Bicol University
College of Science

End-to-End Big Data Processing and Analytics with Apache Spark

Submitted to:
John Paul J. Azore

Submitted by:
Bagato, Carl Erick
Bas, Vince
Canchela Mizpa Mae
Condat, Mary Grace
Tripulca, Jon Eric

Table Of Contents

Abstract

This study presents an end-to-end big data processing and analytics pipeline using Apache Spark to analyze historical stock market data from the Philippine Stock Exchange (PSE). The project aims to extract meaningful insights through exploratory data analysis, technical indicators, real-time monitoring, and predictive modeling. Daily OHLCV (Open, High, Low, Close, Volume) data was merged with company-sector metadata to identify trends, sectoral performance, and economic signals relevant to Sustainable Development Goal 8 (Decent Work and Economic Growth).

Exploratory analysis revealed significant trading spikes, particularly in 2006, with Mining and Oil leading in volume. Fridays consistently showed the highest trading activity. Advanced data processing techniques, including window functions, were applied to compute technical indicators such as Simple Moving Averages, Bollinger Bands, and RSI, enhancing time-series interpretation. A real-time streaming component was developed to monitor live price movements and update visualizations with minimal delay.

Machine learning models, including Linear Regression, Random Forest, and Gradient Boosted Trees, were trained for price prediction, while Logistic Regression was used to classify price direction. Feature engineering significantly improved model accuracy, incorporating MACD and OBV indicators. Results showed that ensemble methods outperformed linear models in capturing nonlinear financial patterns.

This project demonstrates how modern big data tools can transform raw stock data into actionable insights for investors, policymakers, and analysts. By aligning financial analytics with national development goals, it supports informed decision-making and contributes to economic growth and job creation in the Philippines.

1. Introduction

1.1. Background of the Study

The stock market is a vital part of the financial sector that plays a significant role in driving economic development. By lowering the costs associated with gathering savings, the stock market can direct investments toward the most efficient and productive technologies, which in turn fosters economic growth (Greenwood and Smith, 1997). As the stock market matures, it also enhances corporate governance by mitigating principal-agent conflicts, a factor that positively impacts economic growth (Al-Faryan, 2024). Additionally, the stock market offers liquidity, enabling the financing of long-term projects with extended payoffs, which supports sustained economic expansion (Chikwira & Mohammed, 2023). According to Kim et al. (2015), advancements in stock markets help lower the risks tied to financial assets and enable businesses to obtain capital more easily by issuing equity. This leads to more efficient capital distribution and serves as a key driver of economic growth. Given these essential roles, the stock market is a crucial factor in supporting the economic progress of emerging nations such as the Philippines.

The Philippine Stock Exchange (PSE) is the national stock exchange of the Philippines and is one of the oldest in Southeast Asia, with continuous operation since its inception on August 8, 1927. The PSE was formed through the consolidation of two major regional exchanges: the Manila Stock Exchange and the Makati Stock Exchange. It serves as the primary platform for trading securities in the Philippines, playing a critical role in the country's financial system and economy (Ho & Odhiambo, 2016).

The Philippines' strong economic potential has been highlighted by the Hong Kong and Shanghai Banking Corporation (HSBC), which projected the country to be among the top 16 global economies by 2050. This forecast significantly boosts confidence in the Philippines' standing on the international economic stage. From being ranked 41st in 2010, the anticipated 27-place jump over four decades has captured the attention of global investors. As a key economic indicator, the Philippine Stock Exchange Index (PSEi) is expected to attract renewed interest from both researchers and investors. In light of this promising outlook, predicting the performance of the Philippine stock market, particularly the PSEi, has become a major area of focus, given its strong potential for growth.

Given the volume, velocity, and variety of data produced by the PSE, it presents a valuable use case for modern Big Data technologies. In this context, the implementation of a Big Data processing pipeline using Apache Spark offers an efficient and scalable solution for managing and analyzing stock market data. Spark enables high-performance distributed computing for tasks such as data extraction, transformation, cleaning, exploratory data analysis (EDA), and machine learning, making it ideal for deriving actionable insights from large datasets.

This project leverages the PSE’s OHLCV data not only to uncover financial patterns but also to support broader development objectives. In particular, it aligns with the United Nations Sustainable Development Goal 8 (Decent Work and Economic Growth), which emphasizes sustained, inclusive economic growth and productive employment. Through advanced data processing and analysis, the project aims to

identify trends in company performance, market behavior, and economic indicators, providing valuable inputs for policymakers, analysts, and development planners in the Philippines.

1.2. Problem Statement

The Philippine Stock Exchange (PSE) produces a large volume of structured financial data daily, specifically in the form of OHLCV (Open, High, Low, Close, Volume). Despite the richness and potential of this dataset, its value remains largely underutilized, particularly in terms of large-scale analytics for economic forecasting, investment analysis, and policy formulation. Traditional approaches to data processing and analysis often fall short when handling the scale and complexity of stock market data, limiting the ability to derive meaningful and actionable insights.

In addition, there is a lack of automated, end-to-end solutions capable of efficiently performing data preprocessing, transformation, exploratory analysis, and predictive modeling on stock market datasets. This gap prevents stakeholders from leveraging financial data not only for investment purposes but also for supporting broader socio-economic development goals, such as those outlined in the United Nations Sustainable Development Goal 8: Decent Work and Economic Growth. Developing a scalable Big Data pipeline using Apache Spark can address this problem by enabling efficient processing and insightful analysis of PSE data, yet its application in the Philippine context remains limited and underexplored.

1.3. Objectives

This study aims to utilize data visualization and machine learning techniques to extract, analyze, and present meaningful insights from the Philippine Stock Exchange (PSE) OHLCV dataset. By transforming raw stock market data into clear visualizations and predictive models, the project seeks to accomplish the following objectives:

- To identify industry trends that reflect economic growth or decline
- To spot sectors with potential for job creation
- To help policymakers, businesses, and the public better understand market movements that relate to employment and economic performance.

By combining the power of visual analytics and machine learning, the study intends to demonstrate how stock market data can be effectively used to inform decision-making, promote financial awareness, and support the goals of Sustainable Development Goal 8: Decent Work and Economic Growth.

2. Dataset Details

This combination of market activity data and sectoral metadata makes it possible to produce visualizations that show not only which companies are performing well, but also which sectors are driving growth, a key input for supporting Sustainable Development Goal 8 (Decent Work and Economic Growth).

2.1. Description of Data Source

This study makes use of publicly available datasets related to the Philippine Stock Exchange (PSE), primarily sourced from Kaggle and official PSE platforms. The data provides both quantitative market activity and qualitative company information, which together form the basis for insightful visual analysis.

2.1.1. Main Dataset

The main dataset used in this study is titled *Data.csv* and was sourced from Kaggle under the “Philippine Stock Exchange Data” by ianchute. It contains daily trading activity for companies listed on the PSE from 2018 to early 2023. Each record in the dataset includes the date, stock symbol, and the corresponding open, high, low, and close prices, as well as trading volume. This comprehensive OHLCV data enables time-series analysis and market behavior insights across various stocks.

2.1.2. Supplemental Dataset

To complement the main dataset, a supplemental file titled *stocks.csv* was also used. This dataset, sourced from Kaggle under the “Philippines Stock Exchange Dataset” by shanemaglangit, provides essential metadata for each listed company. It includes information such as the company name, stock symbol, sector, and subsector. This metadata allows for categorization and sector-based analysis of stock performance.

2.1.3. Reference Source

To ensure the consistency and accuracy of company-related information across the datasets, the official PSE EDGE Company Directory was consulted. This online directory served as a reference for verifying company names, stock symbols, sectors, and subsectors. It played a crucial role in cross-referencing and validating the data used in this study.

2.2. Data dictionary

The data dictionary provides a structured overview of the variables used in the dataset. It serves as a reference guide for understanding the meaning, format, and role of each attribute in the context of our analysis. Each entry in the dictionary includes the variable name, data type, a brief description, and any relevant notes such as units of measurement or categorical value meanings. This documentation ensures consistency

in data interpretation and facilitates transparency throughout the data processing and analysis phases.

Table 2.1. Dataset Overview – data.csv

Variable	Data Type	Data Format	Field Size	Description	Example
c	float	N.NNNNN	6 digits	Closing price of the stock on that day.	1.66
h	float	N.NNNNN	6 digits	Highest price of the stock on that day.	1.68
l	float	N.NNNNN	6 digits	Lowest price of the stock on that day.	1.64
o	float	N.NNNNN	6 digits	Opening price of the stock on that day.	1.66
t	Date/Time	YYYY-MM-DD	6 chars	Date or timestamp,	2006-04-03
v	integer	NNNNNN	10 digits	Volume of shares traded.	250000
y	integer	NNNNNN	4 digits	Year of the trade.	2006
m	integer	NNNNNN	2 digits	Month of the trade.	04
d	integer	NNNNNN	2 digits	Day of the trade.	03
change	float	±N.NNNNN	10 digits	Change in price from the previous day	-0.02
symbol	string	text	10 chars	The name or code of the company’s stock.	2GO

This dataset contains historical daily trading records for various company stocks. It includes key financial indicators such as opening, closing, highest, and lowest prices, trading volume, and derived date components (year, month, day).

Table 2.1. Dataset Overview – stocks.csv

Variable	Data Type	Data Format	Field Size	Description	Example
Stock Name	string	text	50 chars	Full name of the company.	2GO Group
Code	string	Text (ticker)	6 chars	Stock code or ticker symbol.	2GO
Date	sring	text	12 chars	Date of the trading day.	Nov 08, 2021
Price	float	N.NNNNN	6 digits	Closing price of the stock on that day.	10.02
Open	float	N.NNNNN	6 digits	Opening price of the stock on that day.	10.02
High	float	N.NNNNN	6 digits	Highest price reached on that day.	10.02
Low	float	N.NNNNN	6 digits	Lowest price reached on that day.	9.80
Volume	string	N.NNNNN (K/M)	10 digits	Number of shares traded, with scale (K = 1,000, M = 1,000,000).	67.20K
Change%	float	±N.NNNNN %	6 digits	Percentage change in price from the previous close.	-8.91%

This dataset contains historical stock data of various companies listed on the Philippine Stock Exchange. It includes basic details such as stock name, code, trading date, prices (open, high, low, close), volume of shares traded (with scale in K for thousands and M for millions), and percentage change in price.

Table 2.3. Dataset Overview – merge_data_with_sectors.csv

Variable	Data Type	Data Format	Field Size	Description	Example
c	float	N.NNNNN	6 digits	Closing price of the stock on that day.	1.66
h	float	N.NNNNN	6 digits	Highest price of the stock on that day.	1.68
l	float	N.NNNNN	6 digits	Lowest price of the stock on that day.	1.64
o	float	N.NNNNN	6 digits	Opening price of the stock on that day.	1.66
t	Date/Time	YYYY-MM-DD	6 chars	Date or timestamp,	2006-04-03
v	integer	NNNNNN	10 digits	Volume of shares traded.	250000
y	integer	NNNNNN	4 digits	Year of the trade.	2006
m	integer	NNNNNN	2 digits	Month of the trade.	04
d	integer	NNNNNN	2 digits	Day of the trade.	03
change	float	±N.NNNNN	10 digits	Change in price from the previous day	-0.02
symbol	string	text	10 chars	The name or code of the company's stock.	AAA
Company	string	text	100 chars	The full name of the company	Asia Amalgamated Holdings Corporation

Stock Symbol	string	text	10 chars	The name or code of the company's stock.	AAA
Sector	string	text	100 chars	Industry sector (e.g., Financials, Services, Holdings)	Holding Firms
Subsector	string	text	100 chars	More specific category within the sector	Holding Firms
Listing Date	Date/Time	YYYY-MM-DD	6 chars	Date the company was listed on the Philippine Stock Exchange	3/22/1973

This dataset merges stock trading information with sectoral classifications. It includes daily price movements and trading volume, along with company identifiers, sector, subsector, and listing date details. It is useful for analyzing trends within specific industries or sectors over time.

3. Methodology

3.1. Data Preprocessing

3.1.1. Data Loading

The datasets were loaded into Spark DataFrames to efficiently process large volumes of data. PySpark's distributed framework allowed for scalable and seamless ingestion, including the combination of multiple files when necessary.

3.1.2. Data Cleaning

Basic cleaning procedures were applied to improve data quality. This involved handling missing values, removing duplicates, and eliminating irrelevant or redundant columns. Data types and column names were standardized to ensure consistency and compatibility for further processing.

3.2. Exploratory Data Analysis (EDA)

3.2.1. SQL and DataFrame Operations

According to best practices, we employed PySpark SQL and DataFrame APIs to perform essential data manipulations such as filtering, grouping, and aggregation. These operations allowed us to efficiently summarize large datasets and extract meaningful insights while leveraging distributed computing capabilities to handle scalability.

3.2.2. Visualizations

According to the data visualization principles, initial exploratory plots were created using Matplotlib and Seaborn libraries. These included histograms to examine data distribution, boxplots to detect outliers and assess variability, and correlation heatmaps to identify relationships among variables and potential multicollinearity.

