BIG DATA AND DATABASE MANAGEMENT SYSTEM

21AIE304

NIFTY FMCG SECTOR ANALYSIS

BATCH: B

GROUP: 2

COURSE INSTRUCTOR: Dr. SANJANA SRI J P

JAIDEV – CB.EN.U4AIE21117

CHANDANA – CB.EN.U4AIE21118

PRANISH – CB.EN.U4AIE21137

CHARISHMA – CB.EN.U4AIE21169

TABLE OF CONTENTS:

ABSTRACT:

This report presents a comprehensive analysis of Nifty FMCG utilizing a data-driven approach, integrating technologies such as Spark, MySQL, and Python. The project encompasses data collection, integration with Spark for distributed processing, and subsequent storage in Data Frames. Machine learning techniques are employed for price prediction, with visualizations and analyses displayed through a user-friendly GUI.

The report begins with an introduction outlining the project's objectives and scope. It details the data collection process, including sources and preprocessing steps. Integration with Spark is explained, emphasizing its role in distributed processing. Data Frame creation in Spark is discussed, providing insights into the structured representation of the data.

Machine learning aspects cover techniques employed for price prediction, encompassing features, target variables, and model evaluation. Visualizations, created using Python libraries, are presented to enhance data understanding. A user-friendly GUI integrates analyses and visualizations, fostering an interactive experience.

Key findings from the analysis are summarized, providing insights gained from both visualizations and machine learning predictions. Challenges and limitations are addressed, paving the way for future improvements. The report concludes by emphasizing the project's significance and proposing avenues for future research.

We express my sincere gratitude to Dr. Sanjana Sri for her invaluable guidance throughout this project. Her expertise and encouragement have been instrumental in its success, enhancing both our technical skills and understanding of data analysis and machine learning.

INTRODUCTION:

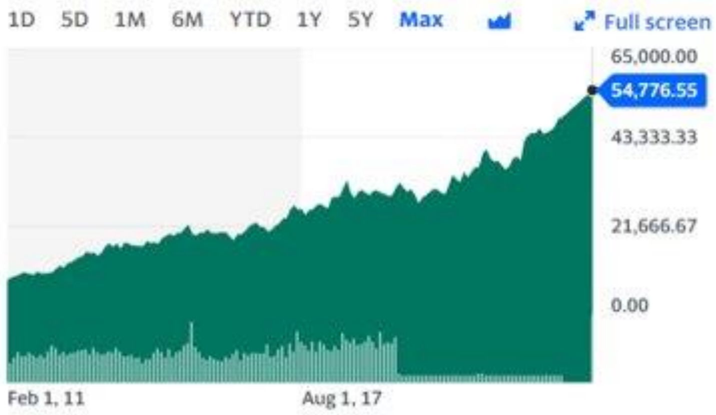
This project undertakes a comprehensive analysis of Nifty FMCG, leveraging a multifaceted approach integrating Spark, MySQL, and Python. The primary objective is to gain insights into the FMCG sector's dynamics by employing data analytics and machine learning. The scope encompasses data collection, integration with Spark for efficient processing, machine learning model development for price prediction, and the creation of a user-friendly GUI for result visualization. By amalgamating these technologies, the project aims to offer a holistic understanding of market trends and facilitate informed decision-making.

MOTIVATION:

In an age dominated by data, the motivation behind this project stems from the profound impact data-driven insights can have on navigating the complexities of the FMCG landscape. The fast-paced and dynamic nature of the sector demands a proactive approach, where understanding market trends becomes synonymous with success. By delving into the depths of Nifty FMCG, we seek to empower decision-makers with the tools to anticipate shifts, capitalize on opportunities, and navigate challenges effectively. The motivation is rooted in the belief that a thorough analysis, powered by technologies like Spark and machine learning, can unveil hidden patterns and provide a competitive edge in an industry where staying ahead is paramount. This project, therefore, serves as a testament to the transformative potential of data analytics, fostering innovation and strategic acumen in the realm of FMCG.

DATA COLLECTION:

The primary source of our Nifty FMCG data is Yahoo Finance, offering a comprehensive dataset spanning from 2012 to December 2023. Yahoo Finance provides a reliable and up-to-date repository of financial information, ensuring the inclusion of relevant data points crucial for our analysis.



Features in our dataset were Date, Open, High, Low, Close, Volume.

DATA PREPROCESSING:

In preparation for analysis, a crucial step involved meticulous data preprocessing. This encompassed addressing missing values to ensure the integrity of our dataset. Null values were systematically converted to zero, a strategy chosen for its simplicity and minimal impact on the overall data structure. By undertaking this preprocessing step, we aimed to create a more robust dataset that could be effectively utilized for subsequent analysis and machine learning endeavours. The focus on data integrity is paramount, and these preprocessing steps lay the foundation for more accurate insights and predictions in the later stages of the project.

Data Loading into MySQL:

Before integrating with Spark, the Nifty FMCG data underwent a crucial phase of loading into a MySQL table. This process involved several key steps to ensure seamless storage and retrieval:

1. Database Schema Design:

- A well-defined database schema was crafted to accommodate the specific attributes and structure of the Nifty FMCG dataset. This schema served as the blueprint for organizing and storing data in the MySQL database.

