Comparison of predictive power of historical and news for stock market movement

Dominik Klepl
Faculty of Engineering Environment and Computing
Coventry University
Coventry, UK
klepld@uni.coventry.ac.uk

Abstract*— Predicting the stock market is known to be complex yet intriguing problem. Two approaches are presented and compared in this paper. There is evidence that using news as source of features provides more predictive power than using previous values of the stock market. In this paper, an attempt has been made to replicate these results. However, on our dataset no difference in predictive power was found.

Keywords— machine learning, stock market, regression, classification, natural language processing

I. Introduction

Ability to predict stock market is attractive for two main reasons. From scientific point of view such predictor might help to understand the mechanism of the stock market better. The second reason is simply beating the market and making a profit off of it. However, according to theories of stock market achieving this should be nearly or completely impossible [1].

Efficient Market hypothesis says that stock price is determined by all available information including previous values and news. The market corrects itself once new information is available, and long-term prediction is therefore unlikely [2]. Random walk hypothesis posits even more extreme idea that even if one would possess all required information prediction would be impossible since the price is determined randomly [3]. Despite the theories' predictions multiple attempts with varying level of success were made to predict the stock market.

Several papers focused on prediction using only historical prices. Patel et al achieved accuracy between 73.3% (Gaussian Naïve Bayes) and 83.56% (random forest) in binary classification of the direction of the stock movement compared to previous values, i.e. up or down [4]. Support Vector Machine based solution proved to be less effective in similar setup with accuracies of 54.73% and 58.52% on testing and training data respectively [5].

Another approach that yields some limited success is based on assumption that large portion of information about stock market is embedded within news articles. Schumaker and Chen [6] used three representations of such texts; bag of words, noun phrases[1] and named entities[2] to predict both the direction of the stock movement and its magnitude. To prove the usefulness of their model, the performance was compared to baseline model trained on historical values. The best directional accuracy was 57.1% using noun phrases compared to 47.8% of the baseline model. The regression models of magnitude had mean squared error of 0.03 and 0.07 for named entities and baseline respectively.

Sentiment and opinion were also shown to be somewhat predictive of stock direction [7]. Predictive accuracy of 0.59% was achieved in this paper.

The aim of this paper is to replicate the results reported in [6] but with using sentiment and subjectivity representation methods similar to those in [7] in addition to n-gram representation which is extended version of bag of words used in [1] as explained in later section. In other words, building predictive models trained on textual representations that perform significantly better than baseline models trained on previous values only. As documented by previous research, not only is predicting changes of stock prices difficult but also predicting just the direction of the movement is difficult. Therefore, we decided to adopt a 2-stage modelling strategy. This means that we first build models to predict the direction of the movement and used these predictions to improve performance of the second stage regression models predicting the magnitude of the change.

The rest of the paper is structured in following way:

- Dataset description and the process of data cleaning
- Detailed process of methods used for feature extraction
- Experimental design starting with system structure, brief overview of used machine learning algorithms and reasoning for choosing those and evaluation methods
- Results of the experiment and selection of two best models
- Diagnostics and interpretation of the best models
- Discussion of the results, limitations of the experiment and suggestions for further analysis

II. Data description and data cleaning

For the experiment, we used dataset published on Kaggle. The dataset consists of 1989 samples.

The daily stock data is the Dow Jones Industrial Average (DJIA) collected from 08/08 2008 to 07/01 2016.

25 most popular news headlines were collected for each day in the same range. The popularity of the news was rated by Reddit users on Reddit WorldNews Channel. The target variable is the difference between opening and closing values reported in DJIA (Figure 1). An example of raw data point is shown in Table 1.

10% of the dataset was held out for evaluation of the models. Since the data is a time series using random splitting is not recommended because values are autocorrelated, i.e. there is interdependency between data points [8]. Instead the testing set is the last, i.e. newest, 10% of the data. There are no missing values in the data. The 25 columns with news headlines were merged into single column separated by whitespace for the purposes of data cleaning and later feature extraction. To obtain text in suitable format for further processing the HTML tags, e.g. " 'b ", punctuation, accents as well as numbers were removed.

