Computational Economics (2021) 57:311–340
https://doi.org/10.1007/s10614-020-10060-y

Predicting Stock Price Falls Using News Data: Evidence from the Brazilian Market

Juvenal José Duarte[1] · Sahudy Montenegro González[1] · José César Cruz Jr[1]

Accepted: 15 October 2020 / Published online: 17 November 2020

Abstract

Market participants use a wide set of information before they decide to invest in risk assets, such as stocks. Investors often follow the news to collect the information that will help them decide which strategy to follow. In this study, we analyze how public news and historical prices can be used together to anticipate and prevent financial losses on the Brazilian stock market. We include an extensive set of 64 securities in our analysis, which represent various sectors of the Brazilian economy. Our analysis compares the traditional Buy & Hold and the moving average strategies to several experiments designed with 11 machine learning algorithms. We explore daily, weekly and monthly time horizons for both publication and return windows. With this approach we were able to assess the most relevant set of news for investor's decision, and to determine for how long the information remains relevant to the market. We found a strong relationship between news publications and stock price changes in Brazil, suggesting even short-term arbitrage opportunities. The study shows that it is possible to predict stock price falls using a set of news in Portuguese, and that text mining-based approaches can overcome traditional strategies when forecasting losses.

Keywords Text mining · Brazilian Portuguese news · Brazilian stock market · Price forecasting · Machine learning algorithms

✉ Sahudy Montenegro González
sahudy@ufscar.br

Juvenal José Duarte
juvenaldrt@gmail.com

José César Cruz Jr
cesarcruz@ufscar.br

[1] School of Management and Technology, Federal University of São Carlos - UFSCar Sorocaba, São Paulo 18052-780, Brazil

1 Introduction

After the 2008 global financial crisis, international interest rates reached historical low levels, especially in developed countries such as the United States, the U.K., and Japan (Del Negro et al. 2019). Even though interest rates in developing countries are still relatively high, several countries tried to use monetary policy to reduce their interest rate levels during the same period. For instance, according to Fund (2019), the Brazilian monetary policy-related interest rate was cut by roughly 84% since 2008, from 13.75 to 2.25% APR, in 2020.

Interest rate cuts directly impact the levels of national debts and indirectly impact firms' and individuals' decisions to invest in real and financial assets. The study developed by Bernanke and Reinhart (2004) highlights that economic decisions are affected by prices and yields of financial assets, which are influenced by changes in interest rates promoted by monetary policy. As an example, we can assume that, when low-risk asset yields decrease, investors become more likely to face more risk to increase their portfolio returns. Therefore, we can expect that investors rebalance their portfolios, including a more substantial portion of assets with higher expected returns, which in turn brings more risk to their investment. Stocks are good examples of alternative investment for those who are willing to diversify their portfolio composition, and are aiming for (possible) high returns.

Due to their high volatility, forecasting stock returns can be considered a hard task for all types of investors. Consequently, several academic and industry studies have already questioned whether or not one can forecast future price movements. Even though this is a widely studied question, it remains with no conclusive answer. The most accepted hypothesis is that the possibility of predicting price movements varies according to the observed market, which may be more or less efficient (Fama 1970). Therefore, the majority of forecast techniques presented in the related literature rely on historical price analysis as the basis of their models.

Even though quantitative approaches are often used in the development of prices/returns forecast models, Economics is classified as a human science, once prices are directly linked to the feelings and reactions of the people involved in the trades (Shiller 2000). The area of Economics that is interested in understanding how human factors relate to trading decisions, especially emotions, is known as Behavioral Economics (Khadjeh Nassirtoussi et al. 2014; Shiller 2000). Therefore, we can assume that rational investors make their decisions based on the available information set and use their best knowledge and skills to decide whether to buy, hold, or sell a certain asset. Risk-averse individuals also try to avoid losses in the market, especially when they believe that losses can occur with a higher (subjective) probability than gains. Such behavior can be verified when an individual's utility function is represented with a steeper curve for losses than for gains. This theoretical framework is presented in various studies, such as in Tversky and Kahneman (1991).

Given the current low-interest rates that induce investment in assets of higher risk in Brazil, we can rely on rationality and on the market efficiency hypotheses

(Fama 1970) to present the following research question: *can investors avoid losses in the Brazilian stock market using information from the news?* Although conclusively answering this question may be difficult, we can observe the relationship between historical prices and the flow of news to determine return patterns. Identifying such patterns in the presence of market anomalies (de Camargos and Barbosa 2003) and using high volumes of daily news can limit manual approaches. This limitation can be overcome using text mining procedures, which can reduce processing time and prevent financial losses (Aase and Öztürk 2011).

