PROJECT DOCUMENT

TITLE : Market Data Analytics Price Challenge Reporting

The Market Data Analytics Price Challenge Reporting project is a comprehensive data-driven solution designed to analyze and evaluate the accuracy, efficiency, and reliability of market data provided by multiple financial data vendors. The system automates the end-to-end process of data collection, cleansing, transformation, aggregation, and visualization, enabling financial auditors and analysts to identify price variations, detect outliers, and assess vendor performance objectively.

ABSTRACT :

The Market Data Analytics Price Challenge Reporting project is an industry-oriented analytical system designed to evaluate the accuracy, efficiency, and reliability of multiple market data providers. In the financial ecosystem, auditors and analysts depend on various vendor feeds for market pricing and valuation. However, due to differences in pricing methodologies and calculation rules, these feeds often produce discrepancies that can impact decision-making and financial reporting. The importance of this project lies in its ability to identify such deviations, assess data consistency, and promote transparency in the pricing mechanisms used by vendors.

The project employs a systematic methodology integrating data engineering, statistical analysis, and business intelligence visualization. Market data was extracted from multiple APIs including Alpha Vantage, Yahoo Finance, Polygon.io, Twelve Data, and Stock Data and processed using Python-based ETL pipelines. The extracted data underwent cleaning, transformation, and aggregation to ensure uniformity and accuracy. Statistical techniques such as mean, median, standard deviation, and percentage variance were applied to detect anomalies, while Power BI was used to visualize Key Performance Indicators (KPIs) like price variance, precision percentage, and outlier distribution across vendors. This combination of automated ETL and visual analytics enables comprehensive monitoring of market data feeds and vendor performance.

The major results from the analysis revealed measurable differences in vendor reliability and pricing accuracy. Vendors with lower outlier percentages and higher precision metrics demonstrated better data stability and quality. The Power BI dashboard effectively visualized these findings, allowing users to interactively filter and explore trends over time. The system successfully identified outlier patterns, validated statistical consistency across vendors, and generated comparative insights that can guide data-driven improvements in vendor selection and pricing validation.

INTRODUCTION :

In the modern financial ecosystem, accurate and consistent market data plays a crucial role in maintaining transparency and efficiency in financial decision-making. Organizations, investors, and auditors rely heavily on market data providers for real-time pricing information of securities, currencies, and other financial instruments. However, since each vendor follows distinct methodologies and data processing algorithms, inconsistencies or deviations often arise between different market feeds. Such discrepancies can lead to incorrect pricing decisions, valuation errors, and misinformed financial analysis.

The Market Data Analytics Price Challenge Reporting project was developed to address these challenges by implementing a data-driven analytical system that measures the efficiency, accuracy, and precision of various market data vendors. The system performs statistical comparison and visualization of data feeds from multiple sources to evaluate their reliability and correctness. By automating data extraction, transformation, and reporting, it ensures that auditors and analysts can monitor vendor performance and identify anomalies in pricing data with greater accuracy and efficiency.

The project integrates three fundamental pillars of modern analytics Data Engineering, Statistical Computing, and Business Intelligence to create a robust analytical framework:

Data Engineering:

This component focuses on the systematic collection, cleaning, transformation, and organization of large volumes of data from multiple APIs such as Alpha Vantage, Yahoo Finance, Polygon.io, Twelve Data, and Stock Data. Using Python-based ETL (Extract, Transform, Load) processes, the raw datasets are standardized and merged into structured formats suitable for analysis. Data engineering ensures data quality, consistency, and readiness for analytical operations by handling missing values, duplicates, currency conversions, and other inconsistencies. It forms the backbone of the project by enabling a smooth flow of accurate and uniform data into the analytical layer.

Statistical Computing:

Statistical computation forms the analytical core of the project. Once the data is processed, statistical techniques such as mean, median, variance, and standard deviation are applied to identify pricing trends, deviations, and anomalies. Price variation is calculated as a percentage difference from the mean price, allowing auditors to detect outliers and inconsistencies across vendors. The project also defines outlier buckets (e.g., <3%, 3–5%, >5%) to categorize vendor performance levels based on deviation severity. This quantitative foundation provides the evidence necessary to assess vendor reliability and performance objectively.

Business Intelligence (BI):

Business Intelligence adds a visualization and interpretative layer to the project by transforming statistical outputs into interactive insights. Using Power BI, the project delivers dynamic dashboards and reports that visualize Key Performance Indicators (KPIs) such as outlier percentage, vendor precision, and price variance trends. These dashboards empower auditors, analysts, and decision-makers to filter data by date, price type, vendor, or exchange, and to drill down into specific performance metrics. The integration of a Q&A visual feature further enhances accessibility, allowing users to explore insights through natural language queries, thereby bridging the gap between complex analytics and business understanding.