[1] Using lexicon to tag nouns in the text and parts of speech (such as adjectives) in close proximity to the noun form noun phrases.

[2] Building on noun phrases, named entities are nouns and noun phrases classified as a e.g. place or person.

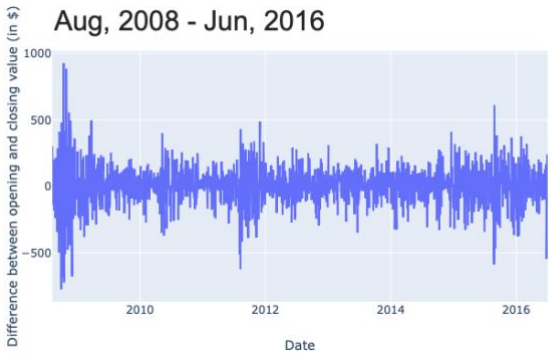


FIGURE 1 THE TARGET VARIABLE, DAILY DIFFERENCE OF THE STOCK VALUE

III. FEATURE EXTRACTION AND SELECTION

Two categories of features were extracted from the clean data: time series features and natural language processing (NLP) features.

There are 11 time series features. 4 of these features are literally time features as they were extracted from the date. These are day of week, month, quarter of the year and year. Furthermore 7 lags of the target variable were created allowing the model to see one week of previous values.

NLP features comprise of 3 methods: n-gram representation, sentiment extraction and subjectivity extraction.

N-gram representation with n = [1,2] was used. The process of creating n-grams begins with tokenizing the texts i.e. splitting on word-level. 1-grams are just single tokens. 2-grams are pairs of tokens located next to each other in the text. Finally, the frequencies of each of the n-grams are computed. We used the n-gram implementation within the Python machine learning package scikit-learn [9]. The vocabulary was learned using only the training set to prevent data leakage. This means that if a new word was present in the testing set it would not be used. Applying this process to the training data resulted in 410 941 unique n-grams. The n-grams were stored separately from other features since n-grams require sparse matrix representation

TABLE I. SAMPLE DATAPOINT FROM RAW DATA

Date	Difference	Headline 1	Headline 25
2008-08-22	165.384	b"British resident held in Guantanamo Bay wins legal battle to force Foreign Office to reveal 'torture' evidence"	b"If you've ever wondered what Kim Jong Il was like in grade school, here you go. Yes, he was quite ronery. Also, ordered his former teacher's whole family killed."

Python-based tool Valence Aware Dictionary and sEntiment Reasoner (VADER) [10] was used for extraction of three sentiment scores: positivity, negativity and compound sentiment score [-1,1]. VADER is using a combination of rule-based and dictionary-based model to produce the sentiment scores. Each word of text is assigned a sentiment value based on its entry in dictionary. This value is then adjusted using set of rules, e.g. a word written in capitals has contains more intensity than a lowercase word. Sentiment scores of all words in a text are then summed and normalized to be in range between -1 and 1 to produce the compound score.

For detailed overview of VADER engine see [10].

Subjectivity scores were computed for each collection of headlines using the Python package TextBlob [11] which uses simple averaging over values of subjectivity of all words in the text extracted from a dictionary.

Finally, 3 lags of all sentiment and subjectivity scores were created to again allow the model to use information from previous days since it is logical to assume that some events of previous days might still play role in determining the price of the stock.

To ensure that the extracted variables will provide information about the target variable, a simple filter-based feature selection was performed. Two filters were applied, near zero variance and unique value ratio filter. Near zero variance simply excludes features with nearly no variance.

