Stock Price Prediction from News Data

1. Introduction/Background

Stock prices are influenced by public sentiment and financial institutions' predictions. This project aims to forecast stock prices by leveraging news data.

Literature Review

Work by Vargas et al. [1] showed that deep learning can effectively extract sentiment from news articles to forecast market movements. Additionally, survey by Zou et al. [2] highlights how sentiment analysis could assist deep learning techniques to lead to accurate stock predictions.

The BERT model, introduced by Devlin et al. [3], revolutionized NLP. For finance specific contexts, a fine-tuned version of BERT, FinBERT [4] was created.

Dataset Description

Stock Data:

We're retrieving stock data from Yahoo! Finance APIs using yfinance library. We're primarily interested in daily open-close stock prices and the volume of the stocks traded each day.

Financial News Data:

Financial news data are gathered from multiple sources:

- **[News Dataset 1]** Daily Financial News for 6000 stocks, 2009-2020: consisting of over 3 million news headlines data from 2009-2020, from multiple financial websites and forums such as Zacks, Investor's Business Daily and Fox Business.
- **[News Dataset 2]** GitHub - pmoe7/Stock-Market-Data: Contains various data for the stock market and stocks including news, price and fundamentals (ratios): consists of news from 2019-2023

- **[News Dataset 3]** Zdong104/FNSPID_Financial_News_Dataset: FNSPID: A Comprehensive Financial News Dataset in Time Series

---

2. Problem Definition

Problem

Stock prices are influenced by various factors beyond historical prices, including macro-economic events and market sentiment. Capturing these influences solely through historical prices is challenging.

Motivation

We aim to combine natural language processing on financial news with time series forecasting of stock data for better estimates.

---

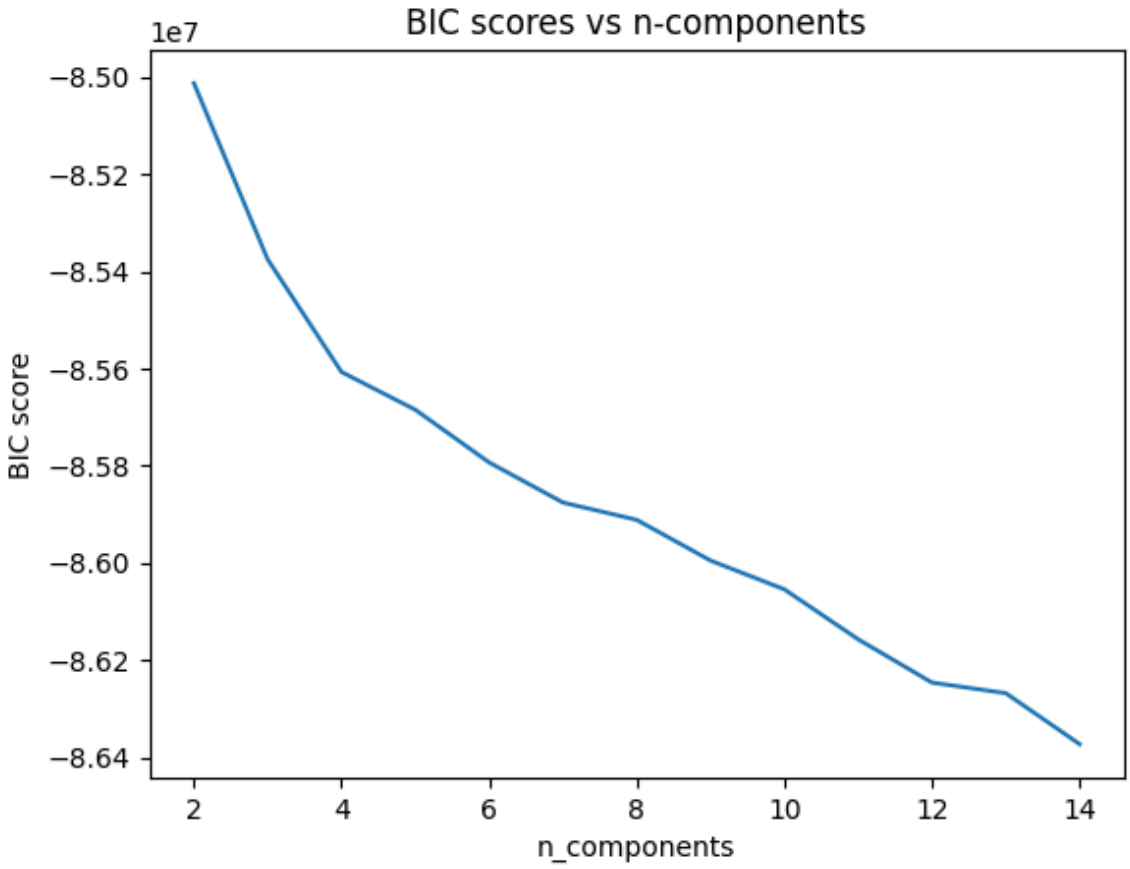
3. Methods

Processing of Financial News Data

Unsupervised Learning

Processing of Financial News Data includes three steps:

- Data cleaning and Exploratory Data Analysis on the datasets
- Data transformation using **Sentence-BERT (SBERT)**: this converted textual data into numerical embeddings that capture semantic meaning. SBERT is able to capture semantic meaning more effectively than traditional bag-of-words or TF-IDF, especially for short texts such as article headlines.
- Unsupervised Learning
    - GMM:- Inititially we intended to use clustering algorithms like K-Means and GMMS. The GMM analysis was performed on sBERT embeddings of randomly sampled 150,000 article headlines from news dataset 3. After reducing dimensionality with PCA to retain 90% of the variance, the resulting 193 features were used to fit GMM models with varying numbers of components. The BIC scores were calculated for each model to assess model fit.

