Is it possible to improve the forecast of the S&P 500 using Sentiment Analysis?

Text Mining and Sentiment Analysis' Project, UniMi

Guglielmo Berzano, ID: 13532A

E-mail: guglielmo.berzano@studenti.unimi.it

October 2024

Abstract

The main objective of this work is to understand the effectiveness of sentiment analysis techniques in improving the precision of 1-step-ahead forecasts of the S&P 500 index. Many papers have recently demonstrated that, by introducing a measure accounting for economic or financial sentiment in standard time series models, forecasts can be substantially more accurate. This study wants to compare the results of augmenting a simple univariate ARIMA$(1, 1, 0)$ with the average daily sentiment of financial news scraped from CNBC, Reuters and the Guardian. In particular, two models will be used to get the sentiment: a rule-based one (VADER) and a transformer-based one (FinBERT). Later, they will be included, one at the time, in the ARIMAX$(1, 1, 0)$ to see whether this is beneficial with respect to a simple ARIMA$(1, 1, 0)$ taken as benchmark.

Keywords: FinBERT, VADER, Sentiment Analysis, S&P500, Finance, Real-Time Forecasting.

1 Introduction

Finding ways to accurately forecast the market has been a dream that dates back to when the financial markets were established. According to the Efficient-Market Hypothesis, market prices are the reflection of all the knowledge available in a certain moment in time and thus it is impossible to precisely forecast future outcomes because future information is missing (Malkiel, 1989). Nevertheless, this did not stop researchers to put effort in developing mathematical models to predict future evolutions in finance.

In this sense, time series models like AutoRegressive Integreted Moving Average (ARIMA (p, d, q)) or more complex Long- Short-Term Memory (LSTM) neural networks have been widely studied to understand their efficacy in forecasting financial markets. In the latest years, researchers from all around the world tried to augment standard models with a measure accounting for sentiment of economic activity to improve the quality of forecasts. Nguyen et al. (2015) opened the way to stock movement predictions using sentiment. Later, Caporin and Poli (2017) demonstrated that news-related variables can improve volatility predictions. The methods used in the literature to get the sentiment vary significantly but in this project I will analyze and compare the results obtained by implementing a rule-based model (VADER), as in Li et al. (2019), with those obtained through a fine-tuned version of BERT for financial data: FinBERT, as in Jihwan et al. (2023).

Valence Aware Dictionary for sEntiment Reasoning, in short VADER, (Hutto and Gilbert, 2014), is a rule-based model developed primarily to analyze social media text. This model includes, on top of words retrieved by word-banks like Linguistic Inquiry Word Count (LIWC), 7,500 *lexical features* that are typical to social medias like emoticons (":-)") and informal acronyms ("LOL"). The researchers acknowledged that the sentiment of such expressions is not precisely defined *a-priori* and thus a *wisdom-of-the-crowd* approach was implemented. In particular, ten independent human raters were asked to give their opinion on a scale from "$[-4]$ Extremely Negative" to "[4] Extremely Positive" to every single *lexical feature*. Subsequently, according to the resulting average, the *lexical-features* received a *polarity* – either positive or negative – and an *intensity*, expressed in form of a value $\in [0, 4]$. Moreover, VADER captures differences that "go beyond what would normally be captured in a typical bag-of-words model" (Hutto and Gilbert, 2014), among the differences are those between lowercase and all caps words ("good" vs "GOOD") and single versus multiple punctuation marks ("!" vs "!!"). This method proved to be very effective in detecting the sentiment of twitter posts and showed interesting results also in other contexts like New York Times Editorials.

FinBERT instead, is a domain specific version of BERT (Bidirectional Encoder Representations from Transformers) developed originally by Araci (2019). As other transformer-based methods, FinBERT is trained in two steps: pre-training and fine-tuning. In the first step, Araci (2019) proceeded by re-training BERT with a large amount of financial-related texts consisting of 400K sentences and around 29 million words. In the second instead, he fine-tuned the model so that it was able to avoid the danger of *catastrophic forgetting*. Another version of FinBERT was developed by Huang et al. (2023) who re-trained BERT from scratch on a large corpus of financial texts including 4.9 billion total tokens. Among tasks learned during the fine-tuning, there is a more complex sentiment classification with respect to Araci (2019). Moreover, Huang et al. (2023) showed that FinBERT improved with respect to BERT not only in the financial sector, but also for topics like environment and governance.

The forecasting exercise was thought as follow: the user knows, at time t, just the closing market value of $t-1$, hence not the opening price of t. At t, they collect the most recent financial-related news and compute, using VADER and/or FinBERT the average sentiment of the market on that day. Finally, they implement a time series model, augmented with the sentiment, to forecast the closing price of that day. Thus, by having access to S&P 500 values up to $t-1$ and news up to t, the forecaster wants to predict the directional change and the point value of the index at time t.