Together, these three components create a comprehensive analytical ecosystem where raw financial data is transformed into actionable intelligence. The project not only automates the complex process of vendor performance evaluation but also enhances data transparency and supports real-time monitoring of market behavior.

Furthermore, this work is closely related to existing research and industry practices focused on data auditing, financial risk analytics, and vendor benchmarking. Prior studies have emphasized the importance of data validation mechanisms to ensure the integrity of financial data, but many lack the integration of automated ETL pipelines with visual reporting. This project bridges that gap by combining data engineering automation with real-time analytical visualization, offering a scalable and reusable framework for auditing financial market data.

METHODS AND MATERIALS :

The Market Data Analytics Price Challenge Reporting project was carried out in several well-defined phases. Each phase involved specific activities, tools, and techniques that helped in collecting, cleaning, analyzing, and visualizing market pricing data. The entire project was implemented using Python for data processing and Power BI for reporting and visualization.

Phase 1: Data Collection

The first phase of the project involved collecting market pricing data from multiple vendor APIs to form the foundation for analysis and reporting. The data was sourced from five reputable market data providers Alpha Vantage, Yahoo Finance, Polygon.io, Twelve Data, and Stock Data. Each of these vendors delivers market feeds related to various financial instruments such as company stocks, currencies, and commodities.

The purpose of this phase was to gather comparable pricing data from different sources in order to evaluate the accuracy, consistency, and reliability of each vendor. Since every data provider applies its own pricing algorithms and update frequencies, it was essential to collect the same type of information from multiple sources for meaningful comparison.

The datasets collected from these APIs included the following key attributes:

- Security ID: A unique identifier for each security (e.g., AAPL for Apple Inc., TSLA for Tesla Inc.).
- Vendor ID and Vendor Code: Used to distinguish between data providers.
- Price Type: Indicates the type of market price such as Open, Close, High, or Low.
- Exchange Code: Represents the stock exchange where the transaction took place (e.g., NASDAQ, NYSE).
- Currency Code: Specifies the currency in which the price is quoted.
- Price Date: The date corresponding to each price record.
- Price: The actual market value of the security for that specific date and price type.
- Conversion Rate: The rate used to convert all prices to a common base currency (USD).

The data collection process was automated using Python scripts that connected to the APIs through HTTPS requests. These scripts utilized Python libraries such as Requests and Pandas to extract, parse, and save the data into structured files. Automation ensured consistency, reduced manual effort, and allowed efficient handling of large data volumes.

Each vendor's dataset was stored in a separate CSV file for clarity and traceability. The files were named according to the respective data providers, as listed below:

- Alpha-vintage.csv
- Yahoo-data.csv
- Poly-io-data.csv
- Twelve-data.csv
- Stock-data.csv

These CSV files served as the raw datasets, representing unprocessed data collected directly from each vendor. No modifications or transformations were applied at this stage. The raw data from this phase formed the input for the next step — data cleaning and pre-processing, where the datasets were standardized, merged, and prepared for analytical processing.

Phase 2: Data Pre-processing and Cleaning

After data collection, the next crucial step was data pre-processing and cleaning. Since the datasets were obtained from multiple vendor sources, it was essential to ensure that all files followed a uniform structure and consistent format before performing any analytical comparisons. This process helped to remove inconsistencies, correct formatting issues, and prepare the data for accurate analysis.

All cleaning and transformation tasks were performed using Python programming, primarily utilizing the Pandas and NumPy libraries. These tools provided efficient methods for handling missing data, managing data types, and performing structured data transformations.

The main steps carried out during this phase are as follows:

Handling Missing Values: The datasets were checked for missing or null entries in critical columns such as Price, Currency Code, and Price Date. Missing values were either filled with appropriate defaults (such as the mean or median value) or removed when they could not be corrected logically. This ensured that the analysis would not be affected by incomplete records.

Removing Duplicates: Duplicate entries were identified and removed to maintain the uniqueness of each record. Each observation was expected to be unique based on the combination of Security ID, Vendor ID, Price Date, and Price Type. Removing duplicate records helped in maintaining data integrity and avoiding redundancy.

Standardizing Formats: The collected datasets had slight variations in format due to different vendor data structures. All date values were standardized into a single format (YYYY-MM-DD), and numerical values such as Price and Conversion Rate were converted to floating-point numbers to ensure consistency in further calculations.

