Topic Modeling and Sentiment Analysis based Tesla Stock price movement prediction

A project for the course Computational Social Science offered in WS23/24

Institute for Statistics
Ludwig Maximilian University of Munich
Pankhil Gawade
Matrikelnummer: 12654141

I Abstract

The aim is to compare the two approaches, i.e., Sentiment Analysis and Topic Modelling, for Tesla Stock price movement prediction.For Sentiment Analysis, I will use the FinBERT. It is a Bidirectional Encoder Representation of a Transformer (BERT) pre-trained on a large corpus of financial data. In contrast, in the case of Topic modeling, I am using gensim library which is general general-purpose library for Topic modeling and NLP. I will use the Hierarchical Dirichlet process (HDP) and Latent Dirichlet allocation (LDA) model from gensim. While embedding the data is converted to a word vector by using Google's word2vec(it takes text corpus as input and outputs word vector.), and classification is performed later to predict the final movement of a stock based on topic features for this purpose I am using a linear model, random forest and XGboost and ANN. I am collecting all the news titles from the media cloud about Tesla for the last 6 months i.e. Aug 23 - Jan 24.Data for the stock price of TSLA (Tesla) is taken from scraping the Yahoo finance website. It contains [Open, High, Low, Close, Volume, and adj Close] for each day of 6 months
Github : https://github.com/Pankhil07/CSS_PROJECT

Table of Contents

II Introduction and Motivation

The interest of humanity in understanding how the stock market behaves is not very recent it dates back to the time of Sir Issac Newton, who invested a significant amount of time in understanding the markets and decoding human behavior in general. Although he was the father of modern physics, Newton was out of luck when it came to incorporating irrational human behavior into his mathematical equations and failed badly in investing and had to incur many losses, his exact quote which states "I can calculate the movement of the stars, but not the madness of men" indicates how complex and intricate it is to understand the market and its behavior in general.

As time passed we came up with many modern theories for economics and finance and thus for the markets as well. The biggest disadvantage of the 19th century was the lack of data collection sources and computing power. As the field of computer science branched out of mathematics thanks to the contributions of people like Von Neuman and Alan Turing there came the possibility of computing and data collection. With recent developments in the fields of AI and Finance, one can say that we are in a much better position than Newton was some hundred years ago .....or are we? This is the question we try to delve deep into in this paper and understand further about our current position.

My main motivation for this topic is the same as what Newton had in the 16th century, which is to understand irrational human behavior. As this paper only touched on the upper fabric of the surface of the vast sea of research and methods to understand markets, it provided me with a starting point to dive deep into this sea and explore further areas and research methods in this field. The course content and exercise gave me a deeper understanding of recent methods in the field that I am using to conduct this analysis.

Predicting the Stock market through Sentiment analysis and Topic modeling is not a new area of research, previously there has been some research done in the field, individually i.e by only using the Sentiment Analysis and Topic modeling, [1] talks about the predicting the stock price using sentiment analysis, their end goal is to use the large time series data corpus for this pur-

pose and feed it into the Deep learning algorithm to get a better result. [2] mainly also includes Topic modeling in the sentient analysis, they do it by introducing their topic model, Topic Sentiment Latent Dirichlet Allocation (TSLDA), which can capture the topic and sentiment simultaneously, the analysis is done for multiple stocks and is compared to various state of the art methods.

The further sections will give you detailed insights into the analysis conducted in this project. Section 3 starts with talking about possible data collection options and ideas then I mention the data collection sources that I used and the limitations to that as well. I will then explain the preprocessing steps undertaken to ensure that the data when it is fed into the model is right format. Then section 4 talks about the methods used, in this case, I discuss in detail the models and procedures I have used in both cases Topic modeling and Sentiment Analysis along with their limitations. Section 5 starts with the results of the analysis conducted and then in detailed discussion about the results.

