Deep Learning Project
Final Report
CSL 4020

Problem Statement

Design a deep learning model to Predicting Stock Prices using Financial News Sentiment Analysis. Our Aim is to study how the sentiments drive the market for the public and develop a trading strategy based on the financial news .

We worked on predicting the change in stock prices after a news article has been published by a news outlet . This would be done on an intra-day basis where the most amount of profit can be made. The financial brokers who hold large volumes of shares already know the news before it has been published therefore this model would be better used by the general public .

Solution Strategy

Our strategy involves using two models , the first model will take the financial news and will then predict the sentiment score for the news.

After aggregating the news and appropriately weighing them , we get the final sentiment score for the day.

The second model will take this sentiment as an input , along with other factors such as gold price, crude oil price and then predict the closing price for the day.

Motivation:

We read many research papers and brainstormed a lot of ideas thus we came to know that the price of the stock is actually affected by the gold prices,crude oil prices etc having high correlation with the stock price.

Efficient Market Hypothesis (EMH):
The Efficient Market Hypothesis posits that all available information in the market is already reflected in stock prices. While this theory suggests that predicting stock prices might not offer significant advantages, we recognize that market inefficiencies and behavioral biases among traders can create pricing irregularities, allowing for exploitable patterns, especially in the short term thus we are predicting prices of the immediate next day .

Need for Sentiment Analysis:
Numerous studies, including [1], have demonstrated the efficacy of sentiment analysis in improving stock price prediction models. By integrating sentiment analysis alongside other market indicators, we aim to enhance the accuracy of our predictions, particularly within the context of intra-day trading where swift reactions to news events are paramount.

Long Term vs. Short Term Trading:
Acknowledging the limitations of predicting stock prices over extended periods, our focus lies on short-term (intra-day) trading. Attempting to forecast stock prices over long-term horizons is fraught with uncertainty due to evolving market dynamics. However, within shorter timeframes, the impact of news events and market sentiment can be more readily discerned and capitalized upon.

Human Sentiments and Market Dynamics:
Human sentiments and behavioral biases play a pivotal role in shaping market outcomes. While the market strives for efficiency, imperfections persist due to the inherent irrationality of some participants. This creates opportunities for informed traders to exploit pricing irregularities and capitalize on predictable patterns in stock returns.

Dataset

We created the dataset and it is now hosted on kaggle , **made available publicly** on kaggle . The dataset contains 1700 financial news articles related to reliance . The data was **scraped** from the internet using python.

- The **initial dataset** only had **news headlines**.
- **Web scraping** was performed on articles using **Beautiful Soup** .
- We also scraped the relevant prices of indicators
    - Gold Price
    - Crude Oil Price
    - Coal Price
    - Petrol/Diesel Price
- We combined the dataset which gave the **best sentiment analysis score** .
- The **dataset** was made **available to be public** for **general use** on **kaggle** .

Major innovations

1. **Understanding Feelings in Financial News**:

We're basically teaching our system to understand the vibes in financial news articles. It's like figuring out if people are feeling positive or negative about certain events and how that might affect stock prices.

2. **Two Heads Are Better Than One**:
Instead of putting all our eggs in one basket, we're using two different models. One looks at the feelings in the news, while the other takes those feelings and mixes them with stuff like gold and oil prices to predict how stocks might behave by the end of the day. It's like having a backup plan to make our predictions more accurate.

3. **Playing the Short Game:**
We're not trying to predict where stocks will be ten years from now because, honestly, who knows? Instead, we're focusing on what happens in a single day. We're trying to ride the waves of the market's emotions and make quick decisions to hopefully turn a profit.

4. **Understanding People's Quirks in Trading:**
We know that sometimes people don't act rationally when it comes to stocks. So, we're taking into account all the weird quirks and behaviors that can affect stock prices. It's like trying to read the room at a party and predicting who's going to dance next.

Results

Sentiment Analysis

For the most basic implementation we used **Sentiment Intensity Analyzer** from **nltk.sentiment** to get 4 scores **positive**, **negative** , **neutral** and **compound** . After performing stock price prediction on this approach we did not get satisfactory results therefore we switched to **Deep Learning Approaches** .

