A MINI PROJECT REPORT ON

Backtested Swing Call Strategies using EMA Algorithms

Submitted Partial in Fulfillment for the Degree of

Bachelor of Technology in Computer Science

Submitted By

Pooja Sawant(58)

Chandraprabha Tawade(64)

Sakshi Tripathi(65)

Under the Guidance of

Prof.Sonal Kadam

USHA MITTAL INSTITUTE OF TECHNOLOGY

S. N. D. T. WOMEN'S UNIVERSITY, MUMBAI - 400049

2024-2025

CERTIFICATE

This is to certify that **Ms Pooja Sawant, Ms Chandraprabha Tawade, and Ms Sakshi Tripathi** has completed the Mid Term Mini Project report on the topic **"Backtested Swing Call Strategies Using EMA Algorithms"** satisfactorily in fulfillment for the Bachelor's Degree in Computer Science and Technology under the guidance of Prof. Sonal Kadam during the year 2024-2025 as prescribed by **Usha Mittal Institue Of Technology, SNDT Women's University**

Guide	**Head of Department**
Prof. Sonal Kadam	Prof. Kumud Wasnik

Principal
Dr. Yogesh Nerkar

Examiner 1	**Examiner 2**

Acknowledgment

We would like to express our sincere appreciation to everyone who gave their invaluable assistance and support in this project. We are highly indebted to Prof. Sonal Kadam for her guidance and constant supervision as well as for providing necessary information regarding the project also for her support in this project. Without her help, this project would not have been possible. Prof. Sonal Kadam.

We would also like to thank our Head of Department Prof.Kumud Wasnik for her in sights in shaping the direction and content of this report. We would also like to extend our gratitude to our classmates who provided valuable feedback and encouragement throughout the process.

Thank you all for your contributions, support, and encouragement

Contents

List of Figures

Abstract

The stock market is inherently dynamic and volatile, making it challenging for investors to make well-informed decisions. This project presents a data-driven approach that leverages Exponential Moving Average (EMA)-based swing trading strategies alongside sentiment analysis to enhance market trend prediction and strategy formulation. By analyzing historical stock data and computing 20-day and 50-day EMAs, the system generates buy and sell signals at key crossover points. The strategy is rigorously validated through systematic backtesting, ensuring its reliability in real-world trading scenarios. To further refine the decision-making process, sentiment analysis is integrated using machine learning techniques that process financial news and social media content, offering real-time insights into market sentiment and investor behavior.

Beyond signal generation, the project emphasizes effective portfolio management by evaluating the performance of multiple stocks and dynamically adjusting asset allocation based on strategy outcomes and sentiment trends. Key metrics such as return on investment, win-loss ratio, and risk-adjusted returns are used to assess the strategy's effectiveness. This holistic approach not only strengthens the predictive power of traditional technical analysis but also enhances adaptability in volatile conditions. By combining technical indicators, sentiment analysis, and portfolio-level risk management, the project aims to deliver a scalable, accurate, and robust trading framework for investors seeking consistent performance and informed decision-making in the stock market.

Problem Statement

The unpredictable nature of financial markets and the reliance on unvalidated traditional indicators often lead to inconsistent trading outcomes and poor investment decisions. This project addresses the lack of reliable, data-driven strategies by developing a back-tested EMA-based swing trading system enhanced with sentiment analysis and portfolio management. By combining historical data analysis, machine learning-driven sentiment insights from financial news and social media, and dynamic capital allocation techniques, the project aims to provide optimized, real-time buy/sell signals and risk-aware portfolio strategies—enabling traders to make more informed, consistent, and adaptive decisions in volatile market conditions.

Literature Survey

A literature survey is a detailed review of existing research on a specific topic, identifying knowledge gaps and findings to establish a strong foundation for further study. It helps in understanding current methodologies, refining research direction, and ensuring new studies build on established knowledge.

The study Stock Market Prediction: Does Trading Volume Help? (2008) investigated the effect of trading volume on stock market predictions using Neural Networks. This model predicted short-term price movements and was simple to implement. However, the inclusion of too many variables increased the risk of overfitting, and trading volume alone did not significantly enhance the accuracy of the prediction.

Stock Price Prediction Using Genetic Algorithms and Evolution Strategies (2012) applied Genetic Algorithms (GA) and Evolution Strategies (ES) to predict stock prices by evolving trading rules. Their approach effectively avoided overfitting and optimized feature selection. However, the model required high computational power and was not validated in live markets.

One of the most comprehensive studies was *Automated Stock Price Prediction and Trading Framework for Nifty Intraday Trading* (2013), which integrated Neural Networks with Sentiment Analysis, Technical Indicators, and an Automated Trading System. This framework provided a fully automated trading system with a high prediction accuracy.

In 2020, a model utilizing Long Short-Term Memory (LSTM) and Support Vector Regression (SVR) was introduced. This approach effectively handled non-linear price movements and long-term dependencies. However, it required significant computational resources and did not incorporate external market factors.

The 2021 study combined Moving Averages with Machine Learning to enhance trend prediction accuracy. Despite reducing latency in trading signals, it was limited by the lagging nature of moving averages and struggled to adapt to sudden market fluctuations.

In 2023, a more advanced model integrated LSTM with Relative Strength Index (RSI) and Exponential Moving Average (EMA) to improve stock price prediction accuracy. This model achieved an accuracy of 80-85%, surpassing previous approaches. However, it lacked fundamental analysis and real-time adaptability. These improvements ensure that the system is not only predictive but also actionable, making it suitable for practical trading applications rather than purely theoretical analysis.

