TradeRight

Final Report

Team Alpine (Team #100)

Introduction

"The average individual investor underperforms a market index by 1.5% per year" is a common statistic available on the internet

TradeRight provides individual investors a shared platform with features to come up with a sound strategy to develop optimal stock portfolios. It analyzes and integrates news and financial data related to stocks and applies machine learning techniques to generate corresponding buy/sell signals. In addition, the platform allows investors to share their investment methodology with other users of this platform. This is favorable because it encourages collaborative building of investment strategies and because unlike wealth managers, the platform would not charge a handsome fee for such interactions.

Problem definition

In the current scenario without an exhaustive market research, it is difficult for an investor to extract gains from stock market trading. The plethora of information responsible for driving the market sentiment makes things more difficult. Moreover, personal research on stock picks often devolves into gut/emotional decisions. Added to that, the fact that wealth managers today charge an exorbitant fee makes it impossible for an average investor to participate in stock market trading. This brings a need for a cost-effective platform that integrates all information driving a stock's price and applies accurate algorithms, thereby predicting a stock's performance and guiding an investor.

The fact that the prevailing business model does not incentivize informed investors and institutional traders to share their insight with public is another concern due to which there is a need for a platform that boosts shared building of investment strategies.

TradeRight mines stock related news and financial data from reputable websites and applies the latest research in language models and machine learning techniques to provide an investor with a clear signal to buy or sell a stock through an interactive user experience. It provides a platform where informed investors can share their portfolio development strategies with fellow investors.

Literature Survey

Textual information about various stocks is available in different forms on the world wide web. It could be in terms of web search queries [1], twitter feeds [5], Facebook posts [4] and news articles [2][3][13] among others. All these sources have been proven useful in reliably predicting the stock price movement.

Many authors [7][14][17] have combined financial text data along with the technical indicators in some cases to improve the prediction of stock price movement the next day. While some papers argue text information alone can be the most important factor and achieve better performance by just using textual indicators compared to using both technical and textual features [14] and call out efficient representation of text is the key to achieve success.

In this regard surveyed papers propose different approaches including deep learning [8][13][14][15][17], SVM [13], reinforcement learning [10] to deal with different indicators. Among the explored papers deep learning consistently performed better than other techniques and were used in the form of RNN [15], CNN [17], ANN [13][8]. This also calls out for

the necessity of better representation of textual features or events in text to better capture the index movement [13].

Since textual features are a critical thing as evidenced from survey, latest language models (LM) such as BERT [18], RoBerta[19] seem obvious to try out for these applications. However, there is very sparse literature in this regard, and we would want to contribute to this aspect and empirically verify if we can better other approaches using state of the art LM.

Proposed Method

Team Alpine has adapted Agile – Sprint based approach. Artefacts to create the application are developed incrementally and integrated together when development is complete.

Architecture

TradeRight application follows standard multilayer architecture as it is expected to scale independently once it starts functioning at its intended scale. The architecture diagram which forms the basis of proposed method is captured in figure 1 below.

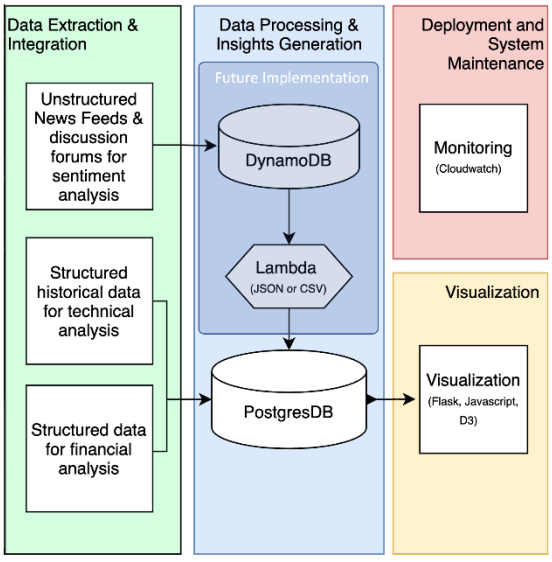

Figure 1 - Proposed Architecture

Note – In future version of the application, the data will be stored in DynamoDB. This data will be processed using Lambda functions to extract the insights into PostgresDB.