III Data collection and preprocessing

Collecting and preprocessing the data is the foundation for any machine and deep learning model. The course introduces a whole range of real data collection schemes that exist and gives a deep dive into each of them, right from the APIs for social media websites such as Twitter/X, Reddit, Youtube, etc apart from that it also introduces the ways collected data from websites such as Google trends, maps and even LLMs as ChatGPT.

In this project, there are two main data collection sources used one being Web scraping yahoo finance site for the stock price of tesla and the other being Mediacloud for news headlines of tesla. The initial decision was to use API for Twitter/X to get the news headlines for Tesla and content related to it eg ElonMusk, SpaceX, etc but after the acquisition of Twitter/X by Elon Musk there have been some policy changes regarding the free availability of Twitter data through APIs and the data is not available for free anymore. So as a countermeasure to the above problem, I diverted towards one of the methods introduced in the lectures, MediaCloud. Even with the MediaCloud data and recent changes in them, I was not able to access the data older than 6 months for free. During the data collection for the stock price, because I had a limit in MediaCloud for 6 months, I had to choose stock price data for 6 months to match the dates and accurate predictions.

While using MediaCloud, I used search phrases such as TESLA and ELON MUSK and for the collections and sources, I used United States- National, The dates used were from Aug 2023 to Jan 2024. The ideal raw data that comes out after this has the following columns id, indexed date, language, media name, media URL, publish date, title, and URL, Now the task was to filter out the relevant columns out of these columns so my initial preprocessing step was to check for any outliers, missing values then once that is dealt with I checked the data types for each column and made sure they are consistent. Then I removed the meaningless column and kept only the ID, published date, and title as they will be used for sentiment analysis and topic modeling purposes. Since the MediaCloud may contain multiple titles from the same dates the additional preprocessing step was to merge all the titles that belonged to the same date.

While scraping the data from Yahoo Finance I got data with the following columns [Open, High, Low, Close, Adj Close], the initial preprocessing steps such as checking for missing data, null values, and datatypes were done. Additionally, merging both Mediacloud and stock price data using the date as a common column was also done to conduct the analysis fairly.

IV Methods used

This course introduces a large number of methods that can be used to conduct this specific analysis. Methods such as LDA (Latent Dirichlet allocation) then HDP (Hierarchical Dirichlet process) are being used from the gensim library and then I tried exploring various pre-trained models for both topic modeling and sentiment analysis. Initially, I tried exploring the BERT pre-trained model but it was trained on a large corpus of unrelated data which might also include financial data, as a measure to get a more accurate prediction I tried to use a model from hugging face which was trained only on large financial data corpus FINBERT.

FINBERT: Financial sentiment analysis by itself is a very challenging task because of the lack of availability of labeled news data and historical news data in particular. With the recent development in the areas of Large Language Models and other forms of Neural Network Architectures, [3] dives deep into language models based on Bidirectional Encoder Representation of Transformer(BERT) for financial NLP tasks. By doing this they were able to achieve state-of-the-art results. This further gives a reference as a pre-trained model for future Sentiment Analysis tasks.

I Sentiment Analysis

Sentiment Analysis is one of the most powerful tools when it comes to getting some sentiment out of the data that you have. In this project, I have used sentiment analysis as follows.

- After the initial preprocessing, the input to the model FINBERT is in terms of strings of csv so the column title is singled out and fed into the FINBERT.
- The result of FINBERT is scores between 0 and 1, if it is closer to zero indicating sentiment is negative, and if closer to 1 indicating it is positive. Additionally, for each row, it outputs the sentiment if it's positive negative, or neutral.
- The dataset with a stock price which had columns [Open, High, Low, Close, AdjClose]. To be able to predict the next day's price if it's higher than the previous one or lower, I added a column called AdjCloseNext

which says if the closing value for the next day is higher than the previous day the value of the label is 1 or if lower then 0.