TITLE, AUTHOR & YEAR	METHODOLOGY	ADVANTAGES	DISADVANTAGES	LIMITATIONS
Stock Market Prediction: Does Trading Volume Help? (2008) By A. U. Khan, T. Bandopadhyaya, and S. Sharma	Focuses on how trading volume affects stock market predictions using Neural Networks.	Can predict short-term price movements. Simple to implement.	Overfitting may occur due to too many input variables. - Trading volume alone does not significantly improve accuracy	Trading volume as a predictor is ineffective.Focus on short-term prediction only
Stock Price Prediction Using Genetic Algorithms and Evolution Strategies(2012) By Ganesh Bonde ,Rasheed Khaled	Genetic Algorithms (GA) and Evolution Strategies (ES) were applied to predict stock prices. The model evolved trading rules using evolutionary techniques.	Avoids Overfitting: Evolutionary approach ensures better generalization. GA and ES optimize feature selection, reducing irrelevant data.	High Computational Cost: GA/ES require extensive computations. No Real-Time Validation: Results not tested in live markets.	The model is not tested on real-time data, meaning it may perform poorly in live markets.Missing other crucial indicators like sentiment analysis, macroeconomic factors, and financial ratios
Automated Stock Price Prediction and Trading Framework for Nifty Intraday Trading (2013) by A. A. Bhat and S. S. Kamath	Combines Neural Networks with Sentiment Analysis, Technical Indicators, and an Automated Trading System to predict and trade stocks.	Fully automated trading system. Accounts for external factors and sentiment.	Limited to the Indian NIFTY market. High reliance on data quality for sentiment analysis.	Only tested on Indian markets. Relies on sentiment data.
Stock Price prediction using LSTM and SVR(2020) by Gourav Bathla	Uses Long Short-Term Memory (LSTM) and Support Vector Regression (SVR) for stock price prediction.	Handles non-linear stock price movement better than traditional models; reduces vanishing gradient problem.	LSTM is computationally expensive; SVR struggles with large datasets.	Ignores sentiment analysis and fundamental analysis, focusing only on price movement.
Prediction of Trends in Stock Market Using Moving Averages and Machine Learning (2021) by N. R. Rao, S. Dinesh, S. R. Samhitha, and S. P. Anusha	Uses Moving Averages combined with Regression models to forecast stock market trends.	Reduces latency in trading signals; improves trend prediction accuracy.	Moving averages are lagging indicators; cannot handle market volatility well.	Model struggles with rapid trend reversals and unexpected market shocks.
Stock market prediction using the LSTM algorithm with the Relative Strength Index (RSI) and Exponential Moving Average (EMA) indicators(2023) by Rahul Maruti Dhokane, Sohit Agarwal	Combines LSTM with Relative Strength Index (RSI) and Exponential Moving Averages (EMA) to improve prediction accuracy.	Enhances prediction by integrating technical indicators; reduces mean absolute percentage error (MAPE).The inclusion of RSI and EMA refines trend detection, making the system more reliable.Giving Higher Accuracy (80-85%)	Requires extensive feature engineering; model training takes longer due to added complexity.	Limited to technical indicators, lacks fundamental analysis and real-time adaptability.

Figure 3.1: Literature Survey

Introduction

The stock market is a dynamic environment where investors strive to make informed decisions by leveraging both historical patterns and real-time insights. This project presents an enhanced swing trading system built around Exponential Moving Average (EMA) crossovers—specifically using 20-day and 50-day EMAs—to generate actionable buy and sell signals. A bullish signal is triggered when the 20-day EMA crosses above the 50-day EMA, and a bearish signal when it crosses below. In addition to this core strategy, risk management mechanisms such as stop-loss and take-profit are implemented to protect capital and lock in gains. Historical stock data is acquired through Yahoo Finance and processed using Python libraries including Pandas, NumPy, Matplotlib, and mplfinance to visualize trends and simulate trade outcomes.

To further refine decision-making, the project incorporates a sentiment analysis module powered by VADER, which evaluates financial news headlines for positive, negative, or neutral sentiment. Headlines are also categorized into themes like Earnings, Growth, and Regulation, providing deeper context for market behavior. Backtesting functionality evaluates trade effectiveness, while performance metrics like return percentage, win/loss ratio, and risk levels are tracked in a virtual portfolio. The system also compares selected stocks (e.g., Reliance) against industry peers such as TCS, Infosys, and Wipro to provide broader market perspective. A visualization dashboard presents technical signals, sentiment distributions, and portfolio trends to help users interpret results intuitively.

This project bridges technical trading logic with sentiment intelligence, delivering a data-rich, user-friendly framework for both novice and experienced traders. Future enhancements may include real-time API integration, deployment on live trading platforms, and the use of machine learning models to further optimize strategy parameters and adapt to evolving market conditions.

Existing System

Current trading strategies rely on technical indicators like RSI, MACD, and Bollinger Bands. However, these methods lack backtesting, leading to unverified trading decisions. Additionally, manual trading strategies are prone to human errors and biases.

- Stock Market Prediction: Does Trading Volume Help? (2008) : Examines the impact of trading volume on stock market predictions using neural networks.Simple to implement but does not significantly improve accuracy.

- Stock Price Predition Using Genetic Algorithms and Evolution Strategies (2012) Uses genetic algorithms (GA) and evolution strategies (ES) to optimize feature selection.Avoids overfitting but has high computational costs and lacks real-time validation.

- Automated Stock Price Prediction and Trading Framework for Nifty Intraday Trading (2013) : Combines neural networks, sentiment analysis, technical indicators, and an automated trading system. Fully automated with 71 percentage prediction accuracy but limited to the Indian NIFTY market.

- Stock Price prediction using LSTM and SVR(2020) : Uses Long Short-Term Memory (LSTM) and Support Vector Regression (SVR) for stock price prediction.