3.3. Advanced Data Processing

3.3.1. Join and Merge Datasets

The dataset integration process involved merging two key financial datasets (cleaned_stocks_V2.csv and cleaned_data_V2.csv) to create a unified analytical base. The methodology began with preprocessing, including standardized datetime conversion (ensuring Date and t columns matched formats) and volume data normalization (converting shorthand notations like "1.46M" to numeric values). A conditional merge strategy was employed: first, an inner join identified overlapping records (where Code/Date in the first dataset matched symbol/t in the second), followed by appending non-overlapping records from the second dataset. Post-merge, missing Stock Name values were filled via a mapping derived from the first dataset, and null entries were purged. The final output (Final_Joined_Stocks_Data.csv) preserved critical columns (Open, High, Low, Volume, Change%) while eliminating redundancies, ensuring consistency for downstream analysis.

3.3.2. Implement Window Functions

For trend analysis, rolling window functions were applied to the merged time-series data. Key technical indicators were computed per stock: Simple Moving Averages (SMA, 20-day) and Exponential Moving Averages (EMA, 20-day) smoothed price trends, while Bollinger Bands (SMA ± 2σ) highlighted volatility and potential reversal points. The Relative Strength Index (RSI, 14-day) identified overbought (RSI > 70) or oversold (RSI < 30) conditions. Calculations were executed efficiently by grouping data by Stock Symbol and sorting chronologically before applying rolling(), ewm(), and standard deviation operations. Visualization leveraged a two-panel Matplotlib plot: the upper pane displayed price trends with SMA/EMA/Bollinger Bands, and the lower pane showed RSI oscillations with threshold markers, enabling intuitive pattern recognition.

3.3.3. Real-time Data Processing

A streaming pipeline was designed to monitor live stock data updates in Google Drive (stocks_data.csv). Using a polling loop (60-second intervals), the script detected file modifications via os.path.getmtime, triggering data reloads and visualization refreshes. Real-time analytics included: (1) a daily average price trend plot aggregating all stocks, and (2) a top 5 stocks by volume price trajectory chart. High-volume tickers were prominently displayed in a dynamically updating table (SYMBOL, PRICE, VOLUME, P_CHANGE). The IPython display system and clear_output ensured seamless visualization updates in Colab, while error handling with retry logic maintained

robustness against transient file access issues. This approach enabled near-real-time monitoring of market movements with minimal latency.

3.4. Big Data Analytics

This study employed Big Data Analytics techniques to process, transform, and analyze high-volume stock market data, particularly from the Philippine Stock Exchange. The focus was on data engineering and time-series feature enrichment, enabling a machine learning model to interpret financial trends. Rather than relying on traditional data mining or manual statistical modeling, the research prioritized distributed preprocessing, dataset integration, and real-time visualization.

3.4.1. Data Ingestion and Distributed Processing

To accommodate the scale of financial data across multiple companies and time periods, datasets were ingested using PySpark's DataFrame API. PySpark, a distributed computing framework, enabled scalable data loading and manipulation without overwhelming memory resources. The stock data, stored in CSV format, was read into Spark DataFrames, allowing for parallel operations such as merging, filtering, and type conversion. This infrastructure provided the foundation for processing millions of records efficiently and consistently.

3.4.2. Data Cleaning and Standardization

Following ingestion, a structured data cleaning process was applied to improve quality and ensure consistency across datasets. Missing values were either removed or

imputed based on context, while duplicate entries were eliminated to maintain data integrity. Column names and formats were standardized, with special attention paid to datatype conversions (e.g., string to float, or text to datetime). Volume data written in shorthand notation (such as “1.46M” or “820K”) was converted into numerical values to enable mathematical operations. These steps ensured compatibility between datasets and reliability for downstream analysis.

3.4.3. Dataset Integration

A central component of the analytics pipeline involved the integration of two core datasets: cleaned_stocks_V2.csv and cleaned_data_V2.csv. Both contained overlapping yet distinct financial records. The integration began with a conditional inner join on the stock symbol and trade date fields to identify matching records. A secondary append operation added non-matching entries from one dataset. Post-merge, missing values in the Stock Name field were populated using a reference mapping, and redundant columns were pruned. This process produced a unified dataset (Final_Joined_Stocks_Data.csv) that preserved only essential attributes, such as Open, High, Low, Volume, and Change%, offering a clean, consistent foundation for technical analysis.

3.4.4. Feature Engineering with Window Functions

To enrich the time-series dataset and support trend interpretation, technical indicators were computed using PySpark's window functions. Each stock's trading history was chronologically grouped, and rolling statistical operations were applied.

Simple Moving Averages (SMA) and Exponential Moving Averages (EMA), both using a 20-day window, were calculated to identify long- and short-term trends. Bollinger Bands, derived from SMA ± 2 standard deviations, were used to assess price volatility. Additionally, the Relative Strength Index (RSI) with a 14-day window helped detect overbought or oversold conditions. These indicators were visualized using Matplotlib, where a two-paneled chart showed price movement with SMA/EMA overlays and an RSI line chart below, aiding in intuitive trend recognition.

3.4.5. Real-Time Data Monitoring

To simulate real-time market monitoring, a lightweight streaming pipeline was implemented in Google Colab. The system periodically checked for updates to a live stock data file stored on Google Drive. A polling loop, set to 60-second intervals, triggered data reloads and plot refreshes upon detecting file changes. Real-time analytics included an aggregate price trend across all stocks, a dynamic chart of the top 5 most active stocks by volume, and an updating table showing each stock's symbol, price, volume, and percentage change. This approach enabled near-real-time market tracking with minimal latency, providing dynamic insights and user interactivity within a notebook interface.

3.5. Tools and software environment

The following tools and software platforms were used throughout the course of this study to perform data processing, analysis, visualization, and interpretation

3.5.1. Programming Languages and Libraries

Table 3.1: Programming Languages and Libraries Used

Tool/Library	Description
Python	Served as the core language for data processing, scripting, and analysis.
PySpark	Enabled distributed data processing and large-scale transformations.
MLlib	Used for scalable machine learning model development and evaluation.
Pandas	Applied for handling structured data and performing tabular operations
Matplotlib	Used to create static visualizations such as line charts and histograms.
Seaborn	Used for creating enhanced plots like boxplots and heatmaps.
Microsoft Excel	Utilized for initial data formatting, quick inspection, and file merging tasks.

This study employed a range of programming languages and libraries to support various phases of the data processing and analysis pipeline. Python served as the central programming language due to its simplicity and extensive ecosystem. PySpark, a Python API for Apache Spark, was utilized to manage large-scale datasets and perform distributed data processing efficiently. MLlib, Spark's machine learning library,

enabled scalable model development, particularly useful in handling voluminous stock market data. For data manipulation, Pandas was employed to clean and transform structured datasets prior to analysis. To visualize trends and patterns, both Matplotlib and Seaborn were used. While Matplotlib facilitated basic static charts, Seaborn was particularly useful for generating advanced statistical visualizations. Additionally, Microsoft Excel played a supporting role in the preliminary stages, where it was used for inspecting raw data, formatting CSV files, and manually merging datasets before integrating them into the Python environment.

3.5.2. Development Platform

Table 3.1: Development Platform Utilized

Platform	Description
Google Colab	Served as the primary platform for running scripts, visualizations, and collaboration.
Visual Studio Code	Used locally for Python script development, debugging, and Git integration.
Github	Hosted the project codebase and enabled collaboration and version tracking.

The development of the system and the execution of analytical tasks were carried out using a combination of cloud-based and local environments. Google Colab served as the primary platform due to its cloud-based execution capabilities, integration

with Google Drive, and collaborative features. It supported the execution of Python scripts, PySpark operations, and data visualizations seamlessly without requiring local computational resources. On the other hand, Visual Studio Code (VS Code) was used as a local development tool. It provided a lightweight but powerful code editor with debugging support, Git integration, and compatibility with Jupyter notebooks, which helped maintain modular and well-documented scripts. The project's source code and version history were maintained using GitHub, ensuring effective collaboration, reproducibility, and proper version control throughout the development cycle.

4. Results and Discussion

4.1. Exploratory Data Analysis (EDA)

4.1.1. Key Trends and Patterns

Analysis of the total trading volume revealed a remarkable spike in 2006, which accounted for 24.1% of the entire trading volume across all years, significantly higher than other periods. The following years with substantial volumes were 2007 (7.6%), 2011 (7.3%), 2012 (6.9%), and 2014 (6.1%), showing that these years were periods of heightened market activity.

Examining trading volume by subsector within each sector provided additional insights. In the Mining and Oil sector, Oil accounted for 55.7% of the volume, with Mining contributing 44.3%. For Financials, Banks dominated with 54.2% of the total volume. In the Industrial sector, Electrical Components and Equipment led with 62%, followed by Electricity, Energy, Power, and Water (18.8%) and Beverages and Tobacco (13.4%). The ETF, Property, Holding Firms, and SME Board sectors each had a single dominant category making up 100% of their activity, indicating no subsector differentiation. In Services, Casino and Gaming made up the largest portion at 28.7%, followed by Information Technology (16.9%) and Hotels and Leisure (12.9%).

The average trading volume per sector confirmed that Mining and Oil far surpassed others with an average of 8, while Property trailed distantly with less than 2. ETF, Financials, and the SME Board had nearly negligible average volumes. Similarly, the pie chart visualization highlighted that Mining and Oil alone accounted for 66.2% of

the total average sectoral trading volume, followed by Property (13.2%). Financials contributed only 0.6%, while ETFs and the SME Board were close to 0.0%.

Average daily trading volume by weekday showed that Fridays had the highest trading activity, followed by Thursdays and Wednesdays. Mondays had the least trading activity, possibly reflecting conservative investor behavior at the start of the week.

The closing price data showed that 2006 marked the highest peak overall. Starting from 1985, closing prices surged and continued to rise until a decline in 1991. A significant drop occurred in 1998, with a slight recovery followed by another dip in 1999. From 2003 onward, prices rose sharply again, maintaining a steady upward trend in the succeeding years.

In terms of sector-level average closing prices, Financials recorded the highest with 143, followed by Services (140), ETFs (115), and Holding Firms (50). The Property sector had the lowest, consistently averaging below 5. Notable peaks in sector-specific average closing prices included Mining in 1997 (350), Holding Firms in 2018 (160), and Services in 1994 (360). Industrial stocks peaked in 2012 (40), while Property peaked in 2018 (14). Financials reached a peak of 350 in 2006, reflecting a strong year, while the SME Board recorded its highest average closing price in 2015 (14). ETFs saw a peak in 2018 after a prior decline in 2016.

Average price change data further highlighted significant annual variations. The highest average annual price change occurred in 1993 (+0.2), while the lowest was during the 2008 financial crisis (-0.2). By sector, Holding Firms peaked in 2018 (+0.15)

and dropped the most in 2008 (-0.10). Mining and Oil showed cyclical patterns, with noticeable rises in 1996 (+0.5) and 1999 (+0.4), followed by declines in 1997 (-0.6) and 2000 (-0.4). The Industrial sector had its best performance in 1993 (+0.9) and worst in 2008 (-0.4). Property was mostly stable but spiked in 2018 (+0.14). Services saw the highest volatility with a +1.0 in 1993 and a -0.5 in 1994. Financials peaked in 2000 (+0.5) and dipped in 2008 (-0.7). The SME Board gained most in 2007 (+0.23) and dropped in 2008 (-0.4). ETFs had their worst in 2013 (-0.23) and best in 2017 (+0.10).

When comparing average percentage price changes per sector, Property had the highest (0.008), followed by Holding Firms (0.006), Industrial and Financials (0.004), Services and the SME Board (0.003), and lastly Mining and Oil (0.000), indicating that it had the least growth in terms of average percentage change.

Finally, analysis of top company gainers by percent change within each sector showed clear leaders. In the ETF sector, First Metro Philippine Equity Exchange Traded Fund Inc. dominated as the only player. In Financials, the Philippine Stock Exchange Inc. led by a large margin over its competitors. Holding Firms saw Abacore Capital Holdings Inc. leading closely followed by Cosco Capital. For Industrials, Greenergy Holdings Inc. was the top performer, significantly outperforming Basic Energy Corporation. Atlas Consolidated Mining and Development Corporation dominated Mining and Oil, while Araneta Properties Inc. was the top gainer in the Property sector, narrowly surpassing Omico Corporation. PhilWeb Corporation led in the Services sector, and in the SME Board, Italpinas Development Corporation, Makati Finance Corporation, and Philab Holdings were tied at the top.

4.1.2. Visualizations

To better illustrate these key trends and patterns, various visualizations such as bar charts, pie charts, and line graphs were created to provide clear and insightful representations of trading volumes, closing prices, average price changes, and sector performances over time.