The architecture is divided into four different sections as follows:

Data Extraction & Integration

Data extraction is done using multiple APIs. We are collecting three types of data for different stocks:

Financial data: Yahoo financial data services are used to extract companies 10K and 10Q data. Once the application matures, scalable data sources can be explored.

Technical data: Yahoo is used to extract from free data source. Same APIs/Services could be expanded to extract financial data at scale.

Social and news data: For this phase of the application, News API and reddit is used to extract ticker related news. We also tried getting the data using Bing APIs, but the data was not well structured, and we did not have a reliable way to associate it with the tickers.

Data statistics

The dataset comprises of technical and financial data related to 6000+ stocks which translates to about 1.2 million rows and 150 MB of data on the disk. About 25K news articles were collected for these stocks and the corresponding size for them was 20 MB.

Data Processing & Insight Generation

This is the heart of the system. Post data collection, different processing techniques are applied, and the data is cleaned up, outliers are removed, formatting is corrected, character encodings are applied. We evaluate different machine learning models for their ability to predict buy/sell signals. We combine news related sentiment as well as a stock's technical indicators like simple moving average, volume, dividends and stock splits as our feature set and train different models and compare their results. We

evaluate VADER to calculate sentiment score for news articles and RoBERTa which is a state-of-the-art Language Model. The combined feature set is used to train different models like Multi-layer perceptron, Random Forest Classifier, SVM, Boosting etc.

Visualization

This layer is handled using Flask, JavaScript and D3 technologies. For future scalability, an AngularJS or React application can be developed.

Deployment & System Maintenance

The application is deployed on the cloud platform Heroku. Heroku was chosen because of its user-friendly, flexible interface for small-scale projects. For any modifications, the Git repo will be updated and then pushed up the corresponding Heroku remote repo.

High Level Design

Following steps outline the high-level workflow and dataflow:

Leverage multiple data sources for the purpose of predicting the sentiment of the stock news and provide a buy/sell signal.

Extract stock price movement data from multiple sources.

Combine the signals extracted from step 1 & 2 above using data science algorithms to generate final signals. We evaluated several algorithms and observed the best results using RoBERTa.

Relay this signal to the application users through visualization.

Combine the signal and portfolio management features into an interactive user interface.

Generating Signals

We make use of multiple approaches in determining the sentiment of text articles to determine buy/sell signals as can be seen in Fig 2 in the appendix.

VADER

VADER [20] makes use of sentiment lexicon and rules to find out the sentiment of sentences and is mainly tuned for text appearing in social websites. VADER not only provides the classification of the sentence but also provides the degree of sentiment with a score. We initially just use VADER to extract the sentiment of the news articles and find best threshold for classification by using the ROC sweep and use the threshold to report the metrics on held out test set.

VADER + Technical Indicators (TI)

In this approach we use technical indicators along with VADER scores to predict buy/sell label. The technical indicators being used include 2/5 days SMA, volumes and dividend splits. Using these as the features we build multiple classifiers including logistic regression, multi-layer perceptron and ensemble techniques such as random forests, Adaboost classifiers and gradient boosting trees.

RoBERTa

Lastly, we explore latest language models (LM) to see if we can get similar performance by just using the news headlines alone. We make use of RoBERTa an optimised version of BERT. RoBERTa is trained on larger training data, longer sequences and uses dynamic masking pattern unlike BERT which uses static masking pattern. We finetune this LM to classify the news article to indicate buy/sell signal.

Data Model

The signals generated from the above step are ingested into our data model which feeds into our user interface. Figure 1 in the appendix captures the intended data model for the system. Currently, we are using a file based approach, in the future we would like to use this data model.

Landing zone

NewsReel, Sentiments, Technicals and Financials are the landing tables for the data from the internet.

Processing Tables

There are two kind of processing tables here. The aggregates are used to aggregate the data so that consolidated value of portfolio, sentiments around a stock can be given. The _signals, which will clearly outline the BUY/HOLD/SELL signal that the system is proposing.