For sentiment analysis we first performed sentiment analysis using basic sequential models such as **GRU** and **LSTMs** on a dataset which gave around **0.75 accuracy**.

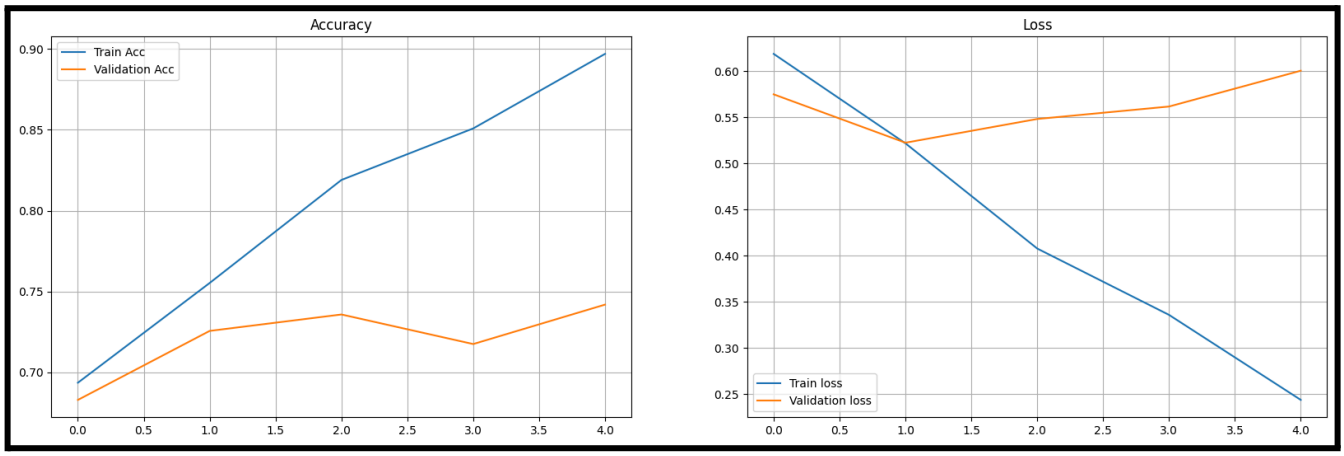
Model used were:

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 25)]              0

 embedding (Embedding)       (None, 25, 128)           6184320

 gru (GRU)                   (None, 25, 256)           296448

 flatten (Flatten)           (None, 6400)              0

 dense (Dense)               (None, 2)                 12802

=================================================================
Total params: 6493570 (24.77 MB)
Trainable params: 6493570 (24.77 MB)
Non-trainable params: 0 (0.00 Byte)
```

Model: Gated Recurrent Network

