ALGO-TRADING MARKET ANALYTICS & PREDICTION SYSTEM

DECEMBER 26, 2025

FLIPKART.PVT.LTD

INTRODUCTION:

In this project, a market analytics and prediction system is developed using historical price data. The system applies a **moving average crossover trading strategy** to generate buy and sell signals.

The strategy is backtested on historical data to evaluate its performance. Key metrics such as **accuracy, profit/loss, drawdown, number of trades, and win ratio** are calculated. Visual tools like price charts and equity curves are used to analyze the strategy's effectiveness.

1. Data Collection & Preparation:

The historical market data used in this project was collected from **Yahoo Finance**, a widely used and reliable source for financial market information. Yahoo Finance provides free access to historical price data for stocks, cryptocurrencies, and other financial instruments.

The dataset includes key attributes such as **Open, High, Low, Close prices**, and **trading volume** recorded over a specified time period.

2. Data Cleaning:

Before performing any analysis, the dataset was examined for quality issues to ensure accuracy and consistency.

1.Checked for missing values:

- Used df.isnull().sum() to identify any missing entries in the dataset.
- Missing values can affect calculations of spending, frequency, and clustering, so they must be handled appropriately.

2. Checked for duplicate records:

- Used df.duplicated().sum() to detect repeated rows.
- Removed duplicates to avoid double-counting transactions.

3. Saved cleaned dataset:

- Missing values can affect calculations of spending, frequency, and clustering, so they must be handled appropriately.
- After cleaning, the processed dataset was saved as 'cleaned_stock_data.csv' for use in feature engineering and analysis.

Summary:

The dataset is now clean, consistent, and ready for feature engineering and exploratory data analysis.

Here is the cleaned Data Set :

"C:\Users\manasa\Downloads\cleaned_stock_data.xls"

3. Feature Engineering :

Feature engineering was done to convert the raw transactional data into meaningful "customer-level features". This helped in understanding the overall behavior of each customer and prepared the data for customer segmentation. The main goal was to combine all purchase records of each customer into one summary record.

Technical Indicators

1. **Exponential Moving Average (EMA)** – EMA_20 and EMA_50 were calculated to capture short-term and medium-term trends in the stock/crypto prices.
2. **Simple Moving Average (SMA)** – SMA_20 was used to smooth out short-term fluctuations and identify trend direction.
3. **Relative Strength Index (RSI)** – Measures the speed and change of price movements to identify overbought or oversold conditions.
4. **Moving Average Convergence Divergence (MACD)** – A trend-following momentum indicator, calculated as the difference between 12-day and 26-day EMAs.
5. **Average True Range (ATR)** – Measures market volatility by evaluating price ranges over time.
6. **Volatility** – Calculated as the rolling standard deviation of returns to capture market fluctuations.

Lag Features

- To provide the model with recent historical context, lag features were created for RSI, MACD, and Volatility over the last 1, 2, and 3 days:
    - RSI_lag1, RSI_lag2, RSI_lag3
    - MACD_lag1, MACD_lag2, MACD_lag3
    - Volatility_lag1, Volatility_lag2, Volatility_lag3