- Then comes the step of merging the two columns concerning the common label date. So now the entire data set containing the title and its sentiments along with stock prices and the additional column we added everything is merged into a single dataset. This explicitly helps with conducting a joint analysis
- Now the task remains is to predict the stock price movement for the next day based on all the data so the price movement upwards is 1 and if downwards then 0, this translates to a classification problem. We consider this problem as a classification task and perform the classification analysis
- All the dataset is divided into train set and test set with 80:20 split. The 4 classification models used are SGD, Xtreme Gradient boosting, Random forest, and Gaussian Process Classifier. Additionally, I have implemented ANN from scratch to see how results for it perform in comparison to classical ML models
- Scores such as Accuracy precision and recall are calculated for each model through classification report and confusion matrix

II Topic modelling

Topic Modelling provides deep insights into the titles of the news, it gives you, for example, the important and relevant topics with knowledge of such topics one can further learn in detail about the title at hand and what are the important factors. Some of the inspiration for the below method was taken from [4] which explores various embedding methods as well in detail. In this project, I conduct the Topic Modelling analysis as follows.

- To get an overview of all the topics that our dataset contains, initially I will be conducting a Topic Modelling analysis using the BERTopic the result of this can shown in the figures below.

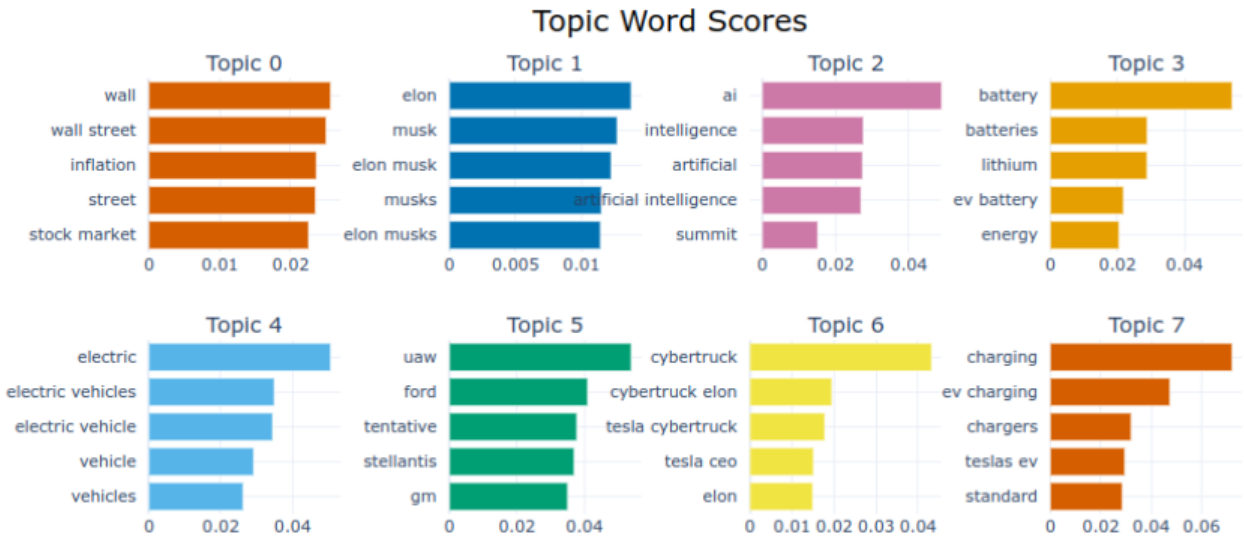
- As a measure to add more features to our dataset and to further have a better generalization performance of the model I am using topic modeling to add additional features. To get the number of topics in the document