```
SentimentRNN(
  (embedding): Embedding(1001, 64)
  (lstm): LSTM(64, 256, num_layers=2, batch_first=True)
  (dropout): Dropout(p=0.3, inplace=False)
  (fc): Linear(in_features=256, out_features=1, bias=True)
  (sig): Sigmoid()
)
```

Model : LSTM



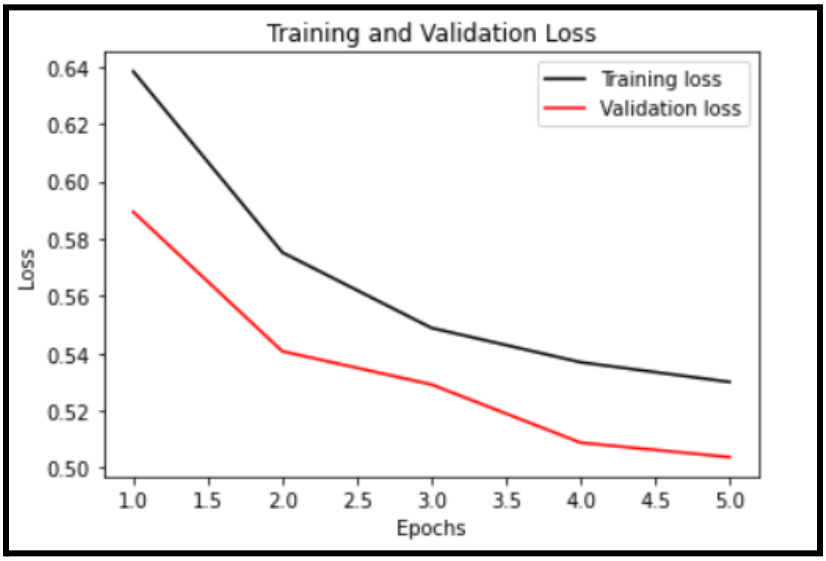
Results from basic sequential models

Now a thing which we can notice by the above graphs is validation loss is not decreasing as we increase the number of epochs although the training loss is decreasing after each epoch. This might be as the model is actually just overfitting to the train dataset and not making its decision on by understanding the meaning of the text.

We have applied multiple embedding techniques with multiple embedding sizes but none gave good results. Therefore we switched to transformer models due to their known ability to capture the meaning of text.

So to do so we did **Transfer Learning on BERT** to perform sentiment analysis .

- First we import the BERT-based **pretrained model** and we load the **BERT Tokenizer .**
- We tokenize ,encode the sequences in the train,validation and test set and then convert them into tensors.
- After training the model on the whole dataset we label our custom dataset.



	precision	recall	f1-score	support
0	0.78	0.69	0.73	833
1	0.73	0.81	0.77	876
accuracy			0.75	1709
macro avg	0.75	0.75	0.75	1709
weighted avg	0.75	0.75	0.75	1709

Results Obtained on Real Dataset

	precision	recall	f1-score	support
0	0.78	0.69	0.73	833
1	0.73	0.81	0.77	876
accuracy			0.75	1709
macro avg	0.75	0.75	0.75	1709
weighted avg	0.75	0.75	0.75	1709

Results Obtained on Custom Dataset

But now as the dataset does not contain the corresponding stock or even the market sector for the news article therefore we switched to the dataset we prepared as explained above which contained news related to "**RELIANCE**".

Now that we have the corresponding sentiment for around 1700 articles related to Reliance we move to the second part that is taking in different features mixed with sentiment corresponding to a particular date and trying to predict the next day's price.

Experiment's Environment:

Approach :-1

Model