All the code is available the GitHub repository for this project that you can access here.

2 Data gathering and preprocessing

News data are retrieved from this Kaggle dataset which contains scraped financial-related headlines and articles' descriptions from December 17th, 2017 to July 19th, 2020, for a total of 649 business days, which increase to 927 if also holidays and weekends are considered. In total there are 51,806 news, 1,276 obtained from CNBC, 17,760 from the Guardian and 32,770 from Reuters organized in three columns: *Date*, *Headline* and (Article) *Description*. Due to limitations in the computational power, however, only headlines were analyzed even though this strategy is not optimal. Indeed, news titles may convey exaggerate sentiment with respect to what it is contained in the actual article as a form of clickbait. Figure 1 displays how many news are available per day. The maximum was reached on March 19th, 2020 during the Covid-19 outbreak when 170 news were published while the average number of news in each day is 55.86.

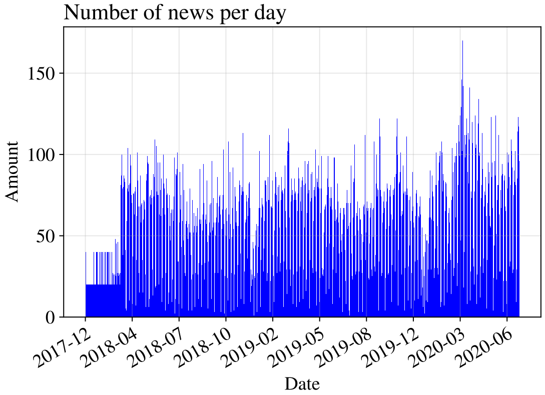

Figure 1: Number of headlines per day.

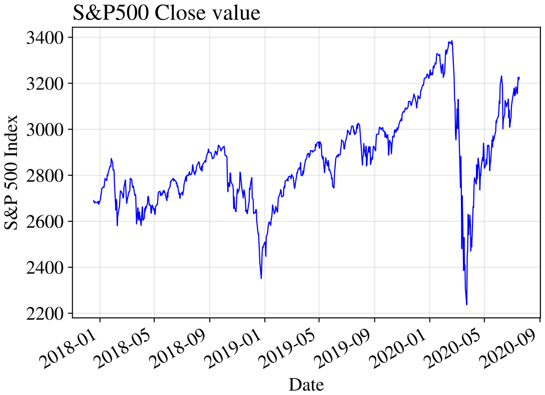

Figure 2: S&P 500 closing value in the analyzed period.

Financial historical data are collected via the `yfinance` library of Python which, in turn, uses the Yahoo! Finance's API. The analysis will focus on the Standard and Poor's 500 index, commonly known as S&P 500, which is a broad-based stock market index that includes and tracks about 500 traded companies in the United States (Ashburn, 2024). Among the sectors included in this index, there are information technology, real-estate and financial services. The `yfinance` library returns a `Pandas` dataframe composed by 8 columns: Date, Open, Close, High, Low, Volume, Dividends and Stock Splits. In particular, this analysis will focus just on the mono dimensional series obtained by the Close column which represents the closing value of the index on each day. Figure 2 presents the S&P 500 close value for the considered time period. The index has visually random walk properties, without significant medium- to long-term trends or cycles. For this reason, successfully and beating the market has always been, and still is to this day, a wishful operation.

3 Sentiment Analysis

In this project I will implement the FinBERT of Huang et al. (2023), available here, via the `pipeline` of the `transformers` Python library, while VADER via the `VADERSentiment` library. Essentially, since these models are ready-to-use, no tokenization or lemmatization are needed. The *Headlines* column of the dataset created earlier is then passed, one row at the time, to the FinBERT's and VADER's functions to compute the polarity. The FinBERT `pipeline` returns the predicted label, among *positive*, *negative* and *neutral* with the certainty that the model has towards that prediction. Instead, the VADER's `SentimentIntensityAnalyzer()` produces a dictionary composed by four key-value pairs. The first three items have, as keys, the labels *positive*, *negative* and *neutral* and, as values, the predicted intensity of their respective keys. Instead, the fourth key-value pair represents the actual prediction of the model: the key is *compound*, and the value is a normalized number between $[-1, +1]$ where -1 stands for extremely negative, $+1$ stands for extremely positive and values around zero stand for neutral. Subsequently, according to the *compound* predictions, I transformed all the values greater than 0.33 into 1, those below -0.33 into -1 and the rest into 0. Figure 3 contains three pie-charts showing, in the first two cases, the percentage of each label according to the two models and, in the third, the differences in labeling. In particular, the third pie-chart's labels "Equal", "Different" and "Opposite" mean respectively that the two models made the same prediction, that one guessed *neutral* while the other either *positive* or *negative* and that the prediction, for the same headline, was completely opposite, one model guessed *negative* while the other *positive*.