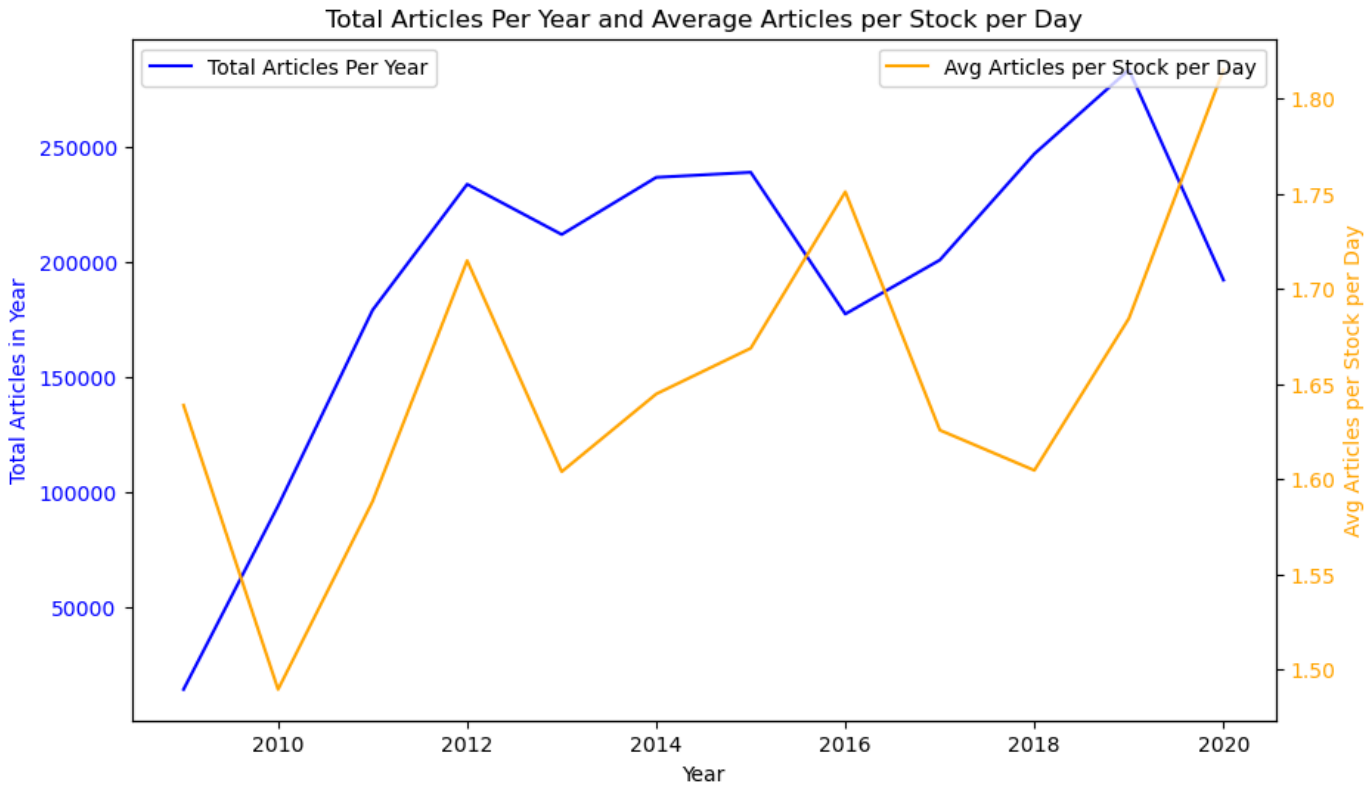


However, the lack of a clear elbow point in the BIC plot suggests that the GMM model may not be capturing the underlying structure of the data effectively. The large number of features (193) may be contributing to this issue, as GMM can struggle with high-dimensional data. To address this, we explored Variational AutoEncoders (VAEs) for unsupervised learning and clustering. VAEs are well-suited for high-dimensional data and can potentially capture more complex patterns in the data.

- Data dimensionality reduction using **Variational Autoencoders (VAE)**: VAE is an artificial neural network which could reduce the high-dimensional SBERT embeddings into lower-dimensional latent space, while keeping the semantic information.
    - VAE is trained by using a collection of specific and random tickers, due to our limitation of time and computational constraints.
    - The fixed specific tickers chosen are KO, MRK, GILD, FDX and TM. They are tickers that are consistently appearing throughout each year in all 3 datasets with a good amount of articles.
    - The amount of articles for each epoch is set at 100,000 articles, which included all articles from the above selected tickers, and the rest were filled randomly every epoch to make 100,000 articles. This randomness allows the VAE model to generalize well, while also being specific enough to our selected tickers.
    - The VAE compressed the BERT embeddings to a 128-dimensional latent space. VAE converged successfully after a few iteration, with relatively low loss.