Currency Conversion: Since the prices from different vendors were sometimes quoted in different currencies, all prices were converted into a common base currency (USD). This was achieved by multiplying each price by its respective conversion rate provided in the dataset. This conversion made cross-vendor price comparison accurate and meaningful.

Merging Vendor Data Files: After cleaning and standardizing the individual datasets, all vendor files were merged into a single consolidated dataset named combined_market_data.csv. This combined file contained consistent pricing records for all vendors and acted as the master dataset for subsequent analysis and reporting.

The data pre-processing and cleaning phase ensured that every record was accurate, uniform, and comparable across all vendors. By the end of this stage, the dataset was free from missing values, duplicates, and inconsistencies, making it suitable for advanced data aggregation, analysis, and visualization in the following phases.

Phase 3: Data Aggregation and Transformation

After completing the data cleaning process, the next step was data aggregation and transformation. This phase focused on deriving meaningful statistical insights from the standardized dataset by performing various calculations and summarizations. The purpose of this phase was to prepare analytical data that could later be visualized in Power BI to assess vendor performance and price accuracy.

The process was implemented using ETL (Extract, Transform, and Load) techniques through Python, primarily utilizing the Pandas and NumPy libraries. The cleaned data from multiple vendors was aggregated and transformed into a structured reporting format containing several Key Performance Indicators (KPIs). The main computations performed during this phase included the following:

Mean Price: The average price was calculated for each Security ID across all vendors for a specific Price Date and Price Type. This helped to establish a benchmark value against which individual vendor prices could be compared.

Median Price: The median value was determined to represent the central tendency of prices across vendors. Unlike the mean, the median is less affected by extreme outliers and therefore provides a stable reference for comparison.

Standard Deviation: The standard deviation was computed to measure the degree of variation or dispersion of prices among vendors. A lower standard deviation indicates consistent pricing, while a higher value reflects greater discrepancies between vendor data.

Price Variation (%): The price variation percentage was calculated for each vendor to determine how far their price deviated from the overall mean price. This metric was critical for identifying vendors whose prices were significantly higher or lower than the market average. This was computed using the formula:

$$\text{Price Variation (\%)} = \frac{(\text{Vendor Price} - \text{Mean Price})}{\text{Mean Price}} \times 100$$

Outlier Buckets: To categorize vendor performance based on deviation levels, each price variation percentage was grouped into outlier categories:
Less than 3% variation: High accuracy (Good performance), Between 3% to 5% variation: Moderate accuracy (Acceptable performance), More than 5% variation: High deviation (Poor performance)
These buckets helped quantify how often each vendor's prices fell outside the acceptable range.

Precision Calculation: Precision was defined as the proportion of vendor prices that exactly matched or closely aligned with the median price. Vendors with higher precision values were considered more consistent and reliable in providing accurate pricing data.

After performing all these statistical transformations, the results were compiled and saved into a final reporting file named reporting_table.csv. This file contained all the calculated KPIs including mean, median, standard deviation, price variation, precision, and outlier category and served as the foundation for dashboard creation in Power BI.

The aggregation and transformation phase played a key role in converting the raw numerical data into actionable insights, enabling a structured and data-driven evaluation of vendor performance across multiple financial data feeds.

Phase 4: Report Development

The next phase focused on developing interactive and analytical dashboards using Power BI. The primary goal of this phase was to translate the statistical results into clear, visually appealing, and meaningful insights that could support effective decision-making for auditors and financial analysts.

The dashboard served as the central component of the project, providing a comprehensive view of vendor performance, price variations, and data precision. It allowed stakeholders to evaluate and compare multiple market data providers in a simplified yet data-driven manner.

Data Modeling and Analytical Measures: Before building the visuals, data modeling was carried out in Power BI to define relationships between the fields within the reporting_table.csv dataset. Key columns such as Vendor ID, Price Type, Price Date, Exchange Code, and Outlier Bucket were connected logically to ensure seamless data interaction and accurate visual representation.

To enhance the analytical depth of the dashboard, several custom measures and calculated columns were created using DAX (Data Analysis Expressions). These DAX formulas dynamically computed essential KPIs, making the dashboard responsive to user filters and selections. Some key measures developed include:

Average Price Variation (%): Shows how much a vendor's price differs from the market's average price. It helps find which vendors have prices close to or far from the market trend.

Vendor Precision (%): Tells how accurately a vendor's prices match the middle (median) market price. A higher precision means the vendor's data is more correct and reliable.