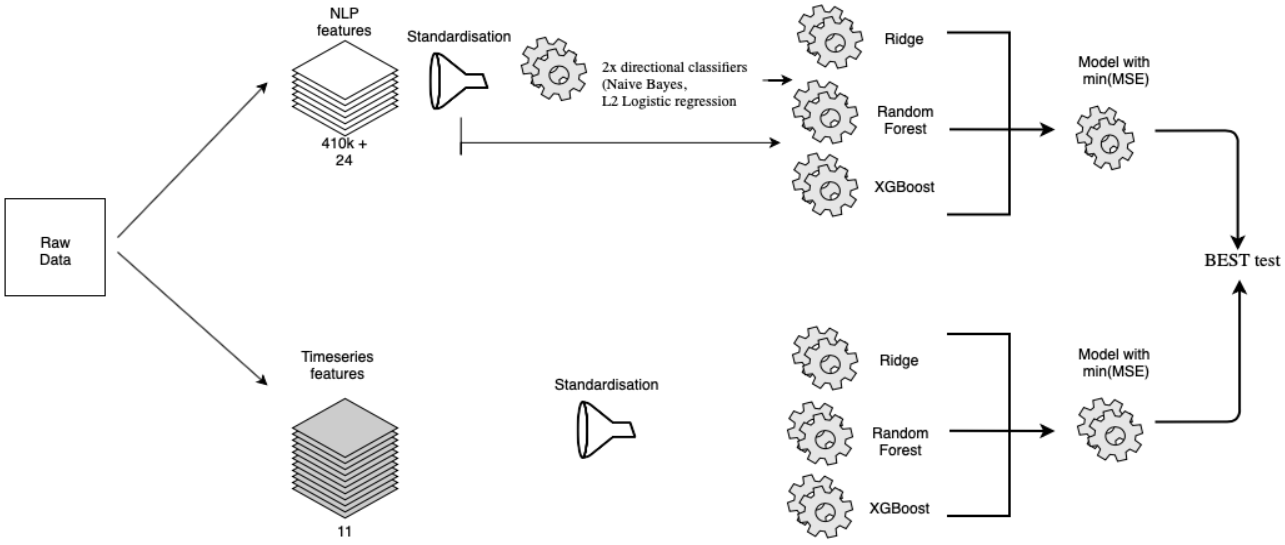


FIGURE 2 OVERVIEW OF THE EXPERIMENTAL DESIGN

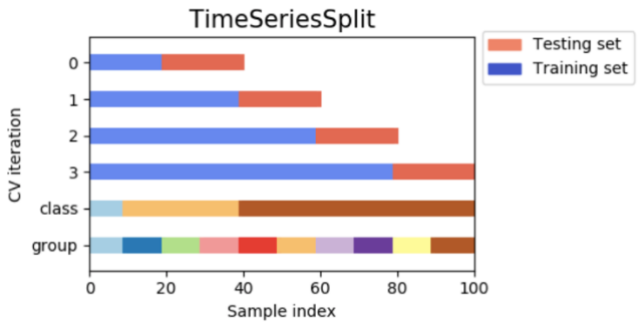


FIGURE 3 VISUALISATION OF CROSS VALIDATION FOR TIMESERIES. TAKEN FROM [8].

Unique ratio filter computes the ratio of unique values to all values. Features with low proportion of unique values were removed.

IV. Experimental design

For easier understanding of the experimental setup, Figure 2 shows an overview of the main points of the experiment.

We trained two separate sets of models using the 90% of the dataset. The same machine learning algorithms were used for both model sets: ridge regression, random forest and Extreme Gradient Boosting Machine (XGBoost). We also used the same training and testing protocol for all models which was a variant of 10-fold cross validation designed for time series data. As mentioned before, traditional random splitting should not be used with timeseries data. The time series cross validation has to ensure that the training folds are older than the samples in testing fold. The process of this cross-validation is illustrated in figure 3. The same folds were used for all models to allow for model performance comparison. Hyperparameter tuning of all models was performed in order to select the parameters combination that minimizes the prediction error.

Mean squared error (MSE) (1) was used as the error function because it assigns weight proportionally to the size of the error. In result this error function penalizes the model for large mistakes more.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (1)$$

Mean absolute percentage error (MAPE) was used for interpretation purposes (2). Unlike MSE, scale of MAPE is independent of the scale of the target variable as it is expressed as percentage. This allows easier understanding of the model performance and also facilitates simple comparison with other works since the reader does not need to know the scale of the target variable and can evaluate quality of the model with ease.

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i} \quad (2)$$

Using the MSE as selection criterion the best model from each set was selected for hypothesis testing, diagnostics and interpretation. The hypothesis that news has higher predictive power than historical values was tested using Bayesian alternative of a t-test, Bayesian Estimation Supersedes the t-Test (BEST) [12]. Traditional t-test is testing whether difference of means of two groups are significantly different from zero. BEST provides the same functionality but instead of a single value of the difference it estimates distributions of the means and standard deviation. Noninformative priors are used. The main advantage of BEST over t-test is that it provides richer information about the samples and differences.