[News Dataset 1] Daily Financial News for 6000 stocks, 2009-2020

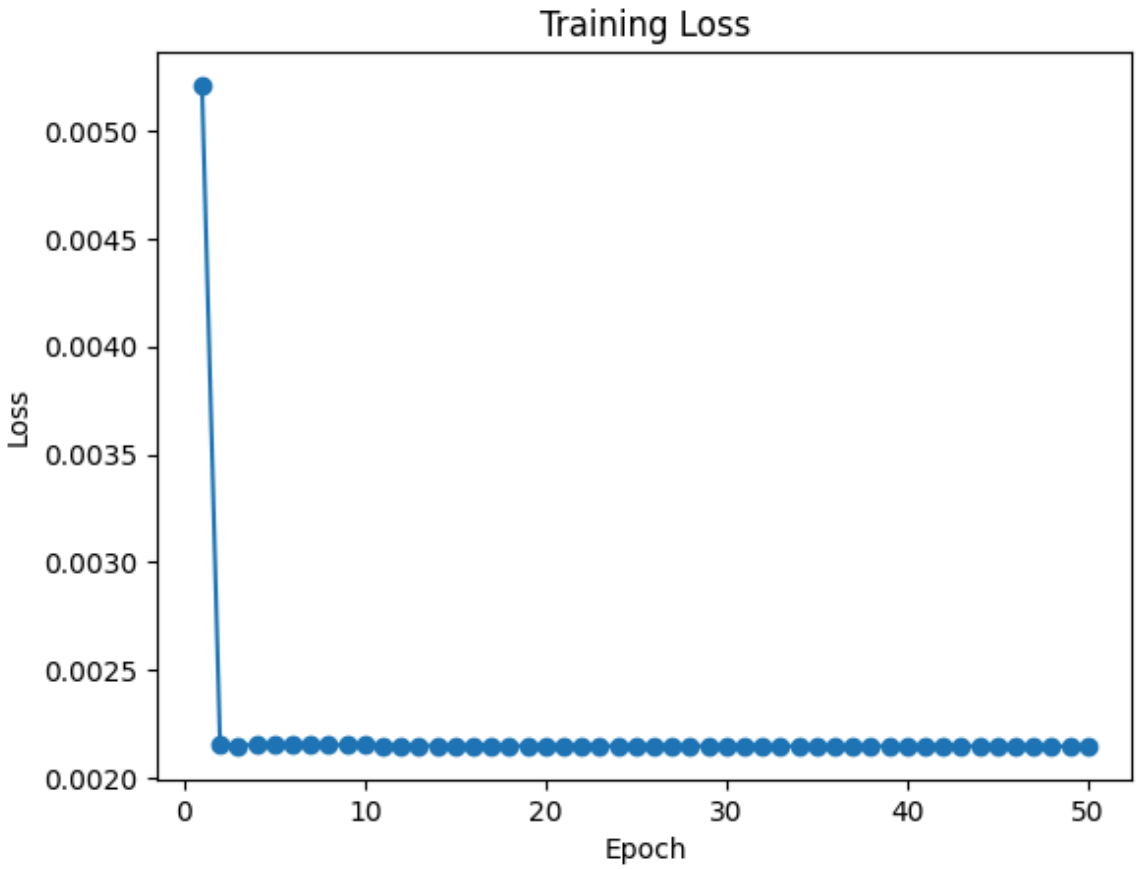
Data was restructured; missing values were removed (only affected less than 0.07% of dataset). To retain the credibility of the news and hence the quality of text data, we also excluded data from sources which include less credible individual users' postings such as Seeking Alpha, TalkMarkets and Vetr.



Average articles per year & average articles per featured stock per day

The resulting dataset includes over 190,000 news headlines daily. On average there are 30 news per stock each year, and 1.6 news per featured stock on a day. On the extreme end, there could be a large amount of news per stock daily, e.g. there are on average 8.4 news for Tesla stock (TSLA) back in 2020, when the stock skyrocketed in price.

SBERT was applied to transform over 2.3 million article headlines into dense 768-dimensional embeddings. After tokenizing with SBERT, VAE is applied to lower the latent space output. The model converged relatively quick, as shown by the loss plot below.

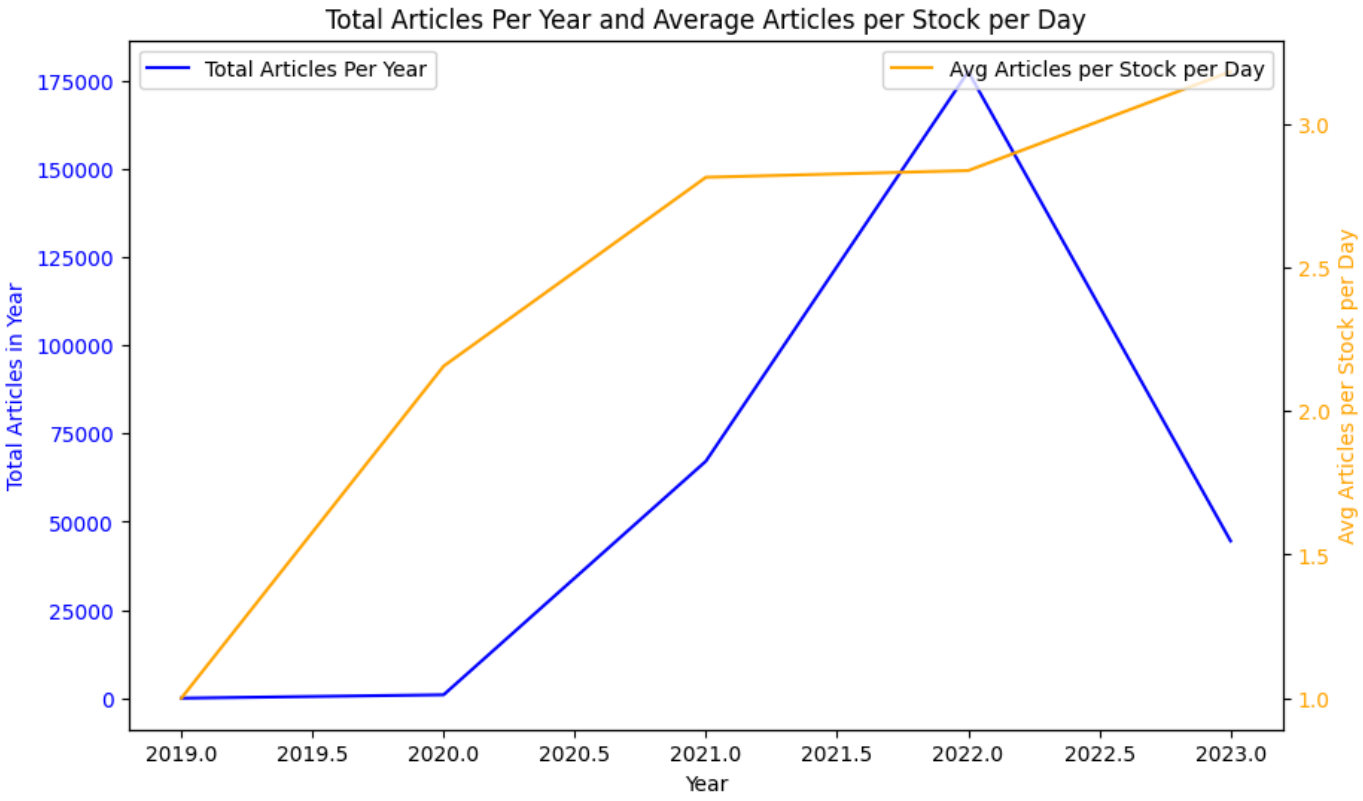


More details on the EDA could be found in the bezinga_news_eda.ipynb.