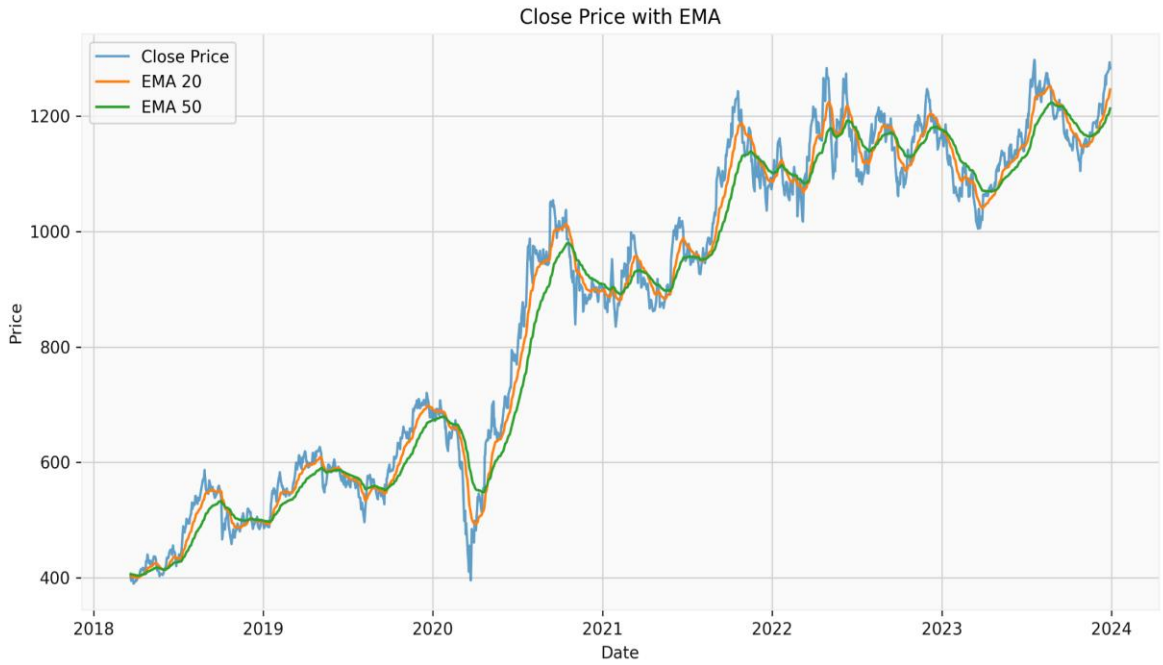
These features allow the model to learn temporal patterns and improve prediction of next-day price movement (up/down).

4. Exploratory Data Analysis (EDA):

Exploratory Data Analysis was done to understand the dataset better and to identify key patterns, trends, and relationships within the data.

1.Price Trend Analysis (Close Price with EMA)

- The line chart shows the daily **Close Price** along with **Exponential Moving Averages (EMA 20 and EMA 50)**.
- EMA lines smooth short-term fluctuations and help identify the overall trend.



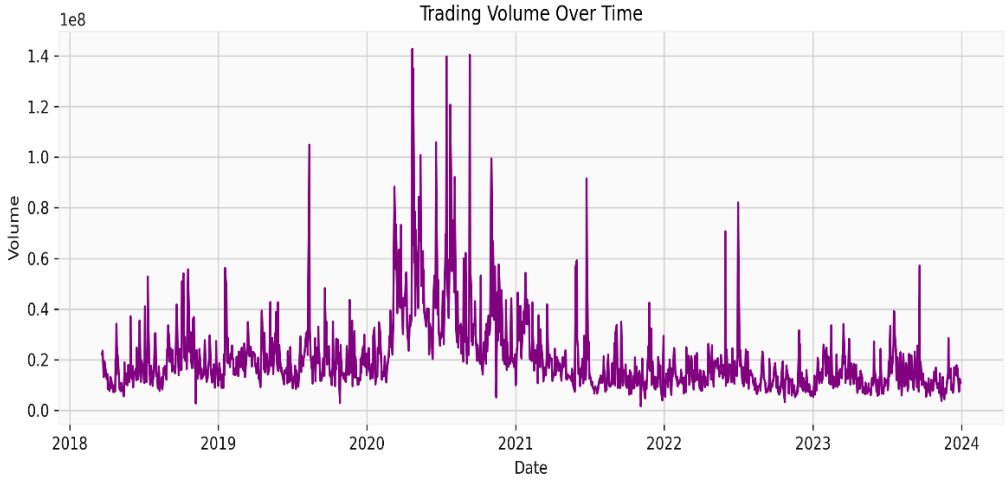
Insight:

Periods where EMA_20 crosses above EMA_50 indicate potential bullish trends, while EMA_20 below EMA_50 indicates bearish trends.

2.Volume Analysis (Trading Volume Over Time):