```
LSTM(
  (lstm): LSTM(1, 32, num_layers=2, batch_first=True)
  (fc): Linear(in_features=32, out_features=1, bias=True)
)
10
torch.Size([128, 1])
torch.Size([128, 32])
torch.Size([128])
torch.Size([128])
torch.Size([128, 32])
torch.Size([128, 32])
torch.Size([128])
torch.Size([128])
torch.Size([1, 32])
torch.Size([1])
```

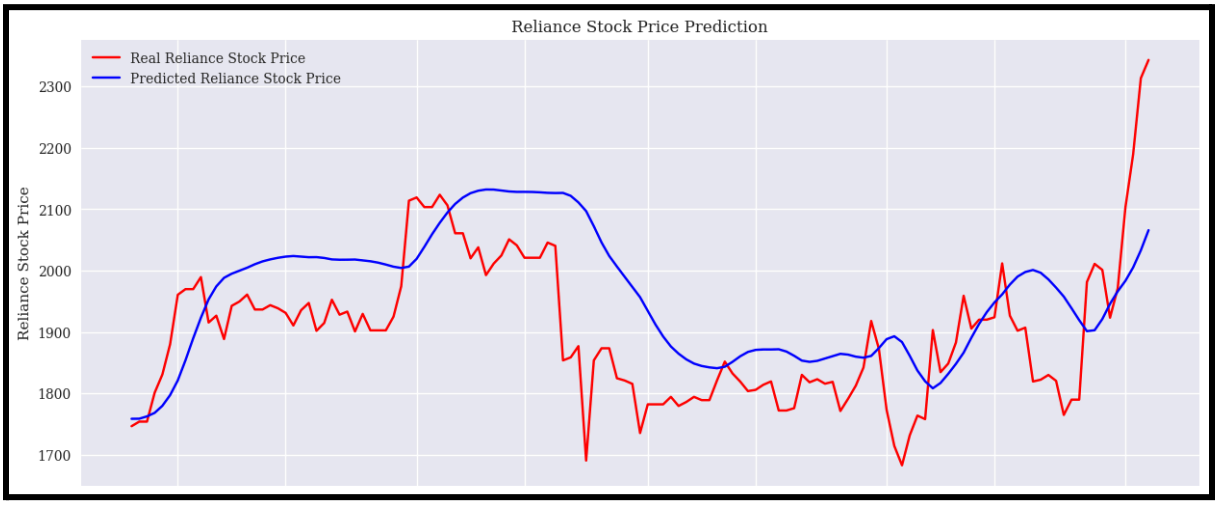
Learning Rate : 0.01

Optimiser : Adam Optimiser

Loss Function : MSE Loss

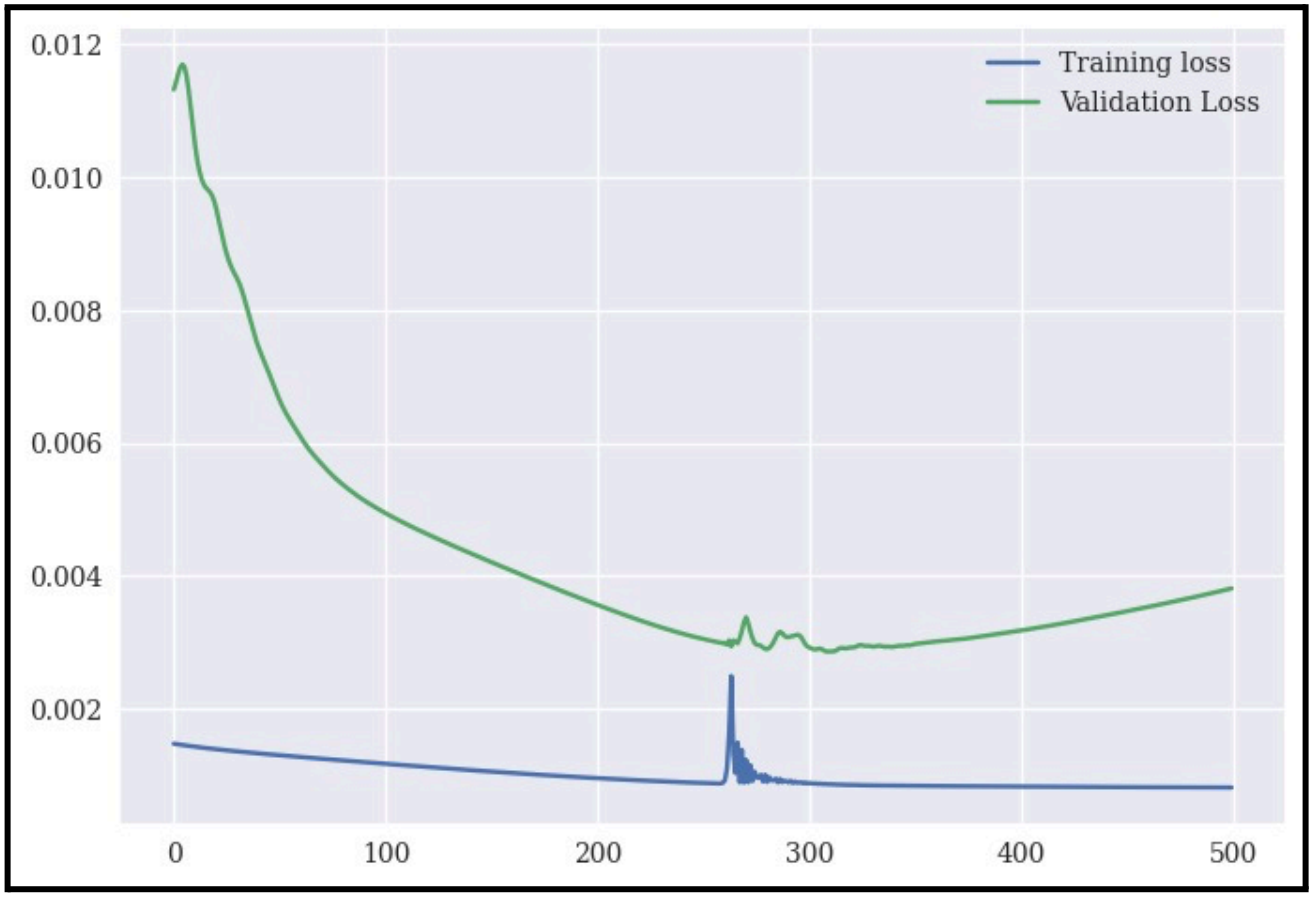
Window size : 100

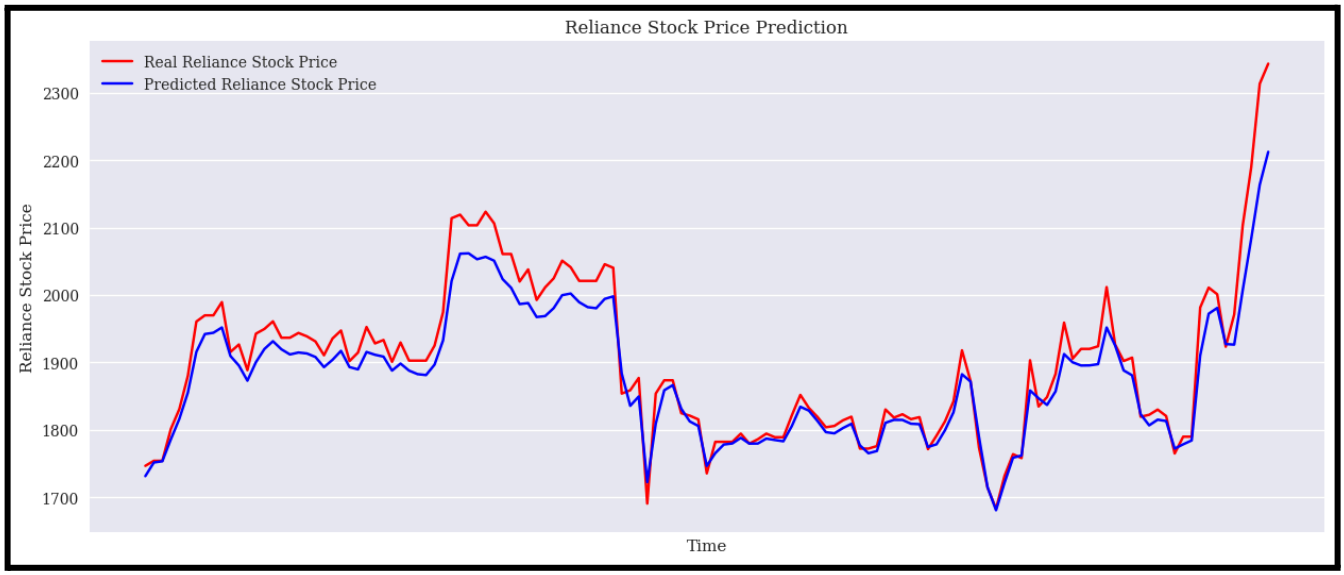
1. First let's take a look at the predictions taking only the closing price of the stock.



Train Score: **27.30 RMSE** ; Test Score: **70.78 RMSE**

2. Now after adding sentiment to the input in the model, meaning in a batch giving a window of closing prices of previous 100 days and last days sentiment improves the predictions a lot.

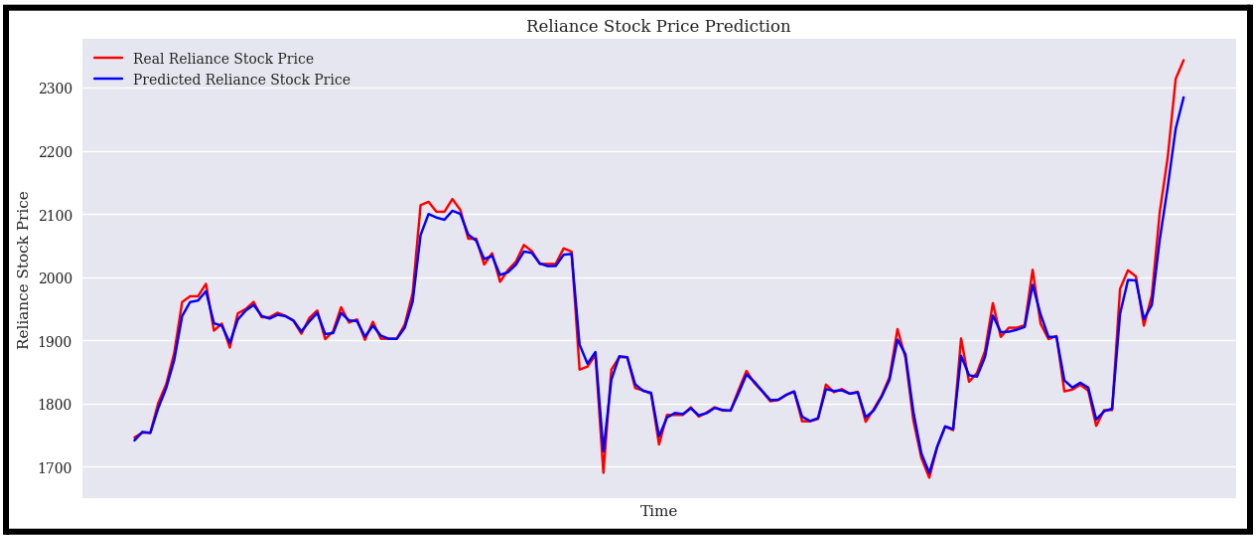




Train Score: **1.87 RMSE** ; Test Score: **35.14 RMSE**

3. Now we applied another method in which we adjusted the sentiments using the percentage change in the stock.

   In our dataset there are multiple articles for a single day and the confidence level of the sentiment for a particular article may be too high but the news might not be too influencing to actually move the stock therefore to take that into account we adjusted the sentiments.



Train Score: **0.95 RMSE** ; Test Score: **14.63 RMSE**