[News Dataset 2] Stock news from 2019-2023

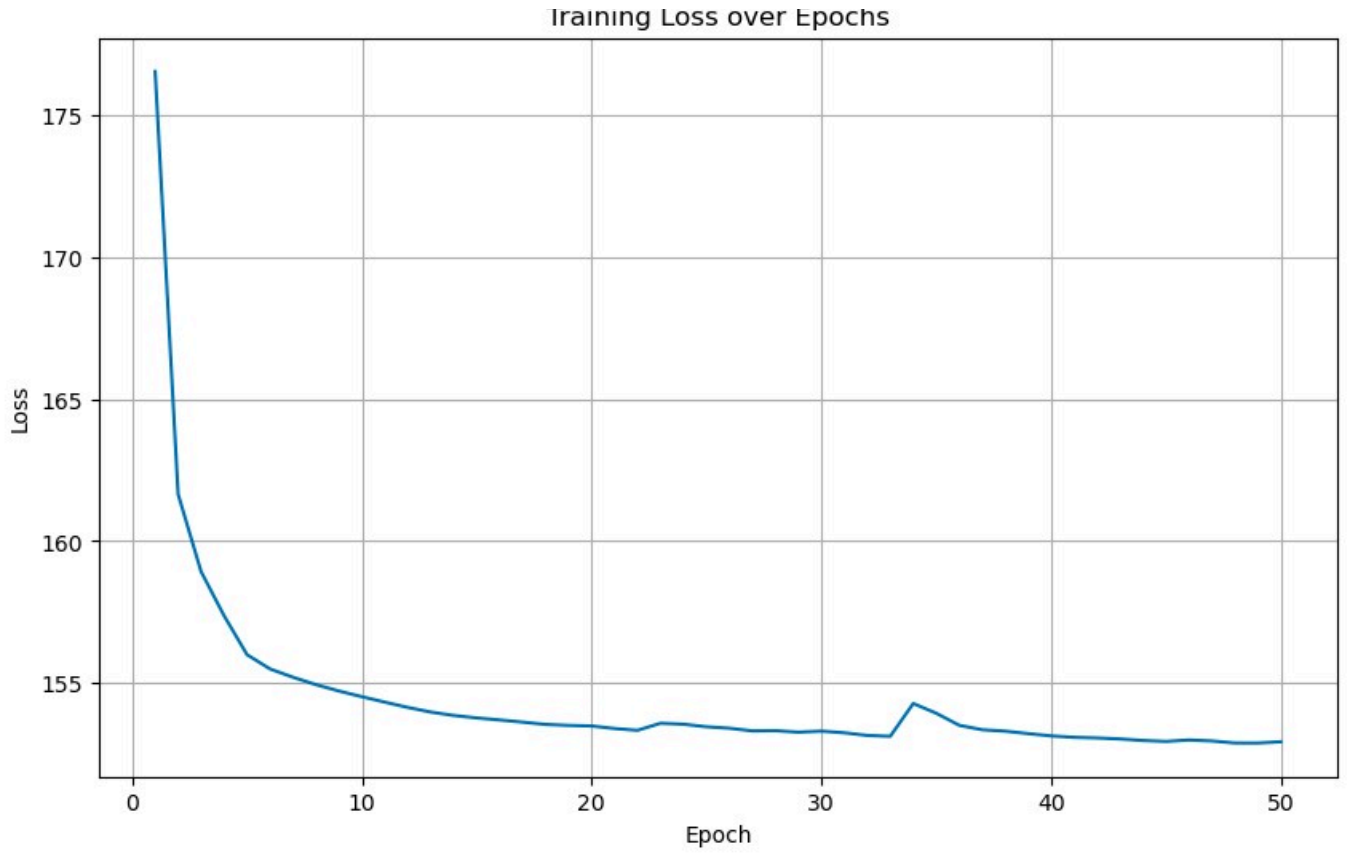
Data was also restructured with missing-values data, unnecessary data such as time, positive, negative and neutral change, removed. The plot shows the amount of stock related articles published per year and the average articles per stock per day. The chart shows the top 5 stocks mentioned per day for each year. The most mentioned stock per day and per year could be up to 35.1 and 4501, respectively.

More detail on news_EDA_2019_2023



	Top 1	Top 2	Top 3	Top 4	Top 5
2019	(L, 1.0)	None	None	None	None
2020	(L, 2.8)	(VNT, 2.7)	(IPGP, 2.6)	(CHD, 2.5)	(NOV, 2.5)
2021	(BIIB, 13.5)	(CERN, 11.7)	(CTXS, 10.6)	(MU, 9.3)	(CCL, 8.9)
2022	(TSLA, 29.3)	(AMZN, 22.7)	(AAPL, 22.3)	(GOOGL, 18.6)	(GOOG, 17.8)
2023	(TSLA, 35.1)	(GOOGL, 23.8)	(MSFT, 23.0)	(AMZN, 21.3)	(AAPL, 18.3)

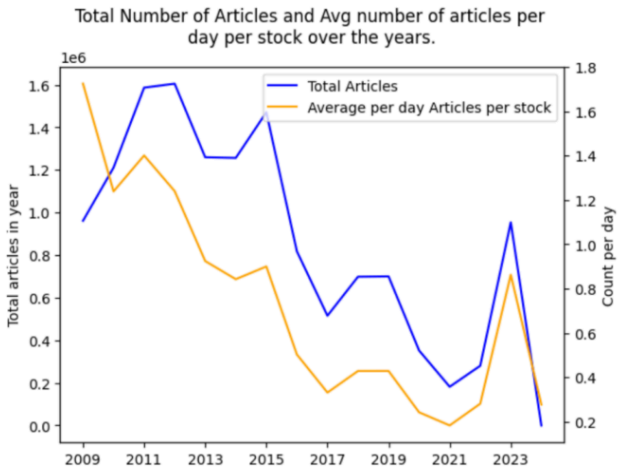
SBERT and VAE was also applied to improve the preprocessed data. In this case, the loss is considerably high, compared to other two datasets. The reason behind it was that the amount of specific tickers in the second dataset was comparably low, which was around 3,000 datapoints, meanwhile the other two had around 40,000 and 20,000, respectively. This result also shows how the consistence of specific tickers affect the loss value of the training.