2. Data Transformation:

- Data transformation steps were implemented to align the raw data with the predefined database schema. This included handling data types, ensuring consistency, and preparing the dataset for efficient storage in a relational database.

3. Loading Data into MySQL:

- Using MySQL's data import tools, the prepared dataset was loaded into the designated table. This step involved mapping the transformed data to the corresponding fields in the MySQL table, facilitating a seamless transfer of information.

Data Integration with Spark:

Spark was instrumental in handling the voluminous Nifty FMCG dataset, providing a scalable and efficient framework for distributed processing. Leveraging Spark's capabilities, the data integration process involved parallel computation across multiple nodes, significantly reducing processing time.

Connecting MySQL with Spark through JDBC facilitated seamless data integration and analysis. The integration process involved several key steps:

1. Configuration Setup:

- JDBC drivers for MySQL were configured within the Spark environment. This entailed specifying the driver class, connection URL, and authentication credentials to establish a secure link between Spark and the MySQL database.

2. Establishing Connection:

- A connection was established using Spark's JDBC API. This connection served as the bridge between the Spark application and the MySQL database, allowing for the efficient exchange of data between the two environments.

3. Data Extraction:

- Spark SQL queries were employed to extract data from the MySQL database into Spark DataFrames. This step involved crafting SQL queries to retrieve specific subsets of data relevant to the analysis, leveraging the power of Spark's distributed computing capabilities.

4. Batch Processing:

- Spark's parallel processing capabilities were harnessed to distribute the data processing workload across multiple nodes. This parallelism significantly improved the efficiency of data retrieval and subsequent analysis, making optimal use of available computing resources.

This integration between Spark and MySQL through JDBC laid the groundwork for a cohesive and efficient data processing pipeline. It seamlessly bridged the gap between the distributed computing capabilities of Spark and the structured relational storage of MySQL, enabling a unified and scalable approach to data analysis.

Exploratory Data Analysis and Financial Metrics:

1. **Percentage Change in Closing Price:**
    - Formula: Percent Change=(Close−lag (Close)lag(Close))×100PercentChange=(lag(Close)Close−lag(Close))×100
    - Inference: This analysis quantifies the day-to-day percentage change in closing prices, providing insights into the volatility of the stock. Positive values indicate price increases, while negative values suggest declines.
2. **Market Sentiment Analysis:**
    - Formula: Sentiment= {"Positive “if Close > lag (Close)"Negative" otherwise Sentiment= {"Positive” “Negative"
      if Close > lag(Close)otherwise
    - Inference: The sentiment analysis categorizes each day's market sentiment as positive or negative based on the comparison of closing prices with the previous day. This aids in understanding trends and potential shifts in investor sentiment.
3. **Relative Strength Index (RSI):**
    - Formula: RSI=100−(1001+RS), where RS=Average Gain Average Loss RSI=100−(1+RS100), where RS=Average Loss Average Gain
    - Inference: RSI measures the magnitude of recent price changes, indicating overbought or oversold conditions. Values above 70 may suggest overbought, while values below 30 may indicate oversold conditions.
4. **Percentage Change in Sales:**
    - Formula: Sales Change=(Volume−lag (Volume)lag(Volume))×100SalesChange=(lag(Volume)Volume−lag (Volume))×100
    - Inference: This analysis gauges the percentage change in trading volume, providing insights into market activity. Higher values may indicate increased trading interest.

5. **Monthly Price Momentum:**

   - Formula: Price Momentum=Close−lag (Close), AvgPriceMomentum=average(PriceMomentum)PriceMomentum=Close−lag(Close),AvgPriceMomentum=average(PriceMomentum)

   - Inference: Monthly price momentum assesses the average change in closing prices from the previous month, offering a perspective on broader market trends.

6. **Rolling Standard Deviation of Closing Price:**

   - Formula: Rolling Std Dev=stddev(Close)Rolling Std Dev=stddev(Close)

   - Inference: Rolling standard deviation measures the volatility in closing prices over a specified rolling window, providing a dynamic view of price fluctuations.

7. **Price and Volume Correlation:**

   - Formula: Correlation=corr(Close, Volume)Correlation=corr(Close, Volume)

   - Inference: Correlation between closing prices and trading volume helps identify potential relationships between price movements and market activity.

8. **Cumulative Returns:**

   - Formula: CumulativeReturn=sum(Return)CumulativeReturn=sum(Return)

   - Inference: Cumulative returns showcase the aggregated performance of the stock over time, aiding in the assessment of overall profitability.

9. **Price Gap Analysis:**

   - Formula: PriceGap=Open−ClosePriceGap=Open−Close

- Inference: Price gap analysis identifies the difference between opening and closing prices, highlighting days with significant gaps and their frequency within each month.

10. **Moving Average Analysis:**

    - Formula: 5DayMovingAvg=avg(Close)5DayMovingAvg=avg(Close)

    - Inference: Moving averages smooth out short-term fluctuations, providing a clearer trend view. The 5-day moving average represents the average closing price over the last five days.

11. **High-Low Range Analysis:**

    - Formula: HighLowRange=High−LowHighLowRange=High−Low

    - Inference: High-Low range measures the daily price range, offering insights into the stock's volatility and potential intraday trading opportunities.