Historical events and empirical studies support the hypothesis that, if not decisive, news reports have a major influence on investor's decisions, especially when they are negative (Shiller 2000). Therefore, several studies have already tried to formulate models based on textual information to predict price trends. Particularly those applying classical classification algorithms were relatively successful (Chowdhury et al. 2014; Aase and Öztürk 2011; Kumar and Ravi 2016; Khadjeh Nassirtoussi et al. 2014).

Our main objective with this study is to integrate textual information to historical stock prices in order to predict financial losses on the Brazilian stock market. We evaluate the performance of several machine learning algorithms, using a method that fits one classifier for each stock, based on news obtained from several public sources. We also aim to answer the following research question: *can textual information help predicting financial losses on the Brazilian stock market?*

Since studies that aim to predict asset return using textual information are rare in the Brazilian stock market (Lopes et al. 2008; Rehbein 2012; Alves 2015), and are usually limited to analyze only few assets, from limited business segments (de Oliveira et al. 2013; de Carvalho 2018; Yim and Mitchell 2005; Lopes et al. 2008; de Araújo and Marinho 2018; Alves 2015), we have various contributions to the current literature in the area. First, we use an extensive variate of stocks (64) and classifiers in our analysis (11). Second, we compared all classifiers to the traditional trading strategies Buy & Hold (B&H) and the exponential moving average (EMA). Third, differently from most studies in the field, we focus on financial loss, which is assumed to be a more clear parameter to indicate the viability of our algorithms as trading strategies. And fourth, we implement an automatic search based only on news published in Portuguese, and using Brazilian websites.

We expect that our results can be widely used by regulators, market participants, and news media. For instance, regulators may use our approach as an additional source of information to anticipate extreme events that could cause circuit breaks in the market. Market participants, such as individual investors and hedging funds, could prevent possible losses by reviewing their trading strategies beforehand. The news industry could follow our results to evaluate the impact of information on investors' decisions.

2 Related Studies

Various studies indicate promising results in forecasting trends based on news mining (Aase and Öztürk 2011; Chowdhury et al. 2014; Azar 2009; Kumar and Ravi 2016; Khadjeh Nassirtoussi et al. 2014; de Araújo and Marinho 2018; Li et al. 2015). Although researchers very often manually label a training dataset for classification, automated methods based on historical prices enable larger training samples with better results (Aase and Öztürk 2011). Recent studies, such as Haddi (2015) and Makrehchi et al. (2013), applied news labeling methods based on historical price series as an alternative approach for prediction using financial news. The authors obtained labels for price direction (fall or rise) from price changes (returns) (Haddi 2015).

Even though Aase and Öztürk (2011) and Azar (2009) analyzed only a few securities, both studies focused on the temporal relationships between the publication of news and the price response. The authors reported that the shorter the delay, the better the results. In addition to individual stock valuation, studies such as Mittal and Goel (2012), Bollen et al. (2011), Makrehchi et al. (2013) and Yu et al. (2013) analyzed the general mood of the stock market using consolidated indices such as the S&P 500 and the DJIA, among others.

Although Shiller (2000) argued that financial assets might remain undervalued or overvalued for long periods, as in the case of market bubbles, Yu et al. (2013) showed that, in the long run, the accumulation of events tends to incorporate greater randomness in pricing, making it difficult to identify patterns. Following these studies, Xing et al. (2018) found that twenty minutes would be the optimal time to verify the impact of publications on prices. However, Yu et al. (2013) highlights that emerging markets tend to be less efficient when reflecting news on prices.

An extensive review of recent studies was presented by Kumar and Ravi (2016). The authors showed that market prediction is one of the main themes in text mining studies applied to finance, particularly in the stock and exchange rate markets. They also presented issues that should be overcome, such as the limitation of studies in the analysis of specific financial securities, and the fact that the body of the news should be preferred over the headline to reduce ambiguity. The limitation of studies to specific assets deserves particular attention since it describes the context in which patterns are observed. As financial markets are adaptive and patterns are hardly permanent, understanding the events behind price swings provides indications of whether the patterns are punctual or recurring, and even if they are extendable to other contexts.