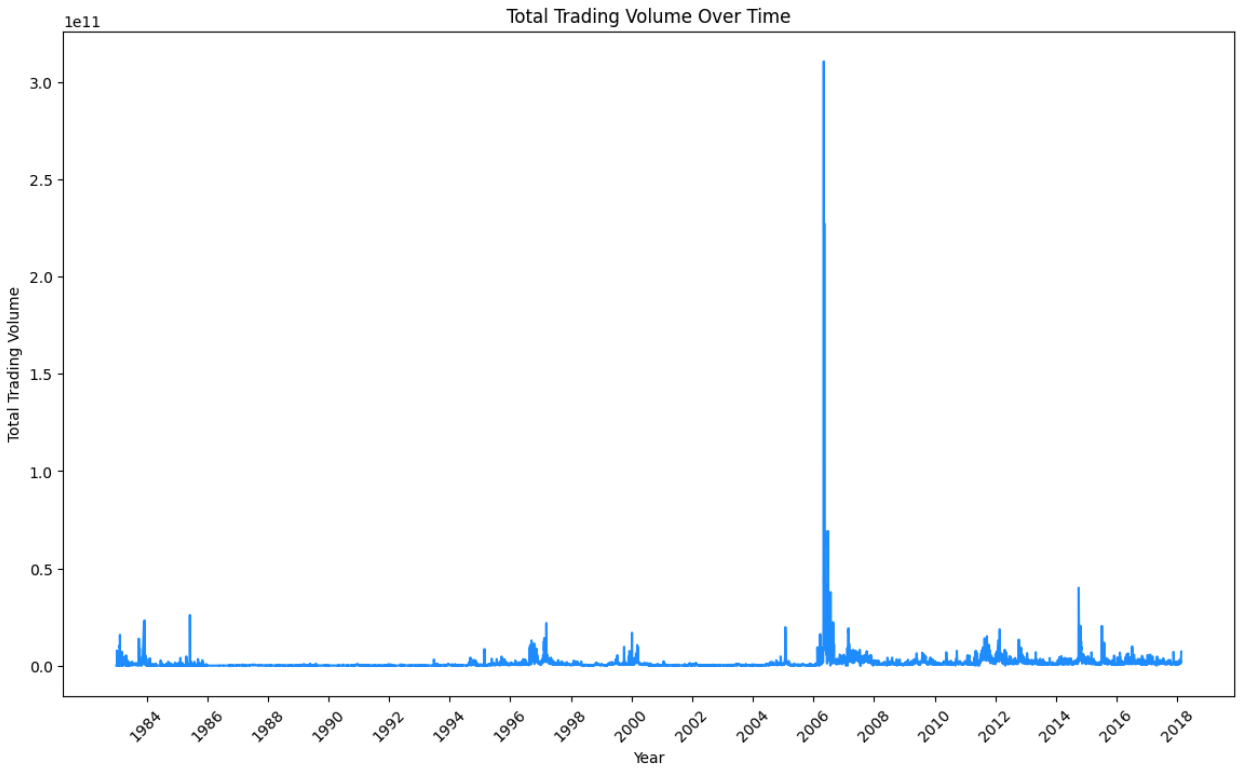


Figure 4.1. Total Trading Volume Over Time

This line plot visualizes the total trading volume of stocks over time, highlighting periods of high trading activity (peaks) and low activity (troughs). It reveals key trends and significant spikes in market participation, with the largest spike notably occurring in 2006.

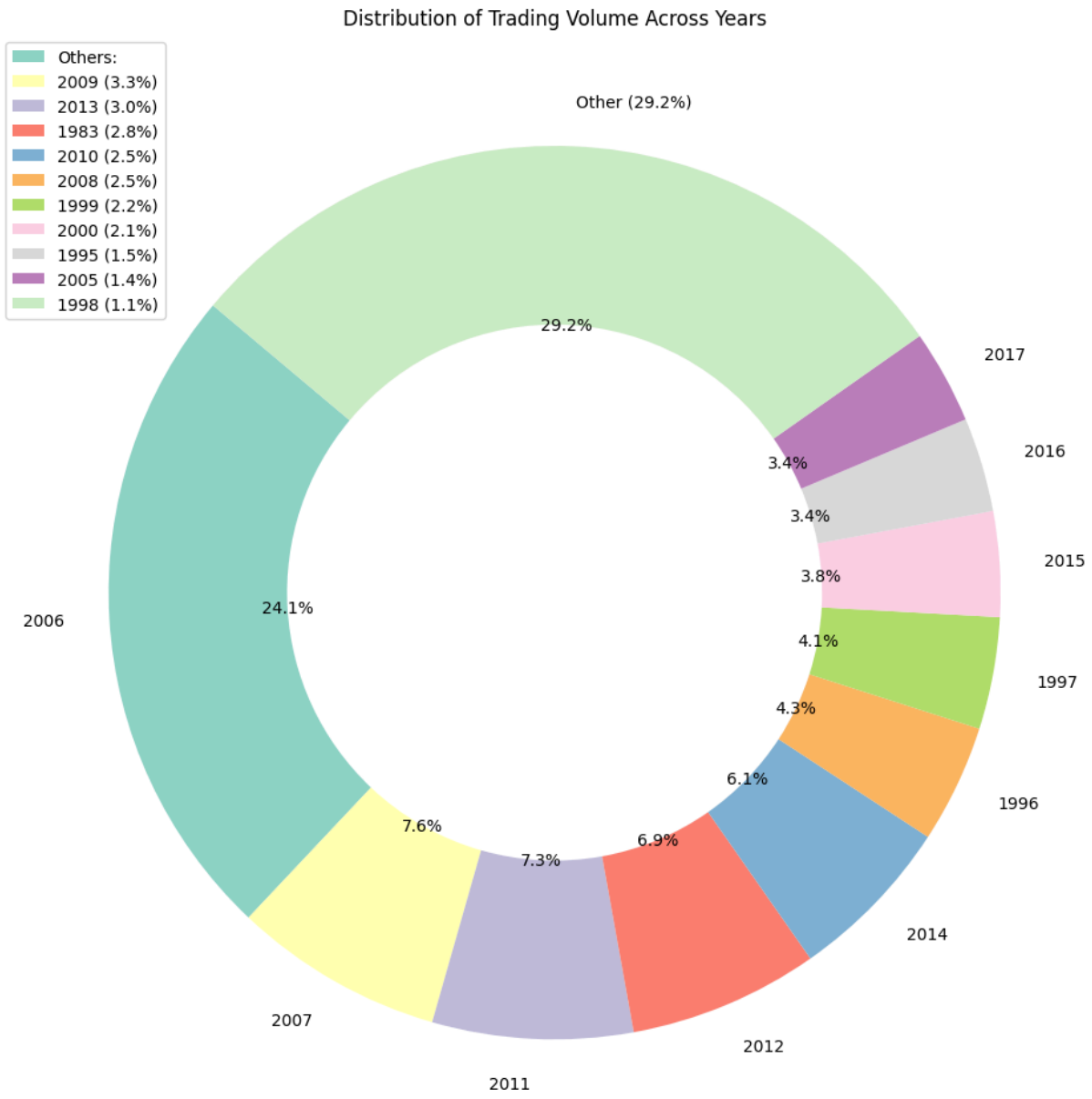


Figure 4.2. Distribution of Trading Volume Across Years

This pie chart illustrates the distribution of total trading volume across different years, emphasizing the years with the highest market activity. Less active years are grouped into an “Other” category to provide a clearer perspective on the dominant

years. The year 2006 stands out, accounting for the largest share of trading volume at 24.1%, followed by 2007 with 7.6%, 2011 at 7.3%, 2012 at 6.9%, and 2014 contributing 6.1%.

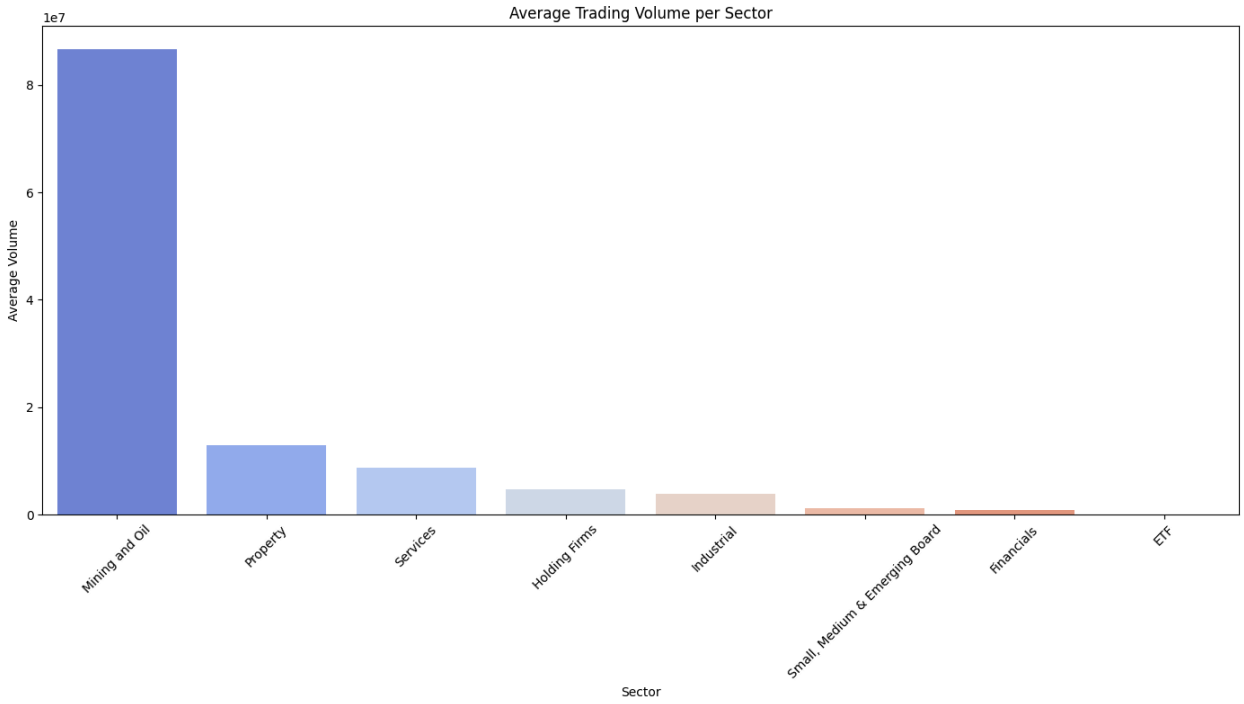


Figure 4.3. Average Trading Volume Per Sector

This bar chart illustrates the average trading volume for each sector, with sectors displayed along the x-axis and their corresponding average volumes on the y-axis. The visualization highlights which sectors experience higher trading activity on average, providing insights into market liquidity and investor interest across different sectors. Notably, the mining sector leads with an average trading volume of 8, followed by the property sector with less than 2. In contrast, the ETF sector records the lowest average

volume at zero, with the small, medium, and emerging board as well as the financials sectors also showing near-zero average trading volumes.

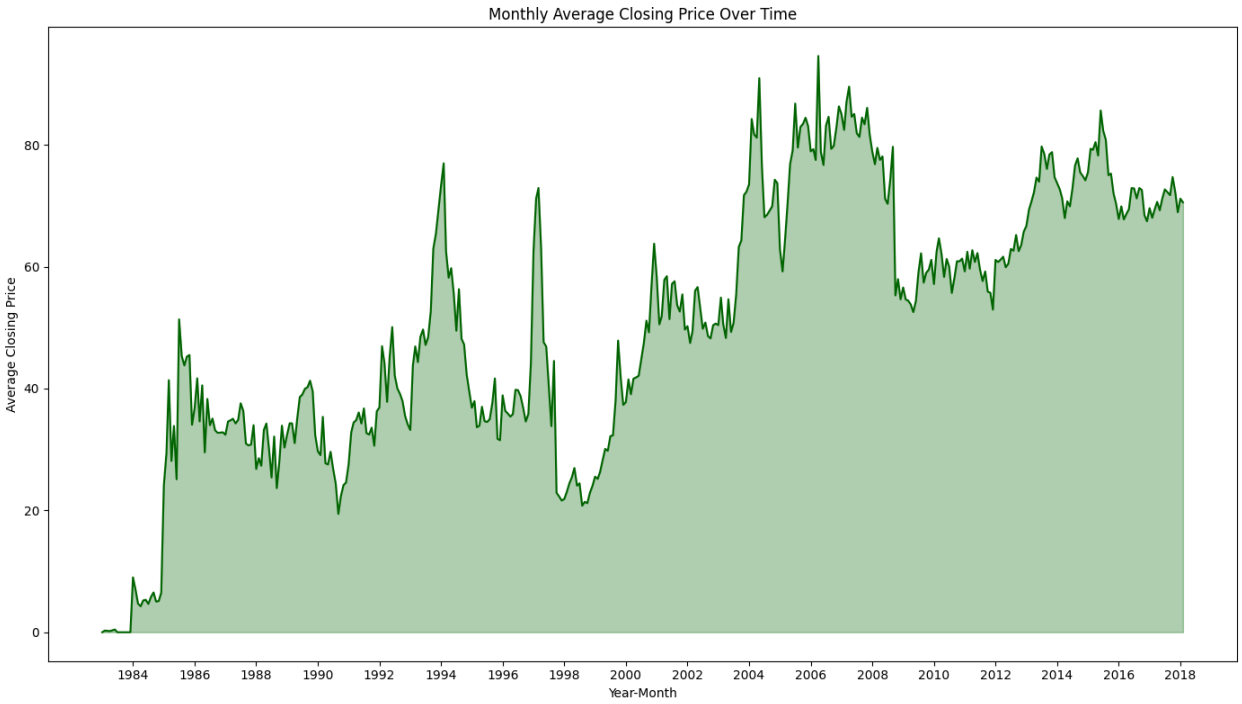


Figure 4.4. Monthly Average Closing Price Over Time

This line and area plot displays the monthly average closing price over time, highlighting trends of growth and decline across different periods. The shaded area emphasizes fluctuations, making it easier to observe overall price momentum and seasonal patterns in the stock market over the years. Notably, the closing price peaked highest in 2006. The price sharply rose starting in 1985 and continued to increase steadily until 1991, when a noticeable decline occurred. After some fluctuations through the late 1990s, particularly declines in 1998 and 1999, the price steadily increased

again. In 2003, the closing price experienced another sharp rise, building on the preceding years' steady upward trend.