- Prediction of Trends in Stock Market Using Moving Averages and Machine Learning (2021) : Uses Moving Averages combined with Regression models to forecast stock market trends.

- Stock market prediction using the LSTM algorithm with the Relative Strength Index (RSI) and Exponential Moving Average (EMA) indicators(2023)
  Combines LSTM with Relative Strength Index (RSI) and Exponential Moving Averages (EMA) to improve prediction accuracy.

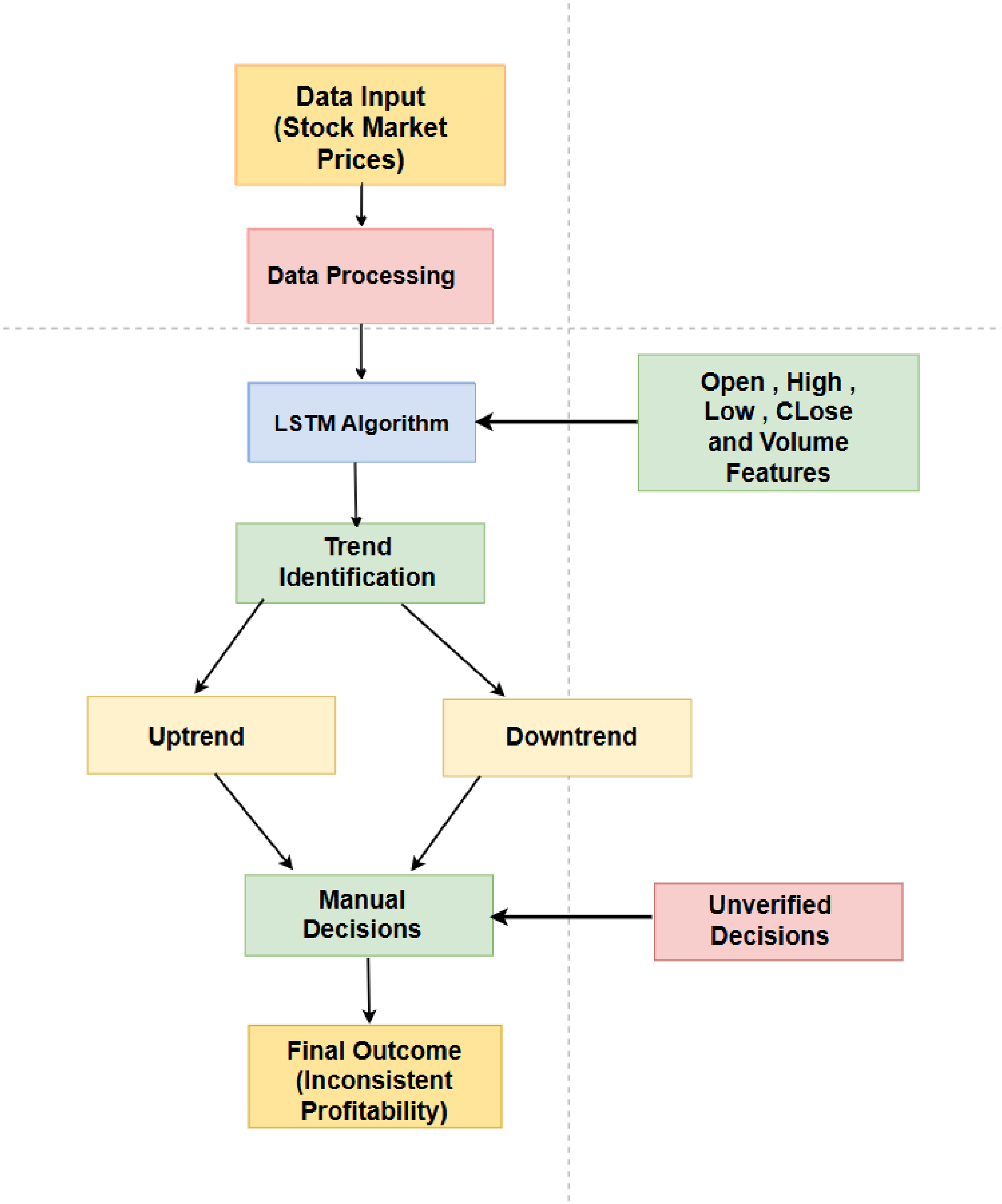


Figure 5.1: Flow of Existing system

Proposed System

This project aims to test a swing trading strategy using Exponential Moving Averages (EMA) and sentiment analysis from news data to enhance trading decisions and to help traders take better actions. The strategy focuses on identifying optimal buy and sell signals based on past stock price trends, allowing traders to capitalize on short-term market movements.Develop a systematic approach to swing trading using the 20-day and 50-day EMA.Identify key buy and sell signals based on EMA crossovers. Backtest the strategy on historical stock data to evaluate its effectiveness. Compare the strategy's performance against market benchmarks. Provide insights into the risks and profitability of this trading method. Lastly the model also provides the users with a holistic portfolio summary for better analysis.

- Step 1: Fetch Stock Data & News Data: Collect historical stock prices (e.g., Reliance, NIFTY 50) for 600 days using Yahoo Finance API. Simultaneously, fetch relevant news articles or headlines affecting the selected stocks.

- Step 2: Preprocessing (Cleaning, Organizing Data): Clean stock price data (remove null values, format dates).

  Clean and preprocess news data (remove stopwords, tokenize, etc.).

  Structure both datasets for further analysis.

- Step 3: Perform EMA Calculation : Plot stock prices using candlestick charts.

  Calculate:

  20-day EMA (faster line - reacts to short-term trends)

  50-day EMA (slower line - shows long-term trend)

  Identify crossovers:

  Buy Signal → 20 EMA crosses above 50 EMA (green dot).