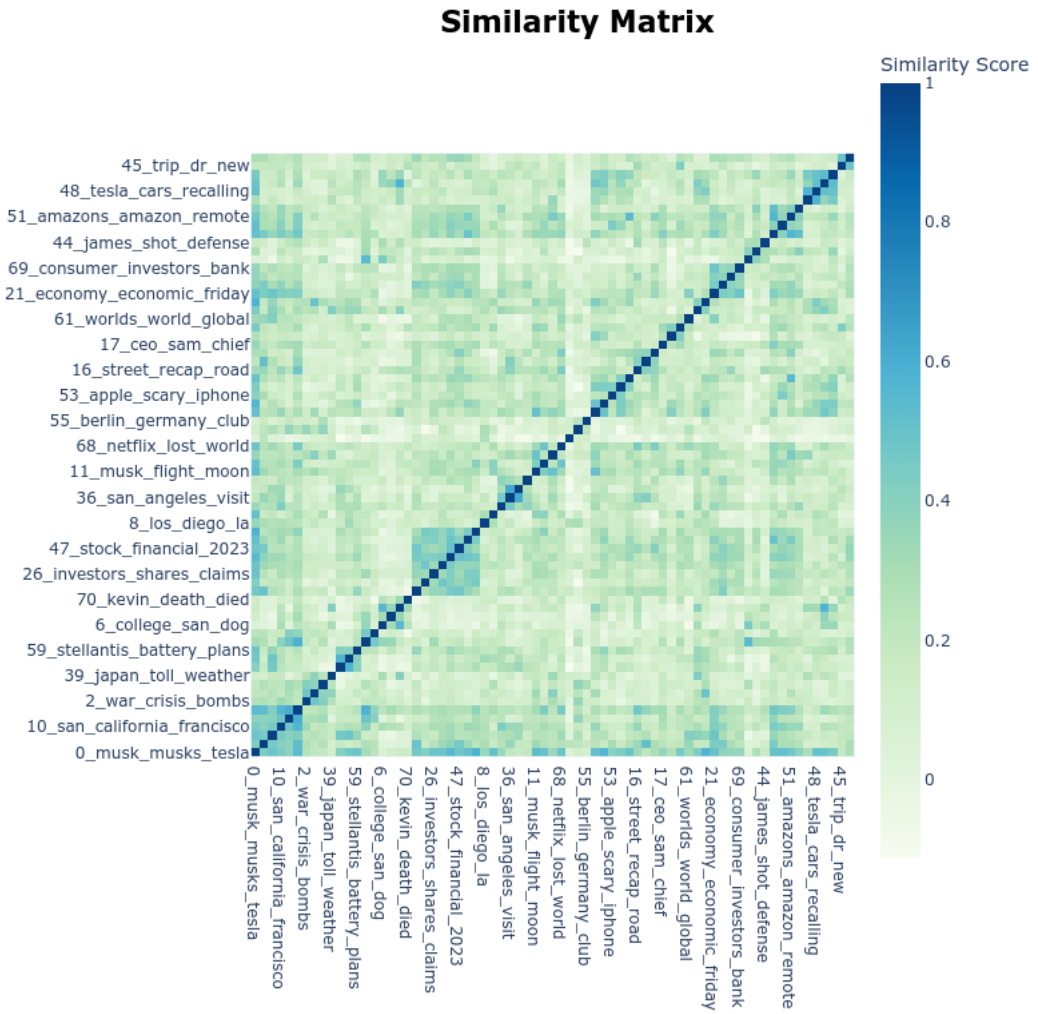
I am using the Hierarchical Dirichlet Process, as HDP only uses a Bag of word embeddings I change this initially and feed data into the model to get the number of topics.

- To model the topics I am using Latent Dirichlet Allocation which will discover clusters in the data and group them.
- After applying LDA we convert data into the topic vectors and this gives us topic features, now we can combine the features and embedding vectors to get more information.
- Using the same classification method as in for sentiment analysis case we get scores for different models.
- One way this project could have moved forward is that I could use different embedding models for our data eg TIDF,doc2vec, Bag of Words, etc, and conduct classification analysis by including the different embeddings as additional features in the dataset, because of the time constraints I am using googles word2vec-news-300 model for embedding and I get 1 additional feature as doc vectors

V Results and Discussion

The results for both the cases are noted below. First, we start with the BERTopic applied to our entire corpus of data to get a better idea of the topics distributed in the dataset. You can see the results for this in the figure below. Topic 0 mainly includes all the words related to Wall Street and the stock market, Topic 1 mostly includes all the information about Elon Musk, and Topic 2 has all the information about AI and its advancement. Topic 3 has information about batteries in particular also about energy. Topic 4 has keywords such as electors vehicles so mainly deals with TESLA, Topic 5,6, and 7 kinds of give us overall information about the cars and development in the direction of EV. These were the topics identified by the BERTopic model when given our data.