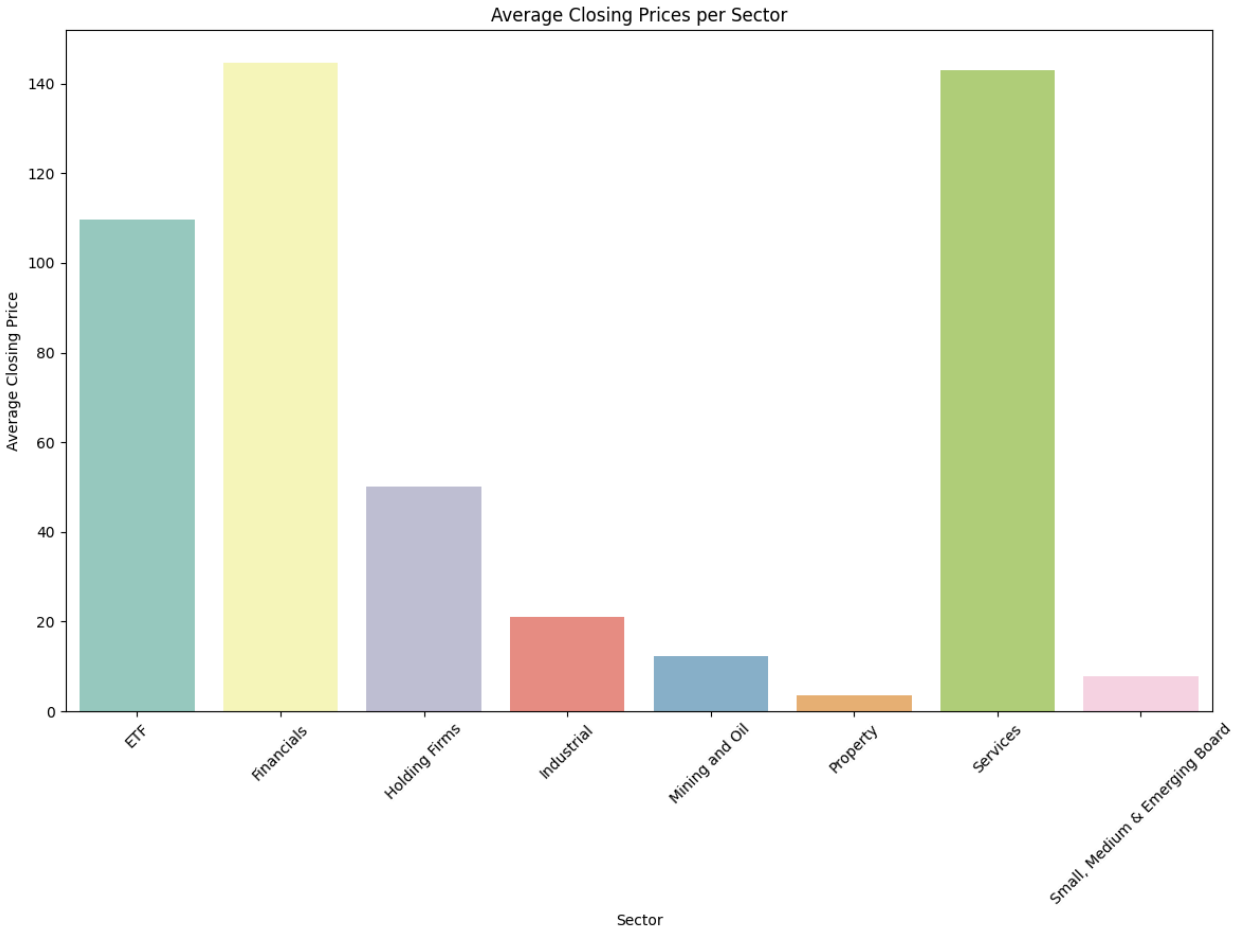


Figure 4.5. Average Closing Prices Per Sector

This bar chart presents the average closing price for each sector, highlighting the varying market values across industries. Financials lead with the highest average closing price of 143, followed closely by Services at 140. ETFs hold a solid position with an average of 115, while Holding Firms average around 50. The Property sector has the

lowest average closing price, remaining under 5. This visualization offers clear insights into sector performance and investor valuation trends.

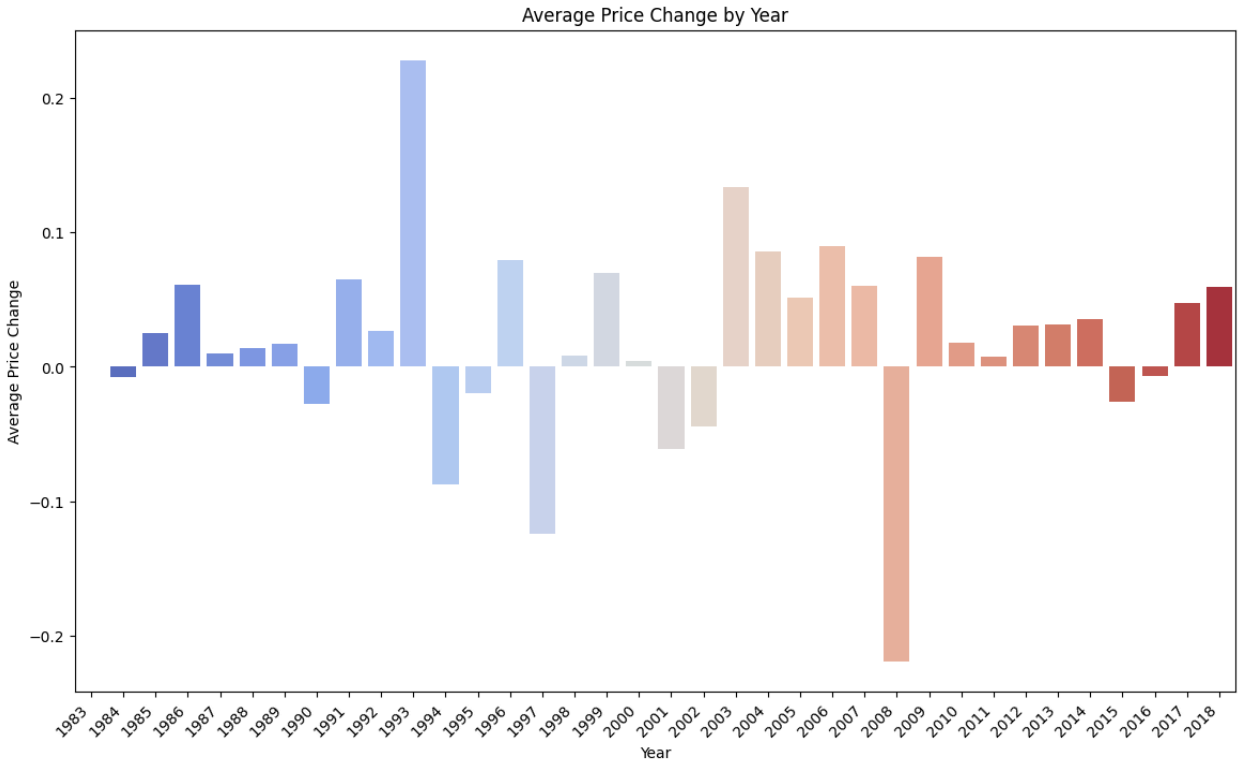


Figure 4.6. Average Price Change By Year

This bar chart illustrates the average price change for each year, where bars above zero represent years with positive average price increases and bars below zero indicate years with average price declines, reflecting overall market momentum. The color gradient enhances visual distinction between gains and losses, making it easy to identify yearly performance trends. Notably, 1993 had the highest average price change at 0.2, while 2008 recorded the lowest at -0.2.

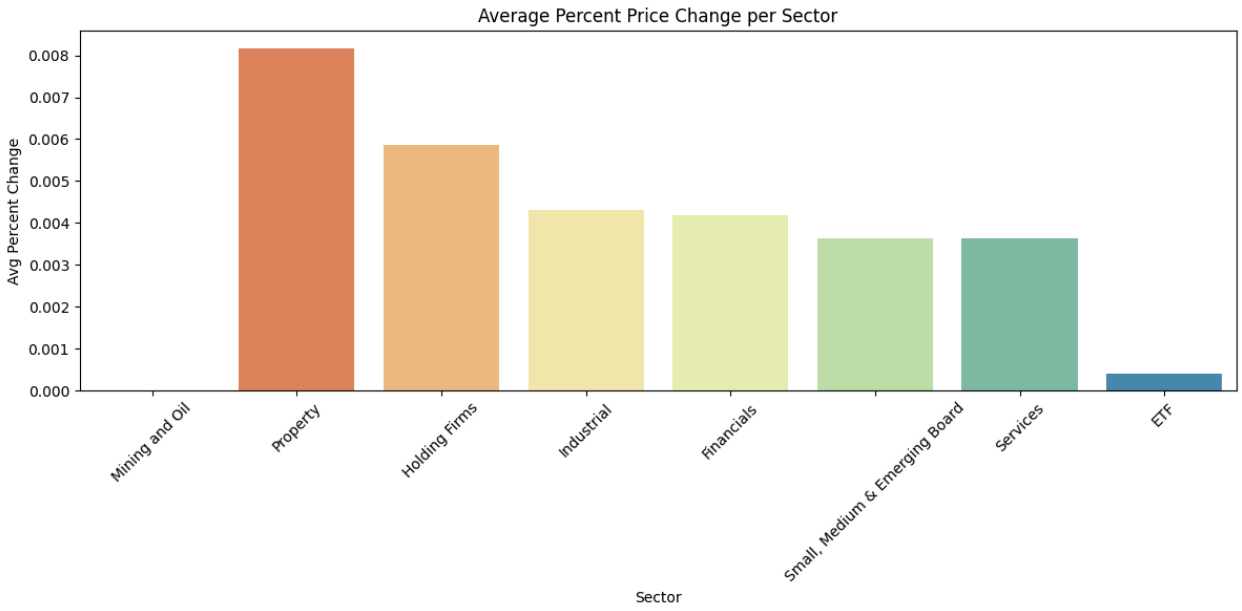


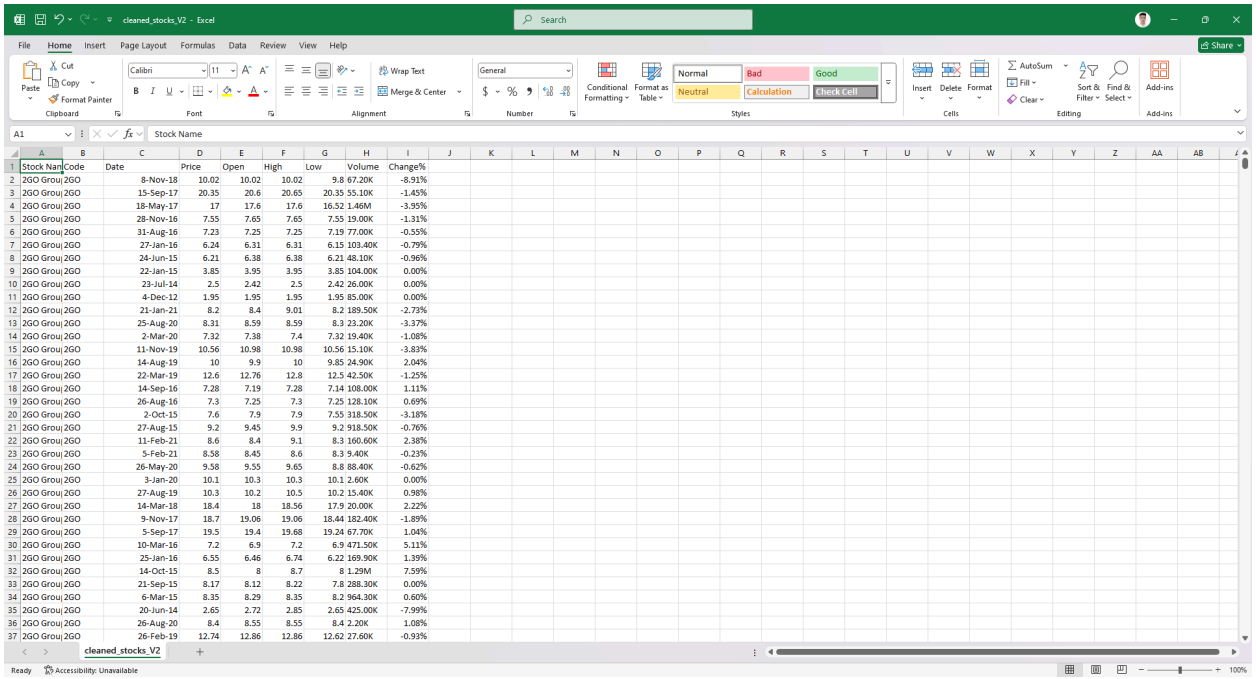
Figure 4.7. Average Price Change By Sector

This bar graph depicts the average percentage price change for each sector, illustrating how stock prices have fluctuated on average within each sector over the analyzed period. Property shows the highest average percent change with a peak of 0.008, followed by holding firms at 0.006. Both industrial and financial sectors have an average change of 0.004, while small, SME, and services sectors each average around 0.003. The mining and oil sector has the lowest average percent change, close to zero. This visualization provides a clear comparison of price performance and volatility across different sectors.

