Predicting the Present with Google Trends and News Headlines

Orest Gkini, Theofilos Belmpas, Dimitrios Chalatsis
EPFL, Switzerland

***Abstract*—In this report, we present our extension of the work done in [1], where the authors employ search engine statistics, specifically Google Trends, to enhance the forecasting of near-term values of economic indicators. We apply their methodology on two datasets of stock prices, and showcase how we manage to improve predictions in periods of uncertainty by utilizing sentimental analysis of financial news headlines.**

I. Introduction

The authors in [1] explore the claim that trends in Google queries can be significant leading indicators of consumer behavior. However, they are not claiming that they can use trends to predict the future, but rather to perform *contemporaneous* forecasting, which in a sense can be described as "predicting the present", by combining an autoregressive model with trends data as exogenous variables. The idea is that if Google queries show an upward for a specific product during the current month, then the company can predict the revenue of this month using sales of previous month combined with the current search trends it is observing.

In this report, we attempt to employ the same technique to predict stock market prices. More specifically, we are interested in the prices of the *S&P 500* market index and *Bitcoin*. Our intuition is that the same approach may not yield the same positive results as in the original paper, since an increase in searches about certain stocks can indicate that people are just checking the prices during a period of economic crisis or because of major societal invents that may negatively impact the economy in the immediate future, e.g. a pandemic. Our approach towards mitigating this phenomenon is to include *sentimental analysis* of news headlines from the same periods, which can give us an insight about whether the increase in search trends is because of investors' confidence or uncertainty.

The rest of this report is structured as follows: in Section II we describe our dataset, in Section III we describe the autregressive models and how we embed the sentimental analysis in the predictions, in Section IV we report our experimental results, and finally, in Section V we provide a brief conclusion.

II. Dataset

Our dataset consists of the following:

- The prices of the *S&P 500* market index for a period of 17 months starting from 01-10-2019 until 31-05-2020.
- The prices of *Bitcoin* for a period of 41 months starting from 01-12-2016 until 30-04-2020.

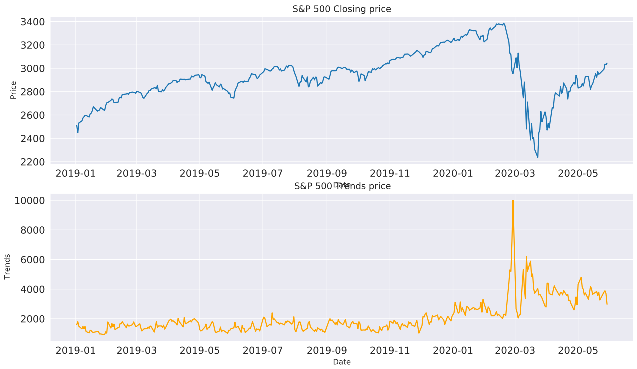

Figure 1. S&P 500 daily closing prices from 01-01-2019 to 31-05-2020 (top) and the Google Trends index for the search term *"S&P 500"* for the same period (bottom).

- The Google Trends data for the search terms *"S&P 500"* and *"Bitcoin"* for the same periods as above. We have limited the location of our search queries to those coming from the United States.
- Two sets of news headlines related to the financial market and bitcoin for the same time periods as above.

A. Data Collection

The S&P 500 and Bitcoin prices were manually downloaded from [2], which provides an interface that allows the user to specify the desired time range and provides the data in a `csv` file format.

To collect the trends we used `pytrends` [3], which provides an unofficial api for gathering data from Google. However, the tool provides the data with a daily frequency only if the requested period is one month (or less), so in order to collect them for our specified time ranges, we request them on a monthly basis and aggregate them.

The news headlines were also manually downloaded from [4], which provides financial related news for more than 6000 stocks, and [5], which includes cryptocurrency related news. We only kept the news for the respective date periods mentioned before.