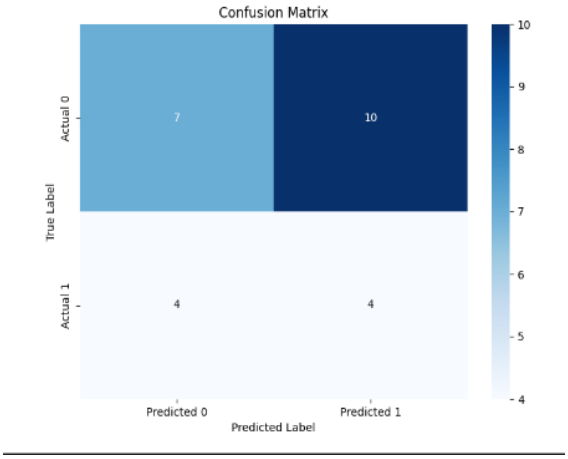




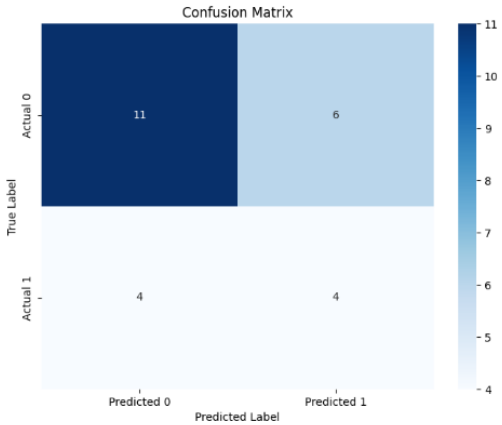
The similarity Matrix gives the idea of how similar the topics are in the given document. The bluer the color the closer the topics are. Using the similarity one can cut down the unnecessary topics or similar topics and thus have a better topic understanding which in turn gives better Topic modeling analysis.

- Sentiment Analysis

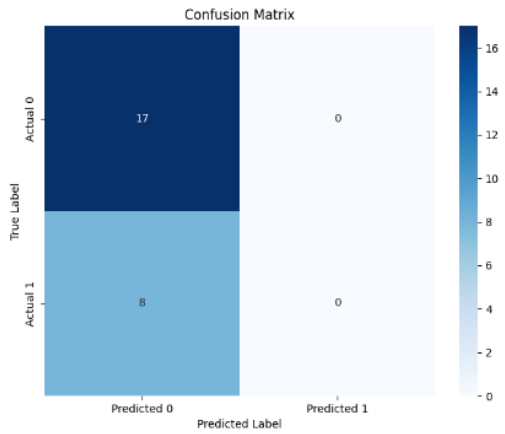
  The classification task to predict the stock price movement was conducted using the following machine learning models Stochastic Gradient Descent, Xtreme Gradient Boosting, Random Forest, and Gaussian Process, Additionally I am using ANN for the classification. The results are obtained via the classification report and accuracy in each case is taken into account. The confusion matrix for each model is plotted below.



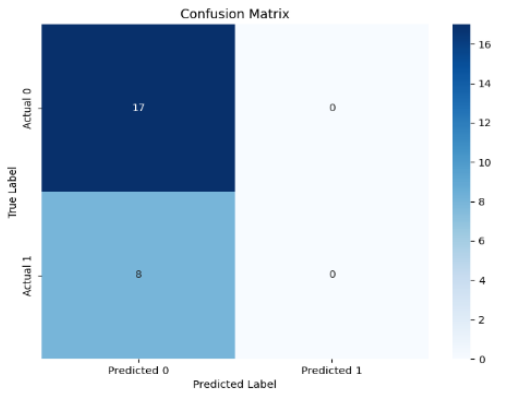
(a) XGB Classifier



(b) RF Classifier



(c) GP Classifier



(d) SGD Classifier

⟶ In the case of the Stochastic Gradient Descent Classifier the accuracy was reported to be 68% with very high precision and recall for all the down movement in stocks but on the other hand model did not have good precision and recall for the case of up movements almost 0.

