Sentiment-Driven Stock Price Prediction: A Machine Learning Approach Using Reddit and Financial Data

by Subhangi Dey

Abstract

This project explores the use of machine learning algorithms to predict stock price movements based on textual data from *Reddit* and stock price data from *Yahoo! Finance*. The analysis captures public sentiment regarding Tesla stock by web scraping posts from relevant subreddits such as *stocks*, *WallStreetBets*, and *cryptocurrency*. Our methodology integrates the financial data, including open and closed prices, to evaluate market reactions. Data preprocessing steps, including cleaning text data with NLP techniques and filtering significant stock price differences, ensure the dataset's quality and relevance. We implemented eight machine learning models, including *Logistic Regression*, *SVM variants, Random Forest,* and *Naive Bayes*, to classify stock price movements. We found that the *Logistic Regression* model performed the best, with an accuracy of 81%. This project demonstrates the potential of combining social media sentiment and historical data to predict stock price changes, providing a foundation for further research and practical applications in financial analytics.

Table of Contents:

1. Introduction

The stock market is a complex system influenced by many factors, including company performance, market trends, and public sentiment. With the rise of social media platforms like Reddit, investor sentiment has become a significant driver of stock price movements. Subreddits such as *WallStreetBets* have gained prominence for influencing market trends, making them valuable data sources for financial analysis.

This project bridges the gap between textual sentiment analysis and traditional stock market indicators. By combining Reddit discussions about Tesla and its historical stock price data from Yahoo! Finance, we seek to predict whether Tesla's stock price will rise or fall. The study involves web scraping Reddit posts, preprocessing and cleaning the collected data, and integrating financial data from the Yahoo Finance API. Various machine learning algorithms are implemented to classify stock price movements, with performance evaluated using metrics like *accuracy, precision, recall,* and *F1-score*.

The results highlight the feasibility of leveraging social media data for stock market predictions. *This project contributes to the growing body of research on financial analytics and sentiment-based stock price prediction by focusing on Tesla stock as a case study.*

2. Objective

This project explores integrating social sentiment analysis and historical financial data to predict stock price movements. By leveraging discussions on Reddit subreddits such as *stocks*, *WallStreetBets*, and *Tesla*, alongside Tesla's open and close prices from Yahoo! Finance, the study aims to classify stock price trends effectively. Advanced preprocessing techniques ensure data quality, while machine learning models like Logistic Regression, SVM, Random Forest, and Naive Bayes are employed to uncover patterns linking sentiment with market behaviour. With an emphasis on practical applications, the project demonstrates the potential of combining unconventional data sources with traditional financial indicators to enhance predictive capabilities in the stock market.

3. Methodology

- Data Collection
    - Web Scraping: Extracted data from Reddit using Python, targeting the keyword "Tesla" in subreddits such as *stocks*, *investing, finance,*

WallStreetBets, *StockMarket, options, financialindependence, cryptocurrency, Daytrading* and *economy*.
    - Columns Collected: *created_at*, *title*, *body*, *num_comments*, and *upvote_ratio*.
- Data Integration
    - Imported financial data using the yfinance library, including *open price* and *close price*.
    - Created a new *target column* by subtracting the open price from the close price. Added a new column, Final_Target and marked it as 1 if the difference exceeded 0.4 and 0 otherwise.
- Data Preprocessing
    - Cleaned textual data from the *body* column.
    - Removed missing values, handled outliers, and normalised features where necessary.
- Exploratory Data Analysis
    - Analysed trends in Reddit data (e.g., comment frequency, sentiment) and stock prices.
- Feature Engineering
    - Merged Reddit sentiment data with stock price data.
    - Generated additional features from stock price trends.
- Model Implementation
  Applied a range of machine learning models to classify the target variable:
    - Logistic Regression
    - SVM
    - Passive Aggressive Classifier
    - Random Forest Classifier
    - Decision Tree Classifier
    - Naive Bayes (MultinomialNB)
- Evaluation Metrics
    - Models were evaluated using accuracy, precision, recall, and F1 score to identify the best-performing classifier.

3.1. Dataset

The project utilised two primary datasets:

1. Reddit Data
    - Source: We obtained Reddit API credentials (client ID, client secret, and user agent) to access Reddit data. Then, we used the Python web

scraping package PRAW (Python Reddit API Wrapper) to download datasets from different subreddits.
    - Subreddits: We collected the data from the following subreddits: *stocks*, *investing, finance, WallStreetBets*, *StockMarket, options, financialindependence, cryptocurrency, Daytrading* and *economy*.
    - Features: We collected the data related to the following attributes of the Reddit posts
        - created_at: Timestamp indicating when the post was created.
        - title: Title of the post.
        - body: Content of the post.
        - num_comments: Number of comments on the post.
        - upvote_ratio: Ratio of upvotes to total votes on the post.
        - subreddit: The subreddit from which each post was extracted.
    - Number of Posts: We collected 805 posts for training our models.
2. Stock Price Data
    - Source: Downloaded the data using the yfinance library of Python.
    - Features:
        - Open Price: Opening price of Tesla stock for the day.
        - Close Price: Closing price of Tesla stock for the day.

3.2. Dataset Preprocessing

The data underwent several cleaning and preprocessing steps to ensure it was in a suitable format for machine learning models. The following steps were performed:

1. **Removing Invalid Entries**

   Rows with the body column containing "[No text content, link post]" were removed. These rows were irrelevant for analysis as they did not contain any text.

