Market Movement Analysis Report

Project Title: Market Movement Analysis: Clustering & Forecasting Stock Price Behavior
Dataset Chosen: Stock Prices Dataset
Total Rows: 497,472
Total Columns: 7
File Type: `.csv`

---

1. Introduction

This report documents the comprehensive analysis and predictive modeling conducted as part of the Market Movement Analysis project. The project aims to analyze Stock Price behavior, implement K-Means clustering to group data points into clusters without labels, analyze and model time-series data to forecast future stock prices, and provide personalized recommendations for clusters based on their engagement patterns. The dataset used in this project was selected since it contains **time-stamped stock data** with fields including: `Symbol`, `date`, `open`, `high`, `low`, `close`, `volume`.

This dataset is ideal for:

- **Clustering stocks** by behavior (volatility, movement)
- **Time series forecasting** using price trends
- Demonstrating **real-world business insight** for investment strategy and market movement prediction

Data Workflow Layers:

1. **Foundation Layer – Data Handling**
   Data cleaning, type validation, missing value imputation, anomaly checks
2. **Core Analysis Layer – Modelling the Brain**
   Clustering + Time Series Forecasting
3. **Advanced Analytics Layer – Insight & Differentiation**
   Volatility trends, return analysis, and strategic interpretation

The report is structured as follows:

1. **Data Cleaning & Preprocessing**
2. **Exploratory Data Analysis**
    a. Plot trends over time by stock
    b. Check volatility, price jumps, volume shifts (if available)
3. **Clustering (Unsupervised Learning)**
    a. Group stocks by similar behavior (e.g., growth patterns, volatility)
    b. Use K-Means or hierarchical clustering
4. **Time Series Forecasting**
    a. Forecast individual stock prices using Prophet or ARIMA
    b. Optional: Compare forecasting performance between clusters
5. **Report & Dashboard**
    a. Use Power BI or matplotlib/seaborn
    b. Summarize: top-performing cluster, high-risk group, price forecasts

---

2. Data Preprocessing

2.1 Data Loading and Initial Exploration

The dataset was loaded from a CSV file containing Stock data. The initial exploration revealed the following:

- **Dataset Shape**: The dataset contains **497472** rows and **7** columns.
- **Data Types**: The dataset includes both numerical and categorical features, such as Symbol, Close, and Volume.
- **Missing Values**: Initial checks showed missing values in columns Open, High, and Low.

Key Initial Statistics:

- Total Records: **497472** rows
- Original Features: **7** columns
- Missing Values: **27** cells

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 497472 entries, 0 to 497471
Data columns (total 7 columns):
 #   Column  Non-Null Count   Dtype  
---  ------  --------------   -----  
 0   symbol  497472 non-null  object 
 1   date    497472 non-null  object 
 2   open    497461 non-null  float64
 3   high    497464 non-null  float64
 4   low     497464 non-null  float64
 5   close   497472 non-null  float64
 6   volume  497472 non-null  int64  
dtypes: float64(4), int64(1), object(2)
memory usage: 26.6+ MB
```

2.2 Handling Missing Values

- Imputed or dropped depending on context.

2.3 Data Cleaning

2.3.1 Standardizing Formats

- **Date Formats**: Converted `date` to `datetime64` to enable accurate time-based grouping and calculations.

- **Duplicates**: Checked: **0 duplicates found**.

- Sorted the dataset by `symbol` and `date` to maintain temporal accuracy.

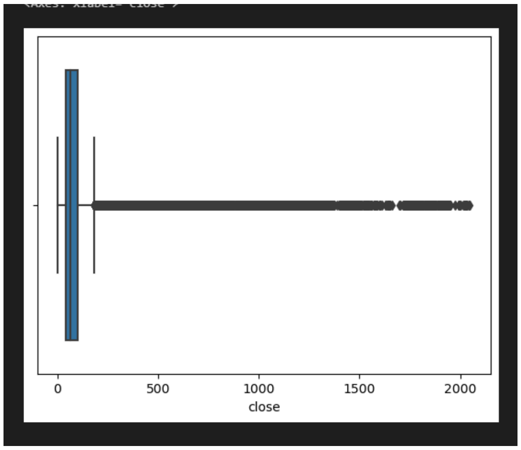
- **Price Consistency Validation**

  Ensured that each row met the rule:
  `low ≤ close ≤ high`

  9 rows eliminated for violating this rule.