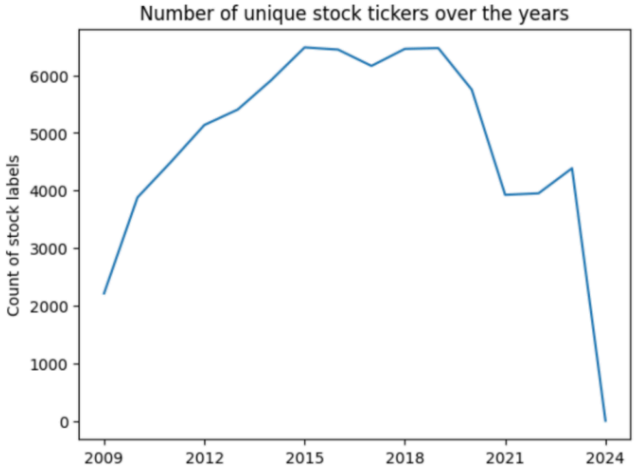


[News Dataset 3] Stock news from 1999-2023

The Financial News and Stock Price Integration Dataset (FNSPID) is a rich repository of financial data, encompassing 15.7 million news articles from 1999 to 2023 for 4,775 S&P500 companies. This dataset integrates quantitative and qualitative information to improve stock market prediction models. Notably, it includes detailed article text and summaries, offering valuable context beyond headline-level analysis. However, stock ticker information is only available from 2009 onward, limiting the historical scope of certain analyses.

To efficiently process this large dataset (close to 23GB CSV), it was converted into an SQLite database. Indices were created on the "Date" and "stock symbol" columns to optimize querying, as these columns will be frequently used for filtering during training and analysis.

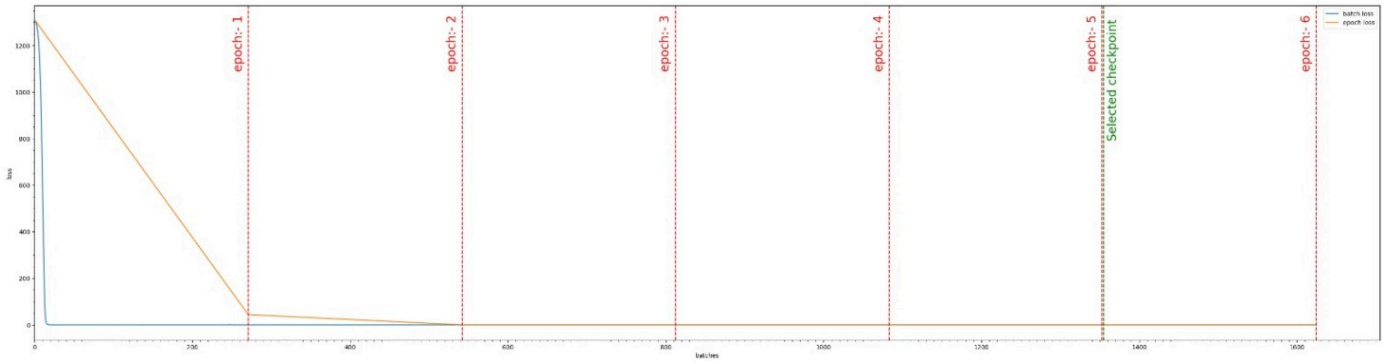




	0	1	2	3	4
2009	(, 949002)	(A, 554)	(CEO, 187)	(CA, 182)	(M, 131)
2010	(, 1106604)	(MS, 602)	(AA, 585)	(EPS, 486)	(BHP, 466)
2011	(, 1393066)	(QQQ, 1167)	(SLV, 1055)	(GLD, 1012)	(USO, 895)
2012	(, 1335803)	(GRPN, 1308)	(EWP, 898)	(GMCR, 831)	(QQQ, 773)
2013	(, 980700)	(GLD, 1046)	(BBRY, 886)	(DISH, 885)	(JCP, 881)
2014	(, 921257)	(GILD, 1162)	(ACT, 1060)	(EBAY, 1060)	(EWJ, 971)
2015	(, 947741)	(GREK, 1876)	(YHOO, 1672)	(GILD, 1270)	(GPRO, 1112)
2016	(, 311401)	(WFC, 2133)	(YHOO, 1569)	(HAL, 1174)	(BABA, 1172)
2017	(, 58792)	(WFC, 1928)	(BABA, 1905)	(RS, 1549)	(T, 1474)
2018	(, 43951)	(SPY, 2764)	(INTC, 2124)	(MU, 1706)	(BABA, 1671)
2019	(, 59729)	(INTC, 2721)	(RSP, 2657)	(GOOG, 2141)	(SLV, 1741)
2020	(DIS, 2874)	(GILD, 2237)	(GOOG, 2081)	(WMT, 2034)	(BA, 2025)
2021	(GME, 2055)	(AMC, 1766)	(DIS, 1741)	(WMT, 1486)	(GOOG, 1472)
2022	(TSLA, 3169)	(AAPL, 3094)	(MSFT, 2679)	(NVDA, 2368)	(BRK, 2364)
2023	(BROGW, 10456)	(BPYPO, 9979)	(BHFAL, 9614)	(PMAY, 9108)	(ACGLO, 9014)
2024	(AMD, 70)	None	None	None	None

Top 5 stock tickers with the most news articles over the years

In this dataset, unlike other two datasets, it converges much faster, even though the model had converged considerably within 1 epoch, we chose the model trained on 5 epochs, as it would have generalized better due to the randomness in the training data every epoch. With high consistency of specific tickers and richer embeddings, which uses article texts and various article summarizations, its converged loss is very low and considerably good. This result tells us that the richness of input does improve the performance of the VAE.