Transaction records

Tables such as Portfolio, Executed Stock Prices, will hold the data related to actual transactions by the users. Team has not yet spent time studying the governance and regulatory compliance of these.

User Experience

Visualization as outlined in Figure 3. Key elements of the visualization are as follows:

Top Signals

This section clearly outlines the market signals as generated by the application. The users of the application will have a choice to heed or ignore these signals.

Portfolio Value

This section showcases the current value of the portfolio and the user defined target portfolio value.

Top Stories

This section will show important news related to stocks belonging to the portfolio of the users.

Published Concepts

This section consists of portfolio concepts published by other users of the platform along with key features of their portfolios and returns.

Stock Analysis, Market Analysis

The user can select the stock and the index of his choice and look at their performance over time.

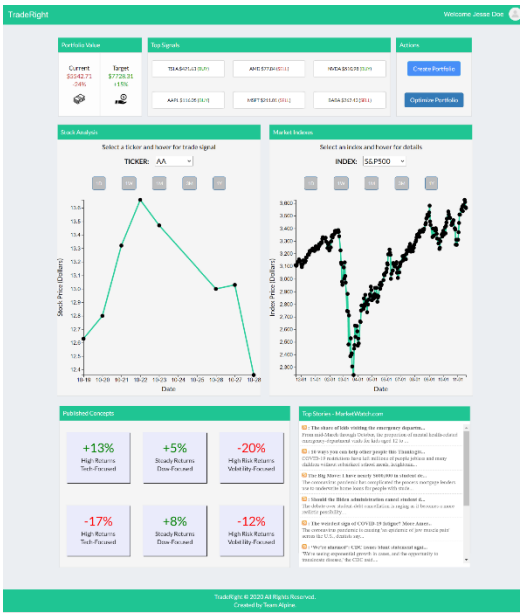

Fig 3: User experience

Intuition

The sentiment of news articles plays major role in people transacting in stock market [2]. While news articles help to a certain degree there are lot of latent factors which determine the movement of stock prices. In our approaches we make use of different NLP techniques which have been shown to perform good for sentiment detection and combine multiple technical indicators to extract the buy/sell label. LM are sparsely used for predicting the stock price movement as evidenced in our survey. Here we make use of RoBERTa LM which has achieved SOTA results and this can help in providing better embeddings for detecting sentiment of news articles. Also combining technical, fundamental and sentimental indicators can help us achieve better results.

Currently there are many intermediaries between investors and the stocks such as brokers, mutual fund managers, mutual fund companies. The expertise is scarce and not every successful investor is working for a financial institute. The intuition behind the concept is to break down unnecessary barriers in financial markets so that investors sitting at the desk can directly build the

optimal portfolio for their needs. The key new concept in the approach is the collaboration, which is backed by data mining and data science, effectively democratizing investment function of the industry.

Experiments/Evaluation

Data Gathering

We explore multiple avenues to gather stock related news, financial and technical data. For technical and financial data, we mainly rely on Yahoo API. For news data we explored Reddit, NEWS and Bing API. While reddit API provided us with lot of textual data related to stock market it is hard to associate it with specific tickers. For example, news articles related to Microsoft can have terms like 'MSFT', 'Microsoft', 'Nadella'. It's hard to find exhaustive list of terms related to a ticker. Similarly, BING API provided data in raw format which could not be associated with ticker in reliable way. Eventually we made use of NEWS API which provides news articles related to specific tickers by taking tickers itself as one of the arguments. Using NEWS API, we gathered news related to various tickers and merged this data with the financial data to create our dataset.

Label Generation

We experimented with two ways of generating labels. First by using simple moving average (SMA). *n*-SMA measures the average of closing value of stock prices for last *n* consecutive days. We assign a label of sell to a particular day *d* if 2-SMA is greater than 5-SMA for day *d* and the opposite for the next day *d+1*. This sell signal captures the profit the person can make by selling before the drop. Similar approach is taken of assigning buy label. But with this approach we were able to label very few days since the above criteria cannot be met always. Later we decided to use this as one of the features for training our models. The thing which worked was to make use of open and close signals to produce labels. If the open is higher than close we would assign a label of sell, so a profit is made, and vice versa. Although this is a straightforward approach, we still wanted to see how it affects our models' performance. Generally, news articles tend to have an immediate effect on the stock's price, and it reflects this label gathering strategy [1][3]. As the project matures, we would like to test more sophisticated ways of assigning labels.