B. Data Exploration and Analysis

Visual exploration. Figure 1 shows the daily S&P 500 closing prices and the corresponding trends data for our specified time period. A noteworthy observation is that around the beginning of March 2020, when the S&P 500 index price was at its peak, the searches experienced a rapid

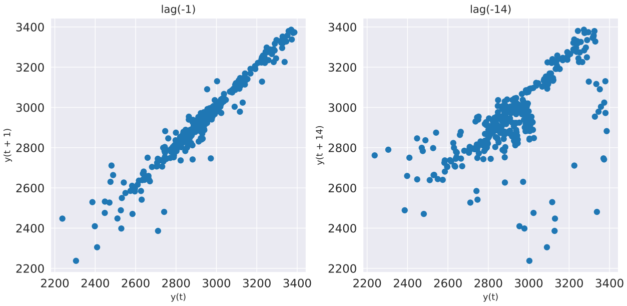

Figure 2. Autocorrelation between the S&P 500 price at time t and the prices at times: (a) $t-1$ (left), and (b) $t-14$ (right).

surge that was immediately followed by a sudden drop in the price, which continued until the price reached its lowest point of the whole period. That period corresponds to the peak of the corona virus pandemic, during which people frequently searched the price of the index while it was decreasing. Therefore, we observe an inverse correlation between prices and trends during that time.

Figure 2 shows the autocorrelation between the price of the S&P 500 index at time t and the price at time $t-1$ and $t-14$. In other words, it shows how correlated is the price of a certain day with the prices of the day before that and two weeks ago. We observe a linear correlation in the former case, which is logical since prices do not vary greatly on a daily basis, and a looser correlation in the latter case.

Statistics. The average number of daily news headlines for the S&P 500 data is 480, while for Bitcoin it is 30.

Missing values. The S&P 500 index prices for the weekends are not available in the dataset since the stock market is closed on weekends. Therefore, we do not consider the trends for the weekends either. However, we also observed that some weekdays (13 values) were missing, too. For those days, we set the price to be equal to the price of the previous open day (for Mondays it is Friday). Bitcoin can be traded on all days so there are no missing days.

III. Model and Methodology

In this section, we describe the autoregression model that was used in the original paper and present our approach to embedding news headlines in the prediction pipeline.

A. Autoregression

The authors of [1] use an autoregressive model which uses values from previous time steps (*lags*) as an input to a regression equation in order to predict the values at the current time step. For example, in the case of daily stock prices a *lag-1* corresponds to the price of the previous day. If we use the *lag-1* and *lag-2* values the regression formula has the following form:

$$p_t = b_1 p_{t-1} + b_2 p_{t-2} + b_0 \tag{1}$$

where p_t indicates the stock price at time t.

Table I
Overall behavior of MAE for S&P 500.

MAE base	MAE trends	MAE news
28.01	28.68	28.52

Table II
Overall behavior of MAE for Bitcoin.

MAE base	MAE trends	MAE news
208.04	208.72	208.56

B. Sentimental Analysis

Sentimental analysis is the process of analyzing a natural language phrase and assigning a value to it that measures how positive (value greater than zero) or negative (value less than zero) that phrase is. For the sentimental analysis of the news headline we use TextBlob [6], which is a Python library for natural language processing tasks. Given a natural language phrase as input, TextBlob returns two values from which we are only interested in the *polarity*, which is a float number in the range $[-1, 1]$, essentially classifying the input as negative, positive, or neutral. We embed this in our predictions by multiplying the trends index for each day with the average news sentiment across all news of that day. Thus, if the trends are going up on a single day because of some bad news, the negative polarity value will reverse the effect of the trends and, thus, drive the model to make better predictions.

C. Training and Predictions

For the training of the autoregressive model we follow the same approach as in the original paper. That is, to predict a value at time t we train the model using the data from time steps k to $t-1$. The value of k must be set so that the model has enough data to train in order to make the initial predictions, thus, the predictions do not start from the first date in our dataset, but from the $(k+1)$-st one.

IV. Experiments

In this section we present the results of our experiments for the three models: (i) baseline model that uses only the lags (*base*), (ii) baseline model enhanced with trends data (*trends*), and (iii) the model that incorporates the news, too (*news*). First, we report the overall results and, then, we focus on periods around *turning points*, meaning points around which there was a sudden change (either drop or increase) in the prices. We use lags 1, 2, 3, 4, and 5 for all models.