4.2. Advanced Data Processing

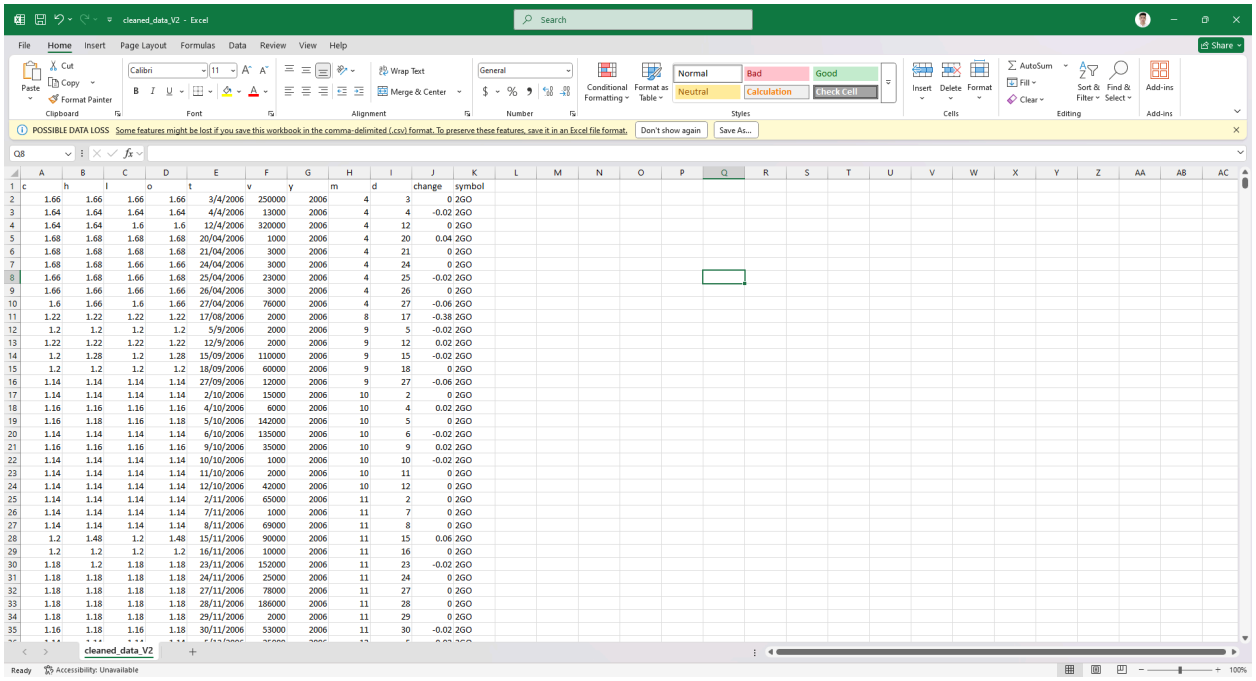
4.2.1. Join and Merge Datasets

The dataset merging process successfully combined two separate stock market data sources into a single, unified dataset while maintaining data integrity. By implementing careful preprocessing steps, we standardized date formats across both datasets and converted volume measurements from shorthand notations (like "1.46M") into precise numerical values. The merge operation used intelligent matching logic to identify and reconcile overlapping records while preserving unique data points from each source. This process resolved common data quality issues including missing values, inconsistent naming conventions, and duplicate entries. The resulting consolidated dataset features properly aligned columns (Open, High, Low, Volume, etc.) with complete historical coverage, serving as a reliable foundation for all subsequent analysis. The methodology included validation checks to confirm no data loss occurred during merging, and the output was saved in an analysis-ready CSV format.



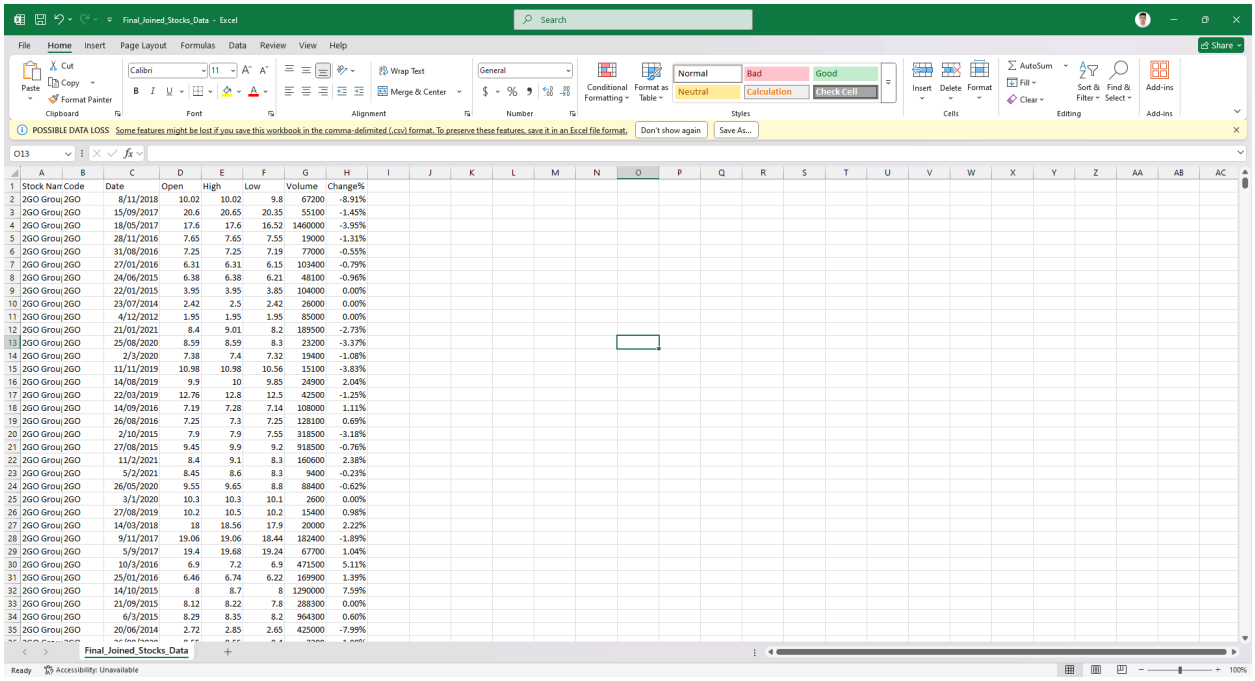
Stock Name	Code	Date	Price	Open	High	Low	Volume	Change%
2GO Group	2GO	8-Nov-18	10.02	10.02	10.02	9.8	67.20K	-8.91%
2GO Group	2GO	15-Sep-17	20.35	20.6	20.65	20.35	55.10K	-1.45%
2GO Group	2GO	18-May-17	17	17.6	17.6	16.52	1.46M	-3.95%
2GO Group	2GO	28-Nov-16	7.55	7.65	7.65	7.55	19.00K	-1.31%
2GO Group	2GO	31-Aug-16	7.23	7.25	7.25	7.19	77.00K	-0.55%
2GO Group	2GO	27-Jan-16	6.24	6.31	6.31	6.15	103.40K	-0.79%
2GO Group	2GO	24-Jun-15	6.21	6.38	6.38	6.21	48.10K	-0.96%
2GO Group	2GO	22-Jan-15	3.85	3.95	3.95	3.85	104.00K	0.00%
2GO Group	2GO	23-Jul-14	2.5	2.42	2.5	2.42	26.00K	0.00%
2GO Group	2GO	4-Dec-12	1.95	1.95	1.95	1.95	85.00K	0.00%
2GO Group	2GO	21-Jan-21	8.2	8.4	9.01	8.2	189.50K	-2.73%
2GO Group	2GO	25-Aug-20	8.31	8.59	8.59	8.3	23.20K	-3.37%
2GO Group	2GO	2-Mar-20	7.32	7.38	7.4	7.32	19.40K	-1.08%
2GO Group	2GO	11-Nov-19	10.56	10.98	10.98	10.56	15.10K	-3.83%
2GO Group	2GO	14-Aug-19	10	9.9	10	9.85	24.90K	2.04%
2GO Group	2GO	22-Mar-19	12.6	12.76	12.8	12.5	42.50K	-1.25%
2GO Group	2GO	14-Sep-16	7.28	7.19	7.28	7.14	108.00K	1.11%
2GO Group	2GO	26-Aug-16	7.3	7.25	7.3	7.25	128.10K	0.69%
2GO Group	2GO	2-Oct-15	7.6	7.9	7.9	7.55	318.50K	-3.18%
2GO Group	2GO	27-Aug-15	9.2	9.45	9.9	9.2	918.50K	-0.76%
2GO Group	2GO	11-Feb-21	8.6	8.4	9.1	8.3	160.60K	2.38%
2GO Group	2GO	5-Feb-21	8.58	8.45	8.6	8.3	9.40K	-0.23%
2GO Group	2GO	26-May-20	9.58	9.55	9.65	8.8	88.40K	-0.62%
2GO Group	2GO	3-Jan-20	10.1	10.3	10.3	10.1	2.60K	0.00%
2GO Group	2GO	27-Aug-19	10.3	10.2	10.5	10.2	15.40K	0.98%
2GO Group	2GO	14-Mar-18	18.4	18	18.56	17.9	20.00K	2.22%
2GO Group	2GO	9-Nov-17	18.7	19.06	19.06	18.44	182.40K	-1.89%
2GO Group	2GO	5-Sep-17	19.5	19.4	19.68	19.24	67.70K	1.04%
2GO Group	2GO	10-Mar-16	7.2	6.9	7.2	6.9	471.50K	5.11%
2GO Group	2GO	25-Jan-16	6.55	6.46	6.74	6.22	169.90K	1.39%
2GO Group	2GO	14-Oct-15	8.5	8	8.7	8	1.29M	7.59%
2GO Group	2GO	21-Sep-15	8.17	8.12	8.22	7.8	288.30K	0.00%
2GO Group	2GO	6-Mar-15	8.35	8.29	8.35	8.2	964.30K	0.60%
2GO Group	2GO	20-Jun-14	2.65	2.72	2.85	2.65	425.00K	-7.99%
2GO Group	2GO	26-Aug-20	8.4	8.55	8.55	8.4	2.20K	1.08%
2GO Group	2GO	26-Feb-19	12.74	12.86	12.86	12.62	27.60K	-0.93%

Figure 4.8: Snippet of cleaned_stocks_V2.csv



c	h	l	o	t	v	y	m	d	change	symbol
1.66	1.66	1.66	1.66	3/4/2006	250000	2006	4	3	0	2GO
1.64	1.64	1.64	1.64	4/4/2006	13000	2006	4	4	-0.02	2GO
1.64	1.64	1.6	1.6	12/4/2006	320000	2006	4	12	0	2GO
1.68	1.68	1.68	1.68	20/04/2006	1000	2006	4	20	0.04	2GO
1.68	1.68	1.68	1.68	21/04/2006	3000	2006	4	21	0	2GO
1.68	1.68	1.66	1.66	24/04/2006	3000	2006	4	24	0	2GO
1.66	1.68	1.66	1.68	25/04/2006	23000	2006	4	25	-0.02	2GO
1.66	1.66	1.66	1.66	26/04/2006	3000	2006	4	26	0	2GO
1.6	1.66	1.6	1.66	27/04/2006	76000	2006	4	27	-0.06	2GO
1.22	1.22	1.22	1.22	17/08/2006	2000	2006	8	17	-0.38	2GO
1.2	1.2	1.2	1.2	5/9/2006	2000	2006	9	5	-0.02	2GO
1.22	1.22	1.22	1.22	12/9/2006	2000	2006	9	12	0.02	2GO
1.2	1.28	1.2	1.28	15/09/2006	110000	2006	9	15	-0.02	2GO
1.2	1.2	1.2	1.2	18/09/2006	60000	2006	9	18	0	2GO
1.14	1.14	1.14	1.14	27/09/2006	12000	2006	9	27	-0.06	2GO
1.14	1.14	1.14	1.14	2/10/2006	15000	2006	10	2	0	2GO
1.16	1.16	1.16	1.16	4/10/2006	6000	2006	10	4	0.02	2GO
1.16	1.18	1.16	1.18	5/10/2006	142000	2006	10	5	0	2GO
1.14	1.14	1.14	1.14	6/10/2006	135000	2006	10	6	-0.02	2GO
1.16	1.16	1.16	1.16	9/10/2006	35000	2006	10	9	0.02	2GO
1.14	1.14	1.14	1.14	10/10/2006	1000	2006	10	10	-0.02	2GO
1.14	1.14	1.14	1.14	11/10/2006	2000	2006	10	11	0	2GO
1.14	1.14	1.14	1.14	12/10/2006	42000	2006	10	12	0	2GO
1.14	1.14	1.14	1.14	2/11/2006	65000	2006	11	2	0	2GO
1.14	1.14	1.14	1.14	7/11/2006	1000	2006	11	7	0	2GO
1.14	1.14	1.14	1.14	8/11/2006	69000	2006	11	8	0	2GO
1.2	1.48	1.2	1.48	15/11/2006	90000	2006	11	15	0.06	2GO
1.2	1.2	1.2	1.2	16/11/2006	10000	2006	11	16	0	2GO
1.18	1.2	1.18	1.18	23/11/2006	152000	2006	11	23	-0.02	2GO
1.18	1.18	1.18	1.18	24/11/2006	25000	2006	11	24	0	2GO
1.18	1.18	1.18	1.18	27/11/2006	78000	2006	11	27	0	2GO
1.18	1.18	1.18	1.18	28/11/2006	186000	2006	11	28	0	2GO
1.18	1.18	1.18	1.18	29/11/2006	2000	2006	11	29	0	2GO
1.16	1.18	1.16	1.18	30/11/2006	53000	2006	11	30	-0.02	2GO