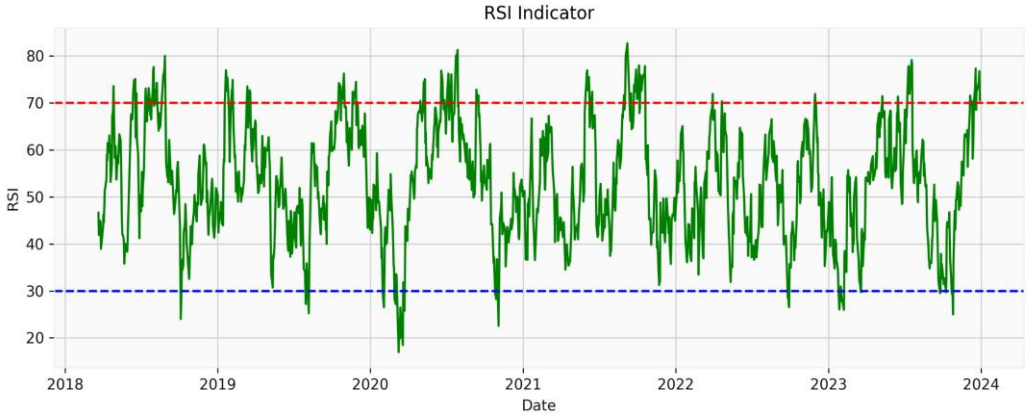
- The line chart shows the **trading volume** for each day.
- Volume indicates market activity; spikes often correspond to significant price movements.

Insight: High volume days may signal strong investor interest and potential market reversals.



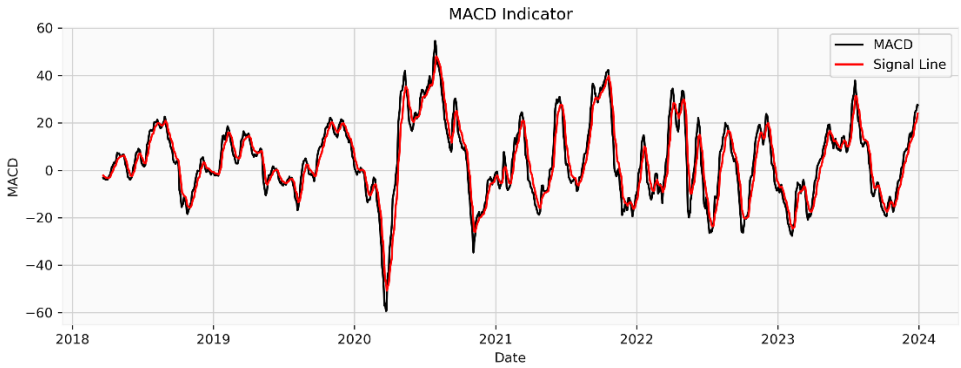
3.RSI Analysis (Relative Strength Index):

- The RSI plot measures momentum and shows overbought (RSI > 70) and oversold (RSI < 30) levels.
- **Insigh**t: Overbought regions may indicate potential price declines, while oversold regions may suggest price rebounds.



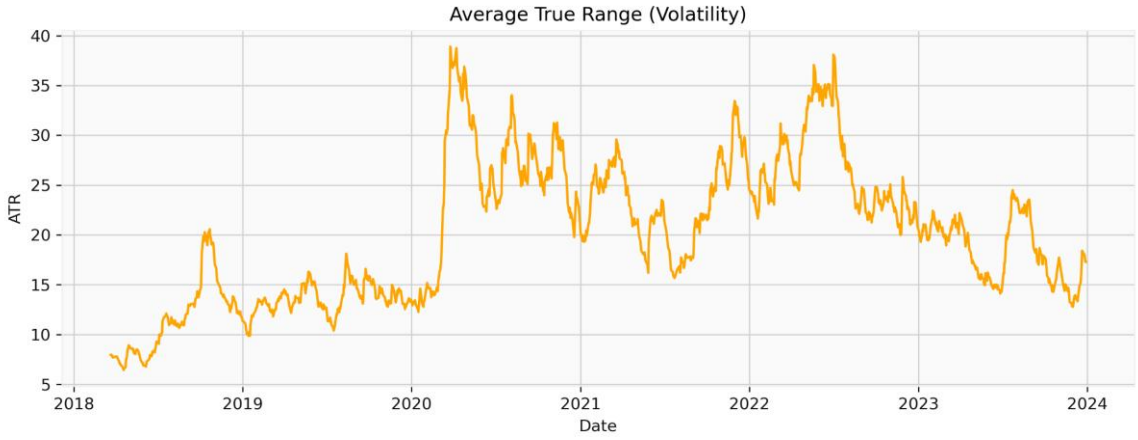
4.MACD Analysis (Moving Average Convergence Divergence):

- The chart shows **MACD line** and **Signal line**.
- **Insight:** When the MACD crosses above the Signal line, it suggests a bullish signal; crossing below indicates a bearish signal.