Once markets are considered accordingly to their efficiency (Fama 1970), the characteristics found in one market are not necessarily the same one can find in others. For instance, the way prices react to news in the U.S. exchanges may not be observed in the Brazilian market. Therefore, academics and market participants should be careful when generalizing patterns that could depend on the asset characteristics of a specific market.

de Oliveira et al. (2013) presented a comprehensive and systematic study about the use of automatic methods to analyze the Brazilian stock market. The authors focused on a single security, the state-owned oil company Petrobrás (PETR4.SA), and used artificial neural networks to predict the direction of price changes. They used a variety of information, including fundamental and technical analysis indicators, and other macroeconomic data. Their model was able to predict price changes in 93.62% of the cases correctly.

The behavior of the same stock, PETR4.SA, was studied by de Carvalho (2018). The author compared the outcomes of a neural network model to those obtained from a random walk model (estimated via a regression model). The later model resulted in better predictions, being able even to anticipate steep falls.

The study developed by Yim and Mitchell (2005) was an attempt to predict firm default using fundamentalist indicators of companies from various sectors in Brazil. The authors implemented two types of Artificial Neural Networks, Multi-Layer Perceptron, and Hybrid Neural Networks. The authors found that their models could predict bankruptcy a year in advance.

The use of textual content for trend identification was also suggested by Lopes et al. (2008). The authors collected opinions from various sources and calculated a daily positivity coefficient based on the available news. They calculated the correlation between trading prices for a group of stocks and the sentiment coefficient expressed in the news. The authors found high levels of correlation between prices and the sentiment in the news. Later, Rehbein (2012) conducted a similar study, incorporating semantic elements in lexical analysis, but with a small data sample.

Textual news data was also used by de Araújo and Marinho (2018) to predict short-term variations in the performance of economic sectors, on the Brazilian stock exchange. The authors translated their dataset from Portuguese to English before implementing predictive methods for tree-based classifiers.

Textual data was also used by Alves (2015) who collected information from Twitter, in Portuguese, to analyze how to buy and sell decisions that were made on the Brazilian stock exchange. The author suggested an alternative approach where he used textual data to check the investors' sentiment before using this metric along with other indicators in his model.

After analyzing previous studies and considering their suggestions for future work, we suggest the following additional approach in our current study:

- We extend the studies of Lopes et al. (2008), de Araújo and Marinho (2018), de Carvalho (2018), Alves (2015), de Oliveira et al. (2013), Rehbein (2012), Yim and Mitchell (2005) to include the application of automatic techniques for stock market analysis, specifically for the Brazilian case, using text data in Portuguese.
- We include a broader set of assets in our analysis, and use all 64 securities that are part of the IBovespa index. These assets represent a wide variety of sectors in the Brazilian economy.

3 The Proposed Method

Before we present the models that will be used to predict financial losses in the Brazilian stock market, we first define the temporal scope and the data set of our analysis. Figure 1 shows the three temporal parameters used in our models: (i) the decision point, or date zero (D_0), (ii) the *publication window*, or the time window from which the information is collected, and (iii) the *return window*, or the future time window for which expected values are predicted. k and p respectively represent the

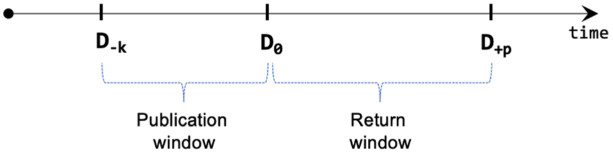

Fig. 1 Rolling windows: D_{-k} past, D_0 today, D_{+p} future

number of days used in the past and future time windows, counted from a certain date D.

As the analysis for a given decision point D_0 moves forward over time, both past and future windows are rolled forward, as well. With that said, we adopt the following definitions:

Definition 1 *Publication window* is the rolling window starting in D_{-k} and ending in D_0. This window contains the set of news articles used in our prediction models.

Definition 2 *Return window* is the rolling window starting in D_0 and ending in D_{+p}. This window contains the expected or predicted financial returns for all assets in our models.

Where

- D_0: a given stock market opening date.
- D_{-k}: the start date for the publication window.
- D_{+p}: the end date for the return window.

We assume that the investor makes their trading decision before the market opens on the day D_0, based on the set of news published until the previous day. Therefore, the investor's trading strategy uses this information set to predict the potential return of their assets. The one-day return is calculated when they buy a particular stock at the opening price, on the day D_0, and sell it at the closing price, on the day D_{+p}. Instead of training a single general predictive model, our approach consists of fitting one classifier for each combination of stock, publication, and return windows. This method permits that the price of each asset, in the short and long run, can be determined by recent and old news.