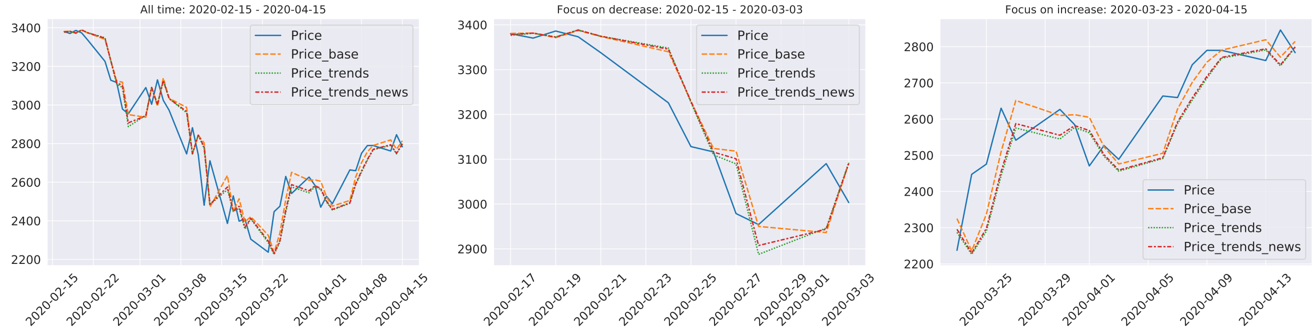

Figure 3. Predictions of our three autoregressive models for the time periods shown in Table III.

Table III
BEHAVIOR OF MAE AROUND TURNING POINTS FOR S&P 500.

Start	End	MAE base	MAE trends	MAE news
2020-02-15	2020-04-15	91.11	92.18	91.68
2020-02-15	2020-03-03	57.05	59.83	58.57
2020-03-23	2020-04-15	72.11	83.59	81.49

Table IV
BEHAVIOR OF MAE AROUND TURNING POINTS FOR BITCOIN.

Start	End	MAE base	MAE trends	MAE news
2017-11-01	2018-02-01	703.63	707.24	706.64
2017-11-01	2017-12-16	509.54	519.77	515.74
2018-01-15	2018-02-05	592.41	591.45	599.48

A. *Overall Results*

Tables I and II show the Mean Absolute Error (MAE) of the models over the whole periods we are considering, for S&P 500 and Bitcoin, respectively. We see that in both cases the base model performs best, but we are mostly interested in in the comparison between the trends model and the news model, to evaluate the impact of our sentimental analysis process. We see that in both cases the news model is marginally better than the trends model.

B. *Turning Points*

Now, we focus on periods around turning points in our time series. Our goal is to evaluate whether embedding sentimental analysis of news headlines in the predictions improves the accuracy of the model in periods with sudden changes in the market. We present our analysis for S&P 500, since our observations are more obvious in this case.

S&P 500. Table III shows the mean absolute errors of our three models for an important turning point in our data. Figure 3 shows a visualization of the actual prices and the predictions

The selected period spans from 2020-02-15 to 2020-04-15, which corresponds to the peak of the coronavirus pandemic (first row of the table). The next two rows contain two smaller sub-periods of that period, where in the first one the price is still decreasing, while in the second the price has started increasing.

We see that again in all cases, the baseline model performs best. However, comparing the trends model and the news model, we can see that in all periods our news-enhanced model achieves improved the accuracy of the predictions, achieving a smaller mean absolute error.

A noteworthy observation that we can make by inspecting the second plot of Figure 3 is that around 2020-02-27, while the price was experiencing a downward trajectory, it suddenly went up, before going down again. Comparing the predictions of the models there, we see that the model with the news made a prediction closer to the actual price than the trends model.

V. CONCLUSION

In this report, we experimented with enhancing an existing autoregressive model that utilizes past data and Google Trends to predict the values of economic indicators, by embedding results from sentimental analysis of news headlines. We applied our approach to two datasets related to stock prices: (i) the S&P 500 market index, and (ii) Bitcoin prices. Our experiments show that our method slightly improves the performance of the model that only utilizes the Google Trends data. A more powerful sentimental analysis based on a neural network architecture may be more accurate and, thus, yield better predictions.

REFERENCES

[1] H. Varian and H. Choi, “Predicting the present with google trends,” *Economic Record*, vol. 88, 04 2009.

[2] “S&p 500 and bitcoin historical price data.” https://finance.yahoo.com/.

[3] “Pytrends.” https://pypi.org/project/pytrends/.

[4] “Daily financial news for 6000+ stocks.” https://www.kaggle.com/miguelaenlle/massive-stock-news-analysis-db-for-nlpbacktests?select=analyst_ratings_processed.csv.

[5] “Cryptocurency cointelegraph newsfeed.” https://www.kaggle.com/asahicantu/cryptocurency-cointelegraph-newsfeed?select=cointelegraph_news_content.csv.

[6] “Textblob.” https://textblob.readthedocs.io/en/dev/.