Stock Movement Analysis Based on Social Media Sentiment

(CapX - Data Science Internship)

Name: Abhash Goyal

GitHub Link:

Video DemoLink:

Linkedin

A sophisticated machine learning system that predicts stock price movements by analyzing various data sources and technical indicators. The project combines data from **Reddit**, **news sources**, and **market data(Yahoo Finance)** to provide comprehensive stock predictions using advanced sentiment analysis and machine learning techniques.

1. SCRAPPING PROCESS

1.1 Reddit Sentiment Scraping Using (Reddit - PRAW)

- **Purpose:** Extract user opinions and discussions on Reddit subreddits (e.g., wallstreetbets, stocks) related to the stock.

- **Process:**
  - PRAW (Python Reddit API Wrapper) was used to query posts matching the stock symbol within the past week.
  - Relevant metadata (e.g., post score, comment count, creation time) was collected.
  - Posts were preprocessed using the same cleaning technique as news articles.
  - Sentiment analysis was conducted on post content (title + body).
  - Data was aggregated to capture daily sentiment trends from Reddit.

1.2 Stock Data Scraping Using *(yfinance)*

- **Purpose:** Retrieve historical stock data (e.g., prices, volume, dividends) for the chosen stock.
- **Process:**
  - The *yfinance* library's *Ticker* object was used to fetch stock history.
  - Data for the past year (*period="1y"*) was retrieved by default.

    - Key stock metrics such as Open, High, Low, Close, and Volume were included.
    - The *index* was standardized to remove timezone information for consistency.

1.3 News Sentiment Scraping Using (NewsAPI)

- **Purpose:** Collect recent news articles related to the stock symbol and analyze sentiment.
- **Process:**
    - The NewsAPI client fetched articles published within the past 7 days (*time_delta* set to 7).
    - Articles were filtered for relevance to the stock symbol.
    - Text data (title and description) was cleaned using regex to remove URLs, punctuation, and other unwanted characters.
    - Sentiment analysis was performed using *TextBlob* to compute:
        - Polarity: Indicates positivity/negativity of the news.
        - Subjectivity: Measures opinion-based content.
    - The sentiment metrics were stored with the publication date for temporal alignment with stock data.

1. 4 Combining Sentiment and Stock Data

- Sentiment data from **NewsAPI** and **Reddit** was combined into a single dataset with fields for:
    - *Average sentiment polarity, subjectivity, and mention counts.*
- Stock data was merged with sentiment data using the date as the primary key.

1.5 Data Cleaning

- ***Removal of URLs:*** Eliminates hyperlinks using regex to focus on meaningful text.
- ***Removal of Special Characters**:* Strips non-alphanumeric symbols.
- ***Removal of Extra White Spaces**:* Standardizes spacing by replacing multiple spaces with a single space.
- ***Conversion to Lowercase:*** Normalizes text by converting it to lowercase.
- ***Handling Missing Values**:* Replaces null text fields with empty strings to prevent errors during processing.

2. FEATURES EXTRACTED AND THEIR ROLE IN STOCK MOVEMENT PREDICTIONS

2.1 Stock Data Features

Stock data is collected using the *yfinance* library, and technical indicators are computed to capture market trends, momentum, and volatility. These features include:

a. Simple Moving Averages (SMA)

- **Description**: Averages of stock closing prices over different time windows (e.g., 20 and 50 days).
- **Purpose**: SMAs smooth out price data to identify trends. For example: If the 20-day SMA crosses above the 50-day SMA, it may indicate a bullish signal.

b. Relative Strength Index (RSI)

- **Description**: Measures the magnitude of recent price changes to evaluate overbought or oversold conditions (values range from 0 to 100).
- **Purpose**:
    - RSI > 70: Overbought (potential reversal or sell signal).
    - RSI < 30: Oversold (potential reversal or buy signal).

c. Moving Average Convergence Divergence (MACD)

- **Description**: Shows the relationship between two EMAs (12-day and 26-day).
- **Purpose**:
    - Positive MACD: Bullish momentum.
    - Negative MACD: Bearish momentum.

d. Bollinger Bands

- **Description**: Consist of a middle band (20-day SMA), upper band, and lower band based on price volatility.
- **Purpose**: Identifies overbought/oversold levels:
    - Price touching the upper band may indicate overbought conditions.
    - Price touching the lower band may indicate oversold conditions.

e. Returns and Volatility

- **Returns**: Percentage change in stock price over time, useful for trend detection.
- **Volatility**: Rolling standard deviation of returns, capturing market instability.

