Deep Learning Final Project

Stock Price Prediction with Social Media Sentiment Data

Gregor Kovač

3 June 2025

Abstract

In this project report we deal with stock price prediction for MAANG companies. We show how this is done using an LSTM network and develop a baseline model. As an addition in hopes to improve the predictions, we incorporate sentiments from Twitter/X posts. We show how we extracted sentiments using the FinBERT model. Then we develop two different architectures combining the baseline model with the sentiments. One uses a single LSTM with more input features, and the other has two LSTM-s, one for stock prices and one for sentiments. We evaluate the results of all three models on the test data set for each stock separately and discuss how our approach could be improved.

1 Introduction

Stock price prediction is one of the hot research topics in machine learning due to its real world application, but it is also one of the harder problems due to the unpredictable nature of the stock market. In this project we used one of the standard deep learning models to predict prices and then explore the idea of using data from social media to improve our predictions. We know that social media had a huge impact on society in the past decades and is quite a good indicator of popular opinions. We believe that we can leverage this to aid stock price prediction.

The source code of the project and all model weights can be found at https://github.com/gregorkovac/lsentiment.

2 Methods

In this section we describe all of the data we used and the models. We first describe the standard stock market prediction approach, then we describe extraction of sentiments from social media data, and lastly we show how we combined the two.

2.1 Stock market prediction

2.1.1 Data

For our stock market data we first wanted to focus on just one company, but later, due to the lack of long-term social media data, decided to take multiple stocks. We chose the five leading tech companies, associated with the acronym MAANG - Meta, Apple, Amazon, Netflix and Google. We obtained historical data for their stocks from Kaggle [1, 2, 3, 4, 5] and joined them into a single data set.

The data contained a *Date* column and multiple prices for each day (*Open*, *Close*, *Low*, *High*, ...). To simplify the problem, we used only the *Close* price, which is the price of the stock when the market closes on a certain day.

We split the data into training, validation and test sets with 70% and 90% splits. We took care that the splitting was not random, but rather temporal. After that we computed the mean and standard deviation of the training set price and applied standardization with these constants to all splits. We then applied a time window of 5 days to each price to generate new columns and prepare our data for the model. In validation and test sets we removed the first few rows to ensure that no data was being leaked from other sets.

2.1.2 LSTM

For our baseline model we used a *Long Short-Term Memory (LSTM) network* [6], which is a standard time series deep learning model. An LSTM is a type of a *recurrent neural network (RNN)*, which uses memory cells and gates to control information flow. It is known for its ability to learn long-term dependencies, which makes it a good model for a time series forecasting problem like ours.

2.2 Sentiment analysis

2.2.1 Data

The second part of our pipeline was obtaining some sort of data from social media to aid our time series forecasting. We chose to use posts from the social media *X* (former *Twitter*), which we will refer to as *tweets* from now on. For our metric we chose *sentiment*, which is commonly used in natural language processing and refers to the emotional tone of a text.

We took data from a financial influencer tweets data set from Kaggle [7]. This data set contains many different columns, but we were most interested in *timestamp* and *desciption* (which is just the text of the tweet). Note that the data set already contained a sentiment column, but we chose to leave it out because 1) we wanted to develop a full pipeline ourselves, and 2) the values in the column were not clean.

2.2.2 FinBERT

Bidirectional Encoder Representations from Transformers (BERT) [8] is a deep learning model specified in understanding context. It is based on a transformer architecture. It is very good at solving many natural language processing tasks, and among those is also sentiment analysis, which is what we needed for our problem. In the project we used *FinBERT* [9], which is BERT fine-tuned to financial texts. We chose this model because our data consisted of tweets from financial influencers.

To create a sentiments data set, we passed each tweet separately through FinBERT to obtain probabilities for three sentiments - positive, neutral and negative. We then computed the mean probabilities of each day.

2.3 Combining the two

We tried two different approaches of combining the original LSTM with sentiments, which we will present in this subsection.

For the data, we merged the stock prices with sentiments to obtain a single data set. We made sure that we properly aligned the values by the timestamp. We also applied the same time window of 5 days to all three sentiment values. Note that we removed the sentiments for the day of each row to prevent data leakage.

At the end, the training data set consisted of 1335 rows, the validation set had 310 rows, and the test set had 105 rows. We do acknowledge that this is not enough data to train a solid model that could be used in the real world.

2.3.1 A joint LSTM

The first (simpler) approach was to just use a single LSTM and concatenate prices and sentiments before the model. In other words, we would still use the same LSTM as before, but it would have more input features.

2.3.2 Two separate LSTM-s

The second idea was to have two LSTM-s: one for stock prices and one for sentiments. The final hidden states of both models were then merged at the end and passed through a multi-layer perceptron to obtain a final prediction.

3 Results

To ensure fairness in comparison and also to get the best performance possible, we ran hyperparameter optimization for all three models. The best chosen parameters for each model can be seen in Table 1. Note that not all parameters were used in all models, since they were not needed. We used mean squared error as our loss and we trained all models with the Adam optimizer and 2000 epochs.

With the chosen parameters, the baseline model achieved a validation loss of 0.0110, the sentiment model with a single LSTM reached a validation loss of 0.0100, and the sentiment model with two LSTM-s got a validation loss of 0.0098. All three models came close.