⟶ For XGBClassifier the accuracy was reported to be 44% with high precision and a bit low recall for all the down movements in stocks on the other hand model did not have good precision but high recall in the case of up movements.

⟶ In the case of Random Forest Classifier the accuracy was reported to be 60% with very high precision and recall for all the down movement in stocks also model did have somewhat good precision and recall for the case of up as well.

⟶ For the Gaussian Processes Classifier the accuracy was reported to be 68% with high precision and recall for all the down movements in stocks but on the other hand model did not have good precision and recall in the case of up movements.

⟶ In the case of ANN, I'm getting the same result as an accuracy of 68%.

- Topic Modeling

  In the case of Topic modeling the classification task to predict the stock price movement was conducted using the following machine learning models Xtreme Gradient Boosting, Random Forest, and AdaBoost, Lr model. Additionally, I am using ANN for the classification. The results are obtained via the classification report and accuracy in each case is taken into account.

  One way to interpret results is when the classification task can predict 1 and 0 correctly, by that I mean being trained on the training data which has the labels for the next day's stock price movements as 1 or 0, 1 being the upward movement of stock and 0 means downward movement of stock. The test data consists of 20% of the dataset and our models predict labels for the given test data. I have mentioned classification metrics for different classifiers below

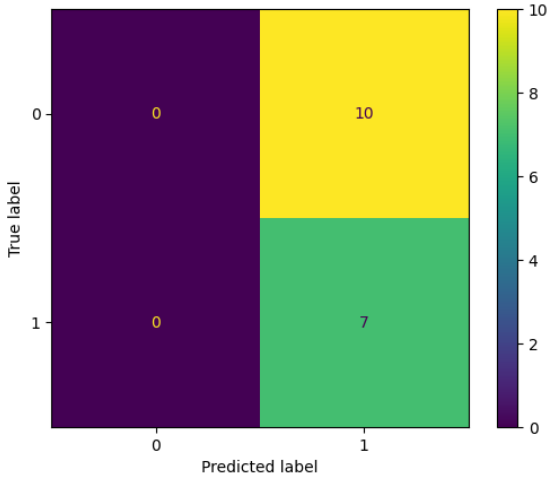
⟶ For the lr model accuracy score is 41% with average precision and recall for both the up and down cases.

⟶ For the rf model accuracy is 65% with above average precision and recall score for up and down case

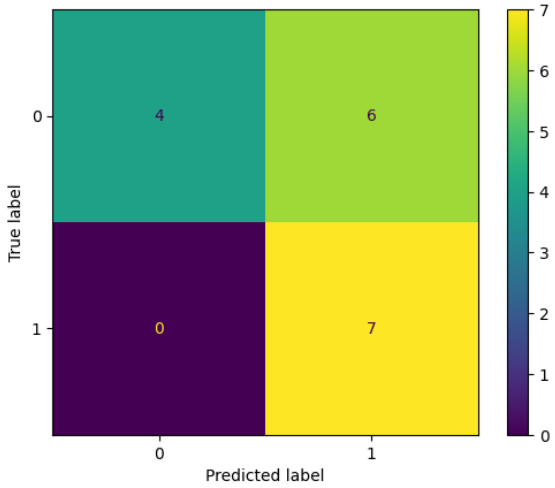
$\longrightarrow$ For the ada model accuracy is 53% with good precision and recall score for up and down cases.

$\longrightarrow$ Xgb model the accuracy is 65% with average precision and recall for both cases.