The flowchart in Fig. 2 summarizes the steps of our method. Our goal is to estimate the expected price of a given stock, s, using news from a publication window of size, k. Additionally, we use the weight function, w, to predict prices in the return window of size p.

The method presented in Fig. 2 consists of three layers: (1) data sources, (2) preprocessing, and (3) setup of the stock market classifier. The data sources are news articles in Brazilian Portuguese, the list of traded stocks on the Brazilian Exchange, $B3$, and the historical price series for all 64 selected assets. Our approach consists of (i) selecting a target asset s among the 64 securities, (ii) using a set of publications

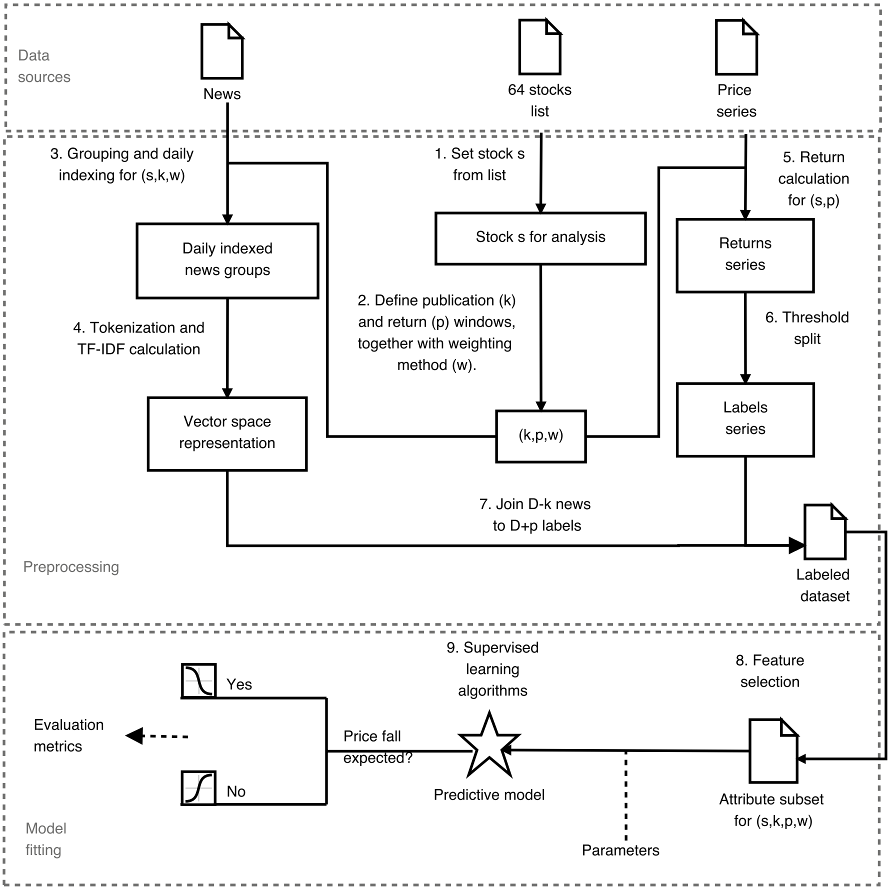
Data sources
News
64 stocks list
Price series
3. Grouping and daily indexing for (s,k,w)
1. Set stock s from list
5. Return calculation for (s,p)
Daily indexed news groups
Stock s for analysis
Returns series
2. Define publication (k) and return (p) windows, together with weighting method (w).
4. Tokenization and TF-IDF calculation
6. Threshold split
Vector space representation
(k,p,w)
Labels series
7. Join D-k news to D+p labels
Preprocessing
Labeled dataset
9. Supervised learning algorithms
8. Feature selection
Yes
Evaluation metrics
Price fall expected?
Predictive model
No
Attribute subset for (s,k,p,w)
Model fitting
Parameters

Fig. 2 Method description to obtain a predictor for a single stock

k, (iii) creating the return window p; and (iv) using the weight function (w). These parameters together define two tasks:

- *The purpose of the prediction* stock and return window to be predicted;
- *The labeled dataset* the news and the weight function.