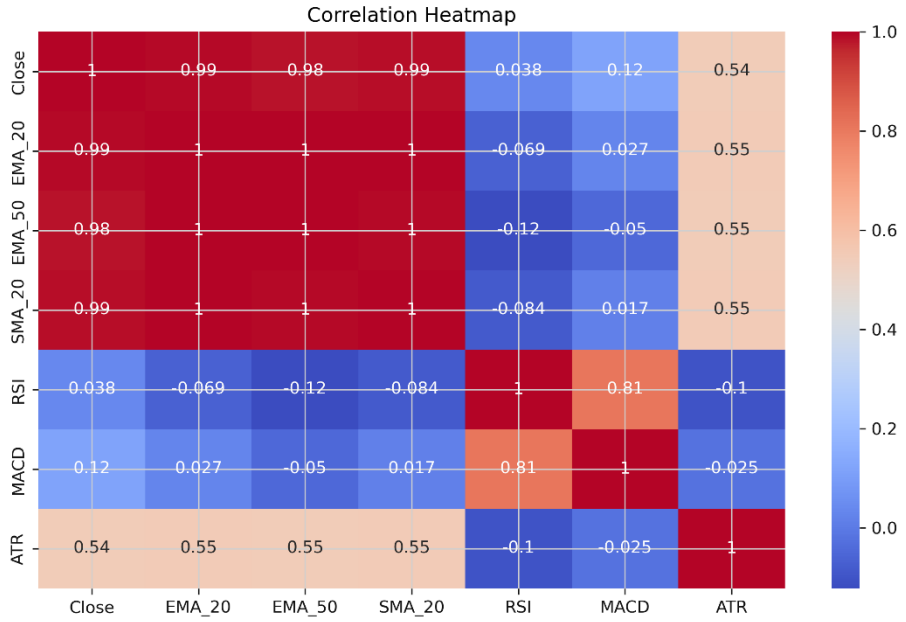
5. Volatility Analysis (Average True Range - ATR):

- The line chart shows the **ATR**, representing market volatility.
- **Insight:** Higher ATR values indicate periods of high price fluctuation, which can influence stop-loss placement and risk management.



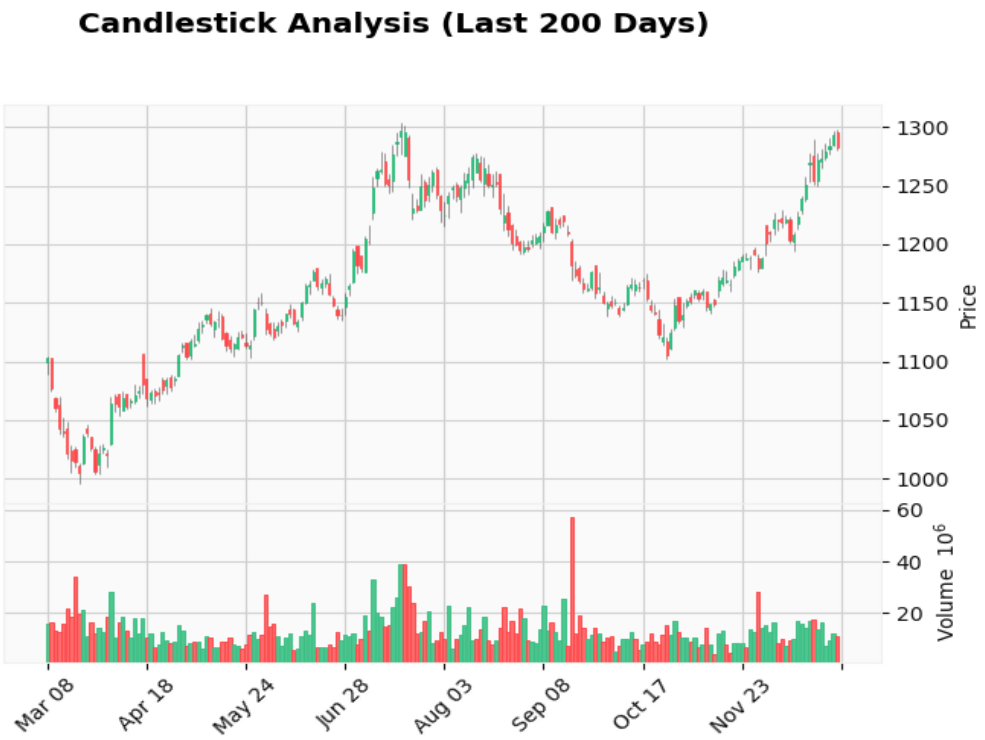
6.Correlation Heatmap:

- The heatmap displays correlations between technical indicators (Close, EMA, SMA, RSI, MACD, ATR).
- **Insight:** Strong correlations between moving averages and close price help identify trend-following patterns. Momentum indicators like RSI and MACD are moderately correlated, providing complementary signals.