2. **Filtering Out Small Price Movements**

   Rows were filtered out where the absolute difference between the *close and open prices* was less than 3. *This was done to focus on significant price movements, ensuring that only entries with a notable impact on prices were considered for analysis.*

3. **Target Column Creation**

A **Target** column was created based on whether the difference between the *close and open prices* was positive. A 1 was assigned for a positive price difference, and a 0 for a negative or zero price difference. This binary classification was used as the dependent variable for the machine learning models.

4. **Text Cleaning**

   The body column containing the Reddit posts' content was cleaned to remove unwanted characters such as mentions (@), URLs, and non-alphanumeric characters. This step helped to standardise the text data for further processing.

5. **Tokenisation and Lemmatization**

   The text was tokenised into individual words, and stop words were removed. Additionally, lemmatisation was applied to reduce words to their base form. This step ensured that similar words were treated as a single entity, enhancing the model's ability to learn relevant patterns in the data.

These preprocessing steps helped ensure that the dataset was clean, consistent, and suitable for machine learning model training, which improved both the data quality and the models' performance.

3.3 Exploratory Data Analysis (EDA)

Text Cleaning and Preprocessing:

We preprocessed the text data to prepare it for analysis. For this, we did the following steps:

1. **Tokenisation:** Splitting the text into individual words.
2. **Lowercasing:** Converting all text to lowercase for uniformity.
3. **Stopword Removal:** Eliminating common words like "the," "and," and "is" that doesn't carry significant meaning.
4. **Removing Non-Alphabetical Characters:** Excluding numbers, symbols, and special characters to focus solely on words.
5. **Lemmatisation:** Reducing words to their root forms (e.g., "running" to "run") to group similar terms.

This cleaning process ensured that only meaningful and relevant words were retained for further analysis.

Visualising Word Frequency:

To gain insights into the dataset, word clouds were created for the most frequent words associated with **positive** and **negative** sentiments:

- **Positive Word Cloud:** Highlights the most common words used in positively labelled texts, revealing key themes and topics associated with positive sentiment.
- **Negative Word Cloud:** Displays the most frequent words in negatively labelled texts, helping identify recurring patterns or concerns.

Findings:

1. The positive word cloud showcased terms that reflect favourable opinions or praise.
2. The negative word cloud highlighted words related to dissatisfaction or criticism.
3. These visualisations provided an intuitive way to understand the content and sentiment distribution within the dataset.

3.4 Models

In this project, several machine learning models were implemented to classify stock price movements effectively. Each model was selected for its unique strengths in handling classification problems, and their configurations were optimised for this specific dataset. Below are the models used, along with their definitions:

1. **Logistic Regression**

   A statistical model that uses a logistic function to model the probability of a particular class or event. It is widely used for binary classification problems and assumes a linear relationship between the features and the log odds of the target variable.

2. **Support Vector Machine (SVM)**

   A supervised learning algorithm that finds the hyperplane which best separates data into classes. Different kernels were used:

   - **Linear Kernel**: Assumes that the data is linearly separable.
   - **RBF Kernel**: A non-linear kernel that maps input features into higher-dimensional spaces for better separation.
   - **Polynomial Kernel**: Extends the linear SVM by applying a polynomial transformation to the input data.

3. **Passive Aggressive Classifier**

   A linear model designed for large-scale learning. It updates its model only on misclassified data points, making it efficient for streaming data or large datasets.

4. **Random Forest Classifier**

   An ensemble learning method that constructs multiple decision trees during training and outputs the mode of their predictions. It is robust to overfitting and works well with non-linear relationships in data.

5. **Decision Tree Classifier**

   A simple, interpretable algorithm that splits the dataset based on feature values to make predictions. It follows a tree-like structure to divide data into homogeneous subsets.

6. **Naive Bayes Classifier**

   A probabilistic classifier based on Bayes' theorem, assuming that features are conditionally independent. It is particularly effective for text classification tasks.

Each model brought unique advantages to the project, enabling a comprehensive dataset evaluation and helping identify the best-performing classifier for stock price prediction.

4. Results

- The Logistic Regression model performed the best overall, achieving the highest accuracy of 81%.
- SVM with Linear Kernel was followed closely, with an accuracy of 78%, demonstrating a balanced trade-off between precision and recall.
- Models such as SVM with RBF Kernel, Random Forest, and Naive Bayes achieved moderate accuracy but varied performance for specific classes.
- SVM with Polynomial Kernel and Decision Tree Classifier showed relatively lower accuracy, indicating potential dataset or model configuration limitations.

This detailed performance breakdown provides insights into each model's suitability for the stock price prediction task.

5. Conclusion

This project successfully integrates social media sentiment analysis and historical financial data to predict stock price movements. By leveraging Reddit discussions and applying rigorous data preprocessing techniques, the study creates a robust dataset for machine learning models. *Among the eight models tested, Logistic Regression emerged as the most effective, achieving an accuracy of 81%.*

The findings demonstrate that combining public sentiment from social media with traditional financial indicators can improve stock price prediction models. While the results are promising, they also highlight challenges such as the need for further refinement of text processing techniques and the potential for bias in social media data.

Future research can expand on this work by exploring additional data sources, enhancing feature engineering, and experimenting with advanced models such as neural networks. Integrating real-time sentiment analysis and predictive modelling also opens new opportunities for applications in financial markets, including trading strategies and investment decision-making tools.