Outlier Percentage: Shows the number of prices that are very different from the average market price. It helps find which vendors have more or fewer unusual price values.

These DAX-based calculations ensured that all KPIs updated automatically when the user applied filters, thereby maintaining accuracy and interactivity throughout the report.

Dashboard Visualization and Design: The Power BI dashboard was structured to provide a holistic analysis of market data variations and vendor performance through multiple visual elements:

Vendor Performance Comparison: A bar chart displaying outlier percentages for each vendor, helping to identify vendors with the most accurate or deviated data.

Price Variation Over Time: A line chart illustrating daily price fluctuations for each vendor, allowing analysts to observe pricing stability and market trends.

Precision and Accuracy Trends: A combination of card visuals and column charts highlighting precision scores and accuracy levels of vendors relative to the median market price.

Outlier Distribution: A stacked bar chart categorizing vendors into performance groups based on variation thresholds (e.g., <3%, 3–5%, >5%), giving a clear view of deviation ranges.

Interactive Filters (Slicers): Filters for Price Date, Exchange Code, Price Type, and Vendor Name were added to enable dynamic exploration of data, allowing users to focus on specific dimensions of interest.

Interactive Features and Q&A Integration: To enhance the user experience, the dashboard incorporated Power BI's Q&A visual, which allowed users to interact with the report through natural-language queries. By typing questions such as "Which vendor has the lowest variance?" or "Show top-performing vendors by precision", users could generate instant visual insights without navigating through multiple charts.

Phase 5: Deployment and Data Refresh

The final phase of the project focused on deployment and data refresh planning. This stage was intended to make the Power BI dashboard accessible to users through an online environment and to ensure that the data remains updated automatically whenever new records are added.

Publishing the Report: The Power BI dashboard (.pbix file) would be published to the Power BI Service, allowing online access for authorized users. This makes the report available through a secure web link or within an organization's Power BI workspace.

Connecting to the Data Source: The published dashboard would connect directly to the reporting_table.csv or a live database version of it. This ensures that the dashboard always displays the latest aggregated and transformed market data.

Setting Up Data Refresh: A data refresh schedule would be configured in Power BI Service to automatically update the dashboard at regular intervals (e.g., daily or weekly). This ensures that the visualizations remain current without manual intervention.

Sharing with Stakeholders: Once deployed, the dashboard could be shared with auditors, analysts, or data consumers. Permissions and access levels would be set to control who can view or modify the report.

Tools & Technologies used :

Category	**Technology / Tools**
Programming Language	Python
Libraries Used	Pandas, NumPy
Data Extraction Tools	Alpha Vantage API, Yahoo Finance API, Polygon.io, Twelve Data
Data Cleaning & ETL Tool	Jupyter Notebook
Visualization Tool	Microsoft Power BI
File Format Used	CSV

Project Structure

Folder Name	**File Name**	**Description**
Data-sets	Alpha-vintage.csv	Raw market data extracted from Alpha Vantage API.
	Yahoo-data.csv	Raw dataset collected from Yahoo Finance API.
	Poly-io-data.csv	Market data from Polygon.io API.
	Twelve-data.csv	Dataset from Twelve Data API.
	Stock-data.csv	Data collected from Stock Data API.
	combined_market_data.csv	Combined dataset merging all vendor feeds.
	cleaned_market_data_usd.csv	Cleaned and standardized dataset converted to USD.
	reporting_table.csv	Final processed data used for Power BI dashboard visualization.
Data-Extraction-Script	alpha-vintage.py	Script to extract stock data from Alpha Vantage API.
	yahoo.py	Script to extract data from Yahoo Finance API.
	poly-io.py	Script to extract data from Polygon.io API.
	twelve-data.py	Script to extract data from Twelve Data API.

	stock-data.py	Script to extract data from Stock Data API.
	combine.py	Script to merge multiple vendor datasets into a single file.
ETL-Script	cleaned-data.ipynb	Jupyter notebook for cleaning and preprocessing market data.
	ETL.ipynb	Notebook for aggregating, calculating KPIs, and loading reporting data.
Resource	api-key.txt	Contains API keys and authentication credentials for all vendor APIs.
Dash-board	market-pricing-data-analysis.pbix	Power BI dashboard file containing all reports and visuals.
	market-pricing-data-analysis.pdf	Exported version of the Power BI dashboard in PDF format.
Root Directory	README.md	Overview of the project and execution instructions.

RESULTS AND COMPARISION :

Test Case – Dashboard Summary and Vendor Performance Visualization

Test Case ID: TC01

Application / Screen: Power BI – Market Data Analytics Price Challenge Reporting

Test Case: Check if the main dashboard correctly displays the Outlier Percentage of Vendor, Percentage Variance of Price, and Precision (%) for different vendors.