7.Candlestick Chart (Last 200 Trading Days):

- This chart provides a visual overview of **price movements using candlesticks**, including open, high, low, and close prices along with volume.
- **Insight:** Candlestick patterns highlight market sentiment and short-term trends, aiding technical analysis.



Conclusion:

"Exploratory Data Analysis (EDA) was conducted to understand price trends, volume behavior, momentum, volatility, and relationships among technical indicators using multiple visualizations and statistical measures."

5. Predictive Modeling:

Objective:

The objective of predictive modeling is to **forecast the next-day price movement (Up/Down)** of the selected financial asset using machine learning techniques.
This helps in making **data-driven trading decisions** instead of relying on intuition.

In this project, the problem is formulated as a **binary classification task**:

- **1 (Up)** → Next day closing price is higher than today
- **0 (Down)** → Next day closing price is lower or equal to today

Target Variable Definition:

- The target variable is created using next-day returns:
- $\text{Target} = \begin{cases} 1, & \text{if } \text{Close}_{t+1} > \text{Close}_t \\ 0, & \text{otherwise} \end{cases}$
- This formulation aligns with **real trading logic**, where the goal is to predict whether the market will move upward or downward the next day.

Feature Selection:

The models use **technical indicators and lag-based features** extracted during feature engineering:

- Trend Indicators:
  - EMA (20, 50)
  - SMA (20)
- Momentum Indicators:
  - RSI
  - MACD
- Volatility Indicators:
  - ATR

- Lag Features:
    - RSI_lag1, RSI_lag2, RSI_lag3
    - MACD_lag1, MACD_lag2, MACD_lag3
    - Volatility_lag1, Volatility_lag2, Volatility_lag3

These features capture **trend, momentum, volatility, and historical behavior**, which are crucial for price prediction.

Train–Test Split Strategy:

- 80% Training Data
- 20% Testing Data
- Data is split **without shuffling** to preserve time-series order.

This avoids **data leakage** and simulates real-world trading conditions.

Models Used:

The following machine learning models were implemented and compared:

- Random Forest Classifier
- XGBoost Classifier
- Logistic Regression

Each model was evaluated using accuracy, precision, recall, and F1-score.

*****Random Forest Classifier*****

What is Random Forest?

Random Forest is an **ensemble learning algorithm** that builds multiple decision trees and combines their predictions to produce a more stable and accurate result. Instead of relying on a single decision tree, it uses a collection (forest) of trees.

Key Characteristics

- Uses **multiple decision trees**
- Reduces overfitting compared to a single tree
- Handles non-linear relationships well
- Works effectively with technical indicators
- Robust to noise in financial data

Outcome in This Project

- Achieved an accuracy of approximately **51%**
- Produced balanced predictions for both upward and downward price movements
- Provided stable performance but limited improvement due to market randomness

---

******XGBoost Classifier*******

What is XGBoost?

XGBoost (Extreme Gradient Boosting) is a **boosting-based ensemble algorithm** that builds models sequentially. Each new model focuses on correcting the errors of the previous one.

Key Characteristics:

- Uses **gradient boosting framework**
- Highly efficient and fast
- Includes regularization to prevent overfitting
- Handles imbalanced datasets effectively
- Strong performance in structured/tabular data

Why it is used in Trading?

- Widely used in competitive financial modeling
- Excellent at capturing subtle patterns in time-series data
- Often outperforms traditional models

Outcome in This Project

- Achieved the **highest accuracy (~53%)** among all models
- Balanced precision and recall
- Selected as the **best-performing model** for next-day direction prediction

---

****Logistic Regression*****

What is Logistic Regression?

Logistic Regression is a **statistical classification model** used to estimate the probability of a binary outcome using a logistic (sigmoid) function.