Regarding the weight function w, the news can be processed based on one of the two approaches: $w_1(k,i)$—the uniform weights function (Eq. 1), or $w_2(k,i)$—weighted by the news publishing date. The second, i.e. the decreasing weights function, is based on the calculation of exponential moving averages with weights assigned according to Eq. 2, where $0 \leq i \leq k$. We use this last equation decreasing values from k to 0.

$$w_1(k,i) = 1, \quad \forall i, 0 \leq i \leq k \tag{1}$$

$$w_2(k,i) = \begin{cases} i > 0 : w_2(k,i) = \dfrac{2}{k+1} * \left[1 - \left(\dfrac{2}{k+1}\right)^i\right] \\ i = 0 : w_2(k,i) = 1 \end{cases} \tag{2}$$

Step 3 initiates the preprocessing stage. It processes all news daily (from D_{-1} to D_{-k}), according to the selected stock s, the security under analysis, and the selected weight function w, producing the daily indexed news database. Then, the news texts in natural language are split into tokens in step 4. The vector space representation is computed based on the TF-IDF frequency for every token.

Given a security s and the return window p, steps 5 and 6 calculate the returns, for each decision point D_0, and turn them into labels. The multiplicative return is the ratio between the closing price at D_{D_0+p} and the opening price at D_0, as presented in Eq. 3. Returns are then transformed into two classes, computed based on a threshold split, as shown in Eq. 4. The resulting classes establish whether or not a downward movement is observed in the return window.

$$Return(D_0,p) = \left(\frac{Price_{close}(D_{D_0+p})}{Price_{open}(D_0)}\right) - 1 \tag{3}$$

$$Label(D_0) = \begin{cases} Return(D_0,p) < 0 : 1 \\ Return(D_0,p) \geq 0 : 0 \end{cases} \tag{4}$$

In step 7, the daily indexed news articles and the labeled price series dataset are joined by date, producing the labeled dataset. As the decision point (D_0) advances through time, the rolling windows allow news stories to affect decisions by influencing vocabulary frequencies on further days. The resulting dataset is high dimensional due to the large volume of news and tokens.

The next steps compute the classification model. Filter feature selection intends to reduce the vocabulary from a generic to a specialized set of tokens most related to s, using F-score. It analyzes the frequency of tokens in relation to the labeled price series dataset. As a result, the labeled dataset is reduced to a subset of features with the most related vocabulary for the asset s, based on k, p, and w. After the reduced dataset is obtained, the next step uses it to forecast the investor's reaction to the news and provides a prediction on whether the prices are expected to decrease or not.

4 Experimental Setup

Even though the experiments were designed to address the main research question, we additionally include other questions related to our method of analysis:

1. What percentage of selected features achieves the best results?
2. Is there any difference between the results obtained from the uniform weights and the decreasing weights functions?

3. How fast does the market react to the news?
4. What is(are) the best algorithm(s) to forecast losses?

We developed four experiments to answer these questions. In the next subsections, we describe the parameters used in our models, the datasets, algorithms settings, and evaluation metrics.

4.1 Parameters Settings

We use four variables to set up our experiments, following the proposed in Fig. 2. The argument s is the stock selected among all 64 securities for a given iteration. We define that the predictions in the publication window are counted in business days, assuming three windows: the previous day ($k = 1$), the previous week ($k = 5$), and the previous month ($k = 21$) news. Return windows for predictions are also based on business days, however, evaluated differently: for the current day ($p = 0$), for the next week ($p = 5$), and for the next month ($p = 21$) returns. We seek for the most relevant past information for the investor's decision, and for the prices to reflect the events reported in the news. Moreover, we use both the uniform weights function $w_1(k, i)$, and the decreasing weights function, based on the publishing date $w_2(k, i)$.

4.2 News Data

Since we were not able to find any public corpus with Brazilian news, we collected data using a crawler and preprocessed the information to produce a new database, available at https://bit.ly/2mXUxix. We started to capturing news data in August, 2016 and ended in May, 2018. In addition to the news text itself, metadata was also stored, such as the source (article's website), publication date (provided by the website), the title, and URL. Data were stored in text files, organized into sub-directories by provider and month of publication, to expedite searches.

The volume of monthly news we collect from each web portal is shown in Table 1. As providers generally publish news with content from various subjects (sports, politics, entertainment, etc.), whenever possible, we considered only political and economic content in our analysis.

We capture data from Advfn, Estadão, G1, and UOL since the beginning of the period. Data from Investing, Exame, Informoney, Último Instante, Valor, and Veja were captured later, as of July 2017, to increase the daily volume of articles.

According to Table 1, the news volume is poorly distributed among providers. G1 accounts for more than a third and Investing less than 1% of the articles -and along the year - smaller volumes at the beginning and the end of the year. The amount of news per provider is naturally heterogeneous, justified by the difference in the size of the portals. Moreover, the difference between months is due to crawler problems caused by significant changes in the providers' portals source code.