2.2 Sentiment Data Features

Sentiment analysis is applied to news articles and Reddit discussions related to the stock. These features provide insights into market sentiment:

a. Sentiment Polarity

- **Description**: Captures the emotional tone of text (range: -1 for negative, 0 for neutral, +1 for positive).
- **Purpose**:
  - Positive polarity may indicate bullish sentiment.
  - Negative polarity may suggest bearish sentiment.

b. Sentiment Subjectivity

- **Description**: Measures the degree of opinion vs. factual content in text (range: 0 to 1).
- **Purpose**: Higher subjectivity indicates opinions, which may reflect emotional reactions to market news.

c. Mention Count

- **Description**: Counts the number of mentions of the stock in news or Reddit discussions.
- **Purpose**: Higher mention counts often correlate with significant stock movements due to increased attention.

2.3 Feature Preparation and Integration

The stock data and sentiment data are combined to form a comprehensive dataset. Missing sentiment values are filled with zero to avoid data loss. This integration ensures that both technical and sentiment-based indicators are included in the predictive model.

2.3 Relevance to Stock Movement Predictions

- **Technical Features**: Indicators like RSI, SMA, and MACD provide insights into price momentum, trends, and market conditions.
- **Sentiment Features**: Polarity, subjectivity, and mentions reflect market psychology and public reaction to news, which often drive short-term price movements.
- **Combined Power**: Integrating technical and sentiment features provides a well-rounded dataset that captures both quantitative trends and qualitative market signals, improving prediction accuracy.

3. MODEL EVALUATION

3.1 Training and Testing the (Random Forest Model)

The Random Forest model is a powerful ensemble machine learning algorithm that is widely used for regression and classification tasks. In this case, it is used to predict stock prices. The steps involved in training and testing the model include:

1. **Splitting Data**: The dataset is split into ***training*** *and* ***testing*** sets. Typically, 80% of the data is used for training, and the remaining 20% is reserved for testing the model. This helps evaluate the model's performance on unseen data, which is essential for generalization.
2. **Model Training**: The Random Forest model is trained using the training data. During this phase, the model learns patterns from the historical stock data and sentiment features to make predictions. It builds multiple decision trees and aggregates their predictions to improve accuracy and reduce overfitting.
3. **Model Testing**: After training, the model is tested on the **test set** to evaluate its predictive performance. The test data is not used during the training phase, allowing for a more accurate assessment of how well the model generalizes to unseen data.

3.2 Model Evaluation Metrics

The primary metrics used to evaluate the model's performance are the ***R^2 score*** *and* ***visual comparison*** of predicted vs. actual prices.

a. R^2 Score (Coefficient of Determination)

- **Definition**: The **R^2 score** measures how well the model explains the variance in the target variable (stock price) based on the input features. It indicates the proportion of variance in the dependent variable (stock price) that can be explained by the independent variables (features like technical indicators and sentiment).
- **Interpretation**:
  - **$R^2 = 1$**: Perfect prediction, meaning all variance in the stock price is explained by the model.
  - **$R^2 = 0$**: The model explains none of the variance and performs no better than predicting the average value.
  - **Negative R^2**: Indicates that the model performs worse than a simple average.
- **train_score**: This score indicates how well the model performs on the training data.
- **test_score**: This score shows how well the model performs on unseen (test) data, offering insights into the model’s ability to generalize to new data.

b. Plotting Actual vs Predicted Prices

- **Actual Prices**: Represented in one color (e.g., blue), showing the true stock price movements.
- **Predicted Prices**: Represented in a different color (e.g., red), showing the model's predicted stock prices.