The performance of the two best models was inspected using visualization methods. Predicted timeseries was first plotted together with the actual timeseries. Next the predicted values were plotted against the true values. Finally, the absolute errors, i.e. difference between predicted and actual value, were plotted over time to see whether the performance changes over time.

To interpret the models, the coefficients of the features were inspected to see the relative importance each feature has on the predicted value.

In the following subsections the model design of the two sets will be described and visualized.

A. Model Architecture

The architecture of the timeseries models is quite straightforward as it follows the traditional training schema. These models were trained only with the 7 lags of target variable and time features extracted from the date. Before each iteration of the cross-validation the training dataset including the target variable was standardized by means of subtracting mean and dividing by standard deviation. This separate standardization was applied in order to avoid data leakage, i.e. model being able to infer structure of testing data from the training data.

The architecture of NLP models is more complex. Same as in the case of the timeseries models the training set was standardized independently in each cross-validation iteration. The NLP models are stacks of two models: directional classifier and regressor. First, the n-gram representation and the sentiment features were used to train binary[3] classifiers, Naïve Bayes and logistic regression with L2 regularization to predict the direction of the stock value. The performance of the classifiers was evaluated using accuracy since the class imbalance was minimal with no-information rate[4] of 0.501. The out of sample predictions were used as a feature for the regressor models together with all sentiment and subjectivity features and their lags.

B. Machine learning algorithms

As described above, three machine learning algorithms were selected to predict the change of the stock price. Furthermore, two other methods were used to train the directional classifiers.

[3] Up or down

[4] If one were to predict the data at random, the probability of guessing correctly is given by this measure.

TABLE 3 CROSS-VALIDATION PERFORMANCE OF DIRECTIONAL CLASSIFIERS ± STANDARD DEVIATION

	Accuracy
Logistic regression	0.518 ± 0.027
Naive Bayes	0.488 ± 0.028

TABLE 4 CROSS-VALIDATION AND TESTING PERFORMANCE OF REGRESSION MODELS ± STANDARD DEVIATION

Set	Model	MSE	MSE test	MAPE test
Timeseries	Ridge	0.867 ± 0.306	1.191	120.273
Timeseries	Random forest	0.855 ± 0.319	1.264	168.445
Timeseries	XGBoost	0.746 ± 0.29	1.183	105.418
NLP	Ridge	0.777 ± 0.277	1.184	117.5
NLP	Random forest	0.944 ± 0.288	1.284	212.443
NLP	XGBoost	0.743 ± 0.284	1.182	106.837

Ridge regression was selected for the problem for several reasons. The method is simple and fast to train but most importantly it is easily interpretable since it is a regularized version of linear regression. The method fits a linear function that describes the relationship between features and target variable the best. The coefficients of the model then describe the slope of the line. However, using linear regression would likely result in overfitting to the training data which is why we used the ridge regression which implements L2 regularization in the model. This means that the model can shrink coefficients of features that do not improve the fit enough but unlike L1 regularization it does not allow the coefficients to drop to zero completely.

Random Forest regressor was also selected partially because it combines the interpretability of decision trees but solves some of their drawbacks. Random forest is an ensemble of decision trees that are trained on different subsets of the data enabled by bagging. Bagging means that each decision tree is trained on random data points sampled from the training set with replacement. The resulting forest of decision trees than delivers the final prediction by majority voting. This method lowers the risk of overfitting compared to using single decision tree. There is, however a potential risk in applying random forest for timeseries prediction as it would be unable to predict values that are significantly different from values that it has seen before. This is caused by the extrapolation problem. Decision trees are essentially learning on binary splits of the data to separate it by the target variable. This means that if the model encounters new value the best prediction it can offer is average of the training values. However, timeseries used in this paper are relatively stationary, i.e. values oscillate around mean, and therefore the extrapolation should not play any significant role.

Extreme gradient boosting (XGBoost) is the third algorithm applied to this problem [13]. This method is also an ensemble method. Unlike random forest, it uses boosting.