Supervised Learning

We chose Long Short-Term Memory (LSTM) and Random Forest for the task of stock price prediction.

Random Forest:

- Random Forest handles nonlinear relationships well and can work well with noisy/incomplete data. Also, Random Forests are in-general robust to overfitting due to ensembling.
- Stock data is often noisy and the amount of stock data per ticker limited, Random Forest based regression is a good choice of model for stock price prediction.
- Random Forest is also faster to train and are effective for short-term predictions.

LSTM:

- LSTMs are designed to capture temporal data and long-term dependencies. LSTMs are also good at integrating multiple temporal features into the prediction process.
- LStMs require more computational resources and training times, but they capture long-term temporal trends.

LSTM with News Headline Embeddings:

- We also added news embeddings from VAEs as features in the LSTM. We wanted to capture the market sentiment using the News headline embeddings.

---

4. Results and Discussion

We tested the model on tickers of stocks having significant amount news headlines. The tickers we analyzed are: MRK, GILD, KO, FDX, TM. We used MAPE (Mean Absolute Percentage Error) and RMSE as

metrics:

- MAPE: Provides scale-independent and interpretable error in percentage terms, making it easy to understand and compare the performance across various tickers and models.
- RMSE: Focuses on the absolute magnitude of the error, penalizing large deviations, and giving more weight on the precision.

The LSTM model performs well on stock market prediction because it is designed to capture long-term dependencies in sequential data, making it effective for understanding time-series patterns in stock prices. The model's memory cells help retain relevant past information and filter out noise, allowing it to identify trends and patterns crucial for predicting future stock movements.

Results:

Following are metrics on the test data (from 2022 to 2023):

1. **MAPE (Mean Absolute Percentage Error):**
    - LSTM significantly outperformed Random Forest, when trained on financial data alone.
    - Incorporating news data into LSTM increased noise and reduced accuracy.

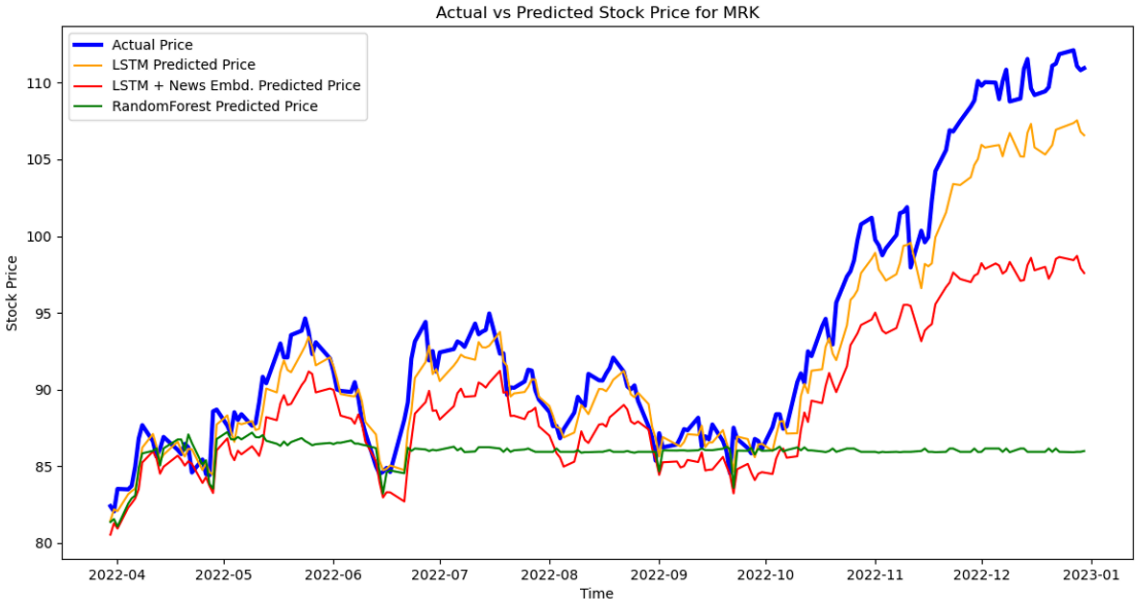
Ticker	LSTM (Financial Data Only)	LSTM + News Data	Random Forest
MRK	1.67%	4.15%	7.39%
GILD	1.50%	1.83%	1.32%
KO	1.37%	4.38%	5.92%
FDX	1.94%	2.37%	2.22%
TM	1.23%	1.57%	1.63%

2. **RMSE (Root Mean Squared Error):**
    - Similar trend is observed in RMSE as well.