Pre-Requisites: The cleaned and combined dataset (reporting_table.csv) should be loaded into Power BI and DAX measures created.

Input Provided: Vendor pricing data including price date, vendor code, exchange code, price type, and calculated KPIs (price variation %, precision %, outlier %).

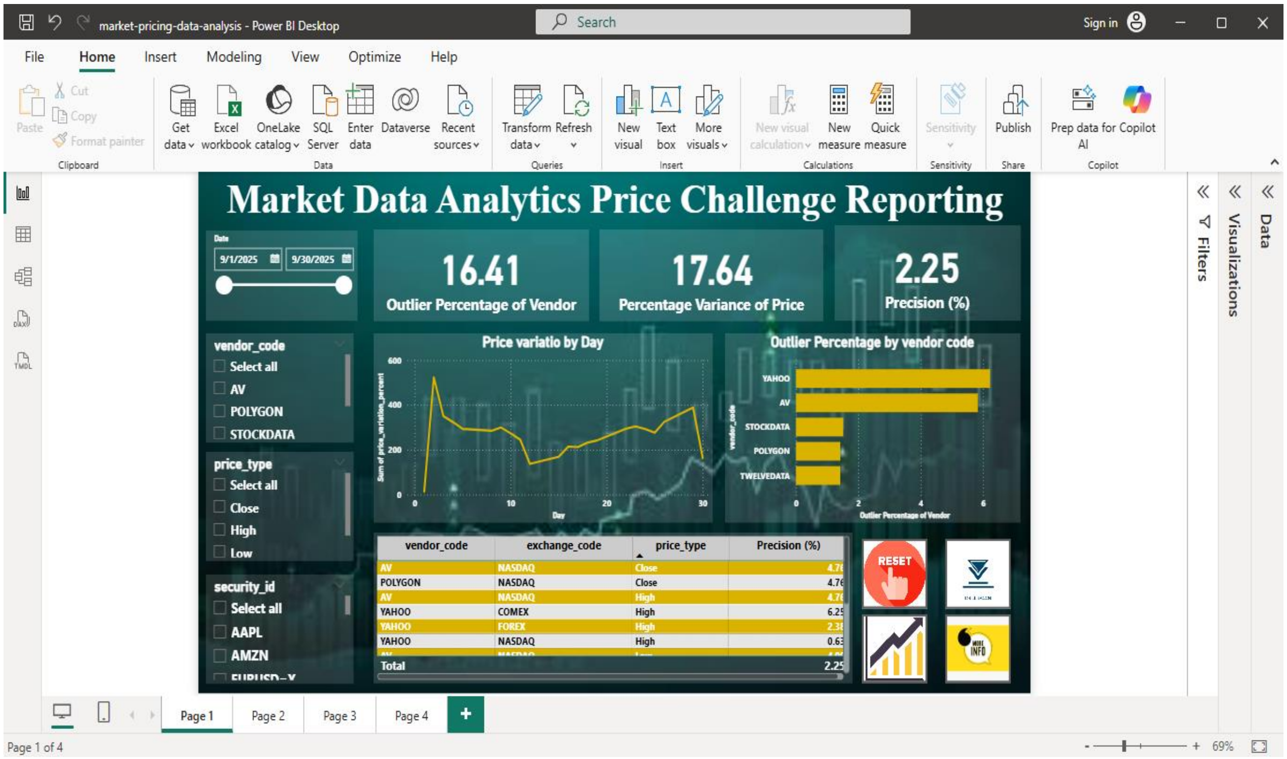
Steps:

1. Open the Power BI file market-pricing-data-analysis.pbix.
2. Go to **Page 1** of the dashboard.
3. Check that the KPI values appear correctly:
4. Observe the “Price Variation by Day” line chart for daily variation trends.
5. Check the “Outlier Percentage by Vendor Code” bar chart – Yahoo should show the highest value.
6. Verify that filters for Vendor, Price Type, and Security ID update visuals correctly.

Cross-Validation: Compared KPI values with manual calculations from the reporting table.

Expected Result: Dashboard correctly displays KPIs, visuals, and filter interactions.