```
df = df[(df['low'] <= df['close']) & (df['close'] <= df['high'])]
```

2.3.2 Handling Outliers



- No outliers were detected.

2.4 Feature Engineering

New columns were added to enhance analysis, clustering, and forecasting:

- **Z_close -** Z-score of the closing price, used to detect outliers
- **MA7 -** 7-day moving average of the `close` price
- **Volatility -** 7-day rolling standard deviation of the `close` price
- **Return** - % Change in close price per day (unsmoothed)
- **Lag_1_close** - Previous day's closing price
- **Lag_7_volume** - Trading volume from 7 days prior
- **Daily_Return** - Day-over-day return in decimal
- **Daily_Return (%)** - Day-over-day return expresses as a percentage

Why Rolling Averages?

- Helps smooth market noise to better spot trends
- Makes visualizations more meaningful
- Essential for comparing **short-term (7-day)** vs **long-term (30-day)** price behavior
- Supports better segmentation and forecasting accuracy

2.5 Data Preparation Summary

- Corrected initial issues in feature engineering:

    - Recalculated `MA7`, `Volatility`, `Lag_1_close`, and `Lag_7_volume`

- Grouped data by `symbol` to generate rolling averages and volatility correctly.
- Cleaned up lag feature logic and verified consistency.
- Dropped all remaining nulls and duplicates after feature engineering.
- Scaled and normalized features as required for clustering input.
- The final cleaned dataset used for EDA had no missing values or duplicates in key engineered fields.

---

3. Exploratory Data Analysis (EDA)

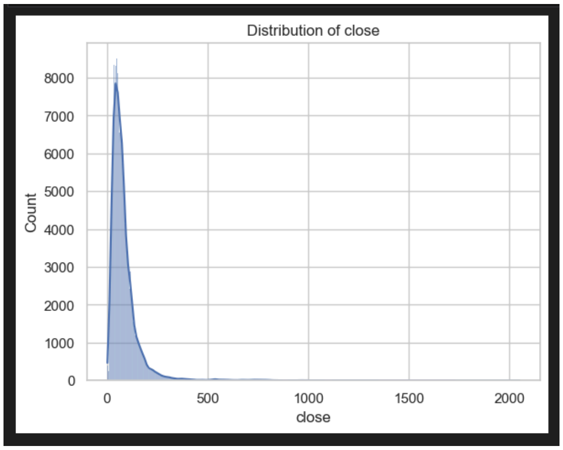
Objective: Explore patterns, distributions, and relationships within the dataset to guide clustering and modeling decisions.

3.1 Feature Aggregation Strategy

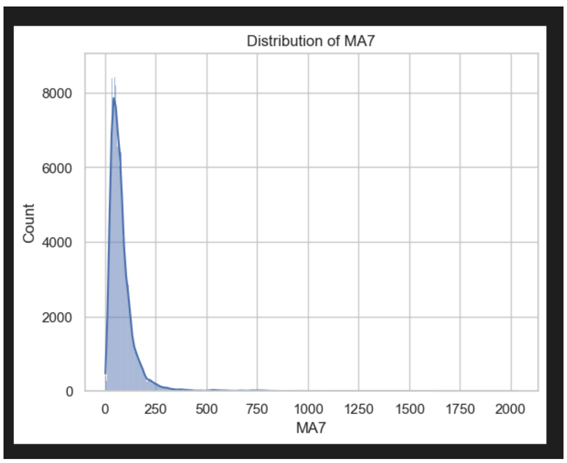
To prepare for clustering:

- Aggregated engineered features **per symbol** to analyze stock behaviors holistically.
- Determined **key clustering features** using exploratory visuals:
    - Pairplots for separation