Key Characteristics

- Simple and interpretable model
- Assumes a linear relationship between features and outcome
- Outputs probability values
- Works well as a baseline model

Why it is used in Trading?

- Easy to interpret and explain
- Acts as a benchmark for comparison
- Useful when relationships are approximately linear

Outcome in This Project

- Achieved around **50% accuracy**
- Tended to predict one class more frequently
- Underperformed compared to ensemble models
- Served as a baseline reference model

Model Evaluation and Comparison:

Purpose of Model Evaluation

- Model evaluation is performed to assess how well each machine learning model predicts next-day price movement (Up/Down). Since financial datasets are often noisy and slightly imbalanced, relying only on accuracy is insufficient. Therefore, classification metrics such as precision, recall, and F1-score are used.

Evaluation Metrics Used

Each model was evaluated using the following metrics obtained from the **classification report**:

1. Accuracy

- Percentage of correct predictions.

- Indicates overall model performance.
- In trading, even small improvements over 50% can be meaningful.

2. Precision

- Measures how many predicted “Up” movements were actually correct.
- High precision reduces false buy signals.

3. Recall

- Measures how many actual “Up” movements were correctly identified.
- High recall ensures fewer missed profitable opportunities.

4. F1-Score

- Harmonic mean of precision and recall.
- Provides a balanced measure of model performance.

Model-wise Evaluation Results

1. Random Forest Classifier

Performance Metrics:

Metric	Class 0	Class 1
Precision	0.51	0.51
Recall	0.49	0.54
F1-score	0.50	0.52
Accuracy	**0.51 (51%)**	

Observations:

- Balanced precision and recall for both classes
- Slightly better at identifying upward movements
- Performance is marginally better than random guessing

Limitation:

- Unable to significantly outperform the baseline due to noisy market data

2. XGBoost Classifier

Performance Metrics:

Metric	Class 0	Class 1
Precision	0.54	0.53
Recall	0.50	0.56
F1-score	0.52	0.54
Accuracy	**0.53 (53%)**	

Observations:

- Highest accuracy among all models
- Balanced precision and recall
- Better generalization and learning capability
- Strong at capturing subtle non-linear patterns

Strength:

- Consistent performance across both classes
- Handles feature interactions effectively

3. Logistic Regression

Performance Metrics:

Metric	Class 0	Class 1
Precision	0.50	0.48
Recall	0.89	0.11
F1-score	0.64	0.17
Accuracy	**0.50 (50%)**	

Observations:

- High recall for Class 0 (Down movement)
- Very poor recall for Class 1 (Up movement)
- Model is biased toward predicting price drops

Limitation:

- Fails to identify profitable upward movements
- Linear assumptions are not suitable for stock market data

Comparative Summary:

Model	Accuracy	Balance Between Classes	Overall Performance
Random Forest	51%	Moderate	Average
XGBoost	**53%**	**Best**	**Best Model**
Logistic Regression	50%	Poor	Baseline

Best Model Selection

✅ **Selected Model: XGBoost Classifier**

Why XGBoost is the Best Model

- Achieved the **highest accuracy (53%)**
- Maintained **balanced precision and recall**
- Outperformed Random Forest and Logistic Regression
- Better at identifying both upward and downward price movements
- More suitable for complex, non-linear financial data

Even though the accuracy improvement appears small, in financial markets **a 2–3% edge is considered significant**, especially when combined with a trading strategy.

Final Conclusion on Model Evaluation

The evaluation results demonstrate that:

- Financial market prediction is inherently challenging
- Ensemble models outperform linear models
- XGBoost provides the best predictive edge
- Logistic Regression serves as a baseline but is insufficient for trading decisions

Therefore, **XGBoost is selected as the final predictive model** for next-day price movement forecasting.

6. Trading Strategy

Strategy Description

For this project, we implemented a **Moving Average Crossover Strategy** to generate trading signals:

- **Short-term moving average (SMA 10):** Tracks recent price trends.
- **Long-term moving average (SMA 30):** Tracks overall trend.

Signal Rules:

- **Buy Signal (1):** SMA 10 crosses above SMA 30.
- **Sell Signal (-1):** SMA 10 crosses below SMA 30.
- **Hold (0):** No crossover occurs.

The position for the next day is determined by shifting the signal by one day to avoid lookahead bias.