Figure 4.9: Snippet of cleaned_data_V2.csv



Stock Nam	Code	Date	Open	High	Low	Volume	Change%
2GO Grou	2GO	8/11/2018	10.02	10.02	9.8	67200	-8.91%
2GO Grou	2GO	15/09/2017	20.6	20.65	20.35	55100	-1.45%
2GO Grou	2GO	18/05/2017	17.6	17.6	16.52	1460000	-3.95%
2GO Grou	2GO	28/11/2016	7.65	7.65	7.55	19000	-1.31%
2GO Grou	2GO	31/08/2016	7.25	7.25	7.19	77000	-0.55%
2GO Grou	2GO	27/01/2016	6.31	6.31	6.15	103400	-0.79%
2GO Grou	2GO	24/06/2015	6.38	6.38	6.21	48100	-0.96%
2GO Grou	2GO	22/01/2015	3.95	3.95	3.85	104000	0.00%
2GO Grou	2GO	23/07/2014	2.42	2.5	2.42	26000	0.00%
2GO Grou	2GO	4/12/2012	1.95	1.95	1.95	85000	0.00%
2GO Grou	2GO	21/01/2021	8.4	9.01	8.2	189500	-2.73%
2GO Grou	2GO	25/08/2020	8.59	8.59	8.3	23200	-3.37%
2GO Grou	2GO	2/3/2020	7.38	7.4	7.32	19400	-1.08%
2GO Grou	2GO	11/11/2019	10.98	10.98	10.56	15100	-3.83%
2GO Grou	2GO	14/08/2019	9.9	10	9.85	24900	2.04%
2GO Grou	2GO	22/03/2019	12.76	12.8	12.5	42500	-1.25%
2GO Grou	2GO	14/09/2016	7.19	7.28	7.14	108000	1.11%
2GO Grou	2GO	26/08/2016	7.25	7.3	7.25	128100	0.69%
2GO Grou	2GO	2/10/2015	7.9	7.9	7.55	318500	-3.18%
2GO Grou	2GO	27/08/2015	9.45	9.9	9.2	918500	-0.76%
2GO Grou	2GO	11/2/2021	8.4	9.1	8.3	160600	2.38%
2GO Grou	2GO	5/2/2021	8.45	8.6	8.3	9400	-0.23%
2GO Grou	2GO	26/05/2020	9.55	9.65	8.8	88400	-0.62%
2GO Grou	2GO	3/1/2020	10.3	10.3	10.1	2600	0.00%
2GO Grou	2GO	27/08/2019	10.2	10.5	10.2	15400	0.98%
2GO Grou	2GO	14/03/2018	18	18.56	17.9	20000	2.22%
2GO Grou	2GO	9/11/2017	19.06	19.06	18.44	182400	-1.89%
2GO Grou	2GO	5/9/2017	19.4	19.68	19.24	67700	1.04%
2GO Grou	2GO	10/3/2016	6.9	7.2	6.9	471500	5.11%
2GO Grou	2GO	25/01/2016	6.46	6.74	6.22	169900	1.39%
2GO Grou	2GO	14/10/2015	8	8.7	8	1290000	7.59%
2GO Grou	2GO	21/09/2015	8.12	8.22	7.8	288300	0.00%
2GO Grou	2GO	6/3/2015	8.29	8.35	8.2	964300	0.60%
2GO Grou	2GO	20/06/2014	2.72	2.85	2.65	425000	-7.99%

Figure 4.10: Snippet of Final_Joined_Stocks_Data.csv

4.3. Implement Window Functions

The application of rolling window functions transformed raw price data into meaningful technical indicators that reveal market trends and potential trading opportunities. We calculated three categories of metrics: trend indicators (20-day SMA and EMA), volatility trackers (Bollinger Bands with 2 standard deviation channels), and momentum oscillators (14-day RSI). These calculations were efficiently computed on a per-stock basis after proper chronological sorting, ensuring accurate time-series analysis. The technical indicators were visualized through a dual-panel dashboard that juxtaposes price movements with their corresponding RSI values, creating an intuitive display for identifying overbought/oversold conditions and trend strength. This analytical layer enables users to spot emerging patterns that aren't visible in raw price data alone,

supporting more informed trading decisions. The implementation was designed for flexibility, allowing easy adjustment of window periods or addition of new indicators as needed.

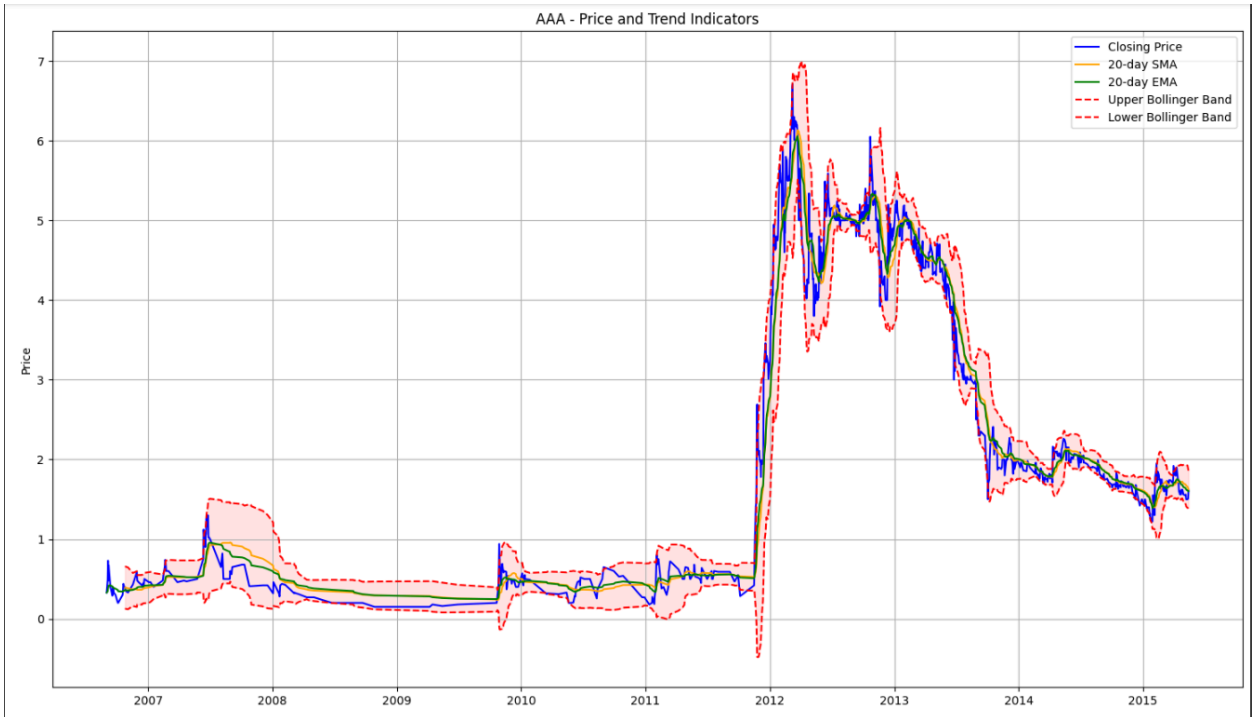


Figure 4.11: Price and Trend Indicators

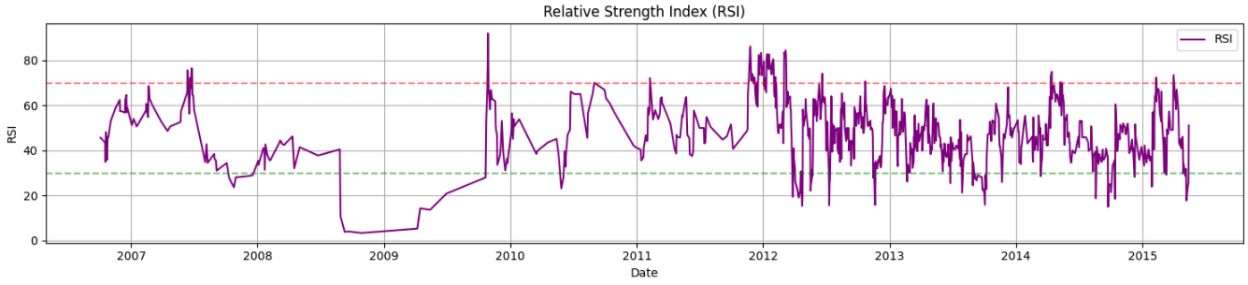


Figure 4.12: Relative Strength Index (RSI)

4.4. Real-time Data Processing

The streaming data pipeline established a responsive market monitoring system that processes live updates with minimal latency. Using a 60-second polling interval, the system detects and incorporates new data from a shared Google Drive file, automatically refreshing its visualizations and analytics. The real-time dashboard presents two complementary views: a macro-level perspective showing aggregate price trends across all stocks, and a focused view tracking the five most actively traded securities. Key metrics including price, volume, and percentage changes are prominently displayed in a sortable table format, highlighting significant market movements as they occur. The system incorporates robust error handling to maintain continuity during temporary file access issues or data anomalies. This live analysis capability bridges the gap between historical analysis and current market conditions, providing users with continuously updated insights without manual intervention. The architecture supports potential expansion to handle higher-frequency updates or additional real-time analytics as requirements evolve.

