**Finance Club Project – Sentiment Analysis**

Name**:** Archit Saraogi Branch: BS MS Physics Enrollment Number: 23324002

Year: Second Year Email: [archit\_s@ph.iitr.ac.in](mailto:archit_s@ph.iitr.ac.in)

Code Explanation

**Sentiment Analysis**:

* Sentiment analysis is a way to analyze the emotions or opinions expressed in text data. In this code, we're using a tool called Vader (Valence Aware Dictionary and sEntiment Reasoner) to analyze the sentiment (positive, negative, neutral) of news headlines.
* We start by importing necessary libraries and downloading a lexicon (a dictionary of words with associated sentiment scores) for sentiment analysis.
* The function analyze\_sentiments takes two parameters: input\_csv (the file containing news headlines) and output\_csv (the file where we'll save the sentiment analysis results).

**Processing News Headlines**:

* We read the input CSV file containing news headlines and make sure all entries in the 'headline' column are strings (text). Any missing values (NaN) in this column are filled with empty strings.
* We then initialize the sentiment intensity analyzer (Vader) that will help us determine the sentiment of each headline.

**Sentiment Analysis Process**:

* For each headline in the dataset, we use Vader to analyze its sentiment. Vader gives us scores for positivity, negativity, neutrality, and a compound score (an overall sentiment score).
* These sentiment scores are stored in a list called sentiment\_scores\_list.

**Storing Sentiment Analysis Results**:

* After analyzing all headlines, we convert the list of sentiment scores into a DataFrame (a table-like data structure).
* We combine this sentiment scores DataFrame with the original DataFrame containing headlines to create a new DataFrame (result\_df) with sentiment analysis results added.

**Saving Sentiment Analysis Results**:

* Finally, we save the updated DataFrame (result\_df) to a new CSV file specified by output\_csv.

**Downloading Financial Data**:

* We then move on to another function (download\_yfinance\_data) to download financial data from Yahoo Finance.
* This function takes parameters such as ticker (the stock ticker symbol), start\_date, end\_date, and output\_csv (the file where we'll save the financial data).

**Aligning Sentiment with Financial Data**:

* Another function (align\_with\_yfinance) aligns the sentiment analysis results (from the first step) with the financial data downloaded.
* It ensures that the sentiment analysis dates match the dates in the financial data, filtering out any mismatched or irrelevant data.

**Machine Learning Model :**

* Training an SVM model using the training data for classification tasks.
* Training a random forest model to improve prediction accuracy and reduce overfitting.
* Building and training a neural network using TensorFlow and Keras for complex pattern learning.
* Making predictions using these trained models on the testing data for evaluation.
* Comparing the performance of different algorithms using metrics like accuracy, precision, recall, and F1 score to determine the most suitable model for the task.

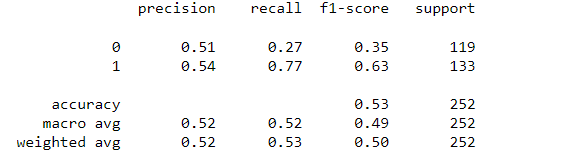
**Backtesting and Calculating Metrics :**

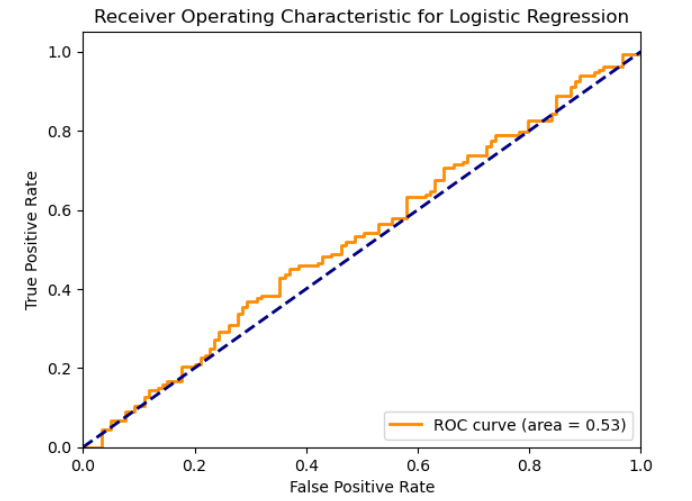
* **Initialization**: Initializes variables like position\_open, position\_price, n\_shares, capital, and investment\_log.
* **Signal Generation**: Generates trading signals based on the indicator column provided (1 for buy signal, 0 for sell signal).
* **Trading Simulation**: Iterates over the DataFrame to simulate buying and selling based on the signals and stop loss criteria.
* **Logging**: Logs each trade with details like 'Buy/Sell', date, number of shares, and price.
* **Final Calculation**: Calculates the final capital after all trades are simulated.
* **Visualization**: Plots the closing prices, trading signals, buy, and sell points for visualization.
* **Return**: Returns the final capital after trading simulation.