Sell Signal → 20 EMA crosses below 50 EMA (red dot).

- Step 4: Perform Sentimental Analysis: Use sentiment analysis (e.g., VADER or TextBlob) to evaluate positive/negative/neutral tone of news headlines.

  Tag each article with a sentiment score.

- Step 5: News Categorization: Classify news into categories such as:

  Earnings, Market Sentiment, Government Policy, Global Events etc

  Use this classification to correlate news sentiment with stock movement trends.

- Step 6: Backtesting (Testing Strategy Performance): Simulate trades based on historical buy/sell EMA signals.

  Calculate: - Profit/loss per trade, Total profit/loss, Success rate, Maximum drawdown

- Step 7:Validating Buy/Sell Signals: Compare generated EMA signals with sentiment-based data.

  Filter signals if sentiment contradicts the technical signal (for example, avoid buying on a negative sentiment day).

- Step 8: Result Analysis + Visualization:

  Visualize: EMA crossovers on price chart, Buy/Sell points, Sentiment overlaid on trading days, Portfolio equity curve

  Analyze: Signal accuracy, Profitability, Risk factors

- Step 9: Generate Portfolio Summary: Summarize:

  Total returns vs. market index (e.g., NIFTY 50)

  Trade statistics

  Risk metrics (Sharpe ratio, Max Drawdown)

  Recommendations for improvement or future research

In this way, all the above steps help in understanding the Proposed Solution better.

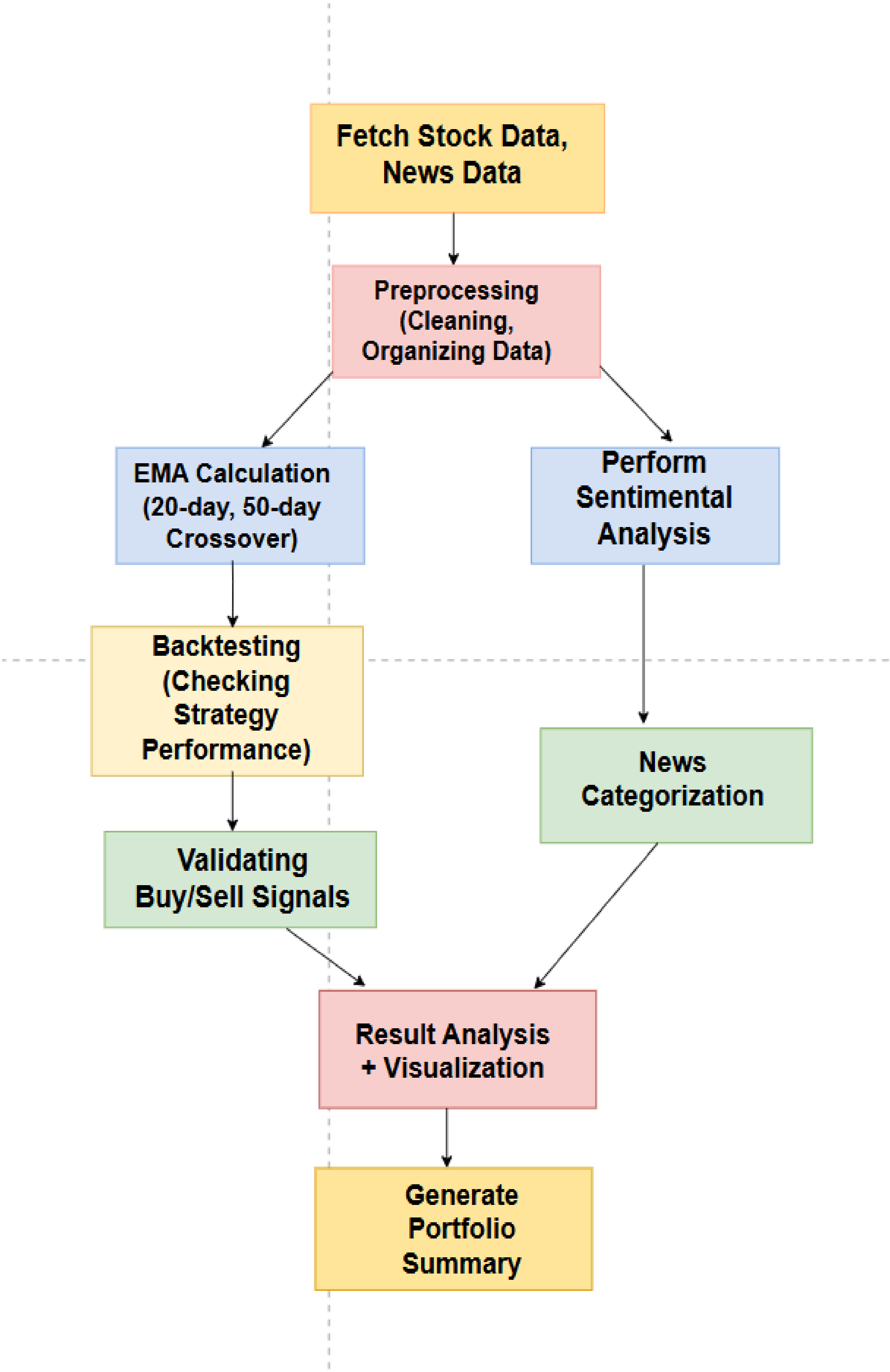


Figure 6.1: Proposed System

Architectural Overview

Detailed Architecture for Backtested Swing Trading Module

1. User Input Module

- **User Interaction:** Users select stock symbols (e.g., RELIANCE), choose a strategy (e.g., EMA crossover), and define the backtest duration (e.g., 600 days).
- **Input Validation:** Ensures correct format (e.g., valid date range, numeric thresholds) and completeness of all inputs.