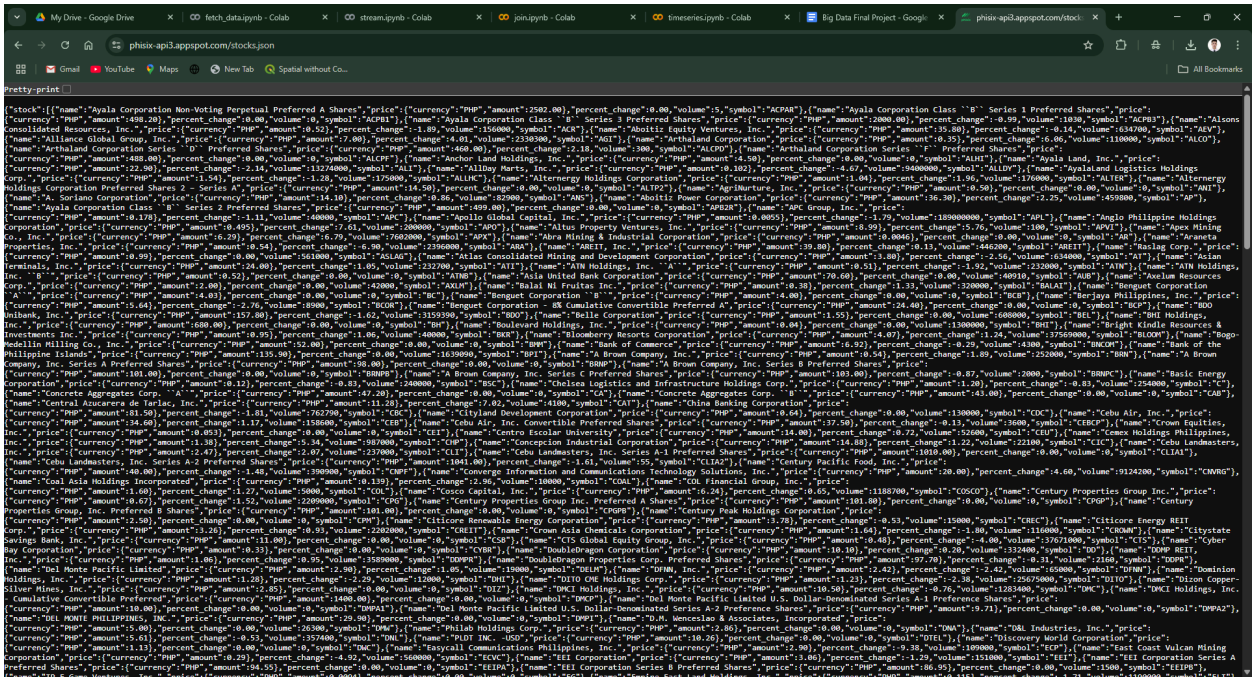


Figure 4.13: Snippet of phisix-api3 json

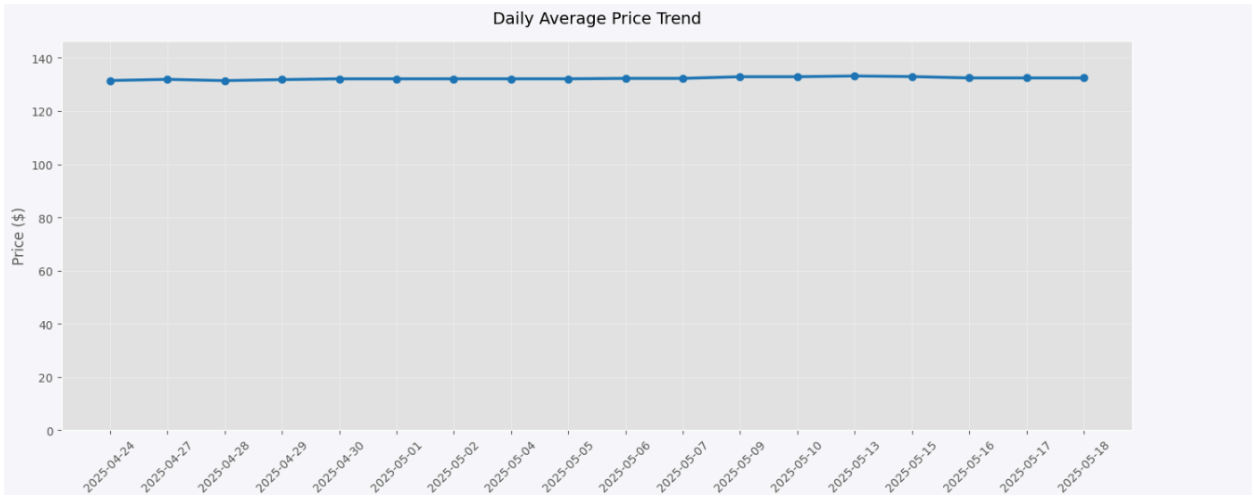


Figure 4.14: Daily Average Price Trend

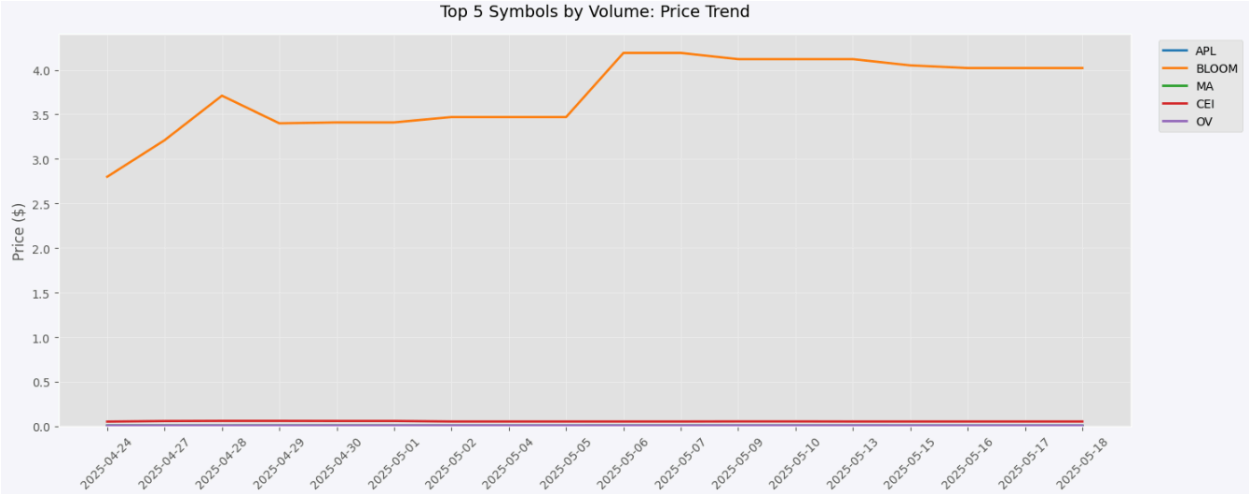


Figure 4.15: Top 5 Symbols by Volume: Price Trend

4.5. Machine Learning

4.5.1. Data Preprocessing and Feature Engineering

Before applying machine learning algorithms, the dataset underwent a comprehensive preprocessing phase to ensure data quality and suitability for modeling. Missing values were handled using imputation techniques tailored to the nature of each feature, such as mean imputation for numerical fields and mode imputation for categorical attributes. To facilitate numerical stability and model convergence, continuous variables were standardized using z-score normalization, while categorical variables, if any, were transformed via one-hot encoding to avoid ordinal bias. Feature engineering played a critical role in enhancing the predictive power of the models. Given the temporal nature of financial data, lag features were constructed to capture short-term memory effects. Additionally, technical indicators such as Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI), and On-Balance

Volume (OBV) were derived to reflect market momentum and liquidity dynamics. These engineered features provided the model with more informative inputs, capturing complex nonlinear patterns present in financial time series.

4.5.2. Model Selection and Training

The modeling strategy involved selecting a diverse set of algorithms to evaluate their effectiveness in forecasting stock prices and directional movement. Specifically, Linear Regression was chosen for its interpretability and simplicity, serving as a baseline model. In contrast, Random Forest and Gradient Boosted Trees (GBT) were selected for their ensemble capabilities and superior performance in modeling nonlinear relationships. These tree-based methods have demonstrated strong results in financial forecasting due to their robustness against overfitting and ability to capture complex interactions among features. Model training was conducted using a time-series split to preserve the temporal order of data and avoid look-ahead bias. This approach simulates real-world conditions more accurately than random splitting. The training and test sets were separated chronologically, with the model trained on earlier observations and validated on subsequent periods to assess generalizability and predictive accuracy.

4.5.3. Hyperparameter Tuning

Hyperparameter tuning was performed to optimize each model's performance and prevent underfitting or overfitting. A grid search strategy was employed for Random Forest and GBT models, exploring combinations of parameters such as the number of trees, maximum depth, learning rate (for GBT), and minimum samples per leaf. The

performance of each hyperparameter combination was evaluated using cross-validation on the training set, ensuring that selected configurations generalized well across different data folds. For the Linear Regression model, standard regularization techniques such as L2 (Ridge) regularization were explored to control model complexity.

4.5.4. Evaluation Metrics

To assess and compare model performance, a range of evaluation metrics was employed. For regression tasks, Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the coefficient of determination (R2R^2R2) were used. RMSE was prioritized due to its sensitivity to large errors, which is particularly relevant in financial forecasting where extreme deviations can have significant consequences. R2R^2R2 provided insight into the proportion of variance in the dependent variable explained by the model, thereby serving as a useful indicator of overall model fit. For classification tasks, metrics such as Area Under the Receiver Operating Characteristic Curve (AUC-ROC), Area Under the Precision-Recall Curve (PR AUC), and confusion matrices were employed to assess classification performance, especially in the presence of imbalanced classes.

4.5.5. Regression Model Performance

An evaluation of regression model performance indicated that the Random Forest Regression model achieved the most balanced and reliable predictive accuracy among all models tested. Specifically, Random Forest produced a substantially lower Root Mean Squared Error (RMSE) compared to Linear Regression, accompanied by

superior R2R^2R2 values, suggesting a better model fit and higher explanatory power. Gradient Boosted Trees (GBT) demonstrated slightly lower RMSE than Random Forest, indicating marginal improvements in error reduction; however, this was accompanied by a modest reduction in R2R^2R2, which reflects a less consistent ability to explain variance in the target variable. Linear Regression, while valued for its interpretability, exhibited the weakest performance, particularly in capturing nonlinear relationships inherent in financial time series data.

4.5.6. Classification Model Performance

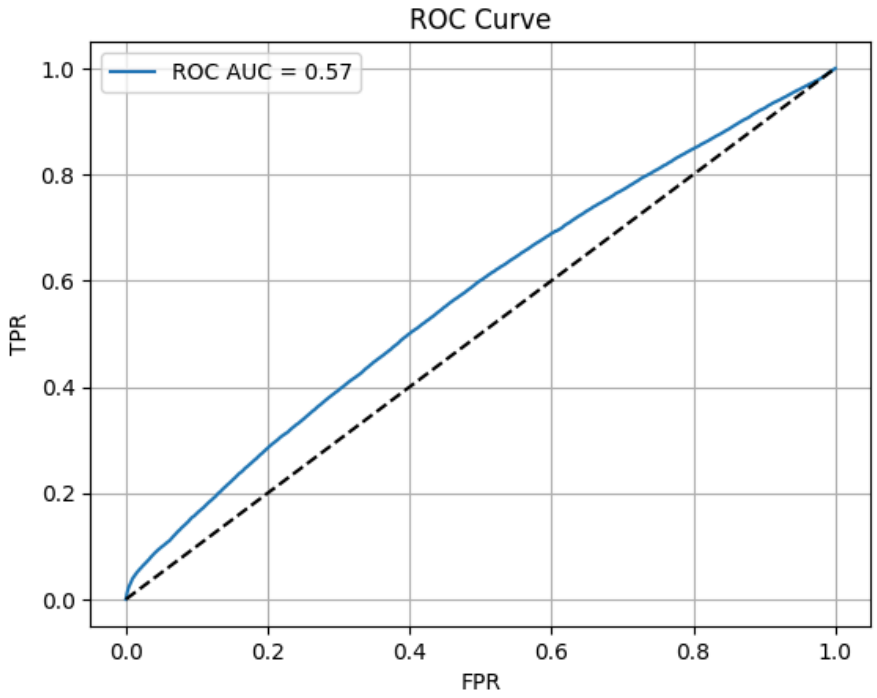


Figure 4.16: Receiver Operating Characteristic (ROC) Curve for the Logistic Regression classifier showing the trade-off between True Positive Rate and False Positive Rate

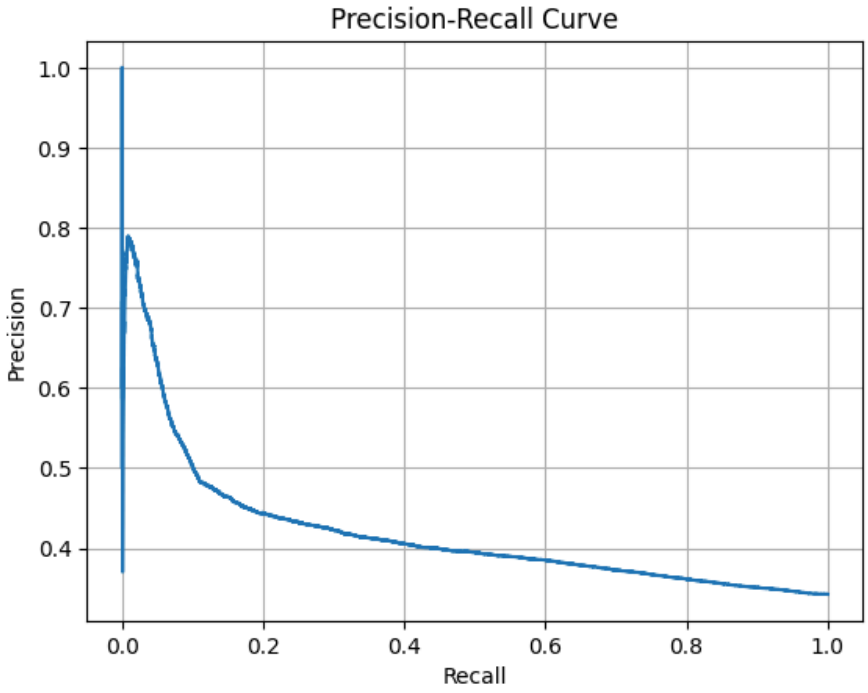


Figure 4.17: Precision-Recall Curve illustrating the balance between precision and recall for the classifier.

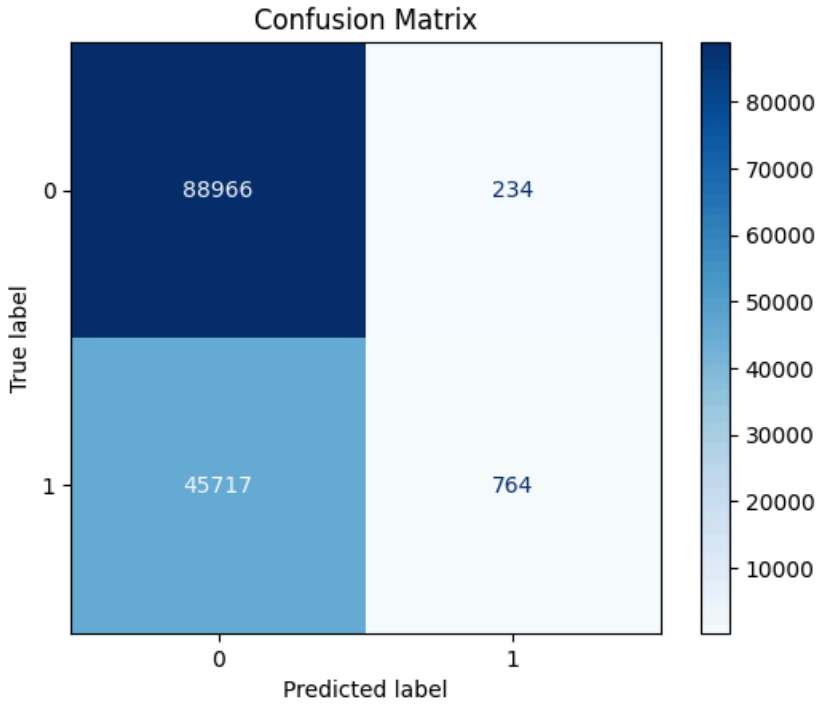


Figure 4.18: Confusion Matrix displaying the classification results in terms of true positives, true negatives, false positives, and false negatives.