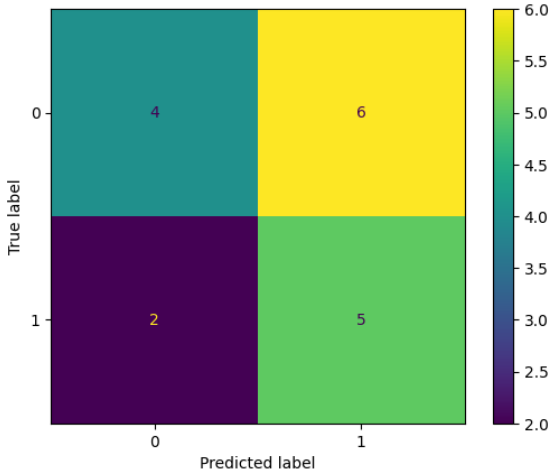
$\longrightarrow$ In the case of ANN accuracy was 60% with very good precision and recall for the down case and almost zero precision and recall for the up case.



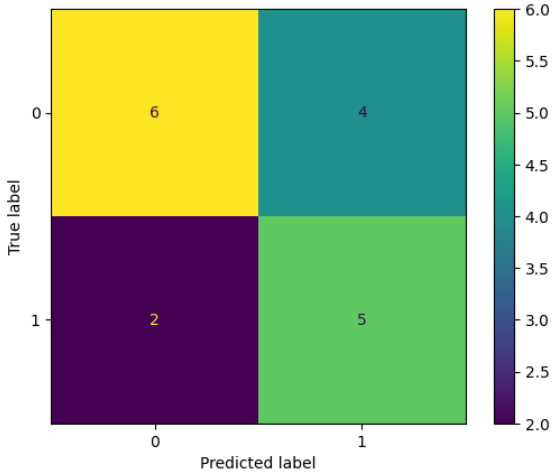
(a) LR Classifier



(b) RF Classifier



(c) ADA Classifier



(d) XGB Classifier

- Discussion and conclusion

  The results discussed above show the classification using different ML algorithms. In the case of Sentiment Analysis where the classifier was trained only on the data obtained from the Yahoo finance website, we can see that the SGD, GP, and ANN perform quite well with an accuracy of 68%. In the case of Topic modeling the models are trained on the data which consists of data from the Yahoo Finance website along with the news data preprocessed and topic modeling applied on it to get topic vectors and doc vectors, results, in this case, tell us that XGB is the best-performing classifier with an accuracy of 65%.

  The biggest challenge I would say is the limitations of the data, as the data from MediaCloud and Yahoo Finance only allowed me to gain access to 6 months of data the training data and test data in the end after preprocessing were very limited which served as an obstacle to getting more accurate results.

  I also developed my own ANN which also helped me understand the process better but for the time series data, I would maybe if possible in the future also extend this analysis to include the ARMA and other econometric models. As we are dealing with time series data one option also was to consider the RNN architecture, but because of the time constraints maybe I would take it up for a later personal project.

  Additionally, I would also like to throw light on a promising architecture for time series data [5] talks about the mixture of ARMA Cell (Autoregressive Moving Average + RNN architecture) and it is from the Statistics Department LMU, maybe this project with enough backing data could be taken into consideration for the applications of ARMA cell.

Bibliography

[1] Saloni Mohan; Sahitya Mullapudi; Sudheer Sammeta; Parag Vijayvergia; David C. Anastasiu. "Stock Price Prediction Using News Sentiment Analysis". In: *2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)* (2019). DOI: 10.1109/BigDataService.2019.00035.

[2] Thien Hai Nguyen Kiyoaki Shirai. "Topic Modeling based Sentiment Analysis on Social Media for Stock Market Predictionv". In: *he 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP)* (2015). DOI: http://dx.doi.org/10.3115/v1/P15-1131.

[3] Dogu Tan Araci. "Financial Sentiment Analysis with Pre-trained Language Models". In: (2019). DOI: https://doi.org/10.48550/arXiv.1908.10063.

[4] Pei G. "Predict Stock Market Trend Using News Headlines". In: (2020).

[5] Christoph Berninger1 Philipp Schiele1 and David R¨ugamer. "ARMA Cell: A Modular and Effective Approach for Neural Autoregressive Modeling". In: (2024). DOI: https://doi.org/10.48550/arXiv.2208.14919.