2. Data Collection and Preprocessing

- **Stock Data:** Retrieved from Yahoo Finance API including OHLCV (Open, High, Low, Close, Volume) data.
- **News Headlines:** Extracted using Google News API or similar services for relevant stock-related news.
- **Preprocessing:** Removal of irrelevant columns (e.g., dividends), rounding of prices, and consistent formatting.
- **Storage:** All data is stored in-memory to maintain a lightweight and efficient pipeline.

3. Strategy Implementation

- **Indicator Calculation:** 20-day and 50-day EMAs are computed using `pandas`.
- **Signal Generation:** Buy signal when 20EMA crosses above 50EMA; sell when 20EMA falls below 50EMA.

- **Risk Management:** Integrated stop-loss and take-profit conditions based on historical price range.

4. Sentiment Analysis Module

- **VADER Analysis:** Each headline is analyzed using the VADER model to assess sentiment polarity.
- **Categorization:** News items are further categorized (e.g., Earnings, Regulatory News, Growth) for deeper insight.

5. Backtesting Engine

- **Signal Execution:** Trades are simulated by iterating through historical EMA signals.
- **Result Compilation:** For each trade, returns and timestamps are logged.
- **Performance Analysis:** Calculates metrics such as win/loss ratio, cumulative return, and average trade return.

6. Visualization

- **Candlestick Charts:** Visual representation of price movement with EMA overlays using `mplfinance`.
- **Signal Markers:** Buy/sell signals highlighted with green/red dots.
- **Sentiment Pie Chart:** Distribution of categorized news sentiments displayed using pie plots.

7. Report Generation

- **Summary Compilation:** Tabular and visual summary of trades, returns, and analysis.
- **Export Options:** Reports are generated in PDF format with graphical summaries and downloadable access.

Process Flow

The complete architecture flows through the following stages:

`User Input` → `Fetch Stock + News Data` → `Preprocess` → `EMA Strategy & Sentiment Analysis` → `Backtest` → `Analyze` → `Visualize` → `Generate Report`

- This detailed architecture outlines each stage of the enhanced swing trading module, integrating both technical indicators and real-time sentiment analysis. The system emphasizes dynamic data handling, with efficient in-memory storage, robust signal generation using EMA crossovers, and risk management via stop-loss and take-profit thresholds. By combining financial news sentiment through VADER analysis and categorized event detection, the model strengthens decision-making for each trade. The modular, database-free architecture ensures clarity, real-time responsiveness, and highly actionable results, offering users a comprehensive and insightful trading experience.

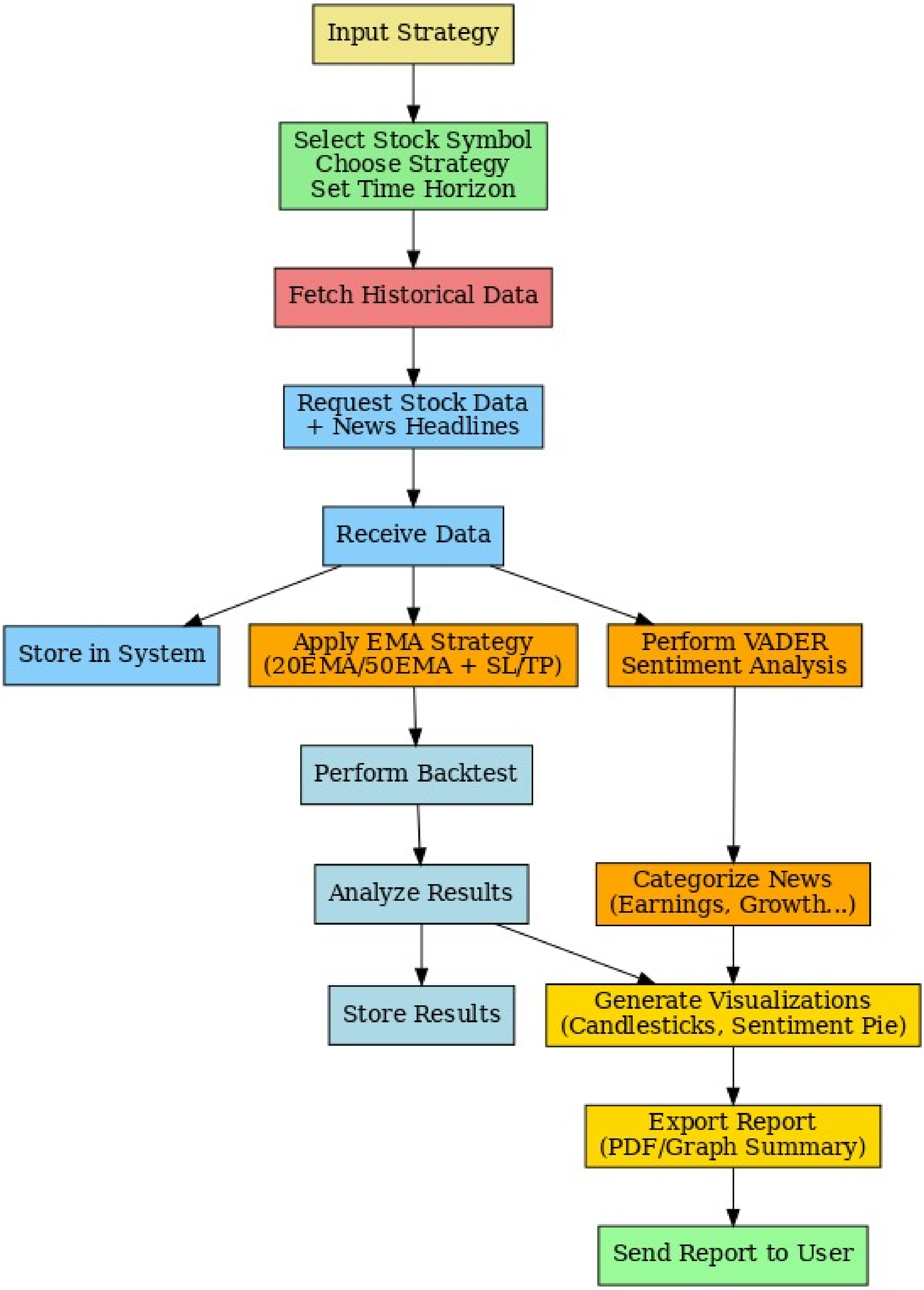


Figure 7.1: Architecture Overview

Hardware and Software requirement

1. Hardware requirements

For Development System:

- **Processor:** Intel Core i5 or higher / AMD Ryzen 5 or higher
- **RAM:** Minimum 8 GB (Recommended: 16 GB for smoother performance during backtesting and visualization)
- **Storage:** Minimum 50 GB of free space (SSD recommended for faster data handling, if working locally)
- **Internet Access:** Required for:
  - Fetching real-time and historical stock market data (using `yfinance`)
  - Accessing financial news via RSS feeds (using `feedparser`)
  - Using Google Colab cloud platform

2.Software Requirements

Development Environment:

- **Operating System:** Windows / Linux / macOS
- **Development Platform:** Google Colab
- **Python Version:** Python 3.8 or later
- **Package Manager:** `pip` (Pre-installed in Colab and local Python environments)
- **Python Libraries Used:**
  - `pandas` – Data manipulation and analysis
  - `numpy` – Numerical operations

- `matplotlib.pyplot` – Data visualization
- `yfinance` – Fetching historical stock market data
- `mplfinance` – Candlestick chart plotting
- `feedparser` – Parsing news headlines from RSS feeds
- `nltk` – Natural Language Toolkit for text processing,specifically sentiment analysis.
- `fpdf` – To generate PDF reports.
- `os` – For creating directories and saving files.

Implementation of the Project

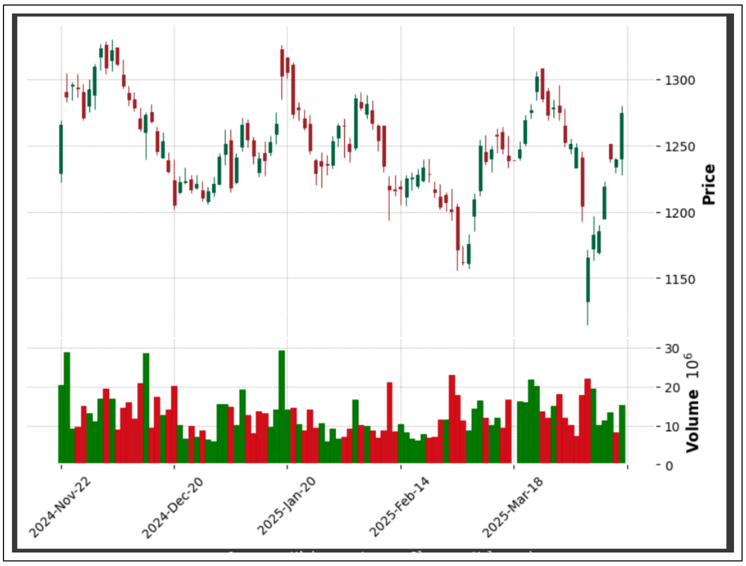


Figure 9.1: Stock Graph/Chart

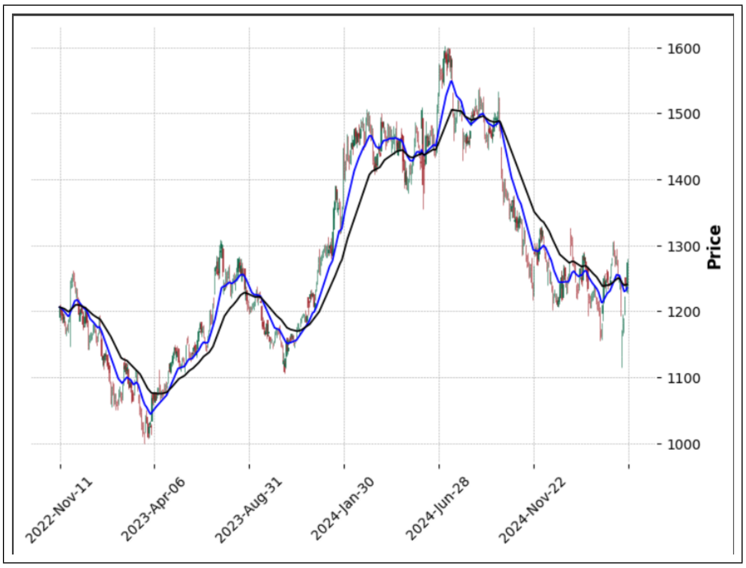


Figure 9.2: Chart with EMAs

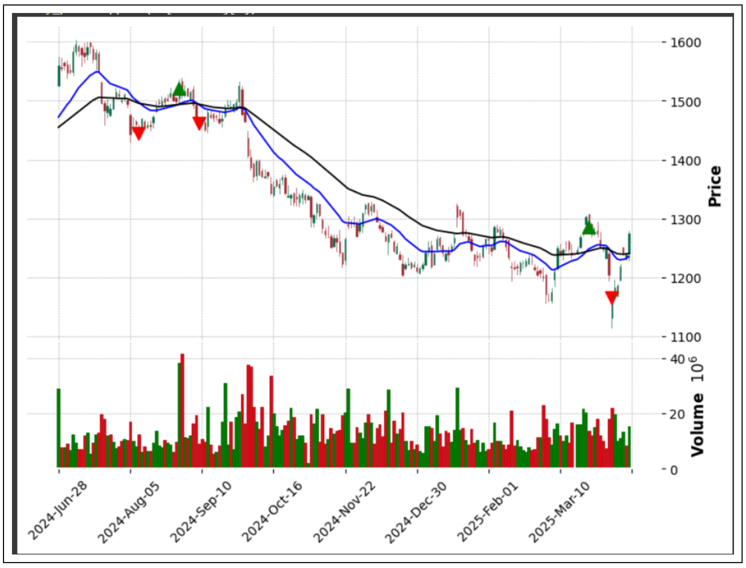
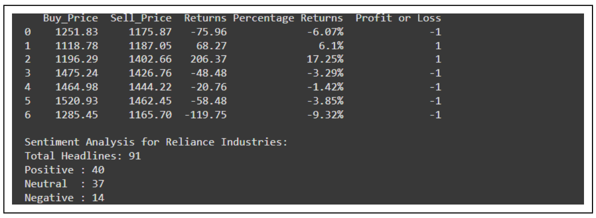


Figure 9.3: Chart with Buy / Sell Signals



	Buy_Price	Sell_Price	Returns	Percentage Returns	Profit or Loss
0	1251.83	1175.87	-75.96	-6.07%	-1
1	1118.78	1187.05	68.27	6.1%	1
2	1196.29	1402.66	206.37	17.25%	1
3	1475.24	1426.76	-48.48	-3.29%	-1
4	1464.98	1444.22	-20.76	-1.42%	-1
5	1520.93	1462.45	-58.48	-3.85%	-1
6	1285.45	1165.70	-119.75	-9.32%	-1

Sentiment Analysis for Reliance Industries:
Total Headlines: 91
Positive : 40
Neutral : 37
Negative : 14