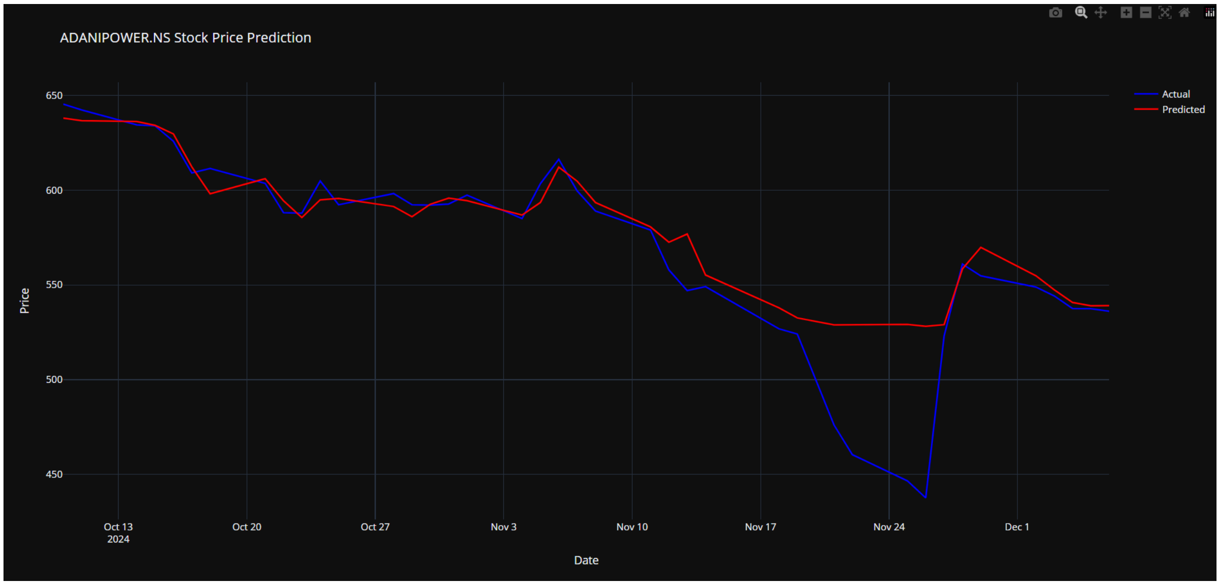


Figure 1: ADANI POWER.NS Stock Price Prediction

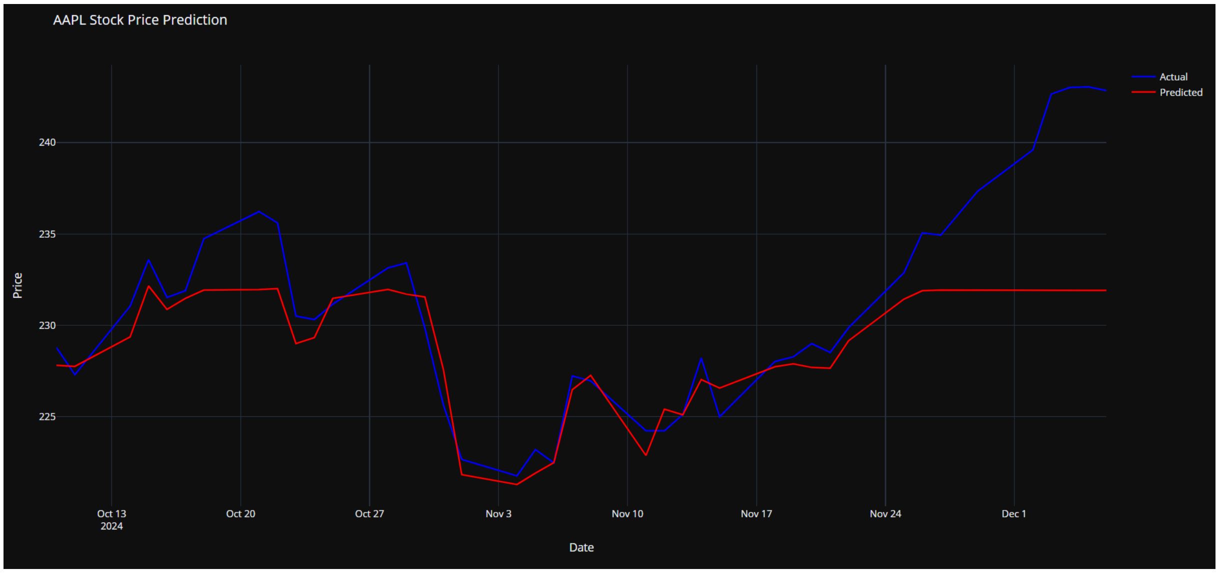
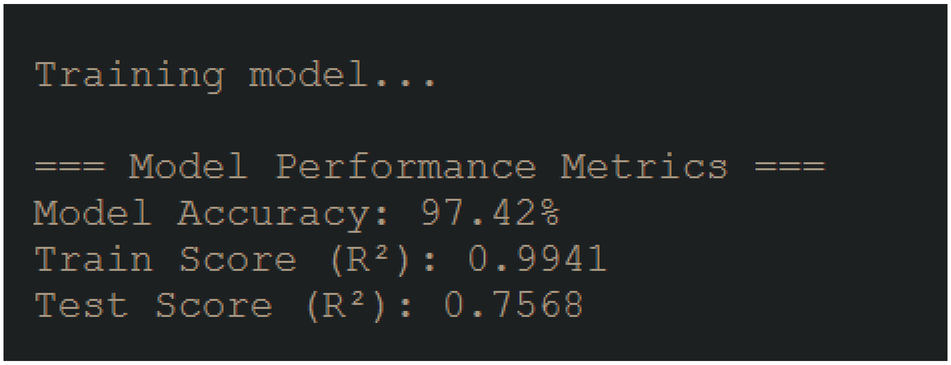