Hyper Parameter tuning

Apart from the baseline VADER experiments we do extensive hyper parameter tuning for rest of the models. For the ensemble models we explore different combination of hyper parameters including number of weak estimators, maximum number of features and depth of weak learners among others.

For RoBERTa as well we explore multiple combination of data and hyperparameters. For the purpose of using RoBERTa for classification we add a linear classification head on top of the embeddings and train RoBERTa. We experiment by training with title alone and title and description of news articles. Adding description degrades the performance of the model indicating description might not contain relevant information and can act as noise. Since RoBERTa is huge in number of parameters and we have comparatively very less data we finetune pretrained RoBERTa using very low learning rate of 2e-5. We also use a validation set to decide on the number of epochs and adjust the learning rate. We stick to 3 epochs and increasing this leads to overfitting. After choosing the best hyperparameters the validation set is added back to the training and the model is trained again with finalized hyperparameters.

Test Bed and Results

For the purpose of evaluation of our techniques we make use of separate unseen test data which is not used for training. Test set is created by temporally splitting the data so we can use the historical news articles to train and predict for the upcoming days. We report the metrics on this

data for the purpose of evaluation. We choose to report accuracy since buying and selling is equally important, and we do not observe any skew in the generated labels. Using these metrics, we would want to conclude which features play a pivotal role in determining the buy/sell signal more accurately and see which of the approaches explored performed the best.

Method	Test Accuracy
VADER (Text Only)	52.89%
Logistic Regression (Text + TI)	54.89%
Multi-Layer Perceptron (Text + TI)	53.91%
Random Forests (Text + TI)	54.27%
AdaBoost (Text + TI)	55.25%
Gradient Boosting Trees (Text + TI)	60.88%
SVM (Text + TI)	57.21%
RoBERTa (Text Only)	**61.81%**

We observe unsupervised VADER to provide an accuracy of 52.89%. Different ensemble and supervised learning approaches improve upon this and provide accuracy in the range 53.91% to 60.88%. Another thing to note is these supervised learning algorithms is that along with sentimental scores we make use of financial data as well and improve upon the baseline. Gradient Boosting trees achieves an accuracy of 60.88% and significantly improves upon the baseline VADER. Adding technical and financial indicators improved the results indicating they are helpful as seen from the accuracies of all models using text and technical indicators. RoBERTa trained with news headlines alone achieves a test score of 61.81%. RoBERTa however outperforms all approaches using text alone. This could be attributed to the exceptionally good representation capability of the model.

We use the algorithms described in multiple papers including SVM, multi-layer perceptron for the purpose of predicting buy/sell signal. Since we do not have access to the data described in the papers we surveyed, we compare the proposed approaches with our approaches on the dataset which we have created. We observe Gradient Boosting trees which make use of both technical indicators and news articles to outperform the SVM and multi-layer perceptron model proposed in multiple papers. Similarly, RoBERTa outperforms all the explored approaches as indicated in results table and we believe using latest language models for financial applications has tremendous potential.

Conclusion and Future Work

The analysis of news articles using RoBERTa has provided us satisfactory results that are well above the baselines and will prove useful in predicting a stock's performance. The integration of these signals with the interactive visualization we have created provides a brief, interactive and personalized view extracted from the abundant information we have analyzed. Equipped with these tools, an investor will be well-informed and have an upper hand when taking decisions about his portfolio. Given the LM are very good at representing textual data, we would want to combine the sentiment embeddings along with technical indicators in the future to see if we can improve our results further.