Figure 9.4: Sentimental Analysis Results

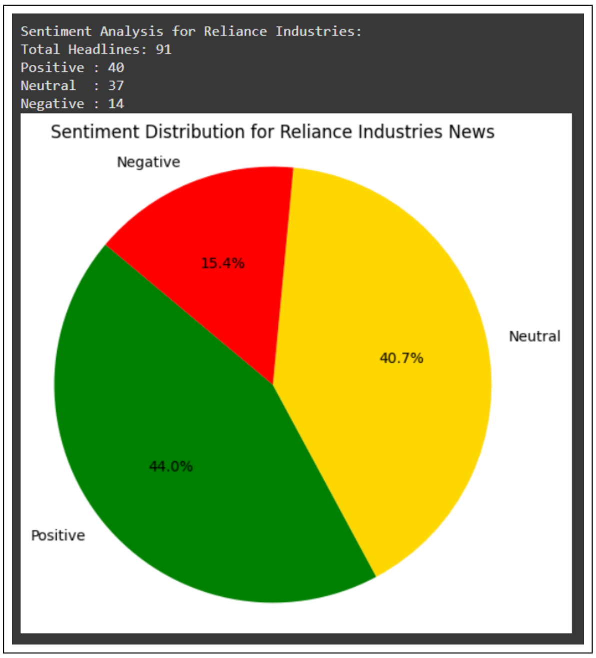


Figure 9.5: Sentiment Distribution

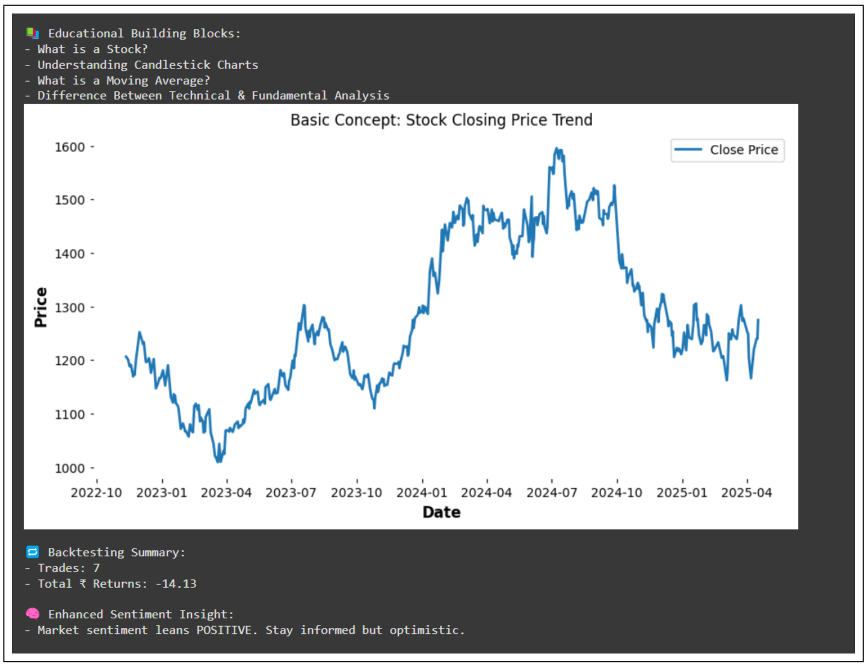


Figure 9.6: Understanding of Basic Concepts

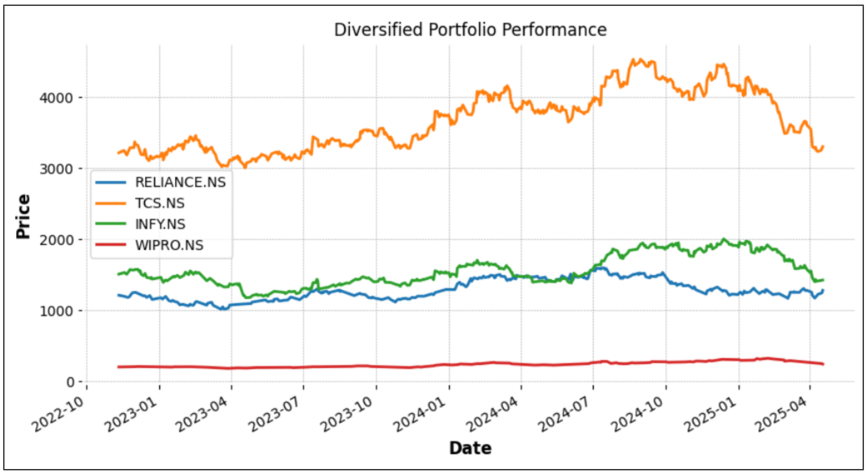


Figure 9.7: Diversified Portfolio Performance

Applications and Future Scope

Applications:

1. Development of swivel trading strategies

- Provides traders with a data-driven framework using EMA crossovers for identifying optimal buy and sell signals.
- Reduces reliance on intuition by following a rules-based system with integrated stop-loss and take-profit logic.
- Enhances decision-making with sentiment-informed trade validation.
- Can be adapted to various stock symbols and market sectors.

2. Strategy Backtesting and Evaluation

- Enables testing of swing trading strategies on historical stock data to measure profitability and consistency.
- Assesses performance through metrics like return percentage, win/loss ratio, and cumulative profit.
- Simulates market conditions to evaluate strategy behavior under varying volatility.

3. Sentiment Analysis Integration

- Uses financial news sentiment scoring to support or filter trade signals based on market sentiment.
- Categorizes news events into themes (e.g., Earnings, Regulation) to understand external influences.
- Adds an extra validation layer for signal reliability and market readiness.

4. Risk Management and Trade Optimization

- Implements and tests stop-loss and take-profit rules to minimize drawdowns and secure profits.
- Measures risk exposure and portfolio performance using a visual risk meter.
- Supports dynamic strategy tuning based on trade results and risk-return metrics.

5. Portfolio Diversification and Peer Comparison

- Allows multi-stock analysis to observe and compare performance across different companies (e.g., TCS, Infosys, Wipro).
- Visualizes diversified portfolio movement over time.
- Aids in selecting better-performing stocks for optimized investment allocation.

6. Educational and Research Applications

- Serves as a learning module for students and traders to understand trading logic, backtesting, and market behavior.
- Offers beginner-friendly insights with tips, alerts, and visual tools to demystify trading.
- Useful for financial research and model validation in academic and professional settings.

Future Scope:

As financial markets evolve and technology advances, there is substantial potential to enhance and extend the capabilities of this project. Future improvements may include:

- **Strategy Optimization Engines:** Use reinforcement learning or genetic algorithms to optimize EMA lengths, stop-loss levels, and risk-reward ratios.
- **Real-Time Trading with API Integration:** Connect the system to live trading platforms (e.g., Zerodha, Upstox) for automated execution of trade signals.

- **Advanced Technical Indicators:** Incorporate additional tools such as RSI, MACD, Bollinger Bands, and Fibonacci levels to enhance signal strength and reliability.