Figure 1: AAPL Stock Price Prediction

Figure 1 illustrates the actual (blue line) vs. predicted (red line) stock prices for Apple Inc., showing a close alignment with minor deviations and an overall upward trend toward the end. Figure 2 depicts the actual vs. predicted stock prices for Adani Power Ltd., where the predicted values align moderately well with the actual prices but exhibit notable discrepancies during sharp declines or rises, with an overall downward trend followed by a recovery spike.

3.3 Performance Metrics

- *Model Accuracy:* Indicates that the model's predictions are highly close to the actual stock prices, with a 97.42% accuracy rate.
- *Train Score*: The model explains 99.41% of the variance in the training data, indicating a very good fit.
- *Test Score*: The model explains 75.68% of the variance in the test data, suggesting reasonable performance but some room for improvement.



```
Training model...

=== Model Performance Metrics ===
Model Accuracy: 97.42%
Train Score (R²): 0.9941
Test Score (R²): 0.7568
```

Figure 3: Performance Metrics

4. CHALLENGES

5.1 Scrapping

1. I was using Twitter API initially but due to its limitation of 100 tweets and so many restrictions afterwards I decided to go with Reddit.
2. Twitter API was expensive so I used Reddit APIs.
3. The data from reddit was very unorganised.
4. Handling multiple API failures gracefully (NewsAPI and Reddit).
5. Managing API rate limits and timeouts.
6. Dealing with authentication using environment variables.

5.2 Feature Extraction

1. Timezone handling between different data sources.
2. Merging sentiment data with stock data on dates.
3. Handling missing data and NaN values.
4. Text cleaning and normalization.

5.3 Technical Calculations

1. Calculating complex technical indicators (RSI, MACD, Bollinger Bands).
2. Ensuring proper window sizes for rolling calculations.
3. Dealing with data lag in indicators.

5.4 Sentiment Analysis

1. Combining sentiment from multiple sources (news and Reddit).
2. Converting text data into meaningful numerical features.
3. Handling different sentiment scales and aggregation.

5.5 Model Evaluation

1. Feature selection and engineering.
2. Maintaining temporal order in train-test split.
3. Balancing model complexity with performance.
4. Avoiding data leakage.

5. FUTURE IMPROVEMENTS

- Integration of additional data sources (e.g., Twitter, financial reports).
- Enhancing the feature set with alternative indicators of economic data.
- Advanced model techniques like LSTM or Transformers for time-series prediction.