12. **Trading Volume Trends:**

    - Formula: VolumeTrend=Volume−lag(Volume)VolumeTrend=Volume−lag(Volume)

    - Inference: Volume trends highlight changes in trading activity, indicating periods of increased or decreased market participation.

These analyses collectively provide a comprehensive understanding of various aspects of Nifty FMCG's historical performance, aiding in strategic decision-making and trend identification.

Machine Learning:

In the pursuit of predicting Nifty FMCG prices, we employed the Linear Regression algorithm within the Spark environment. This regression technique aims to establish a linear relationship between independent features and the target variable, facilitating price predictions based on historical data.

Feature Selection:

We identified essential features crucial for predicting closing prices. The chosen features, including "Open," "High," "Low," and "Volume," were assembled into a feature vector using Spark's Vector Assembler. This transformation facilitated the integration of these features into a format suitable for the machine learning model.

Data Splitting:

To assess the model's generalization performance, we divided the dataset into training (80%) and testing (20%) sets. This partitioning allowed us to train the model on one subset and evaluate its predictive capabilities on unseen data, ensuring an unbiased assessment of its efficacy.

Linear Regression Model:

A Linear Regression model was selected for its simplicity and interpretability. The model, instantiated with Spark's Linear Regression class, was configured with the label column ("Close") and feature column ("features"). This established a foundation for capturing linear relationships between the chosen features and the target variable.

Model Training and Prediction:

The Linear Regression model was fitted to the training data using the fit method, learning the underlying patterns in the historical dataset. Subsequently, predictions were generated on the test data, offering insights into the model's ability to generalize to unseen market conditions.

Graphical User Interface (GUI) for Interactive Data Visualization:

Our Graphical User Interface (GUI) enhances the analytical experience by providing interactive and dynamic visualizations of the Nifty FMCG analysis. The GUI is built using Python, HTML, CSS, JavaScript leveraging popular visualization libraries such as Matplotlib and Plotly to create immersive and responsive plots.

Key Features:

1. **Time Series Visualization:**
    - Interactive time series plots offer a comprehensive view of the historical trends in Nifty FMCG closing prices. Users can zoom in, pan across time intervals, and hover over data points to access specific information.
2. **Sentiment Analysis:**
    - The GUI incorporates a sentiment analysis plot, depicting the positive and negative market sentiments over time. Users can explore sentiment trends, correlate them with price movements, and gain a nuanced perspective on market dynamics.
3. **Technical Indicators Overview:**
    - Multiple technical indicators, such as Relative Strength Index (RSI), Moving Averages, and Standard Deviation, are presented in interactive plots. Users can customize the view, toggle between indicators, and explore their impact on price movements.
4. **Volume Trends:**
    - Interactive volume trend charts offer insights into trading activity. Users can delve into patterns of increased or decreased market participation, potentially uncovering significant market events.

User Interaction:

- **Zoom and Pan:**
    - Users can dynamically zoom in on specific time periods and pan across the visualizations for a closer inspection of data points.
- **Tooltips:**
    - Interactive tooltips provide detailed information when hovering over data points, allowing users to extract specific values and insights.
- **Toggle Options:**
    - Toggle switches empower users to customize the displayed information. For instance, users can choose specific technical indicators or sentiment analysis metrics to focus on.

Enhanced User Experience:

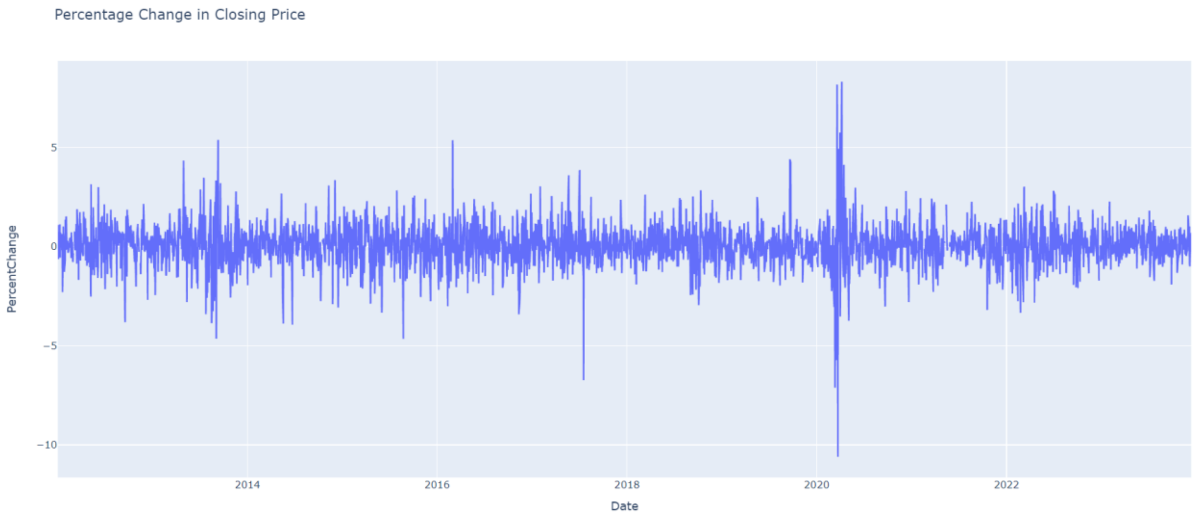
The GUI aims to enhance user experience by providing a user-friendly interface that caters to both novice and experienced users. By offering interactive features, our GUI transforms data exploration into an engaging and insightful process, fostering a deeper understanding of the intricacies within the Nifty FMCG dataset.