Distribution of efforts

Here are how the activities are divided among our team members:

1. **Data extraction and processing** – Sahil (Lead) and Eshwar
2. **Machine Learning models** – Eshwar (Lead) and Sahil
3. **Data model** – Ganesh (Lead) and Amrith
4. **Frontend development** – Don (Lead) and Aarushi
5. **Integration** – Amrith (Lead), Ganesh and Sahil
6. **Documentation, Poster and Report** – Aarushi (Lead) and Eshwar

The team also is cross trained on all the aspects of the project and takes up various activities as required.

Appendix

Data Model

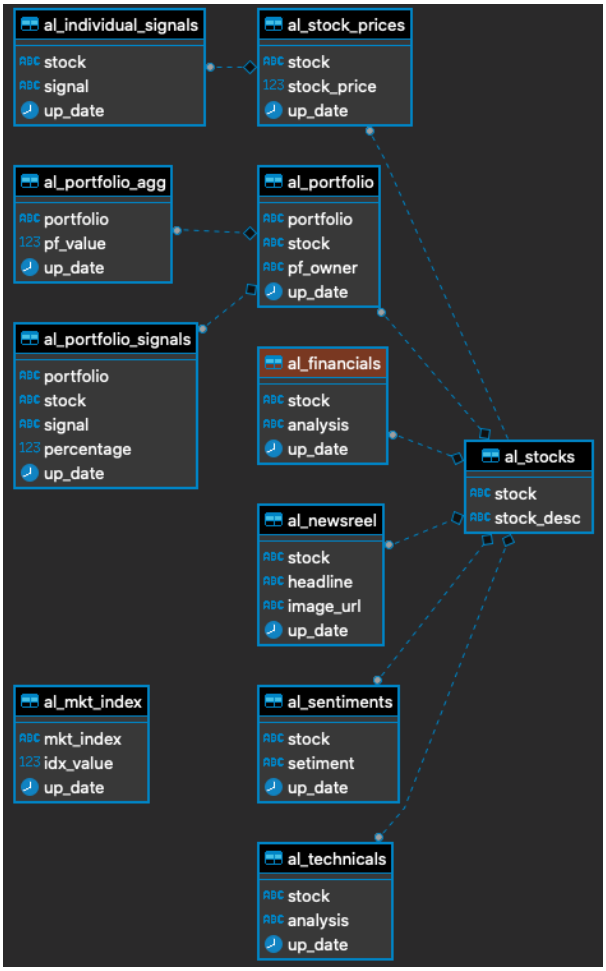

Fig 1 Data Model

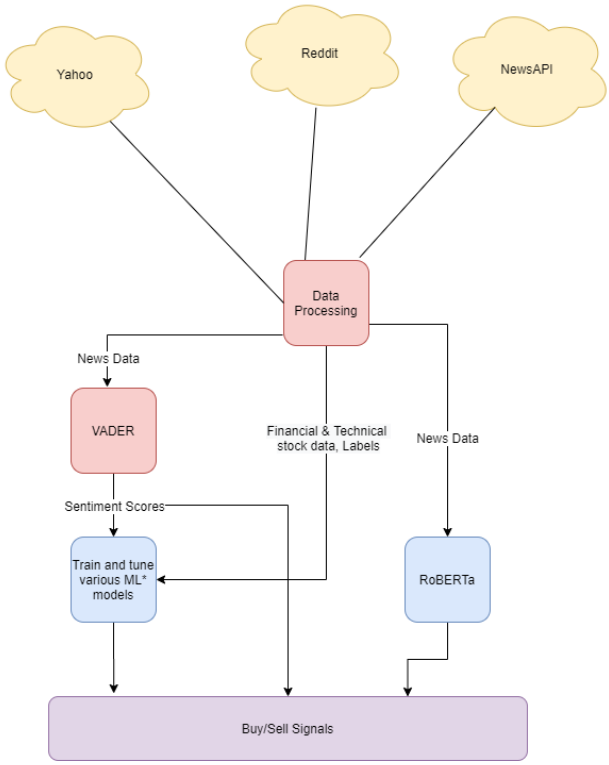

Fig 2: Generating signals

References

[[1] Bordino, Ilaria, Stefano Battiston, Guido Caldarelli, Matthieu Cristelli, Antti Ukkonen, and Ingmar Weber. "Web search queries can predict stock market volumes." PloS one 7, no. 7 (2012): e40014.

