure of the results. This flexible design allows for both programmatic integration and interactive exploration, making it suitable for a wide range of analytical use cases.

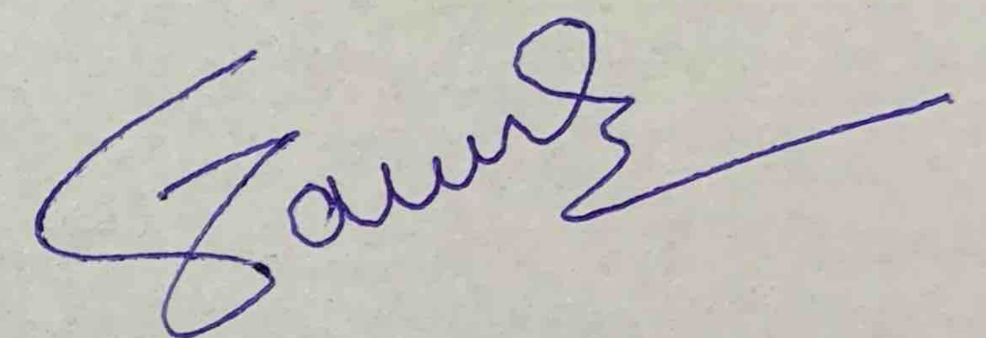
# Directions for Future Work

* Complete benchmarking across domains.
* Develop API/UI access.
* Risk Register.
* DVC
* Finalize evaluation and documentation.
* Publish code as library with more updates.

**Supervisor’s Rating of the Technical Quality of this Dissertation Outline**

EXCELLENT / GOOD / FAIR/ POOR (Please specify): \_\_\_\_\_\_\_Excellent\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Supervisor’s suggestions and remarks about the outline (if applicable).**

Date\_\_\_\_\_\_\_24/May/2025\_\_\_\_\_\_  (Signature of Supervisor)

Name of the supervisor: Janardan Jayaraman

Email Id of Supervisor: Janardan.Jayaraman@bny.com/JanardanJ@gmail.com

Mob # of supervisor: +1-412-519-6004

# Bibliography / References

The following are referred journals from the preliminary literature review.

1 Brown, T., Mann, B., Ryder, N., et al. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33, 1877–1901. https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html

2 Chen, M., Tworek, J., Jun, H., et al. (2021). Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374. “https://arxiv.org/abs/2107.03374”

3 McKinney, W. (2010). Data structures for statistical computing in Python. Proceedings of the 9th Python in Science Conference, 51–56. “https://conference.scipy.org/proceedings/scipy2010/pdfs/mckinney.pdf”

4 Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830. “https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html”

5 Sculley, D., Holt, G., Golovin, D., et al. (2015). Hidden technical debt in machine learning systems. Advances in Neural Information Processing Systems, 28, 2503–2511. “https://proceedings.neurips.cc/paper/2015/hash/86df7dcfd896fcaf2674f757a2463eba-Abstract.html”

7 Zaharia, M., Chen, A., Davidson, A., et al. (2018). Accelerating the Machine Learning Lifecycle with MLflow. SIGMOD Conference. <https://dl.acm.org/doi/10.1145/3183713.3190662>

8. Jain, A., Kumar, R., et al. (2025). An LLM-Based Approach for Insight Generation in Data Analysis. arXiv preprint arXiv:2503.11664.  
<https://arxiv.org/abs/2503.11664>

9. Singh, P., Zhao, L., et al. (2024). Towards Automated Cross-domain Exploratory Data Analysis through Large Language Models. arXiv preprint arXiv:2412.07214.  
<https://arxiv.org/abs/2412.07214>