RESULTS:

Data Visualizations and Analytical Insights:

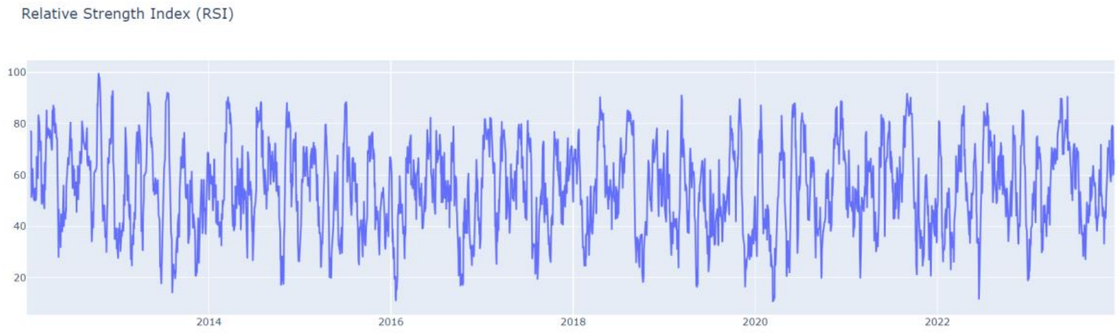
Percentage Change in Closing Price

We see the percentage change in the market, and we notice there’s significant percentage drop in 2020 (Covid) followed by a positive rally in 2021.

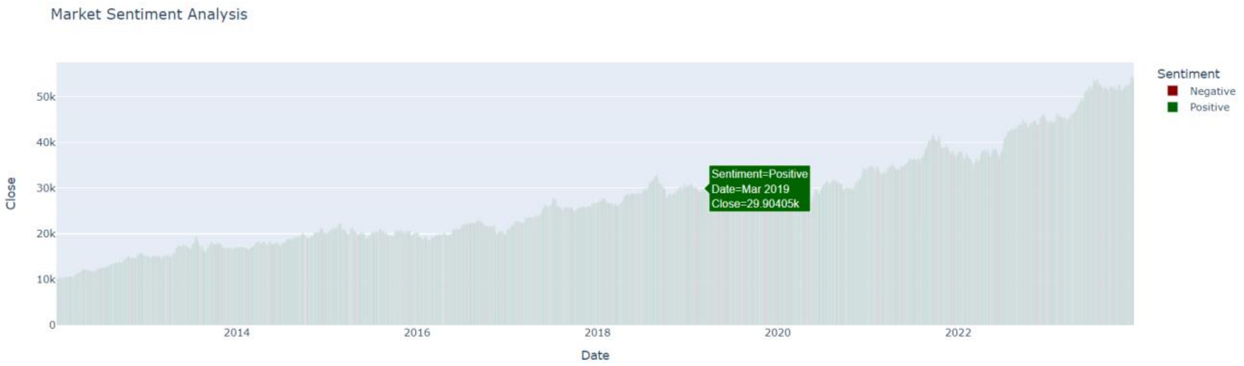


Relative Strength Index (RSI):

Values above 70 indicate possible uptrend and below 30 indicate downtrend. Traders infer and trade based on these values.

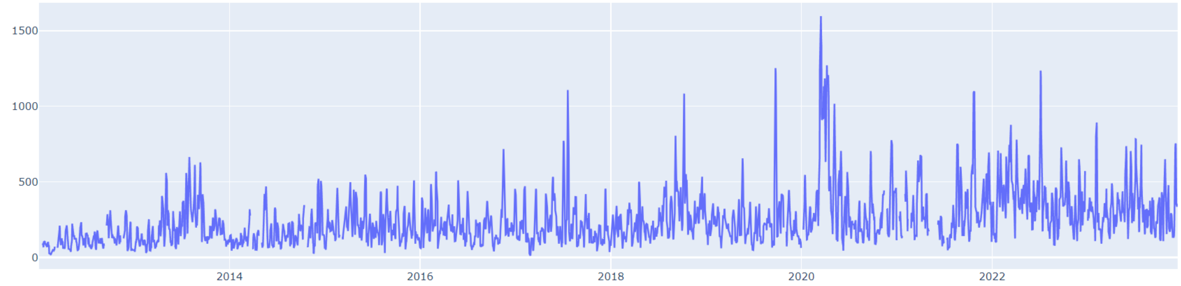


Market Sentiment Analysis:



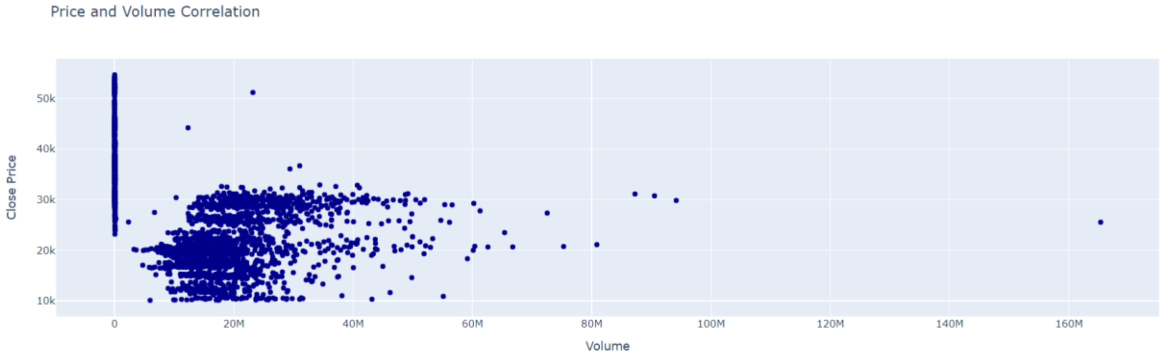
Rolling Standard Deviation of Closing prices:

We notice there’s more volatility during the pandemic period and in 2022 due to fear of Recession.



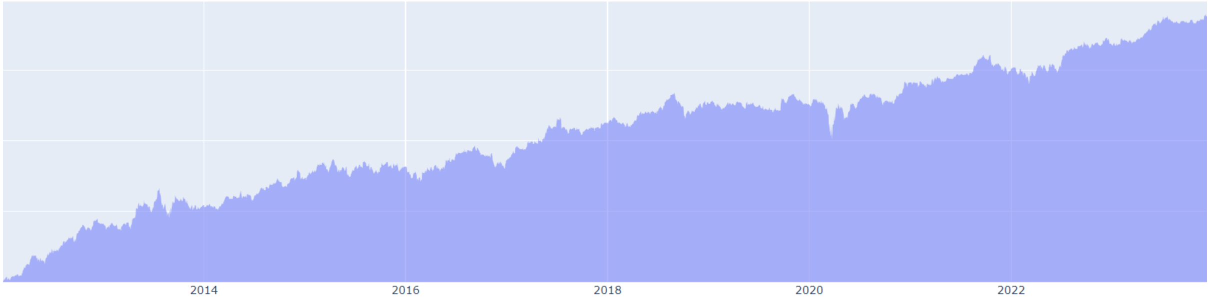
Price and Volume Correlation:

\We notice that the daily price and volume are closely related to each other.



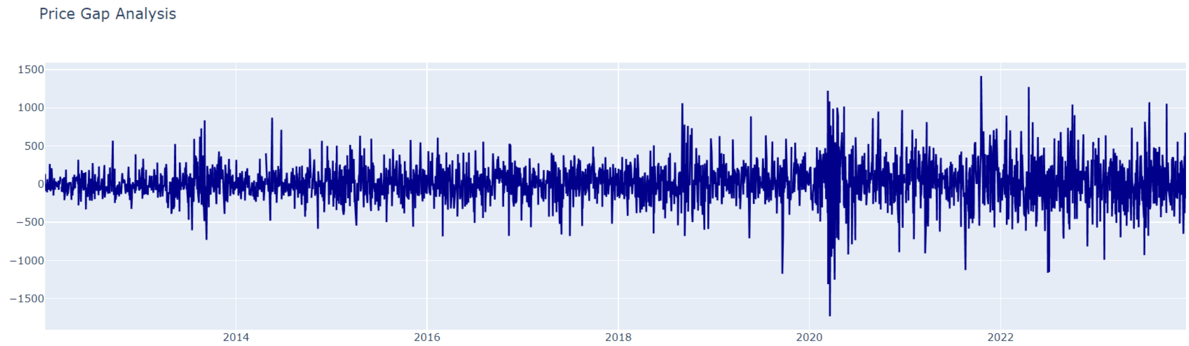
Cumulative returns:

We see that Nifty FMCG has given 569% returns in total since 2012.

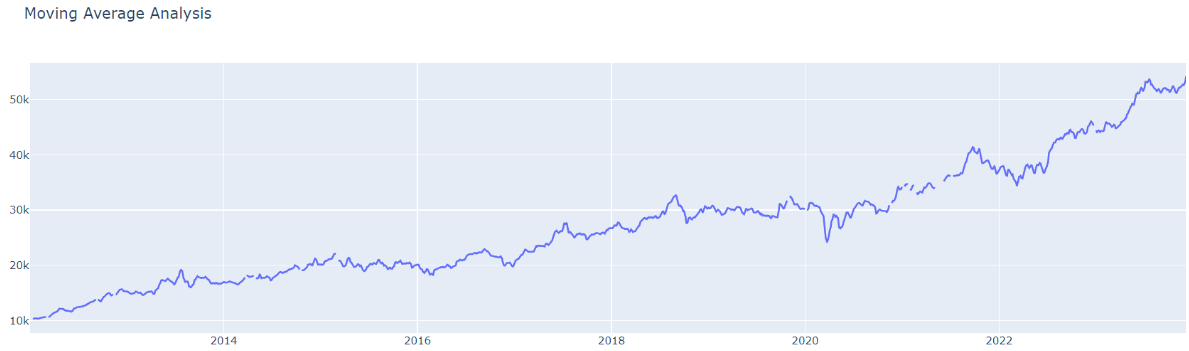


Price Gap analysis:

The more the price gap, we infer that the volatility is higher than usual.