- **Scalability Across Asset Classes:** Extend functionality to support cryptocurrencies, commodities, and forex for broader market application.
- **Social Media Sentiment Analysis:** Include real-time data from platforms like Twitter and Reddit to better capture public opinion and emerging market sentiment.
- **Deployment as an Interactive Dashboard:** Build a web-based interface using Streamlit or Dash to allow users to interact with the model in real time.

Conclusion

Thus, our project successfully implements an enhanced Python-based swing trading strategy that combines Exponential Moving Average (EMA) crossovers with risk management techniques, sentiment analysis, and multi-stock portfolio tracking. By leveraging tools such as NumPy, Pandas, Matplotlib, mplfinance, and yfinance, the system automates stock data retrieval, technical signal generation, trade execution, and result visualization in a structured and reproducible manner.

The inclusion of real-time sentiment analysis through VADER and categorized news tagging enhances the model's ability to respond to market sentiment, while the integration of stop-loss and take-profit logic introduces practical risk control for more realistic performance evaluation. The system's ability to simulate trades, analyze win/loss ratios, and compare returns across industry peers provides users with a deeper level of insight into market behavior.

Although the current implementation shows great potential to improve trading decisions and minimize emotional bias, future enhancements, such as the application of machine learning for adaptive signal filtering, the inclusion of additional technical indicators and the extension to real-time trading APIs, could significantly increase the accuracy and applicability of the model in the real world. Overall, the project contributes to the evolving landscape of algorithmic trading by offering a modular, scalable, and data-driven framework for modern stock market analysis and strategy execution.

References

1. A. A. Bhat, S. S. Kamath, "Automated Stock Price Prediction and Trading Framework for Nifty Intraday Trading," IEEE, 2013.

2. G.Bathla, "Stock Price Prediction Using LSTM and SVR," in Proceedings of the Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC), IEEE, 2020.

3. Bonde and R. Khaled, "Stock price prediction using genetic algorithms and evolution strategies, 2012."

4. N. R. Rao, S. Dinesh, S. R. Samhitha, and S. P. Anusha, "Prediction of Trends in Stock Market Using Moving Averages and Machine Learning," in Proceedings of the International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), IEEE, 2021.

5. R. M. Dhokane and S.Agarwal, "Stock Market Prediction Using LSTM Algorithm in Association with RSI and EMA Indicators," Gyan Vihar University Research Journal, vol. 2023, Oct. 2023.

6. Stock Market Prediction: Does Trading Volume Help? (2008) By A. U. Khan, T. Bandopadhyaya, and S. Sharma