TABLE 5 DIRECTIONAL ACCURACY OF ALL MODELS ON TESTING DATA

Set	Model	Accuracy
Timeseries	Ridge	0.543
Timeseries	Random forest	0.508
Timeseries	XGBoost	0.533
NLP	Ridge	0.542
NLP	Random forest	0.513
NLP	XGBoost	0.534

Boosting is sequential method of fitting multiple models where each model tries to improve its performance by focusing on errors of the previous model. The version of XGBoost used in this paper utilizes linear regression as its learners. This method includes regularization as well to reduce the overfitting of the model.

V. Results

All the models were fitted using the scikit-learn API [9] for Python 3.6. All experiments were run using Kaggle's cloud computing platform. Model interpretation plots were created in R programming language, version 3.6.1. (Action of the Toes) [14] .

The output of feature selection filter is reported in table 2. Positivity feature was tagged as both having low variance and low proportion of unique values and was therefore removed together with its lags.

The performance of directional classifiers measured in accuracy is reported in table 3. Logistic regression trained on all NLP features achieved the highest cross validation accuracy. Therefore out-of-sample predictions of this model were used in the second stage of the NLP models.

The cross-validation and test results of both timeseries and NLP regression models are reported in table 4 and visualized

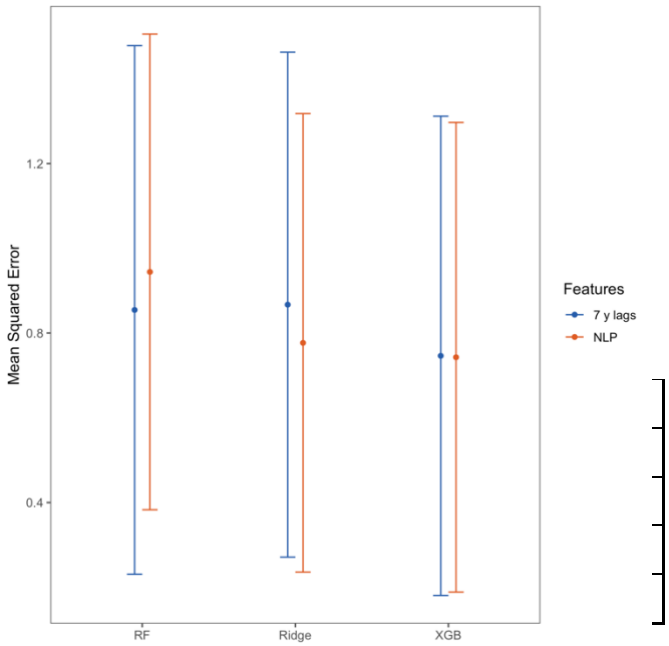
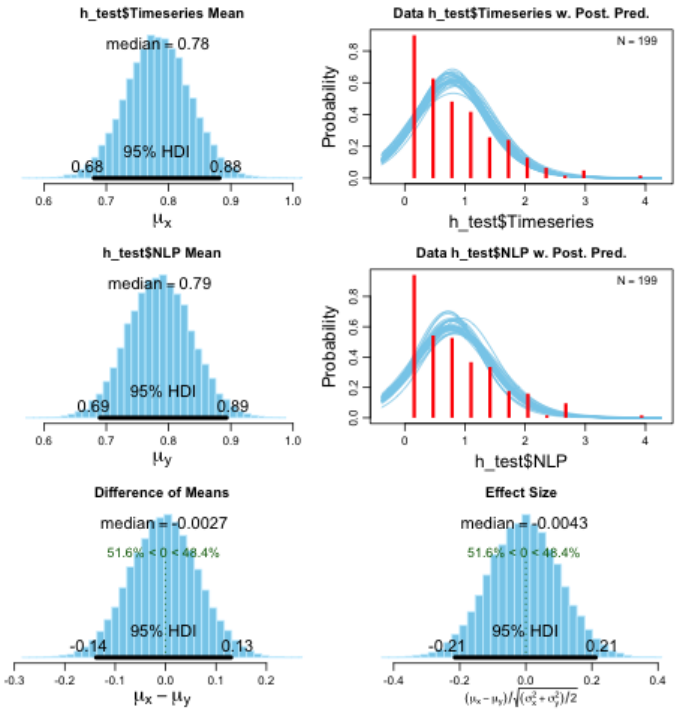