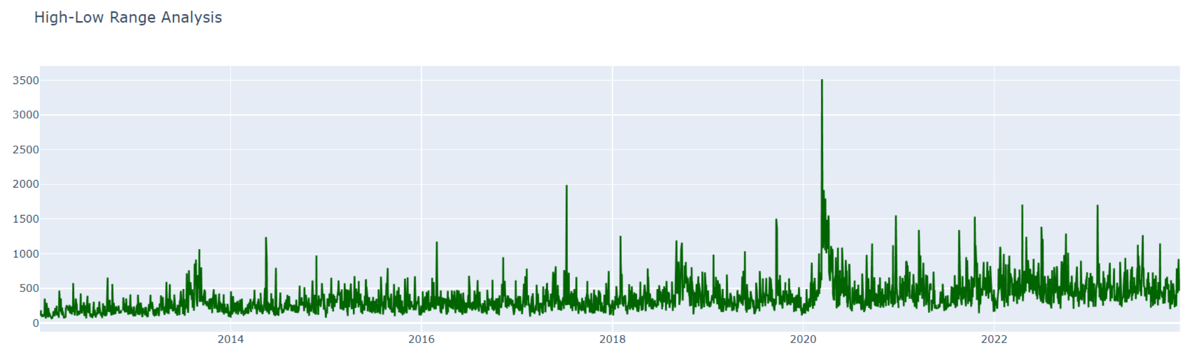


Moving average analysis:



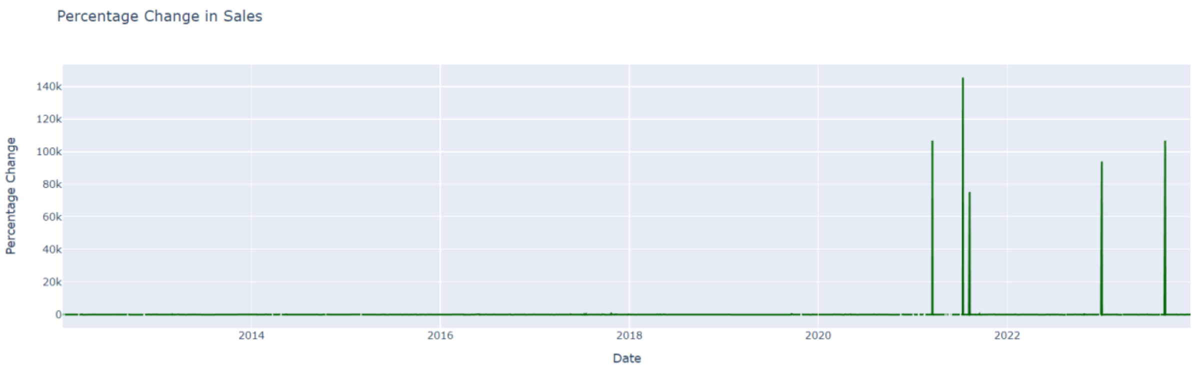
High – Low range analysis:

The gap up openings were more during the pandemic and post pandemic period.



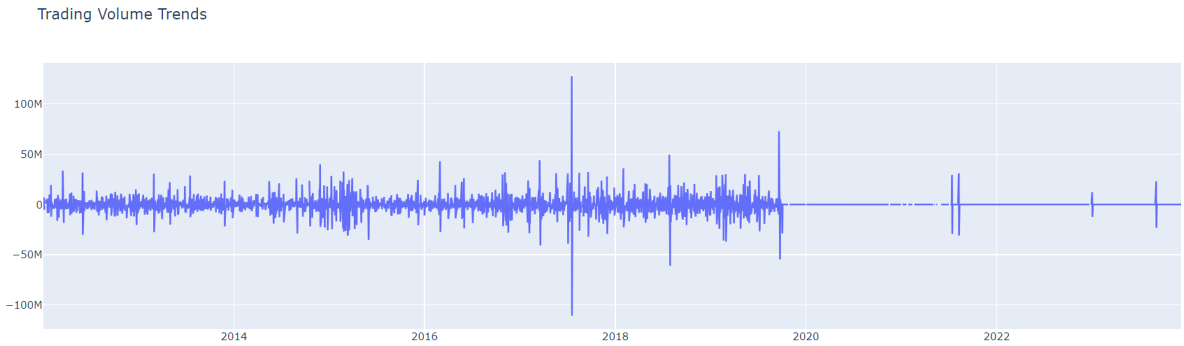
Percentage Change in Sales:

The consumer goods sales peaked after the pandemic period when things were getting back to normal.



Trading volume Trends:

This indicates the volume of amount that is been pumped into the market. We notice that a pump and dump has taken place in this sector during 2017.