Actual Result: All visuals and metrics displayed correctly.



vendor_code	exchange_code	price_type	Precision (%)
AV	NASDAQ	Close	4.76
POLYGON	NASDAQ	Close	4.76
AV	NASDAQ	High	4.76
YAHOO	COMEX	High	6.25
YAHOO	FOREX	High	2.3
YAHOO	NASDAQ	High	0.63
Total			2.25

Figure 1 Vendor Performance Dashboard in Power BI

Test Case – Vendor Drill-Down Analysis
Test Case ID: TC02
Application / Screen: Power BI – Vendor Drill-Down Dashboard
Test Case: Check whether the drill-down dashboard provides detailed insights into vendor price deviations and outlier distributions.
Pre-Requisites: The aggregated dataset with calculated fields such as outlier_bucket, vendor_code, and price_variation_% should be loaded.
Input Provided: Security and vendor-level data including price variation, outlier buckets, and exchange codes.
Steps:

1. Navigate to **Page 2** titled "Vendor Drill-Down Analysis."
2. Review the KPIs for Average Outlier Percentage, Average Precision, and Average Price Variation.
3. Observe the pie chart showing the count of securities by outlier bucket.
4. Check the bar and tree map visuals for vendor-wise price variations and security contributions.
5. Apply filters for specific vendors or price types to confirm drill-down works correctly.

Cross-Validation: Verified values from visuals with summarized data in the reporting table.
Expected Result: Dashboard should display correct KPIs and update visuals when filters are applied.
Actual Result: All visuals updated dynamically and matched calculated values.