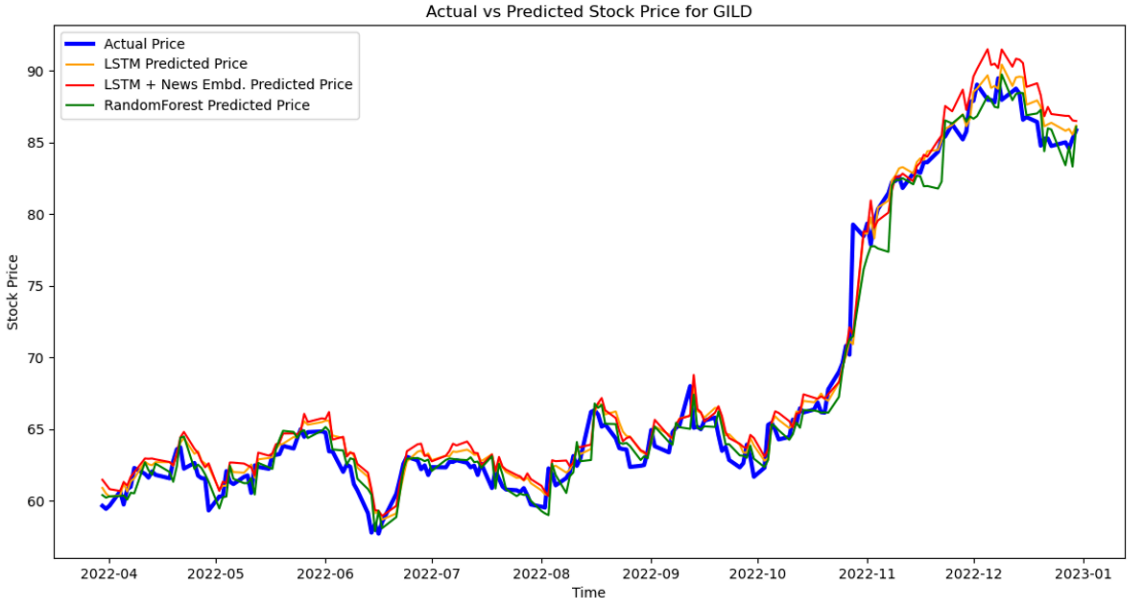
Ticker	LSTM (Financial Data Only)	LSTM + News Data	Random Forest
MRK	3.01	6.36	10.94
GILD	1.19	3.70	1.34
KO	1.37	4.38	5.92

Ticker	LSTM (Financial Data Only)	LSTM + News Data	Random Forest
FDX	5.81	6.09	6.70
TM	2.65	7.25	3.30