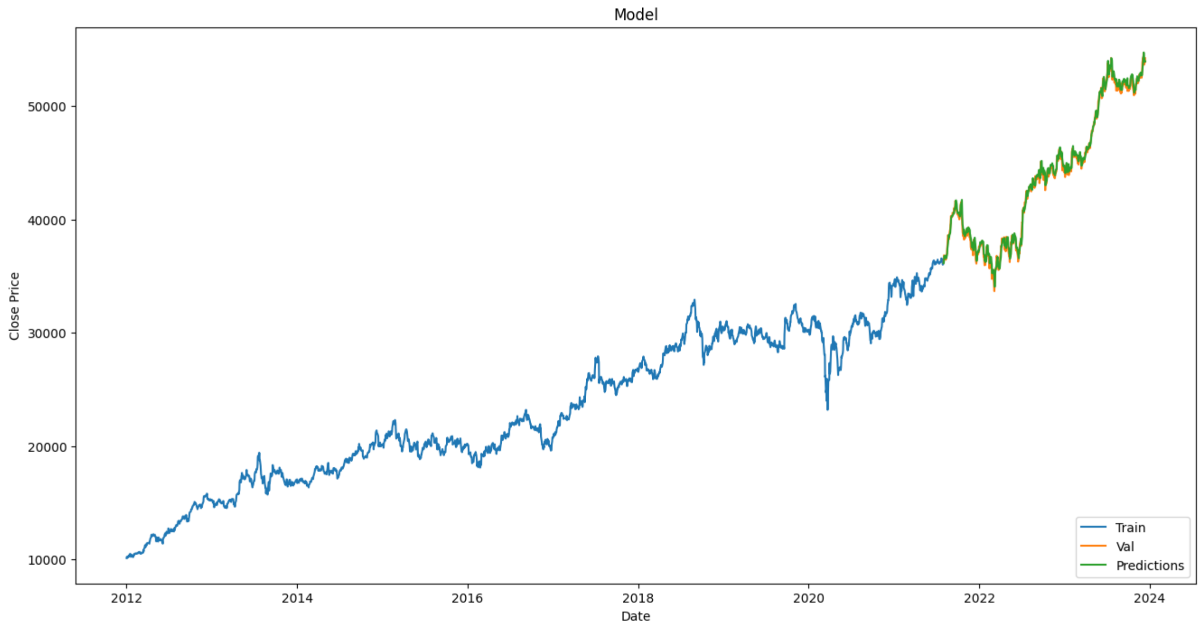
Machine Learning Predictions:

Model Evaluation:

To quantify the model's performance, we employed the Root Mean Squared Error (RMSE) metric. The Regression Evaluator in Spark allowed us to assess the disparity between predicted and actual closing prices.

Results and Inferences:

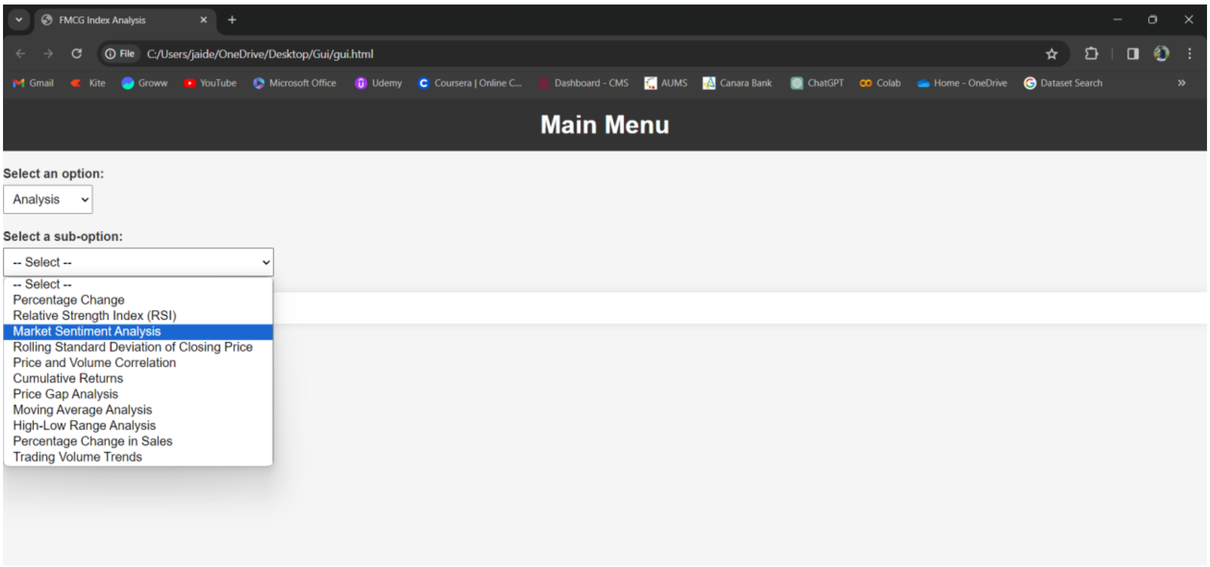
The trained Linear Regression model demonstrated promising predictive capabilities, as evidenced by the RMSE evaluation on the test data. We got our RMSE value around 100 which is quite high but that's because of the type of data and the fact that stock's price is influenced by other factors which are not in anyone's control.