FIGURE 4 PERFORMANCE OF ALL REGRESSION MODELS WITH 95% CONFIDENCE INTERVALS




```
 Bayesian estimation supersedes the t test (BEST) - two sample

data: h_test$Timeseries (n = 199) and h_test$NLP (n = 199)

  Estimates [95% credible interval]
mean of h_test$Timeseries: 0.78 [0.68, 0.88]
mean of h_test$NLP: 0.79 [0.69, 0.89]
difference of the means: -0.0027 [-0.14, 0.13]
sd of h_test$Timeseries: 0.63 [0.54, 0.71]
sd of h_test$NLP: 0.63 [0.55, 0.72]

The difference of the means is greater than 0 by a probability of 0.484
and less than 0 by a probability of 0.516
```

FIGURE 5 RESULTS OF THE BEST TEST, TESTING THE DIFFERENCE BETWEEN NLP AND TIMESERIES MODELS

in figure 4 for easier understanding of the results. The directional accuracy of these models is reported in table 5.

Based on these results the timeseries' and NLP's XGBoost models were selected for hypothesis testing and interpretation. The results of BEST test are shown in figure 5.

The inspection of the two best models is visualized in figure 6 and 7 for timeseries and NLP models respectively. The coefficients of the two models are visualized in figure 8 and 9.

VI. Discussion

As reported in the previous section the results of BEST test indicate that there is no difference in predictive power of the best performing timeseries and NLP models. However, when we look at the cross-validated MSE of the compared models we can see that NLP performs slightly better. The models perform almost identically in terms of directional accuracy. Therefore the findings reported by [1] were not replicated on

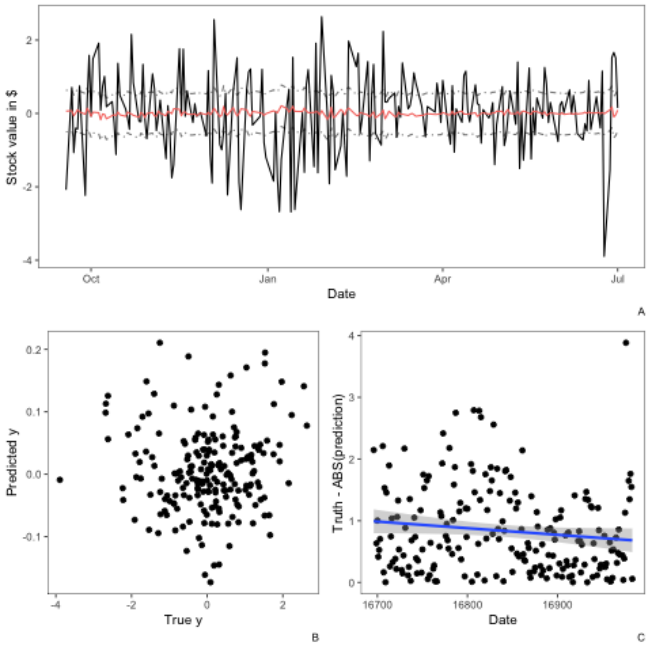


FIGURE 6 INSPECTION OF PREDICTIONS OF THE TIMESERIES XGBOOST. A. PREDICTIONS AND REAL VALUES, B. PREDICTIONS PLOTED AGAINST REAL VALUES, C. THE ABSOLUTE VALUE OF PREDICTION ERRORS OVER TIME AND LINEAR REGRESSION SHOWING NO SIGNIFICANT RELATIONSHIP BETWEEN ERROS AND DATE.