The first two panels show that the percentages of sentences labeled as *positive*, *negative* and *neutral* are very similar for both models. In particular, according to VADER, there are 1% less *neutral* sentences with respect to FinBERT, around 3.5% less *negatives* and 4.5%

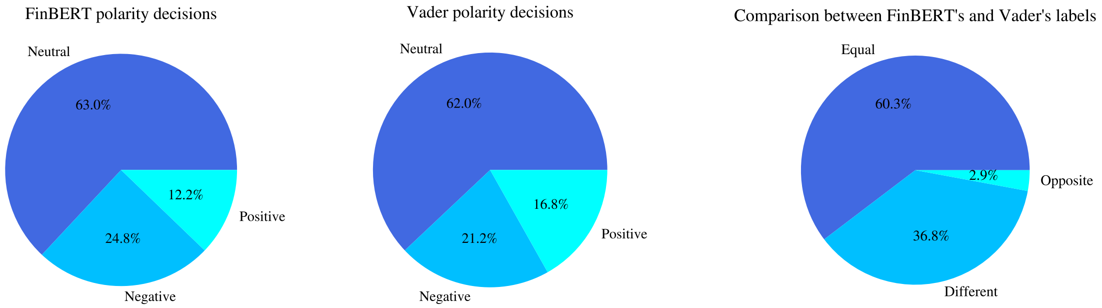

Figure 3: Polarity percentages per model and difference in “opinions”.

more *positives*. These differences are further analyzed in the third panel which shows that 60.3% of sentences have been labeled the same way, 36.8% received different labels while the remaining 2.9% were labeled in an opposite way. One of the reasons may be that VADER was not purposely built to classify financial text but social media posts instead. This may have provoked a difficulty for the model to correctly classify the headlines’ sentiment.

After having computed the sentiment for every headline, I took the daily average to obtain the sentiment that the market had on each day. Since news were retrieved also on days in which the stock exchange was closed, in order to align the daily sentiment data to the S&P 500, I moved the sentiment of closed-market days to the nearest open-market day and took the average. For example, the sentiment of Saturdays and Sundays was brought ahead and averaged to the one of Mondays. Moreover, there were five non-consecutive days in which the market was open without any news available. In these cases, I used, as sentiment of t, the average between $t-1$ and $t+1$. Figure 4 shows the difference between the average daily sentiment computed with the two models while Figure 5 depicts the correlation between the S&P 500 and the two computed sentiments.

By looking at Figure 4, it is possible to notice that VADER’s sentiment tends to be closer to zero and to lay above the one of FinBERT. Even though the standard deviations are quite similar, $\sigma_V = 0.092$ for VADER and $\sigma_F = 0.089$ for FinBERT, the averages are quite different: $\overline{s}_V = -0.039$ and $\overline{s}_F = -0.122$ for VADER and FinBERT respectively. Since, on average, the FinBERT’s sentiment is more different from zero relative to VADER’s and tends to stick slightly more to its mean, it will certainly be more impactful – either positively or negatively – in the final model. Figure 5 illustrates that the correlation between the S&P 500 and VADER is lower with respect to the one with FinBERT. This may signal that FinBERT captures a sentiment that is more closely related to the evolution of this economic indicator.

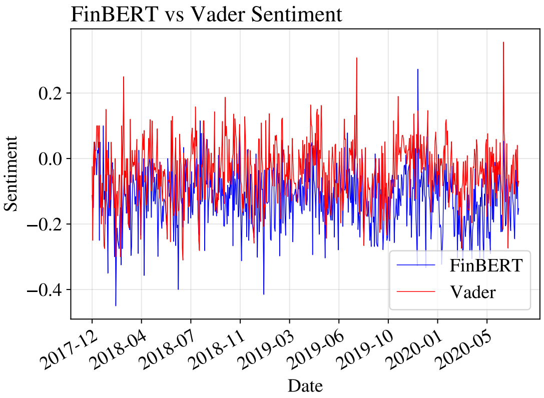

Figure 4: FinBERT vs VADER Daily Sentiment.