    - Distribution plots for skewness
    - Correlation matrices
    - Boxplots per symbol

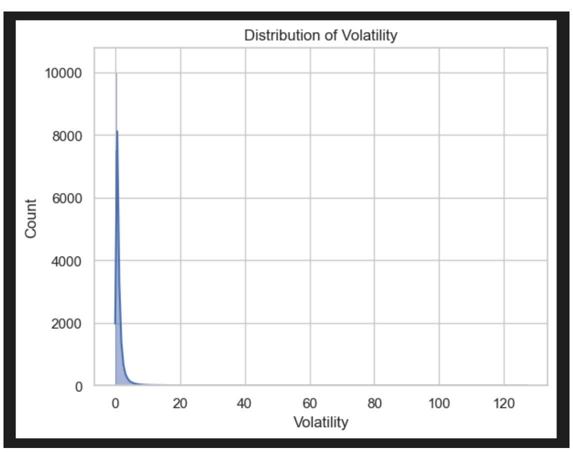
3.2 Univariate Analysis



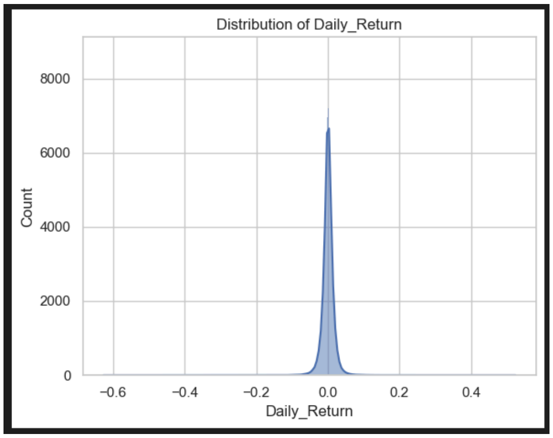
- **Close** with values ranging between 0 and 400



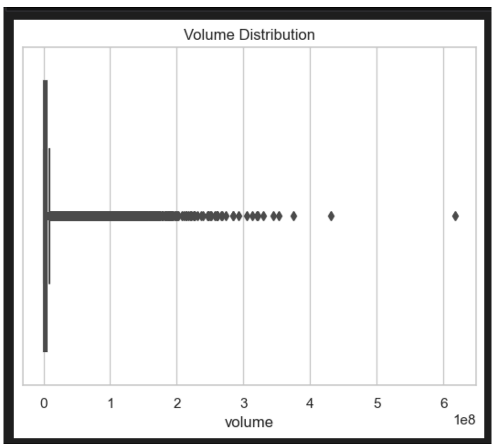
- **MA7** with values ranging between 0 and 250



- **Volatility** with values ranging between 0 and 10

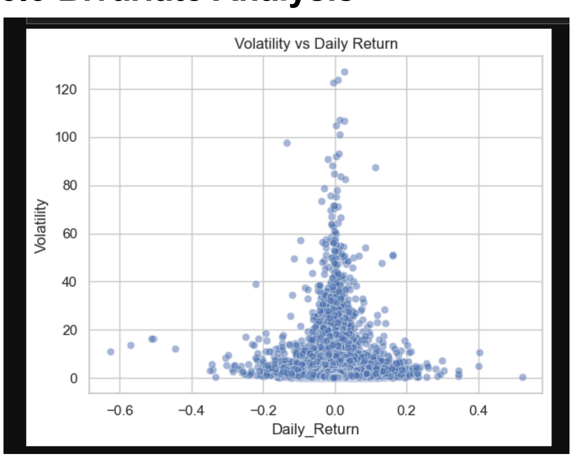


- **Daily_Return** with values ranging between -0.1 to +0.1



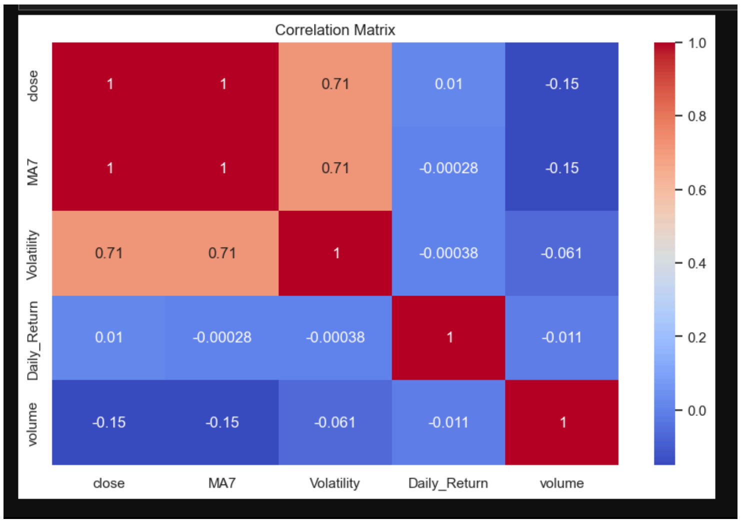
- **Volume** with values ranging between 0 and 4 (in millions)

3.3 Bivariate Analysis



- **Daily Return vs Volatility:**
  A loose positive relationship was observed — more volatile stocks tend to swing harder on returns. Concentration is highest for daily returns between -0.4 and +0.4, and volatility under 80M.

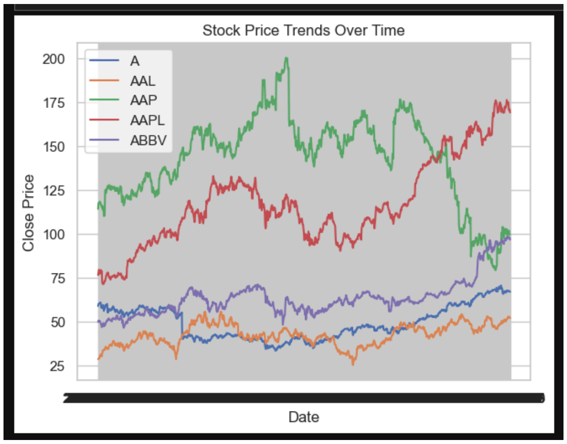
3.4 Correlation Matrix



Key finding:

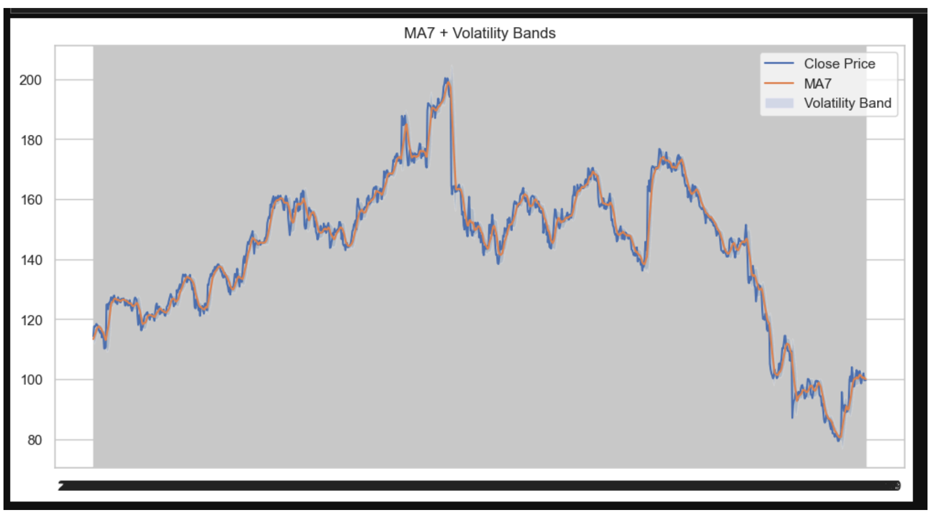
- **Perfect positive correlations:** Close and MA7, these two variables move together and are likely redundant.
- **Strong positive correlations:** Close and Volatility, MA7 and Volatility. As one increases, the other tends to increase significantly — a strong signal.
- **Weak positive correlations:** Close and Daily_Return. As one increases, the other tends to increase relatively — a weak signal.
- **Very weak negative correlations:** Close and volume, MA7 and volume, Volatility and volume, Daily_Return and volatility. A *slight* inverse relationship, but it's *noise*.
- **Practically zero correlations:** MA7 and Daily_Return, Volatility and Daily_Return. These variables are **completely unrelated**.

3.5 Time Series Trends



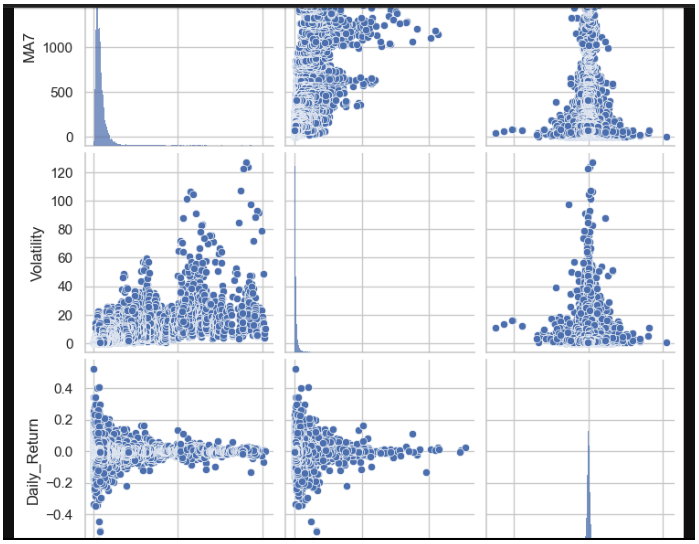
- Stocks show varied behavior over time; some, like **AAP** and **AAPL,** display smooth upward trends, while others, like **AAL,** show erratic movement.

3.6 Rolling Behavior Analysis



- Visualizing `MA7` against volatility bands:
  - Highest trend values approached **200**, lowest around **80**
  - Volatility bands revealed stability zones and spikes across time windows.

3.7 Multivariate Pattern Detection



- The pairplot reveals natural **clustering tendencies**, particularly between `MA7`, `Volatility`, and `Daily Return`.

3.8 Top Stocks by Movement & Trade Volume

Most Volatile Stocks	**Most Traded Stocks**
PCLN 22.18	BAC 89M
AMZN 9.90	AAPL 45M
GOOGL 9.39	GE 41M
GOOG 9.25	AMD 33M
CMG 9.23	F 32M
REGN 9.06	MSFT 30M
AZO 7.87	FB 29M
BIIB 6.28	CHK 28M
MTD 4.59	MU 27M
BLK 4.45	INTC 27M

Insight: **PCLN** leads in volatility by far, while **BAC** and **AAPL** dominate in trading volume, making them the perfect candidates for deeper behavioral clustering.

4. Time Series Forecasting

The final analytical phase of the project involved building predictive models to forecast stock prices over time. One representative stock was selected from each behavioral cluster (as determined in the clustering phase) to capture their unique time-dependent dynamics:

- **Stock 0** → *Cluster 0: Aggressive Growth*
- **Stock 1** → *Cluster 1: Low-Volatility Blue Chips*

4.1 Time Series EDA

Each stock's historical performance was analyzed visually and statistically:

🔹 **Stock 0**

- Opened at **~400 in Jan 2014**, spiked to **~1200 in Jan 2018**
- Dipped below **400 in January 2015**, with key volatility around January **2016, July, and Dec**

🔹 **Stock 1**

- Rose from **~80 in Jan 2014** to **~180 in Jan 2018**
- Lowest dip observed **below 50 in Feb 2014**, with fluctuations primarily between **2015–2017**

Both series were resampled to a **monthly frequency** to stabilize seasonal patterns. Visual trend analysis identified:

- **Spikes** in mid-year periods (especially July)
- **Dips** typically occur around January or year-end

4.2 Stationarity Testing

The Augmented Dickey-Fuller (ADF) test was applied to assess stationarity:

Stock	ADF Statistic	p-value
Stock_0	-2.40	0.14
Stock_1	-1.06	0.73