But this looks too good to be true, which is as we adjusted the test confidence levels as well which led to memory leakage leading to such results.

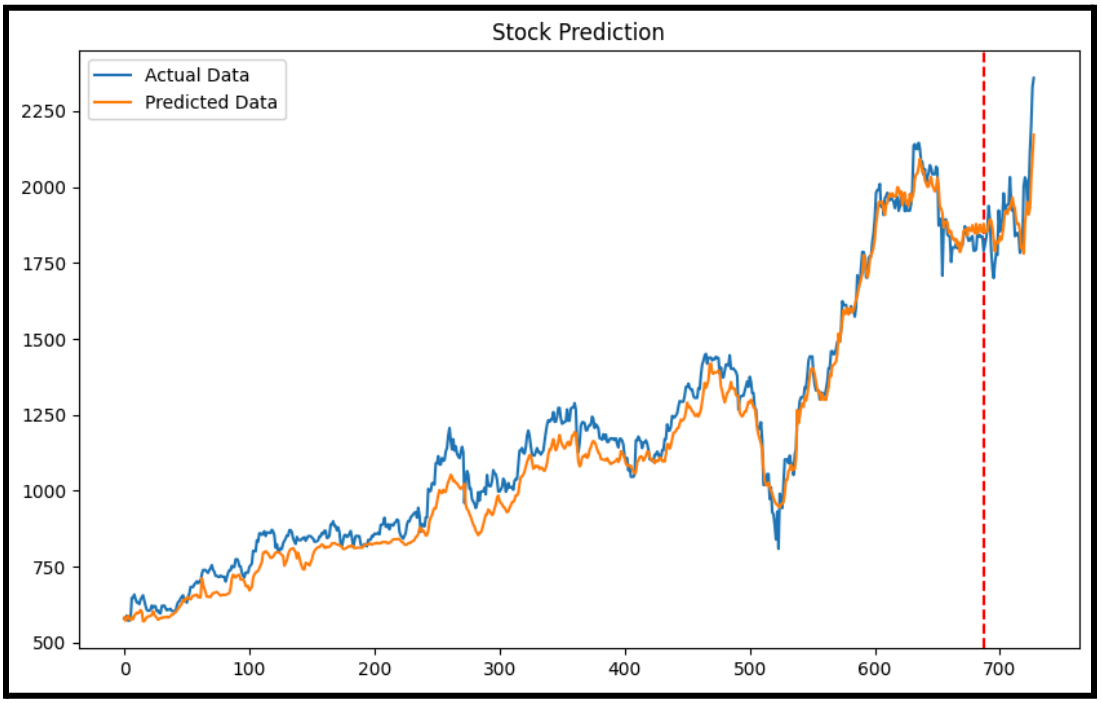
Approach 2 -

Model : -

To improve the performance we have included multiple features like Gold price,Coal Price and Crude Oil price as the main source of reliance stock depending upon its petroleum industry and we have also tried to make the Lstm model more complex and add regularization using dropout . and also included lag1 as it time series data and current stock price depend on its previous day value.

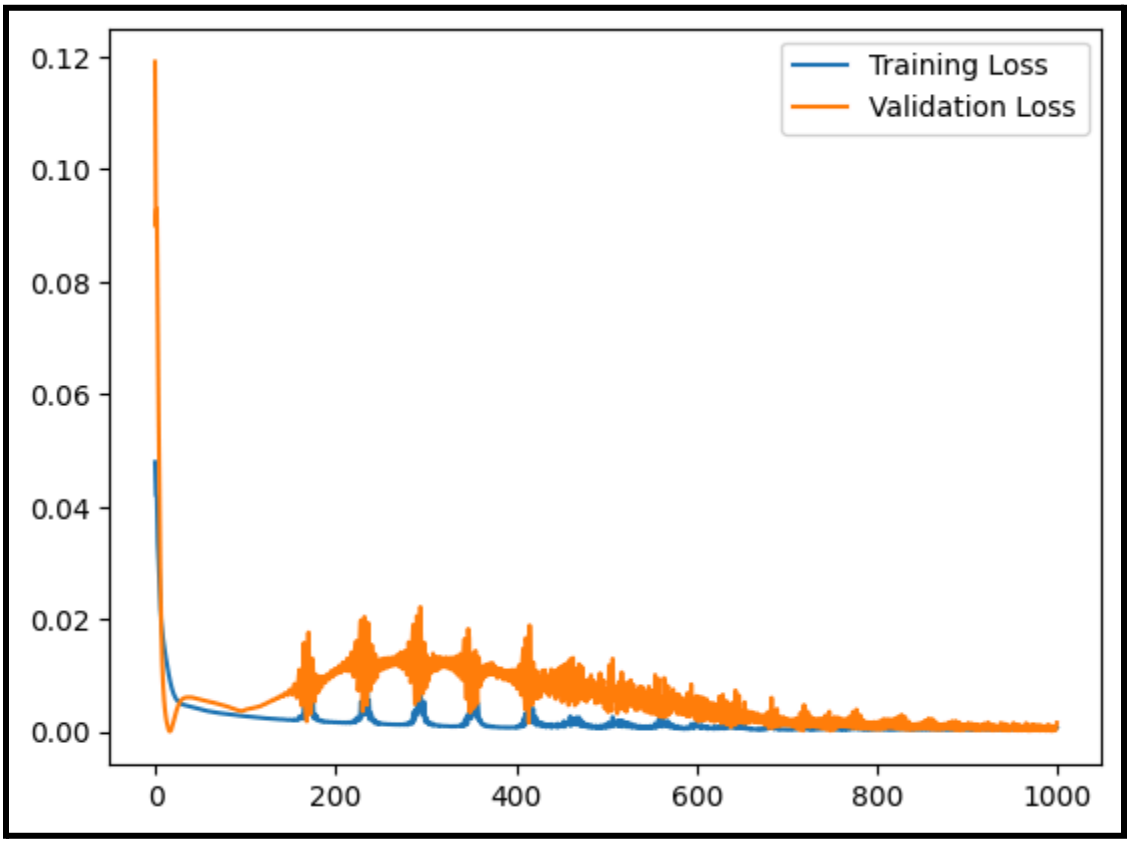
```
LSTM(
  (lstm): LSTM(5, 2, batch_first=True, dropout=0.2)
  (fc): Sequential(
    (0): Linear(in_features=2, out_features=128, bias=True)
    (1): ReLU()
    (2): Linear(in_features=128, out_features=1, bias=True)
  )
)
```

The Actual vs predicted



Total Rmse - 65.241

The results were improved as the model attempted to predict the next day's value by incorporating various features and enhancing the model architecture.



Analysis of the solution with discussion on the possible weaknesses

Strengths:

1. **Integration of Multiple Factors:** By incorporating sentiment analysis of financial news along with other market indicators like gold and crude oil prices, the model learns much more about the market. This also resembles real life much better since a human would have had access to this data.