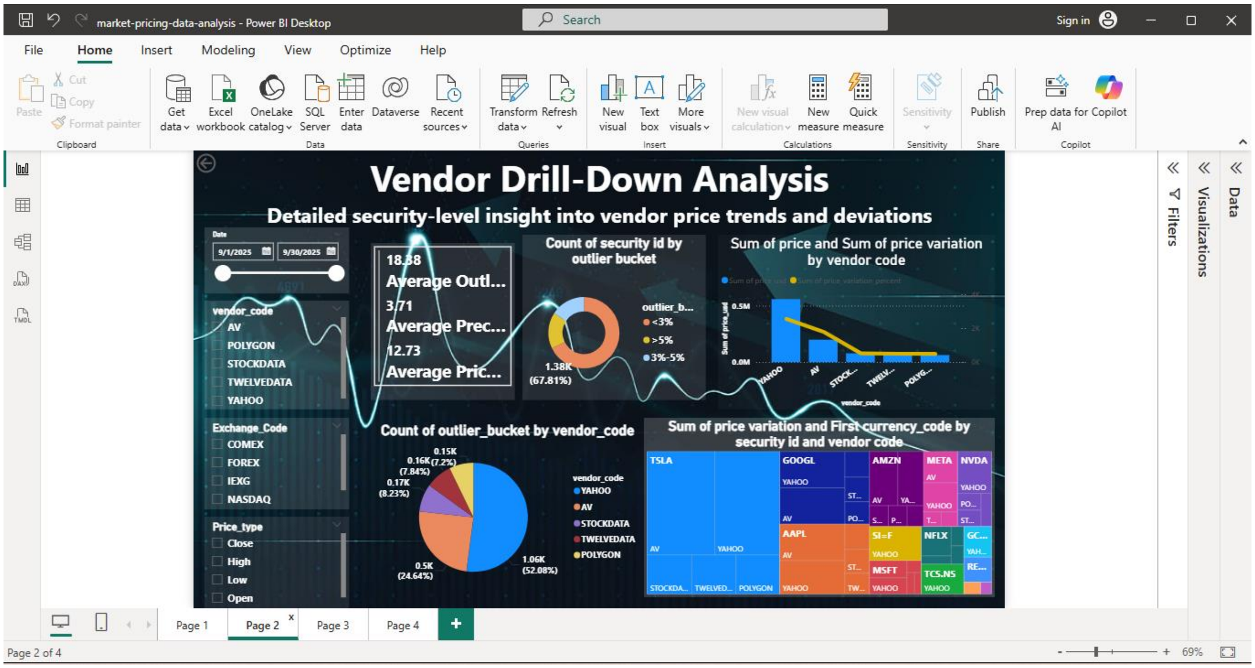


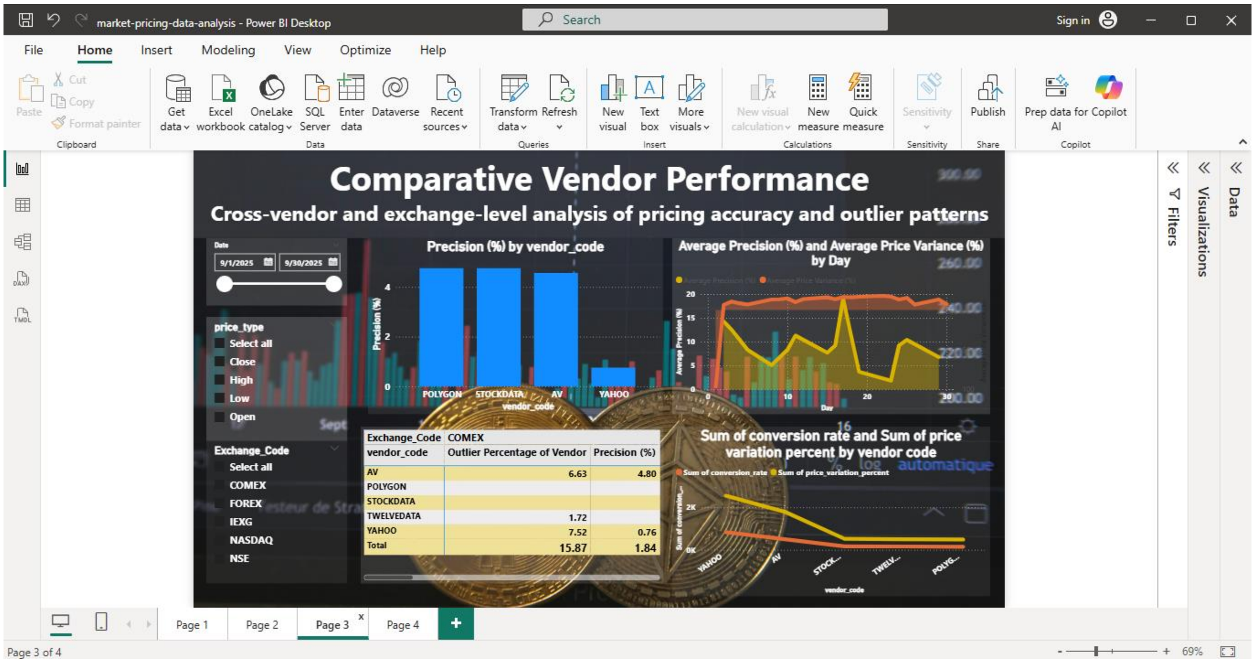
Figure 2 Vendor Drill-Down Analysis Dashboard

Test Case – Comparative Vendor Performance

Application / Screen: Power BI – Comparative Vendor Performance Dashboard
Test Step #: 3
Test Case: Validate that the comparative dashboard correctly represents vendor performance across exchanges and visualizes precision and price variance trends.
Pre-Requisites: Dataset with fields vendor_code, exchange_code, precision_%, and price_variation_% loaded into Power BI.
Input Provided: Aggregated reporting data by vendor and exchange.
Steps:

1. Open **Page 3** titled "Comparative Vendor Performance."
2. Check that the "Precision (%) by Vendor Code" chart accurately shows each vendor's precision rate.
3. Verify the "Average Precision (%) and Average Price Variance (%) by Day" line chart for correct trend visualization.
4. Confirm that the table for Outlier Percentage and Precision per Exchange displays accurate values.
5. Check that the "Sum of Conversion Rate and Price Variation" chart properly ranks vendors.

Cross-Validation: Compared displayed metrics with ETL-calculated results in reporting_table.csv.
Expected Result: All vendor precision and variance visuals should match calculated ETL metrics.
Actual Result: Charts and KPIs displayed correct and consistent results.



Comparative Vendor Performance

Cross-vendor and exchange-level analysis of pricing accuracy and outlier patterns