Signal Rules:

- **Buy Signal (1):** SMA 10 crosses above SMA 30.
- **Sell Signal (-1):** SMA 10 crosses below SMA 30.
- **Hold (0):** No crossover occurs.

The position for the next day is determined by shifting the signal by one day to avoid lookahead bias.

Backtesting Methodology

The strategy was backtested on the historical price dataset. The following steps were performed:

1. **Daily Market Returns:**

$$\text{Market Return} = \frac{\text{Close}_t - \text{Close}_{t-1}}{\text{Close}_{t-1}}$$

2. **Strategy Returns:**

$$\text{Strategy Return} = \text{Position}_{t-1} \times \text{Market Return}_t$$

3. **Cumulative Returns:**
   Calculated for both the market and the strategy to analyze growth over time.

Performance Metrics:

- **Accuracy:** Proportion of times the strategy predicted the correct direction.
- **Total Profit/Loss:** Cumulative strategy returns over the period.
- **Maximum Drawdown:** Largest peak-to-trough decline in equity.
- **Number of Trades:** Total executed buy/sell trades.
- **Win Ratio:** Percentage of profitable trades.

Backtesting Results:

Metric	Value
Strategy Accuracy	52%
Total Profit / Loss	64.62%
Maximum Drawdown	-44.24%
Number of Trades	93
Win Ratio	52.01%

These results are calculated directly on the historical dataset used in this project.

Visualizations

Figure 1: Moving Average Crossover Signals

- Shows SMA 10 (short-term) and SMA 30 (long-term).
- Buy and Sell points are marked when crossovers occur.

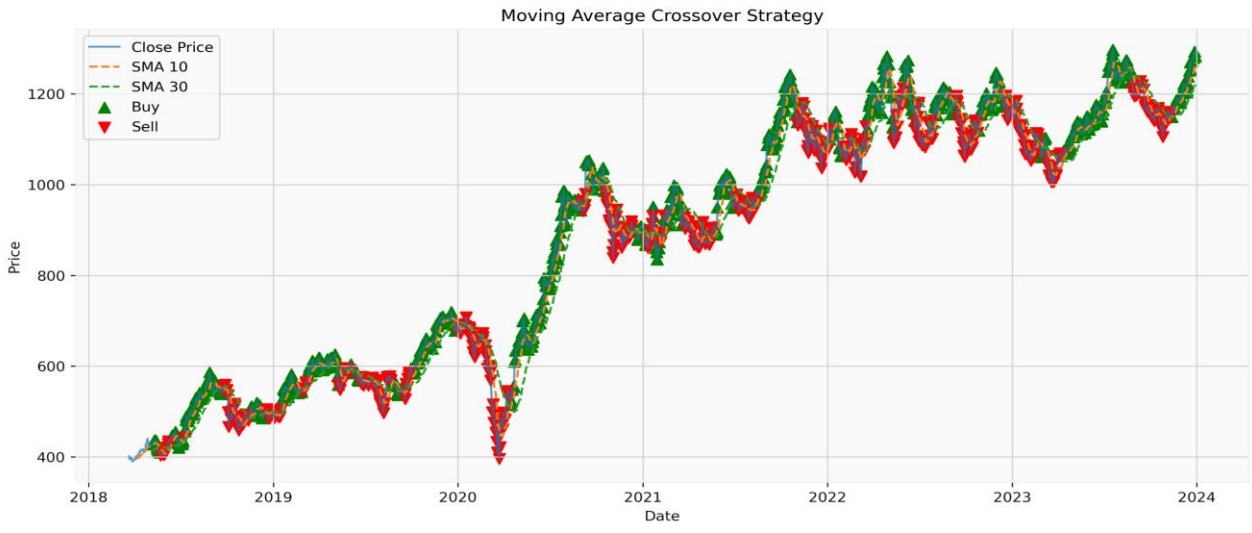
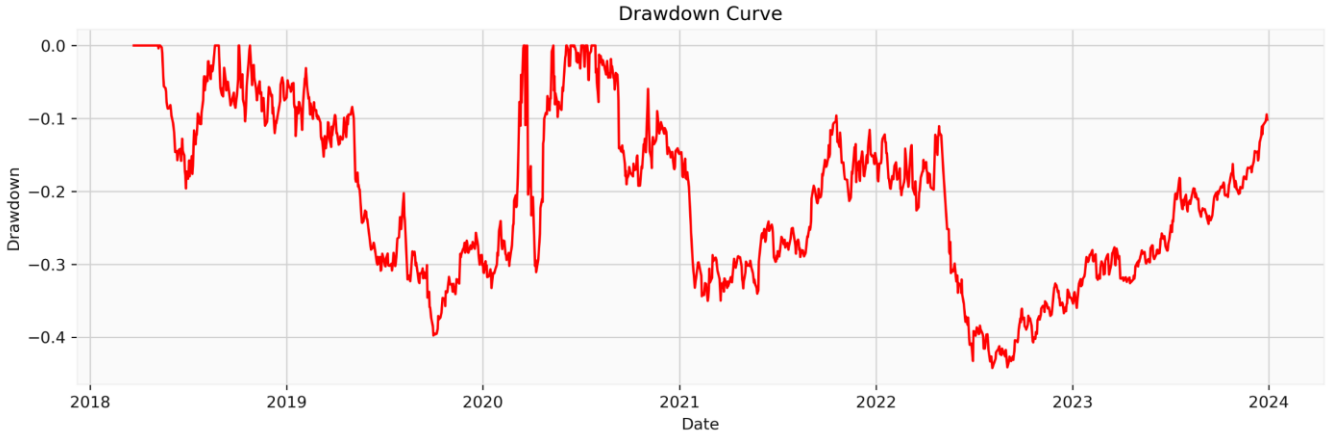