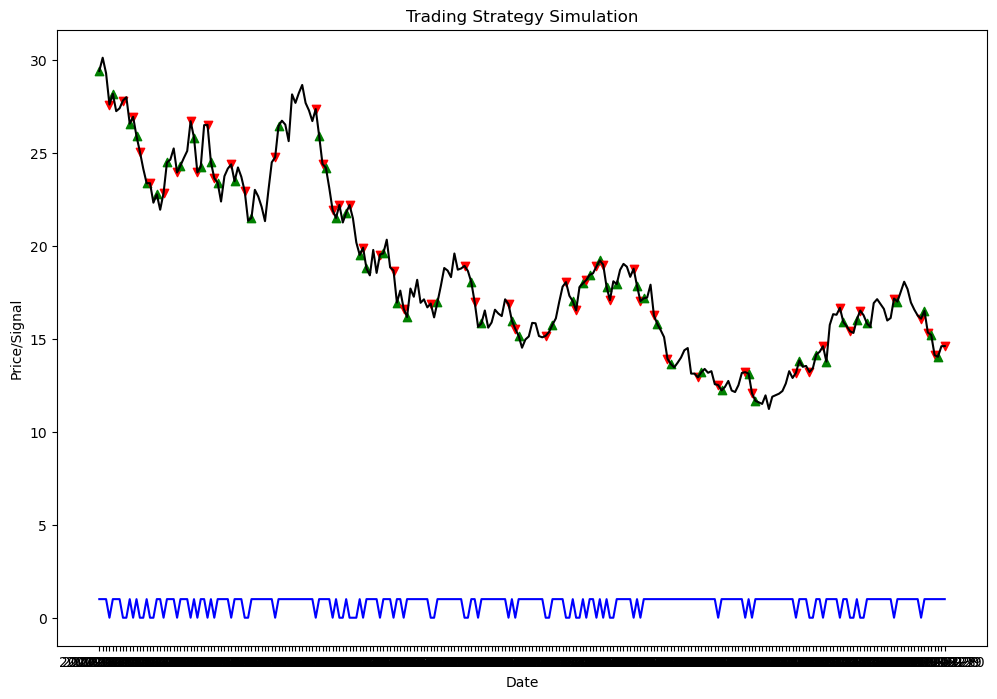
Comparision between different ML models and their results:

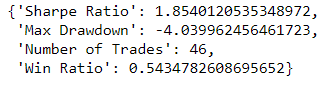
**Logistic Regression:**

**Annual Returns=** 0.1782



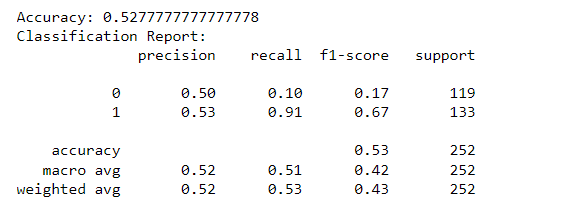


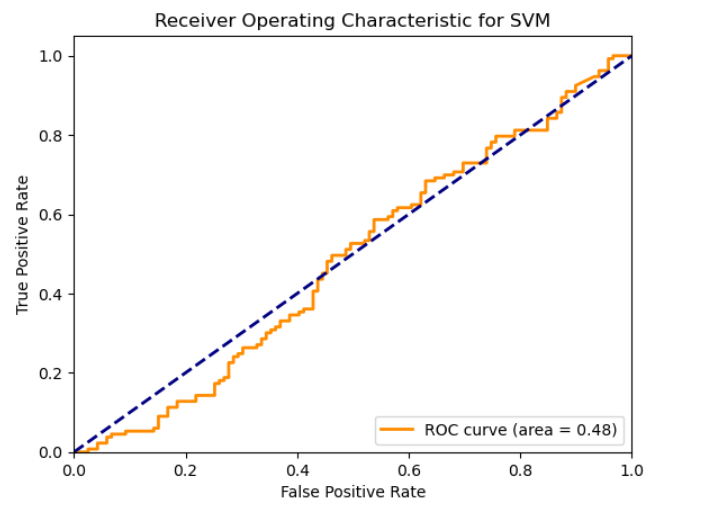


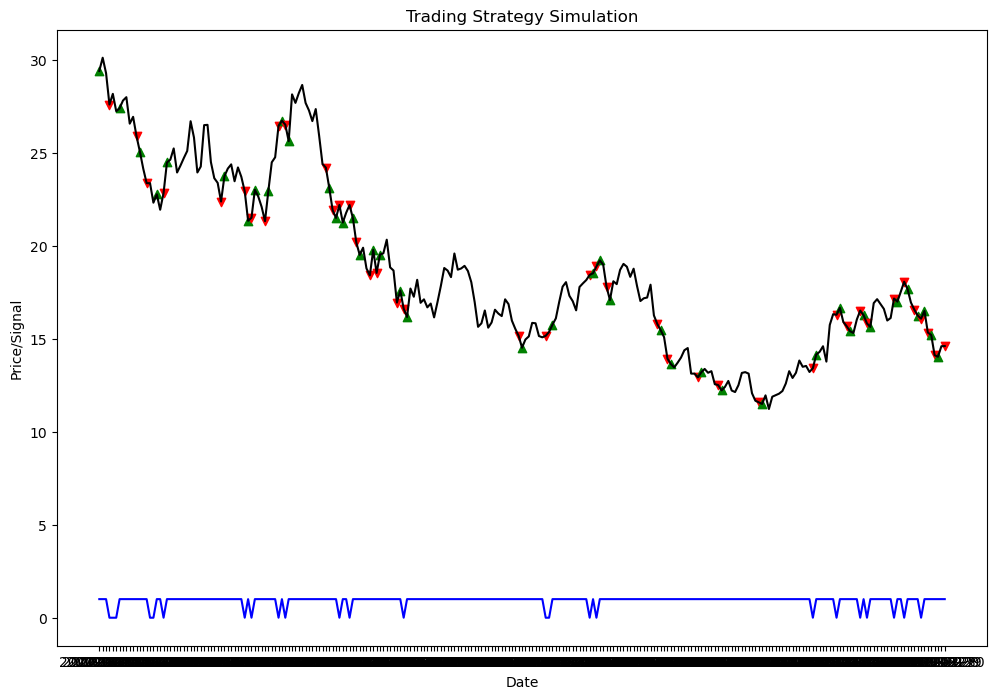


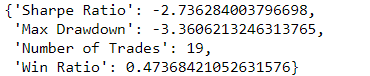
**Support Vector Machines:**

**Annual returns: -0.49**



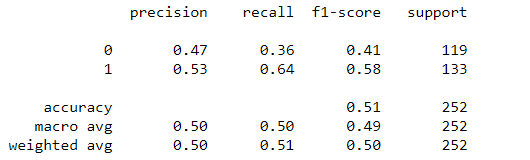


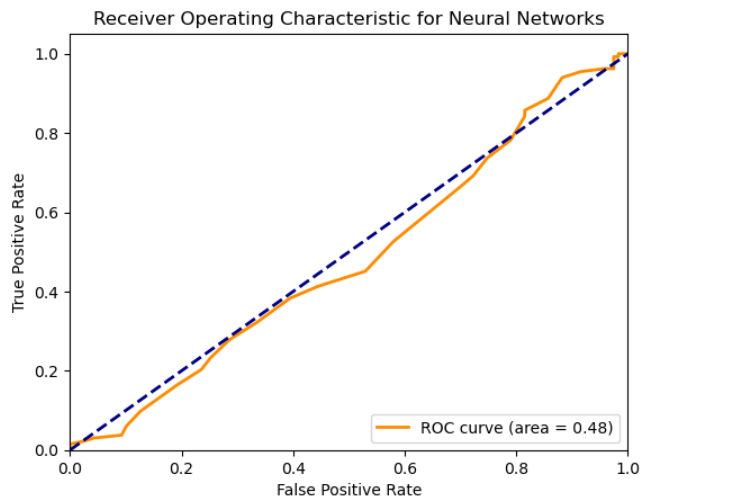




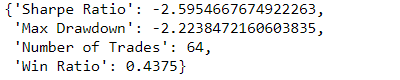
**Neural Networks:**

**Annual returns: -0.45**

****

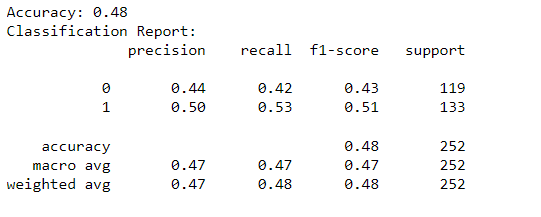


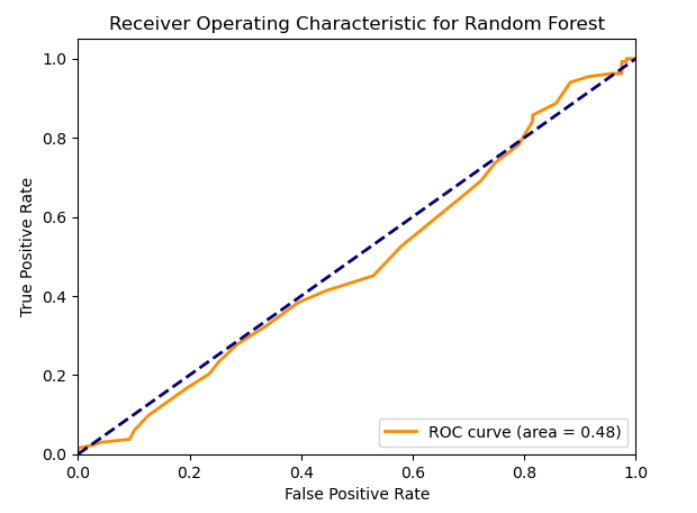


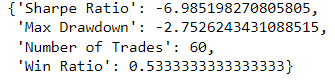


**Random Forests:**

**Annual returns: +0.892**

****

****



Conclusion:

Judging on return and accuracy the preferred model of sentiment analysis should be logistic regression.

Although sentiment analysis in itself isn’t fully accurate combining it with other indicators might produce fruitful results.