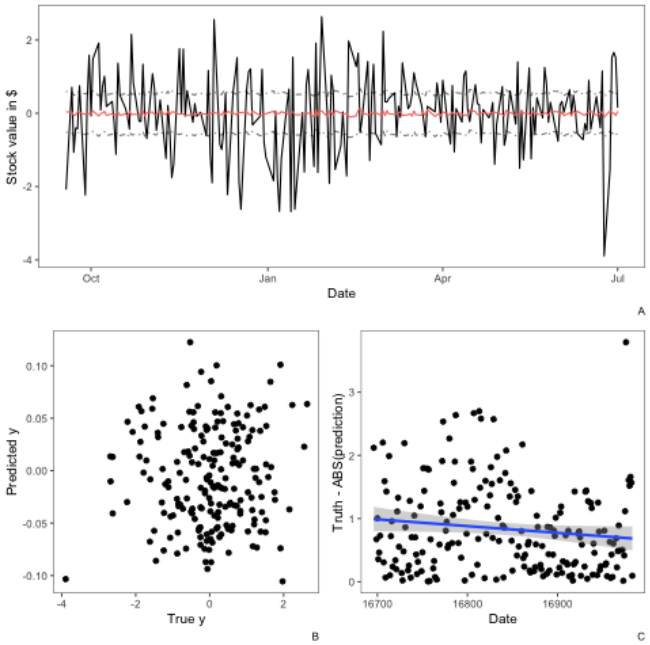


FIGURE 7 INSPECTION OF PREDICTIONS OF THE NLP XGBOOST. A. PREDICTIONS AND REAL VALUES, B. PREDICTIONS PLOTED AGAINST REAL VALUES, C. THE ABSOLUTE VALUE OF PREDICTION ERRORS OVER TIME AND LINEAR REGRESSION SHOWING NO SIGNIFICANT RELATIONSHIP BETWEEN ERROS

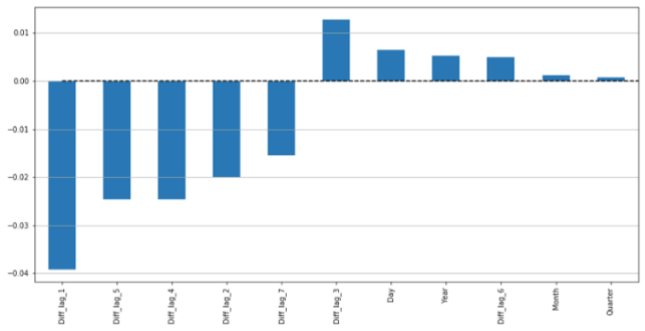


FIGURE 8 COEFFICIENTS OF THE TIMESERIES XGBOOST

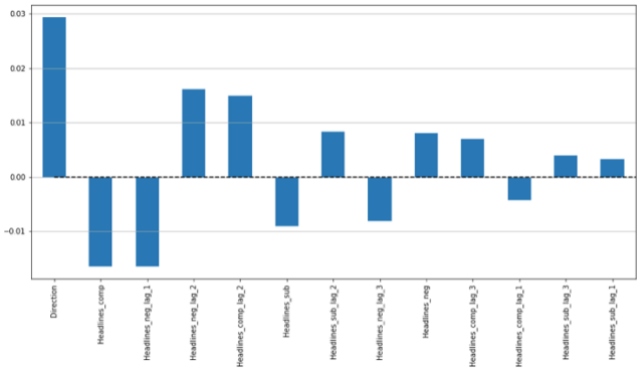


FIGURE 9 COEFFICIENTS OF THE NLP XGBOOST

our dataset completely. However, it is still interesting to see that information embedded in news headlines provides the same level of predictive power as previous values.

This experiment also showed that stock market prices react not only to financial events as showed in [1] but also to events reported in non-financial section of media. This might be caused by the nature of our dataset since we did not predict stock of any company but rather the industrial average. It might be therefore reacting to events that stock of single company would not.

In comparison to predictive accuracies reported in some papers our models are lacking significantly. Some of the previous works showed directional accuracies of up to 83.56% [4] when using historical data and 59% [7] with news data. In contrast, our best model reached directional accuracy of 54.3%. Although other works reported also the regression errors of their models it is not possible to compare the quality of the models since the information about the scale of the predicted variable is not reported. Looking at the regression errors, mainly the MAPE values, we can however conclude that our models are not performing very well. When looking just at the best model (NLP XGBoost) the predicted value is on average 106.83% off the actual value. A possible explanation might relate to the nature of the used dataset. While previous research modelled change of stock price as reaction to a news article being published we were trying to predict the sum of reactions to the 25 most popular news of the day. In other words, our dataset might be lacking in detailed information in comparison to other datasets.