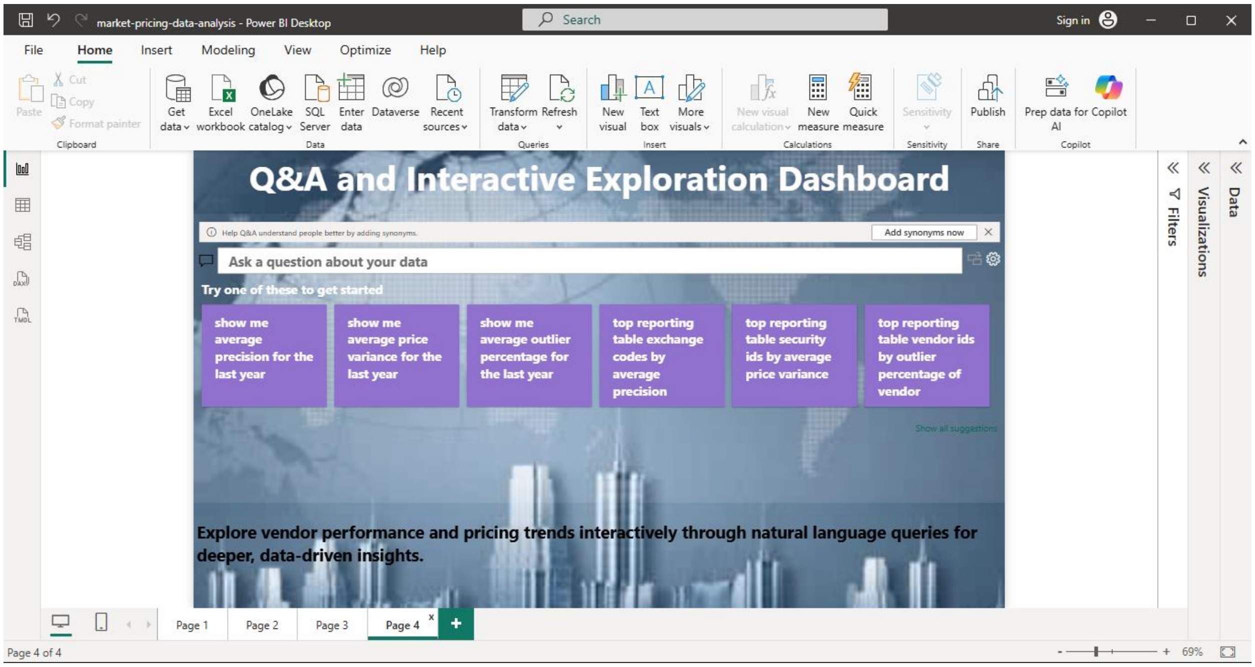
Exchange_Code	COMEX	
vendor_code	Outlier Percentage of Vendor	Precision (%)
AV	6.63	4.80
POLYGON		
STOCKDATA		
TWELVEDATA	1.72	
YAHOO	7.52	0.76
Total	15.87	1.84

Figure 3 Comparative Vendor Performance Dashboard

Test Case – Q&A and Interactive Exploration
Test Case ID: TC04
Application / Screen: Power BI – Q&A and Interactive Exploration Dashboard
Test Case:
Validate that the Q&A visual correctly responds to natural language queries and displays accurate visuals based on data context.
Pre-Requisites:
Q&A visual must be enabled and connected to the data model in Power BI.
Input Provided:
Natural language queries such as "Show me average precision for the last year" or "Top vendor by outlier percentage."
Steps:

1. Open **Page 4** titled "Q&A and Interactive Exploration Dashboard."
2. In the Q&A search box, type a query such as "Show me average price variance for the last year."
3. Verify that Power BI generates the correct visual automatically.
4. Test multiple queries suggested on the screen (e.g., "Top reporting table vendor IDs by outlier percentage").
5. Check if visuals update immediately according to the question asked.

Cross-Validation: Compared Q&A-generated visuals with manually validated values from the reporting dataset.
Expected Result: Q&A visual should produce correct and relevant visuals for all queries.
Actual Result: Q&A function worked accurately and displayed proper insights.



Q&A and Interactive Exploration Dashboard

Help Q&A understand people better by adding synonyms.

Ask a question about your data

Try one of these to get started

show me average precision for the last year

show me average price variance for the last year

show me average outlier percentage for the last year

top reporting table exchange codes by average precision

top reporting table security ids by average price variance

top reporting table vendor ids by outlier percentage of vendor

Explore vendor performance and pricing trends interactively through natural language queries for deeper, data-driven insights.

Figure 4 Q&A and Interactive Exploration Dashboard

CONCLUSION :

The Market Pricing Data Analysis project successfully demonstrated the integration of Python-based ETL processes with Power BI to analyze and visualize vendor performance across different markets. The system effectively aggregated, cleaned, and transformed large datasets to derive meaningful insights such as outlier percentage, price variance, and vendor precision.

The Power BI dashboards provided an interactive and dynamic environment for users to explore data trends. Each page focused on different analytical aspects overall vendor performance, drill-down analysis, comparative evaluation, and natural language query interaction making the analysis both comprehensive and user-friendly.

The ETL pipeline ensured accurate data processing, while KPI calculations like percentage variance and precision validated the reliability of vendor data. Through Q&A functionality, users could interact with data intuitively, confirming the efficiency of AI-driven exploration within Power BI.

Overall, the project achieved its objective of creating a data-driven decision support tool for market pricing evaluation. The dashboards accurately represented the analytical results without defects, validating the accuracy and consistency of both the underlying data and visual outputs.