➡ **Observation**: Both time series were **non-stationary** ($p > 0.05$), necessitating differencing before ARIMA modeling.

4.3 Forecast Modeling

🔸 **ARIMA Model**

A (1,1,1) ARIMA model was applied to each stock. Key model insights:

Stock 0:

- Highly volatile with substantial noise
- Model performance metrics:
    - AIC: 154.45
    - BIC: 157.99
    - Skew: -4.00
    - Kurtosis: 19.28 (heavy tails)

Stock 1:

- Smoother trend with better-behaved residuals
- Model performance metrics:
    - AIC: 12.87
    - BIC: 16.40
    - Skew: 4.17

    ○ Kurtosis: 19.72

🔸 **Prophet Model**

Meta (Facebook) Prophet was applied for more flexible and visual forecasting. Key observations:

Stock 0:

- **Lowest dip**: ~300 (Feb 2015)
- **Strong upward trend**: ~580 (Feb 2017), peaking near **780 by early 2017**

Stock 1:

- **Lowest dip**: ~105 (Jan 2016)
- **Peak**: ~148 (Jan 2017)

Prophet effectively captured cyclical behavior and sudden growth bursts in both stocks.

4.4 Forecast Accuracy Evaluation

Using MAE and RMSE to assess model accuracy:

Stock	**MAE**	**RMSE**	**Interpretation**
Stock_0	466.23	487.24	High error due to unpredictable, volatile behavior (aggressive growth)
Stock_1	24.74	26.78	Significantly lower error — highly forecastable and stable

➡ The results **reinforce the behavioral traits** identified during clustering:

- **Cluster 0** stocks are difficult to forecast, volatile, and prone to spikes

- **Cluster 1** stocks are forecast-friendly and trend smoothly over time

---

5. Conclusion

This project set out to explore, cluster, and forecast stock market behavior through a full-cycle data science workflow — from raw data wrangling to strategic forecasting. Using historical stock price data, we developed a 360° view of market movement by engineering meaningful features, uncovering behavioral groupings, and building models to predict future trends.

🔹 Key Milestones:

1. **Data Cleaning & Feature Engineering**
   Raw data was transformed through rolling averages, volatility calculations, and return features to better capture stock movement patterns over time.