[2] Schumaker, Robert P., and Hsinchun Chen. "Textual analysis of stock market prediction using breaking financial news: The AZFin text system." ACM Transactions on Information Systems (TOIS) 27, no. 2 (2009): 1-19.

[3] Khedr, Ayman E., and Nagwa Yaseen. "Predicting stock market behavior using data mining technique and news sentiment analysis." International Journal of Intelligent Systems and Applications 9, no. 7 (2017): 22.

[4] Karabulut, Yigitcan. "Can facebook predict stock market activity?." In AFA 2013 San Diego Meetings Paper. 2013.

[5] Souza, Thársis Tuani Pinto, Olga Kolchyna, Philip C. Treleaven, and Tomaso Aste. "Twitter sentiment analysis applied to finance: A case study in the retail industry." arXiv preprint arXiv:1507.00784 (2015).

[6] Tetlock, Paul C. "Giving content to investor sentiment: The role of media in the stock market." The Journal of finance 62, no. 3 (2007): 1139-1168.

[7] Picasso, Andrea, Simone Merello, Yukun Ma, Luca Oneto, and Erik Cambria. "Technical analysis and sentiment embeddings for market trend prediction." Expert Systems with Applications 135 (2019): 60-70.

[8] Qiu, Mingyue, and Yu Song. "Predicting the direction of stock market index movement using an optimized artificial neural network model." PloS one 11, no. 5 (2016): e0155133.

[9] Kirlić, Ajla, Zeynep Orhan, Aldin Hasovic, and Merve Kevser-Gokgol. "Stock market prediction using Twitter sentiment analysis." Invention Journal of Research Technology in Engineering & Management (IJRTEM) 2, no. 1 (2018): 01-04.

[10] Pendharkar, Parag C., and Patrick Cusatis. "Trading financial indices with reinforcement learning agents." Expert Systems with Applications 103 (2018): 1-13.

[[11] Kearney, Colm, and Sha Liu. "Textual sentiment in finance: A survey of methods and models." International Review of Financial Analysis 33 (2014): 171-185.

[12] Guo, Li, Feng Shi, and Jun Tu. "Textual analysis and machine leaning: Crack unstructured data in finance and accounting." The Journal of Finance and Data Science 2, no. 3 (2016): 153-170.

[13] Ding, Xiao, Yue Zhang, Ting Liu, and Junwen Duan. "Using structured events to predict stock price movement: An empirical investigation." In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1415-1425. 2014.

[14] Vargas, Manuel R., Beatriz SLP De Lima, and Alexandre G. Evsukoff. "Deep learning for stock market prediction from financial news articles." In 2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), pp. 60-65. IEEE, 2017.

[15] dos Santos Pinheiro, Leonardo, and Mark Dras. "Stock market prediction with deep learning: A character-based neural language model for event-based trading." In Proceedings of the Australasian Language Technology Association Workshop 2017, pp. 6-15. 2017.

[16] Zuo, Y., & Kita, E. (2012). Up/Down Analysis of Stock Index by Using Bayesian Network. Engineering Management Research, 1(2). doi:10.5539/emr.v1n2p46

[17] Gudelek, M. U., Boluk, S. A., & Ozbayoglu, A. M. (2017). A deep learning based stock trading model with 2-D CNN trend detection. 2017 IEEE Symposium Series on Computational Intelligence (SSCI). doi:10.1109/ssci.2017.8285188

[18] Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[19] Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. "Roberta: A robustly optimized bert pretraining approach." arXiv prepri nt arXiv:1907.11692 (2019).

[20] Gilbert, C. H. E., and Erric Hutto. "Vader: A parsimonious rule-based model for sentiment analysis of social media text." In Eighth International Conference on Weblogs and Social

Media (ICWSM-14). Available at (20/04/16)
http://comp. social. gatech. edu/papers/icwsm14.
vader. hutto. pdf, vol. 81, p. 82. 2014.