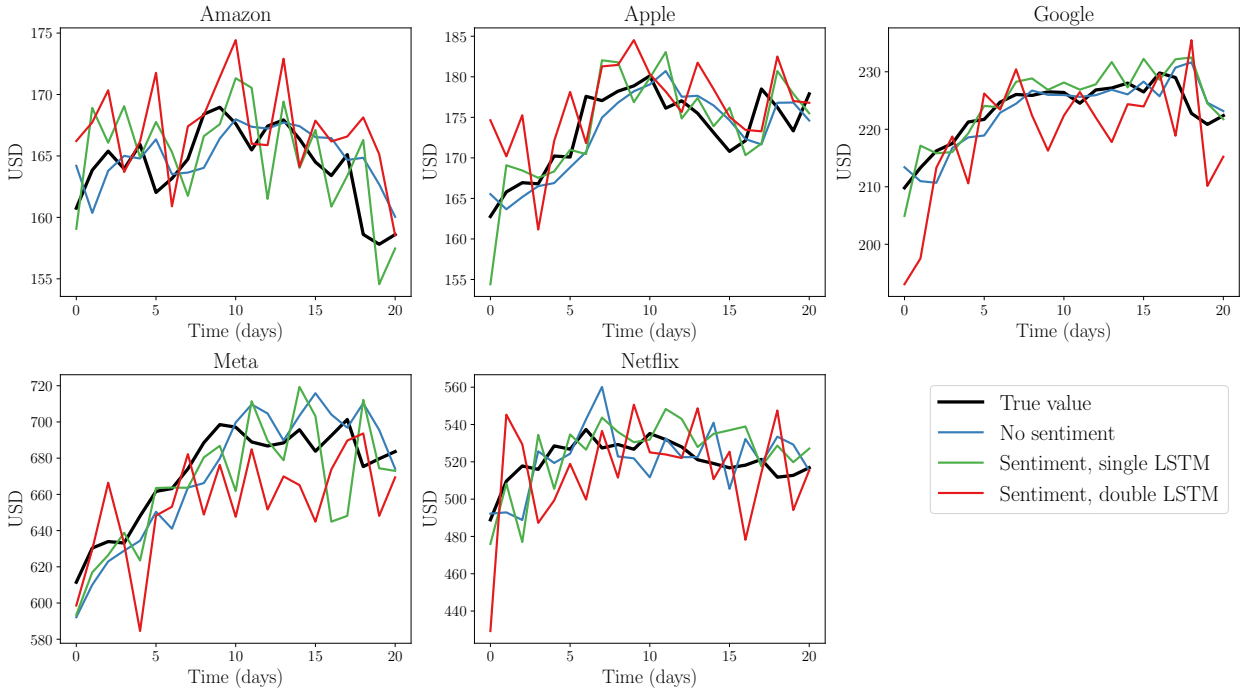

Figure 1: Results of all three models on the test set.

We then retrained all three models on the combined training and validation sets and evaluated them on the test set. The results can be seen in Figure 1. We visualized results for each of the MAANG companies separately. To make the results more representative, we converted them back to the real currency (USD) by destandardizing.

4 Discussion

We can observe that performance did not improve when using sentiments compared to the baseline model. In fact, it gets worse. There may be several reasons for this. One of them is that the sentiment data wasn't representative enough to actually improve predictions. In other words, it may have just added noise to the model. We compressed a day's worth of data into three sentiment probabilities, which resulted in a loss of information. We could have tried to create more complex features given the big tweet data size. We could have also tried using some other models to gain more insight into the problem and possible solutions. Another quite possible reason is that tweets just cannot be used to help much with stock prediction. Surprising things are constantly happening in the market which really cannot be predicted.

Otherwise, we drove this project to a close with an entire pipeline implemented. We are looking forward to possibly improving it in the future.

References

[1] *Amazon Stock Data 2025*. Abdul Moiz. Kaggle. https://www.kaggle.com/datasets/abdulmoiz12/amazon-stock-data-2025

	baseline LSTM	single LSTM	double LSTM
lstm1_hidden_size_1	500	350	150
lstm1_num_layers	1	1	1
lstm2_hidden_size	.	.	150
lstm2_num_layers	.	.	1
mlp_layers	.	.	1
mlp_hidden_size	.	.	400
lr	$4 \cdot 10^{-5}$	$3 \cdot 10^{-3}$	10^{-4}
batch_size	64	64	64
use_scheduler	no	yes	yes
scheduler_step	.	950	450
scheduler_gamma	.	0.76	0.72
weight_decay	0.0002	0	0
dropout_rate	0.4	0.2	0.3

Table 1: Chosen hyperparameters for all three models

[2] *Apple updated stock complete dataset*. Matif Latif. Kaggle. https://www.kaggle.com/datasets/matiflatif/apple-stock-complete-dataset

[3] *Google Stock Price (2018 - 2025)*. Soroush Esnaashari. Kaggle. https://www.kaggle.com/datasets/soroushesnaashari/google-stock-price-2018-2025

[4] *Meta Stock Data 2025*. Umer Haddii. Kaggle. https://www.kaggle.com/datasets/umerhaddii/meta-stock-data-2025

[5] *Netflix Stock Data - Live and Latest*. Kalilur Rahman. Kaggle. https://www.kaggle.com/datasets/kalilurrahman/netflix-stock-data-live-and-latest?select=Netflix_stock_history.csv

[6] *Long Short-Term Memory*. Sepp Hochreiter, Juergen Schmidhuber. 1997. Neural Computation, Volume 9, Issue 8, Pages 1735 - 1780.

[7] *Financial Tweets*. Amy. Kaggle. https://www.kaggle.com/datasets/amulyas/financial-tweets

[8] *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. 2019. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Volume 1. Pages 4171 – 4186.

[9] *FinBERT: Financial Sentiment Analysis with Pre-trained Language Models*. Dogu Araci. 2019. ArXiv.