2. **Short-Term Focus:**Since the effects of a news die out really quickly our model predicts price of the next day thus capitalizing on the news before the market absorbs the sentiment.

Weaknesses:

1. **Data Quality and Availability:** The effectiveness of the model heavily relies on the quality and availability of data, particularly for sentiment analysis of financial news. Inaccurate or limited data could undermine the model's predictive capabilities and introduce biases.

2. **Market Volatility and Unforeseen Events:** Despite its focus on short-term trading, the model may struggle to account for sudden market volatility and unforeseen events. Events like unexpected economic announcements or geopolitical developments could disrupt the predictive accuracy of the model.

3. **Human Factors:** While the model attempts to account for human sentiments, it may still struggle to fully capture the complexity of human behavior in trading. Behavioral biases, market sentiment shifts, and other human-driven factors may not be fully quantifiable, limiting the model's predictive power.

Conclusion

In conclusion, the inclusion of sentiment analysis alongside other market factors significantly enhances the predictive capabilities of the model for short-term stock price prediction. By leveraging sentiment analysis of financial news alongwith indicators such as gold and crude oil prices, the model achieves a more

comprehensive understanding of market dynamics, leading to more reliable predictions..

While challenges such as data quality, model complexity, and market volatility persist, the demonstrated effectiveness of the solution highlights its potential as a valuable tool for traders seeking to capitalize on short-term market opportunities.

In essence, by combining sentiment analysis with other market factors, the proposed solution offers a compelling approach to short-term stock price prediction, empowering traders with actionable insights and enhancing their ability to navigate the complexities of financial markets effectively.

Team Members

Team Member 1	Manan Jain	B21AI021	jain.67@iitj.ac.in
Team Member 2	Yash Mangal	B21AI047	mangal.7@iitj.ac.in
Team Member 3	Anshu Raj	B21AI048	raj.29@iitj.ac.in
Team Member 4	Saksham Jain	B21EE059	jain.77@iitj.ac.in

References

1. Narayana Darapaneni, Anwesh Reddy Paduri, Himank Sharma, Milind Manjrekar, Nutan Hindlekar, Pranali Bhagat, Usha Aiyer, and Yogesh Agarwal. 2022. "Stock Price Prediction using Sentiment Analysis and Deep Learning for Indian Markets." arXiv:2204.05783 [q-fin.ST].
2. Robert P. Schumaker and Hsinchun Chen. 2009. Textual analysis of stock market prediction using breaking financial news: The AZFin text system. ACM Trans. Inf. Syst. 27, 2, Article 12 (February 2009), 19 pages. https://doi.org/10.1145/1462198.1462204
3. Zhuang Liu, Degen Huang, Kaiyu Huang, Zhuang Li, and Jun Zhao. 2021. FinBERT: a pre-trained financial language representation model for financial text mining. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI'20). Article 622, 4513–4519.
4. Social signals and algorithmic trading of Bitcoin David Garcia, Frank Schweitzer Chair of Systems Design, ETH Zurich
5. Paper Summaries

Code Files

The code files have been hosted on google drive . Below are the explanations for the same .

- **Bert SIA** contains the work done on **Bert and Sentiment Intensity**
- **Analyzer**; it requires three datasets.
    - Gold-data -> Contains the Gold Data and the scraped data
    - financial-news-data -> Contains the data containing only the scraped data
    - india-financial-news-headline-sentiments -> Contains the financial news headline data
- **GRU** -> This file contains the **GRU for performing the sentiment analysis**.
- **LSTM_Price_Prediction_With_Sentiment_and_indicators**-> For Stock Price Prediction using LSTM Comprehensive Notebook .For Various Experiments with adding additional market data with closing price and sentiment which moves Reliance stock such as gold price, coal price and petroleum price.
- **LSTM Sentiment Analysis** -> Implemented a LSTM model for sentiment analysis on Financial_News_dataset
- **LSTM_price_prediction_with_and_without_sentiment** -> Implemented a LSTM model for price prediction and multiple experiments including price prediction using only closing price and implementing sentiment with close price.
- **Web Scraping** -> For performing web scraping using Beautiful Soup