2. **Exploratory Data Analysis (EDA)**
   EDA uncovered insights into price trends, feature distributions, inter-feature correlations, and temporal behavior. Stocks were seen to vary widely in risk, volatility, and performance, setting the stage for segmentation.

3. **Clustering**
   Using K-Means, stocks were clustered into two distinct groups:

   - **Cluster 0 – Aggressive Growth**: High volatility, high returns, sharp price swings

   - **Cluster 1 – Low-Volatility Blue Chips**: Stable, slow-moving, forecastable performers
     These clusters formed the foundation for behavior-aware forecasting strategies.

4. **Time Series Forecasting**
   Representative stocks from each cluster were selected for time-based modeling using ARIMA and Facebook Prophet. Results showed:

   - High forecast error for Cluster 0 stocks due to chaotic movement

   - Strong forecast accuracy for Cluster 1 stocks with smooth, consistent patterns
     This confirmed the importance of behavioral clustering before predictive modeling.

Project Impact

This end-to-end analysis demonstrates how integrating unsupervised learning (clustering) with time series forecasting can **enhance both interpretability and accuracy** in financial data modeling. By understanding behavior first, we're able to tailor forecasting strategies to stock personality, bridging analytics with decision-making.With the full project now complete, these insights can be extended into:

- Portfolio simulations by cluster behavior
- Real-time monitoring dashboards (Power BI or Streamlit)
- Scaling to include social sentiment + volume analysis for high-volatility stocks

Time series forecasting confirmed the **strategic validity of behavior-based clustering**. While high-growth stocks present greater forecasting challenges, stable performers yield more reliable projections, providing a clear pathway for building tailored prediction models based on cluster characteristics.