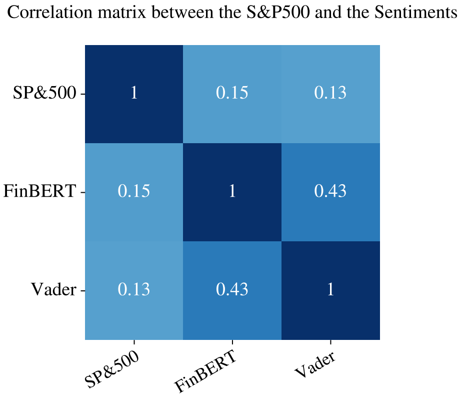

Figure 5: Correlation between the sentiments and the S&P 500.

4 Forecasting with Sentiment

Forecasts are made using the `forecast` package of `R`, see Hyndman and Khandakar (2008), with a univariate Auto Regressive Integrated Moving Average eXogenous model, ARIMAX $(1, 1, 0)$ defined as:

$$y_t = c + y_{t-1} + \beta x_t + \phi_1 \Delta y_{t-1} + u_t \, ,$$

where y_t is the S&P 500 at time t, c is a constant, x_t is the exogenous variable at time t which, in this case, is the financial sentiment, multiplied by a coefficient β, $\phi_1 \Delta y_{t-1}$ is the autoregressive element where $\Delta y_{t-1} = y_{t-1} - y_{t-2}$ multiplied by the coefficient ϕ_1 and u_t is the error at time t distributed as a Guassian white noise with mean 0 and variance σ^2. This model was chosen for its simplicity in interpretation, rapidity of estimation and accuracy in forecasts. I am following the result obtained by Serafini et al. (2020) who discovered that, for Bitcoin price prediction, an ARIMAX performs as well as, and in some cases even outperforms, a LSTM neural network. Moreover, since the scope of this project is not to find the best time series model to predict the price but just to visualize the differences between adopting or not a regressor accounting for sentiment, the simplest time series model for nonstationary series, hence the ARIMAX$(1, 1, 0)$ will be employed.

The forecasting exercise starts on March 2$^{\text{nd}}$, 2019, so that the forecaster has access to 300 observations and 300 average daily sentiments. Three models will be estimated each day: ARIMAX$(1, 1, 0)$ with FinBERT's sentiment, ARIMAX$(1, 1, 0)$ with VADER's sentiment and ARIMA$(1, 1, 0)$ as benchmark. Each series will be transformed in natural logarithm before estimating the models. Errors are measured in terms of Root Mean Squared Prediction Error (RMSPE) and Mean Directional Accuracy (MDA). The MDA is particularly useful when working with financial data because the objective may not be to predict the exact value that the market will have in the future but just whether it is expected to increase or decrease. Essentially, each time that the model correctly predicts the sign change of the series, thus $\text{sign}(\hat{y}_t - y_{t-1}) = \text{sign}(y_t - y_{t-1})$, 1 is stored inside a vector ν. Otherwise, 0 is stored. Taking the mean of the vector ν yields the MDA. Results are displayed as follows.

Table 1: Results for 1-step-ahead forecast. Best values in bold.

Model	RMSPE	MDA
ARIMAX$(1,1,0)$, FinBERT	**48.500**	**0.625**
ARIMAX$(1,1,0)$, VADER	48.980	0.604
ARIMA$(1,1,0)$	49.103	0.573

The ARIMAX$(1,1,0)$ augmented with the FinBERT's sentiment is clearly the best model out of the three analysed both in terms of RMSPE and of MDA. The largest improvement is observed in the MDA which, as mentioned, is typically the most relevant metric for these exercises. Numbers show an increase of 5% with respect to the benchmark for the model augmented with the VADER's sentiment and an increase of 9% for that augmented with the FinBERT's one. For the RMSPE, the improvement is substantially more constrained: just 0.3% better than the benchmark for VADER and 1.24% for FinBERT. In general, augmenting the base model with one regressor accounting for sentiment improves the quality of both the point forecasts and of the directional change, regardless of the sentiment analysis model chosen. Even though the model augmented with FinBERT's sentiment performs better, the VADER model may be an interesting choice since it still provides better results compared to the benchmark and is extremely efficient in terms of computational cost.

Figure 6 depicts the actual S&P 500 values and the forecast obtained by the ARIMAX $(1,1,0)$ with the FinBERT's sentiment. The left panel shows the performance over the entire forecasting period while the right panel focuses on the last four months and a half, during the outbreak of COVID-19.

Visually, the series of predicted values follows extremely closely the one of actual values. This was expected since the value of the RMSPE in Table 1 is relatively low. However, by zooming in, right panel, it is possible to visually notice that error is still quite high, both in the point- and in the directional-predictions. However, it is true that the sentiment augmented models are able to capture a decent amount of additional information, especially regarding directional change, with respect to the one grasped by the benchmark.