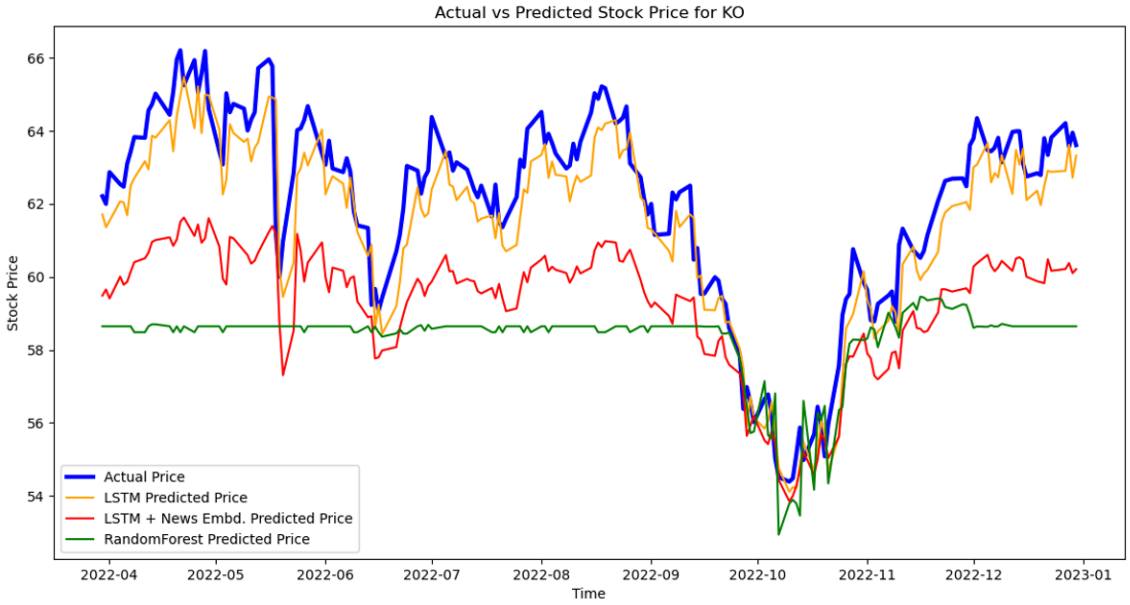
Predictions



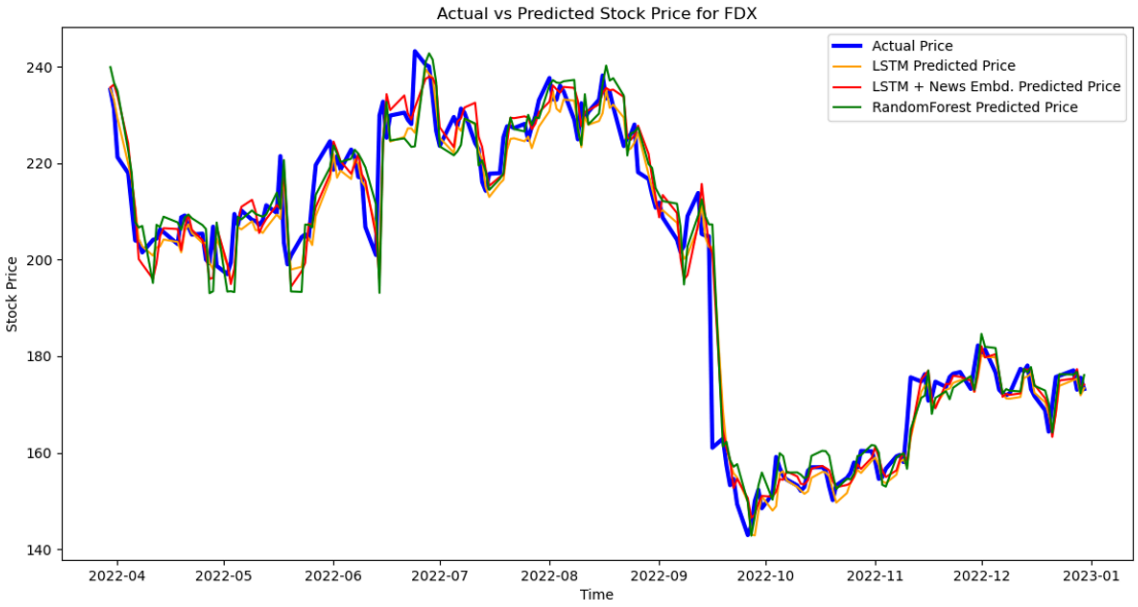
Stock price predictions for MRK



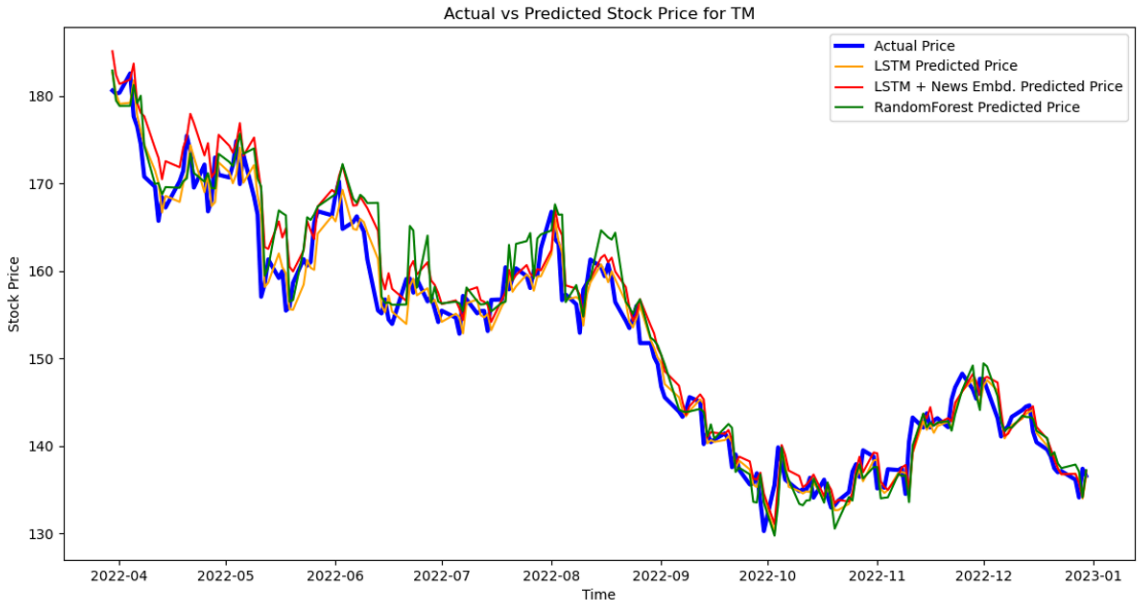
Stock price predictions for GILD



Stock price predictions for KO



Stock price predictions for FDX



Stock price predictions for TM

We can see from the above prediction graphs that Random Forest based regression often fails to capture the temporal nature of the stock price (i.e. it is not able to capture the fact that if the prices were high on the preious X days, there is a high chance that the price would be high on the current day).

Discussions:

Why news embeddings did not help much?

As we can see, using the embeddings of news headlines related to the ticker did not help much in improving the performance of the LSTM model and in-fact the LSTM model trained only over financial data ended up performing better. We attribute this to the following:

- Stock news could sometimes include contradictory or irrelevant information, confusing the prediction model.
- Many times, the news headlines are speculative and not completely factual. Also, news headlines might be able to provide an insight on the direction of the price change, but not the magnitude.
- Stock prices are volatile and there a lot of factors that directly or indirectly impact stock prices.
- The news data might be noisy and has weak/inconsistent correlation to the Close price.
- Financial data is directly tied to the prediction target (future Close prices) and contains clear temporal patterns.

Why LSTM worked well?

- stock prices are inherently sequential and the current price are greatly influenced by historical data. LSTMs could capture such relationships (which could be linear or non-linear) well due to their memory states.
- Unlike traditional RNNs, LSTMs are better suited at addressing vanishing gradient problem, which helps them in learning long term patterns.
- LSTMs can smooth out the noise by focusing on the inherent long-term patterns.

Future Steps

- More financial and macro-economic indicators could be incorporated in the model for a better prediction.
- More analysis needs to be done to on leveraging news embedding to their maximum potential.
- New sequence-to-sequence learning methods such as Transformers could be used for such tasks.

---

5. References

1. M. R. Vargas, B. S. L. P. de Lima and A. G. Evsukoff, "Deep learning for stock market prediction from financial news articles," 2017. doi: 10.1109/CIVEMSA.2017.7995302.
2. J. Zou, Q. Zhao, et al, "Stock Market Prediction via Deep Learning Techniques: A Survey," 2023. doi.org: 10.48550/arXiv.2212.12717
3. Devlin, J., Chang, M., Lee, K., & Toutanova, K. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," 2018. doi: 10.48550/arXiv.1810.04805.
4. Dogu Araci, "FinBERT: Financial Sentiment Analysis with Pre-trained Language Models," 2019. doi: 10.48550/arXiv.1908.10063

---

6. Planning & Contribution

Gantt Chart planning here.

Name	Proposal Contribution
Gia Khanh Dao	Researched & performed data preprocessing for dataset1. Performed sBERT and VAE training on dataset 1. Wrote report on the mentioned sections. Managed

Name	Proposal Contribution
	project Gantt chart. Did video final presentation with Raj.
Adithya Manjunatha	Researched and proposed the unsupervised learning section, along with the architecture diagram, proposal video presentation; Unsupervised model and training Architecture & code; Training and inference of dataset 3; SQLite DB setup for large dataset.
Raj Shah (rshah647)	Researched about various LSTM architectures, on how to integrate news embeddings, implemented and experimented with various LSTM architectures and Random Forest regression, updated streamlit website.
Raj Jignesh Shah (rshah629)	Researched about various LSTM architectures, on how to integrate news embeddings, implemented and experimented with various LSTM architectures and Random Forest regression, created video for supervised method, results, created video for final project presentation.
Nguyen, Thong Q	Searched on dataset, Worked on preprocessing data, applying sBERT and VAE on the second data set, writing reports