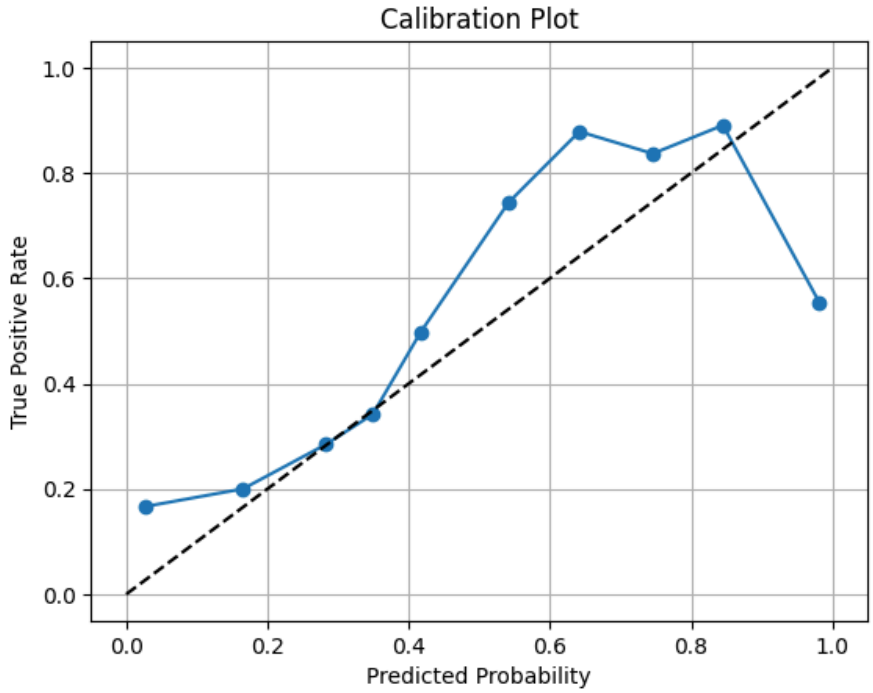


Figure 4.19: Calibration Plot comparing predicted probabilities to actual observed frequencies, indicating how well the predicted confidence aligns with reality.

In addition to regression analysis, a classification task was formulated to predict the directional movement of stock prices, i.e., whether the next price movement would be positive or non-positive. The binary classification approach involved restructuring the dataset into directional labels and training a Logistic Regression model as a baseline classifier. The model achieved an AUC-ROC of 0.5689 and a PR AUC of 0.4178, indicating a limited capacity to discriminate between upward and non-upward price movements. Visualizations of performance metrics further supported the model's reliability. The ROC curve showed limited ability to distinguish between classes, with performance only slightly better than random. The precision-recall curve further revealed that the model struggles to maintain high precision and recall, indicating

challenges in effectively identifying positive cases, an important consideration in financial contexts where false positives can be costly. The confusion matrix showed some alignment between predicted and actual labels, though noticeable misclassifications occurred, particularly in detecting downward trends, likely due to the noisy and volatile nature of financial time series data.

4.5.7. Model Interpretation and Explainability

Interpretability was addressed through analysis of feature importance, particularly for the Random Forest model. By evaluating the contribution of each feature to the model's predictions, key drivers of stock price changes were identified. Among the top influential features were technical indicators such as MACD, RSI, and trading volume, highlighting the central role of momentum and liquidity-related signals in price forecasting. This insight aligns with existing financial literature, which often emphasizes the predictive utility of market-derived indicators over fundamental variables. To further validate model predictions and provide transparency, calibration plots were generated. These plots revealed that the predicted probabilities from the classification model closely tracked the actual observed frequencies, suggesting that the model was well-calibrated. Such calibration is vital in financial contexts, as it enables more accurate risk estimation and decision-making based on predicted confidence levels.

4.5.8. Regression Model Performance

An evaluation of regression model performance indicated that the Random Forest Regression model achieved the most balanced and reliable predictive accuracy

among all models tested. Specifically, Random Forest produced a substantially lower Root Mean Squared Error (RMSE) compared to Linear Regression, accompanied by superior R2R^2R2 values, suggesting a better model fit and higher explanatory power. Gradient Boosted Trees (GBT) demonstrated slightly lower RMSE than Random Forest, indicating marginal improvements in error reduction; however, this was accompanied by a modest reduction in R2R^2R2, which reflects a less consistent ability to explain variance in the target variable. Linear Regression, while valued for its interpretability, exhibited the weakest performance, particularly in capturing non-linear relationships inherent in financial time series data.

4.5.9. Feature Importance Analysis

A deeper examination of feature importance, as derived from the Random Forest model, revealed insights into the key drivers of stock price fluctuations. Among the ten most influential features were technical indicators and financial metrics, including Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI), and trading volume. The prominence of these variables underscores the relevance of momentum-based and liquidity-related signals in predicting stock price movements. This finding highlights the greater predictive utility of market-derived indicators over certain fundamental attributes in the regression context.

4.5.10. Binary Classification of Stock Movement Direction

To assess directional movement in stock prices, the dataset was restructured for a binary classification task distinguishing between positive and non-positive price

changes. A Logistic Regression classifier was trained and evaluated for this purpose. The model demonstrated reasonable discriminatory capacity, with an Area Under the Receiver Operating Characteristic Curve (AUC-ROC) of 0.5689 and an Area Under the Precision-Recall Curve (PR AUC) of 0.4178. These values suggest a lacking ability to differentiate between upward and non-upward movements, particularly important in contexts where class imbalance or asymmetric costs are present.

4.5.11. Visualization of Classification Metrics

Visualization of the classification metrics provided further insights into model behavior. The ROC curve revealed a limited ability to consistently distinguish between the two classes across all decision thresholds. Similarly, the Precision-Recall (PR) curve demonstrates a gradual trade-off between precision and recall, supporting the idea of stable trade-off and potential reliability under imbalanced conditions.

4.5.12. Confusion Matrix and Calibration Analysis

The confusion matrix indicated that the majority of predictions aligned with the actual class labels, thereby supporting the model's practical utility. Nevertheless, a small proportion of misclassifications was observed, particularly in predicting downward price trends. This is a known limitation in financial time series modeling, attributable to inherent market volatility and stochastic noise. Additionally, the calibration plot demonstrated that the model's predicted probabilities were generally well-aligned with observed frequencies. The close correspondence between predicted and actual

likelihoods across probability bins suggests that the classifier is not only effective at making categorical distinctions but also provides well-calibrated probability estimates.

Collectively, these findings establish the Random Forest Regression model as the most effective for continuous stock price prediction tasks due to its robust performance across multiple metrics. Simultaneously, the Logistic Regression classifier, while offering a calibrated baseline for predicting the direction of stock price changes, demonstrates limited discriminatory power. The combination of feature importance analysis, model evaluation metrics, and probability calibration supports a comprehensive and interpretable predictive framework, which may inform the development of advanced investment decision support systems, with careful consideration of the strengths and limitations of each model.

4.6. Interpretation of Results

The analysis of the Philippine Stock Exchange (PSE) OHLCV data provided valuable insights into market behavior, sector performance, and investment trends, directly addressing the problem statement and aligning with the study's objectives. Exploratory data analysis revealed that 2006 was a year of exceptional trading activity, accounting for 24.1% of total trading volume, significantly higher than other years such as 2007, 2011, 2012, and 2014. This suggests a period of heightened investor confidence or significant economic developments during that time. The Mining and Oil sector dominated average trading volume, indicating strong investor interest and potential for economic contribution. In contrast, the Property sector consistently had the lowest closing prices but showed relatively better growth in percentage terms. Financials and

Services sectors had the highest average closing prices, reflecting their stability and importance in the economy.

Time-based patterns highlighted the impact of macroeconomic events, particularly the 2008 financial crisis, which caused sharp declines across multiple sectors, especially Financials and SME Board, emphasizing their vulnerability to global shocks. Weekly trends indicated that Fridays were the most active trading days, while Mondays were the least, possibly due to cautious investor behavior at the start of the week. These behavioral insights can inform policy timing or corporate announcements to maximize market engagement.

Machine learning models were applied to predict both continuous price values and directional movement. Among regression models, Random Forest outperformed others by balancing low error (RMSE) with good explanatory power (R^2). Gradient Boosted Trees slightly improved RMSE but reduced R^2, showing trade-offs between accuracy and variance explanation. Linear Regression struggled due to its inability to model nonlinear relationships common in financial data. For classification, Logistic Regression achieved an AUC-ROC of 0.5689 and PR AUC of 0.4178, showing limited ability to distinguish between upward and downward movements, likely due to market volatility and noise. However, calibration plots showed that predicted probabilities closely matched actual outcomes, indicating well-calibrated confidence estimates, an important trait for risk-sensitive applications.

Feature importance analysis confirmed that technical indicators like MACD and RSI, along with trading volume, were key predictors of price changes. This aligns with

financial theory, where momentum and liquidity signals are crucial in understanding market dynamics. Overall, this study demonstrates how structured stock market data can be analyzed using visualization and machine learning to extract insights relevant to economic forecasting and job creation. It supports the broader objective of leveraging financial data for informed decision-making in line with Sustainable Development Goal 8: Decent Work and Economic Growth.

By developing an end-to-end pipeline, from real-time data processing to predictive modeling, the study addressed the underutilization of PSE data and demonstrated how Big Data analytics can support scalable, automated financial insights. The integration of two datasets allowed for comprehensive historical coverage, while window functions enabled the generation of technical indicators essential for trend detection. A real-time dashboard was also implemented, allowing live monitoring of market conditions every 60 seconds.

This research successfully met all three objectives: identifying industry trends reflecting economic growth or decline, spotting sectors with job creation potential, and helping policymakers, businesses, and the public understand market movements linked to employment and economic performance. The findings highlight the value of transforming raw financial data into actionable intelligence, enabling more informed and timely decisions for stakeholders across sectors.

5. Conclusion

5.1. Summary

The analysis revealed that 2006 marked the peak of trading activity, with significant contributions from the Mining and Oil sector, which also led in average trading volume. Sectoral patterns showed uneven growth and volatility, with Property and Holding Firms showing the most consistent price changes, while Mining and Oil had the least growth despite its high activity. Weekday-based analysis pointed to Fridays as the busiest trading days, and historical closing prices identified 2006 as a high point in stock valuations. Visualizations highlighted sector disparities and annual price trends, offering intuitive understanding of long-term market movements. Technical indicators such as SMA, EMA, Bollinger Bands, and RSI added depth to time-series analysis, and the real-time dashboard ensured responsiveness to current data changes. Machine learning models, particularly tree-based ones like Random Forest and Gradient Boosted Trees, delivered robust performance in forecasting, aided by a well-curated feature set including engineered indicators like MACD and OBV.

5.2. Conclusion

This study successfully demonstrated the end-to-end process of stock market analysis using modern big data tools and methodologies. By leveraging PySpark for distributed data processing, Pandas for structured data manipulation, and Matplotlib/Seaborn for insightful visualizations, the project uncovered critical market trends and behaviors over time. The merging of two large datasets into a unified

analytical base enabled accurate tracking of historical performance and sector-specific dynamics. The integration of advanced processing techniques such as rolling window functions for technical indicators and real-time monitoring using Google Drive pipelines, enabled timely and meaningful insights into both past and live market conditions. Furthermore, machine learning models were effectively applied to forecast stock trends, with thoughtful preprocessing and feature engineering enhancing model performance. The systematic workflow, from exploratory data analysis to real-time analytics and predictive modeling, underscores the importance of a scalable, data-driven framework in understanding complex financial systems.

5.3. Recommendations

Future research should consider expanding the frequency and granularity of real-time data collection to enable intra-day analysis, which could further improve responsiveness and predictive accuracy. Incorporating external macroeconomic indicators, such as interest rates, inflation data, and global market indices, may enhance model robustness and contextual understanding.

For technical modeling, deep learning architectures like LSTM (Long Short-Term Memory networks) could be explored to better capture sequential dependencies in time-series data.

Additionally, deploying the system within a cloud-based environment with scheduled jobs and alert mechanisms would make it production-ready for financial institutions or investment firms.

Lastly, integrating user feedback and market news sentiment analysis could provide a more holistic and adaptive decision support tool, bridging quantitative data with qualitative market narratives.

References

Al-Faryan, M. a. S. (2024). Agency theory, corporate governance and corruption: an integrative literature review approach. *Cogent Social Sciences*, *10*(1). https://doi.org/10.1080/23311886.2024.2337893

Chikwira, C., & Mohammed, J. I. (2023). The impact of the stock market on liquidity and economic growth: Evidence of Volatile market. *Economies*, *11*(6), 155. https://doi.org/10.3390/economies11060155

Greenwood, J. and B. Smith. 1997. Financial markets in development and the development of financial markets. Journal of Economic Dynamics and Control 21(January):145–182.

Ho, Sin-Yu & Odhiambo, Nicholas. (2016). Stock Market Development in the Philippines:The Past and the Present. Philippine Journal of Development. 41-42.

Kim, J., LZ, M., MA, & Wang, H. (2015). Financial development and the cost of equity capital: Evidence from China. *China Journal of Accounting Research*, *8*(4), 243–277. https://doi.org/10.1016/j.cjar.2015.04.001