Finally, we have seen that some machine learning models were slightly better fitted for solving our problem (figure X). The confidence intervals of all models are overlapping but those of random forest are definitely the widest. The cause of this might lie in the problem of extrapolation discussed in the overview of the algorithm.

The confidence intervals are however, relatively wide in all models. Possible explanation is that the performance of the models relied heavily on the data points used in training them. In other words, some of the data points were influencing the models significantly more.

A more detailed inspection of the two best models revealed that the models are too conservative in their predictions (figure X, A). The distribution of the prediction errors does not follow any linear pattern which indicates that there are no structured errors in the performance of the both models (figure X, B and C)

The timeseries XGBoost assigned the most importance to the smallest lags which seems logical that the newest values are the most similar to the current. The time feature did not contribute to the model very much therefore the regularization reduced their coefficients to very low values.

The NLP XGBoost used the predictions of the directional classifier the most. This again makes sense since this coefficient indicates to the model whether its prediction should be positive or negative. The second largest coefficient is the sentiment compound score and the first lag of negativity score. This indicates that the stock market reacts more to negative events and therefore the model was able to find this pattern better.

The results show that there are some serious limitations either in the dataset or used methods. While we covered some of them in this discussion, to explore all of them would require more detailed analysis of the models which is beyond the scope of this paper. Finding the most influential data points might be a sensible problem to analyze. Furthermore, more detailed interpretation of the models is advisable to investigate what did the models actually learn from the data. One of the options might be counterfactual plotting since the best performing models, XGBoost, are class of statistical models for which the technique is the perfect fit. Counterfactual plots are generated from artificial data points where all features are kept constant and only the feature of interest is varied.

VII. Conclusion

In conclusion, we failed to replicate the results that using news for stock market prediction is more useful than just using previous values of the stock. Instead our findings indicate that news provide approximately the same information for the prediction. This finding should be interpreted with extreme conservatism as neither of the models would be particularly useful in real life application. In other words, using the models reported in this paper in real life would likely result in serious financial losses of the potential user.

References

[1] R. P. Schumaker and H. Chen, "Textual analysis of stock market prediction using breaking financial news: The AZFin text system," *ACM Transactions on Information Systems (TOIS)*, vol. 27, no. 2, p. 12, 2009.

[2] B. G. Malkiel and E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *The journal of Finance*, vol. 25, no. 2, pp. 383-417, 1970.

[3] E. F. Fama, "Random walks in stock market prices," *Financial analysts journal*, vol. 51, no. 1, pp. 75-80, 1995.

[4] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques," *Expert Systems with Applications*, vol. 42, no. 1, pp. 259-268, 2015.

[5] K.-j. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1-2, pp. 307-319, 2003.

[6] R. Schumaker and H. Chen, "Textual analysis of stock market prediction using financial news articles," *AMCIS 2006 Proceedings*, p. 185, 2006.

[7] Y. Kim, S. R. Jeong, and I. Ghani, "Text opinion mining to analyze news for stock market prediction," *Int. J. Advance. Soft Comput. Appl*, vol. 6, no. 1, pp. 2074-8523, 2014.

[8] C. Bergmeir, R. J. Hyndman, and B. Koo, "A note on the validity of cross-validation for evaluating autoregressive time series prediction," *Computational Statistics & Data Analysis*, vol. 120, pp. 70-83, 2018.

[9] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825-2830, 2011.

[10] C. J. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," in *Eighth international AAAI conference on weblogs and social media*, 2014.

[11] *TextBlob: Simplified Text Processing*. [Online]. Available: https://textblob.readthedocs.io/en/dev/

[12] J. K. Kruschke, "Bayesian estimation supersedes the t test," *Journal of Experimental Psychology: General*, vol. 142, no. 2, p. 573, 2013.

[13] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016: ACM, pp. 785-794.

[14] R. C. Team, "R: A language and environment for statistical computing," 2013.

VIII. APPENDIX

All used code is published on author's GitHub, accessible on following link: GitHub