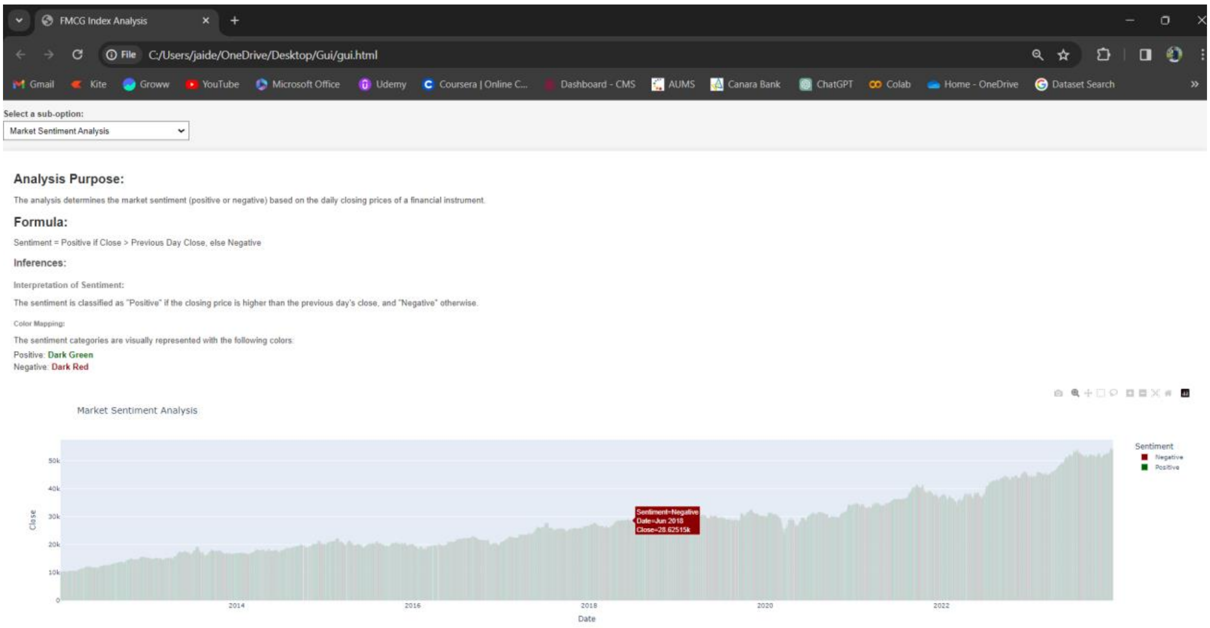
We see that the predicted prices closely align with the original stock’s price.

The Graphical user interface has been designed with a drop-down feature to view all the analysis and results.

On choosing the analysis we wish to see,



It displays the idea behind the analysis and a guide on how to infer from the visualization. The plots are interactive, and we can choose any specific day and view the results on that specific day.



Select a sub-option:
Market Sentiment Analysis

Analysis Purpose:

The analysis determines the market sentiment (positive or negative) based on the daily closing prices of a financial instrument.

Formula:

Sentiment = Positive if Close > Previous Day Close, else Negative

Inferences:

Interpretation of Sentiment:

The sentiment is classified as "Positive" if the closing price is higher than the previous day's close, and "Negative" otherwise.

Color Mapping:

The sentiment categories are visually represented with the following colors:
Positive: Dark Green
Negative: Dark Red

CONCLUSION:

In conclusion, our comprehensive analysis of Nifty FMCG using a synergistic blend of Spark, MySQL, Python, and machine learning techniques has yielded valuable insights into the dynamics of the FMCG sector. Through meticulous data collection, integration, and exploratory analysis, we've uncovered trends, patterns, and market sentiments that provide a holistic understanding of Nifty FMCG's historical performance.

The integration of machine learning, particularly the Linear Regression model, has enabled us to make informed predictions about closing prices. The evaluation metrics, such as Root Mean Squared Error (RMSE), attest to the model's ability to capture underlying patterns in the data. This predictive capability serves as a valuable tool for investors seeking data-driven insights for strategic decision-making.

The Graphical User Interface (GUI) enhances the accessibility of our analyses, providing an interactive platform for users to explore time series trends, sentiment analysis, and technical indicators. The dynamic visualizations not only facilitate a deeper understanding of the data but also empower users to tailor their exploration based on specific areas of interest.

While our analysis has provided valuable insights, it is crucial to acknowledge certain challenges. Overfitting and underfitting were addressed through careful model tuning, emphasizing the iterative nature of the machine learning process. Additionally, the quality of predictions is intricately linked to the relevance of selected features, necessitating ongoing refinement and adaptation to evolving market conditions.

In essence, this report serves as a stepping stone in the exploration of data-driven insights within the financial domain. By leveraging cutting-edge technologies and methodologies, we've uncovered valuable patterns and trends, providing a foundation for strategic decision-making in the complex landscape of the FMCG market.

FUTURE WORKS:

1. **Advanced Models and Feature Engineering:**

   - Explore advanced machine learning models like ensemble methods and deep learning for improved predictive accuracy. Emphasize innovative feature engineering techniques to incorporate additional relevant market indicators.

2. **Real-Time Integration and GUI Enhancements:**

   - Implement real-time data integration for continuous model updates and responsiveness to market changes. Dynamically enhance the Graphical User Interface (GUI) with more interactive features and customization options to cater to diverse user preferences.

3. **Comprehensive Analysis and Model Optimization:**

   - Integrate natural language processing into sentiment analysis and incorporate external data sources for a more holistic market understanding. Optimize processes for scalability, focus on model interpretability, gather user feedback for GUI improvements, and establish continuous monitoring mechanisms for timely updates, ensuring an adaptive and user-centric analysis framework.

Reference:

https://kontext.tech/article/610/spark-scala-load-data-from-mysql

https://spark.apache.org/docs/latest/

https://dev.mysql.com/doc/

https://www.hitbullseye.com/Stocks-and-Shares.php

https://finance.yahoo.com/