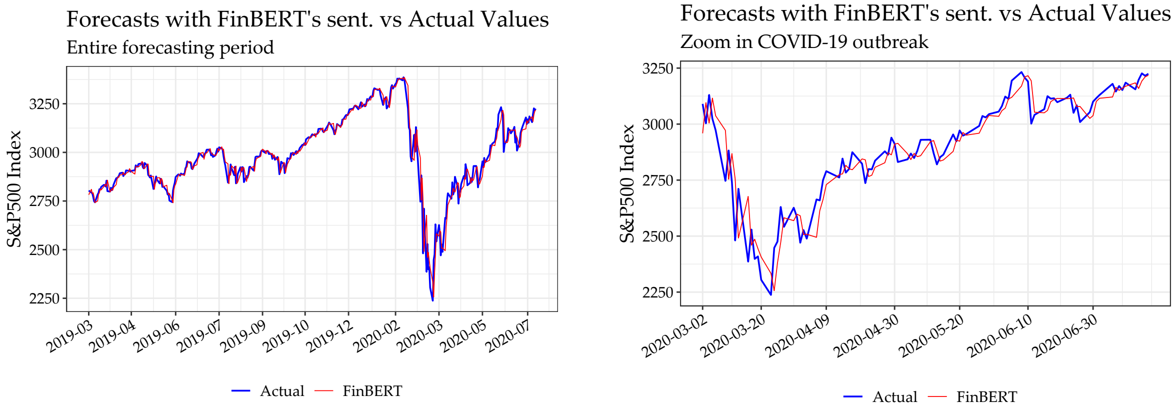

Figure 6: Forecast for ARIMAX$(1, 1, 0)$ with the FinBERT sentiment vs actual values.

5 Conclusion

The reason behind this work can be found in the interest that the scientific literature is developing towards sentiment analysis techniques for forecasting. In particular, this study began by analyzing a Kaggle's dataset comprising of more than 53,000 news scraped from CNBC, Reuters and the Guardian. News headlines were then passed to FinBERT and VADER to compute the individual sentiment. Next, the daily average was stored and was placed side by side to the S&P 500 closing value according to the date. Later, an ARIMAX$(1, 1, 0)$ was run including the average daily sentiment obtained by the two models – one at a time – and finally compared to a standard ARIMA$(1, 1, 0)$. Results show that including the sentiment can improve the mean directional accuracy up to 9% and reduce the RMSPE up to 1.24% [1].

[1] The `R` file containing the code for this project includes also a part in which I tried to improve the results. I was able to reach a RMSPE of 31.973 and a MDA of 0.719 by inserting, as another regressor, the opening price of t. This improved the overall results by 26.14% for MDA and by 34.97% for RMSPE with respect to the original benchmark.

References

Araci, D. (2019). Finbert: Financial sentiment analysis with pre-trained language models.

Ashburn, D. (2024). S&P 500. https://www.britannica.com/money/SandP-500, Accessed on October 24th, 2024.

Caporin, M. and Poli, F. (2017). Building news measures from textual data and an application to volatility forecasting. *Econometrics*, 35(5).

Huang, Allen, H., Wang, H., and Yang, Y. (2023). Finbert: A large language model for extracting informationfrom financial text. *Contemporary Accounting Research*, 40(2):806–841.

Hutto, C., J. and Gilbert, E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, volume 8.

Hyndman, R. J. and Khandakar, Y. (2008). Automatic time series forecasting: the forecast package for R. *Journal of Statistical Software*, 27(3):1–22.

Jihwan, K., Hui-Sang, K., and Sun-Yong, C. (2023). Forecasting the S&P 500 Index Using Mathematical-Based Sentiment Analysis and Deep Learning Models: A FinBERT Transformer Model and LSTM. *Axioms*, 835(12).

Li, X., Li, Y., Yang, H., Yang, L., and Liu, X.-Y. (2019). DP-LSTM: Differential privacy-inspired LSTM for stock prediction using financial news. *arXiv preprint, arXiv:1912.10806.*

Malkiel, Burton, G. (1989). Efficient market hypothesis. *Finance*, pages 127–134.

Nguyen, T. H., Shirai, K., and Velcin, J. (2015). Sentiment analysis on social media for stock movement prediction. *Expert Systems with Applications*, 42(24):9603–9611.

Serafini, G., Yi, P., Zhang, Q., Brambilla, M., Wang, J., Hu, Y., and Li, B. (2020). Sentiment-driven price prediction of the bitcoin based on statistical and deep learning approaches. *International Joint Conference on Neural Networks (IJCNN).*