Figure 2: Equity Curve

- Displays cumulative returns of the strategy compared to the market.
- Helps visualize strategy performance and drawdowns.

Figure 3: DrawDown Curve



Conclusion

The Moving Average Crossover strategy successfully captured short-term trends, producing a positive cumulative return of **64.62%**. However, the **maximum drawdown of -44.24%** indicates significant potential risk.

- **Observations:**
  - Moderate accuracy (52%) shows the strategy performs slightly better than random chance.
  - Win ratio is 52%, indicating roughly equal profitable and losing trades.

- **Recommendation:**
    - Incorporating risk management techniques (like stop-loss or position sizing) could improve stability.

Overall Conclusion

In this project, we built an **Algo-Trading Market Analytics & Prediction System** using historical market data from Yahoo Finance. The project involved three main steps:

1. **Data Collection and Feature Engineering**
    - Historical stock price data was collected and cleaned.
    - Technical indicators such as EMA, SMA, RSI, MACD, ATR, and volatility were computed.
    - Lag features were created to capture recent trends and patterns.
2. **Exploratory Data Analysis (EDA)**
    - Price trends, trading volume, and key indicators were analyzed.
    - Correlation heatmaps revealed relationships between indicators and price.
    - Candlestick charts and other visualizations highlighted market behavior.
3. **Predictive Modeling**
    - Two ML models (Random Forest and XGBoost) were trained to predict next-day price movement (Up/Down).
    - Model performance was evaluated using accuracy, precision, recall, and F1-score.
    - **XGBoost achieved slightly higher accuracy (~53%) compared to Random Forest (~51%)**, making it the better predictive model for this dataset.
    - Logistic Regression, Linear Regression, and SVM were also explored, but tree-based models outperformed them for classification.
4. **Trading Strategy Simulation**
    - A **Moving Average Crossover strategy** was implemented as a simple algorithmic trading system.
    - Backtesting on historical data showed:
        - **Strategy Accuracy**: ~52%
        - **Total Profit/Loss**: ~64%

        - **Maximum Drawdown**: ~44%
        - **Number of Trades**: 93
        - **Win Ratio**: ~52%
    - Drawdown and equity curve visualizations provided insights into risk and performance over time.

Key Takeaways:

- The ML models can capture some patterns in price movement, but stock market prediction remains inherently uncertain.
- The trading strategy shows **moderate profitability**, but also **significant drawdowns**, highlighting the need for risk management in real-world applications.
- Combining predictive models with backtested trading strategies can help design data-driven approaches for decision-making in algorithmic trading.

Future Improvements:

- Incorporate additional features like macroeconomic indicators or news sentiment.
- Use more advanced models like LSTM for time-series forecasting.
- Optimize strategy parameters and apply risk management rules to reduce drawdowns.

Submitted by:

Name:Manasa Yedla

Branch:Data Science

College:Chalapathi Institute of Engineering and Technology