We collected only public news from the websites. Portals Estadão and G1 provide exclusive content for subscribers who are charged monthly fees. Therefore, we did not use information from the subscribers' area on those websites.

Table 1 Monthly news volume by portal

Year/Month	br.advfn.com	br.investing.com	estadao.com.br	exame.com.br	g1.globo.com	infomoney.com.br	ultimoinstante.com.br	uol.com.br	valor.globo.com	veja.com.br	Total
2016/08	0	0	0	0	59	0	0	8	0	0	67
2016/09	1071	0	980	3	6431	0	0	605	0	4	9094
2016/10	2104	0	1721	1	13,420	1	0	1199	0	2	18,448
2016/11	2383	0	1655	2	12,645	0	0	1252	0	3	17,940
2016/12	2331	0	1652	1	10,597	0	0	1246	0	7	15,834
2017/01	1504	1	1076	7	6950	0	0	895	0	3	10,436
2017/02	2354	0	1566	20	9357	1	0	1024	0	15	14,337
2017/03	2753	0	1751	21	11,004	4	0	1190	1	22	16,746
2017/04	2513	0	1669	18	3971	4	0	1209	1	11	9396
2017/05	2822	0	2043	13	2153	6	0	1555	19	12	8623
2017/06	2592	0	1729	32	2039	8	0	1520	37	10	7967
2017/07	2548	7	1525	889	3227	12,296	250	1354	2373	2214	26,683
2017/08	3163	27	1792	1612	3538	4276	945	1419	4217	3936	24,925
2017/09	1921	14	1487	1161	2580	2951	513	1112	3274	2665	17,678
2017/10	2462	25	1641	0	3265	3348	788	1300	4092	1937	18,858
2017/11	2048	25	1383	0	2733	4535	588	1065	3831	1711	17,919
2017/12	1683	19	1152	0	2270	2945	75	944	3121	145	12,354
2018/01	437	26	1544	0	2723	3916	679	1208	3551	0	14,084
2018/02	0	20	1096	0	1410	2655	513	942	2972	0	9608
2018/03	0	2	123	0	61	735	2	83	470	0	1476
2018/04	0	0	0	0	0	1	0	0	33	0	34
2018/05	0	0	44	0	8	52	0	0	310	0	414
Total	36,689	166	27,629	3780	100,441	37,734	4353	21,130	28,302	12,697	272,921

Despite the filters applied during the data collection, we used a further analysis to find irrelevant articles such as error pages, links in other languages, advertising, and other websites that did not refer to the news. Because the number of problematic cases was low compared to the full body of the story, they were removed manually.

For preprocessing, the words were converted to lower case, the URL has been discarded, special characters have been removed and stop words have been excluded. The list of stopwords in Portuguese has been retrieved from Python's Natural Language Toolkit (NLTK).

4.3 Price Series Data

We downloaded price time series from Yahoo[1]. We used daily open and close prices and trade volumes.

Our analysis includes all securities that were included in the IBovespa index portfolio in December 2018[2]. The 64 securities are listed in Table 2, which also contains the price direction balance, informing the number of negative and positive returns obtained in the training and test samples. The content in Table 2 is ordered according to the last column, which is the average percentage of positive records (losses) in the dataset, where the first stocks are the ones with the least price falls on the period.

According to Table 2, the number of days with price increases compared to falls is reasonably well distributed for short term returns, for both the training and test data. The most significant difference was MGLU3, in the training set, with 63% up movements and 37% down. For long term windows, such as D_{+5} and D_{+21}, we can find a more uneven distribution between up and down movements. For instance, BRFS3 had only 21 negative samples for the $21d$ test set, in contrast to 117 positive samples.

4.4 Training and Test Sets

For this research, we presuppose that past events may influence future price movements to some extent. On the other hand, prices in the past cannot be influenced by news coming in the future, since the market prices reflect only the information available by the time it is negotiated (Fama 1970). This fact imposes temporal dependence when sampling the data for training and tests, as well as in cross-validation and grid search.

Traditional K-Fold sampling expects independence between the records, so that subsets of data may be iteratively picked for validation, from any position. For that reason, traditional K-Fold is not suitable for time series analysis. The library Scikit Learn offers an alternative K-Fold implementation where validation rounds are performed on growing subsets of the training data, as shown in

[1] Accessed on link: http://finance.yahoo.com.

[2] Accessed on: http://www.bmfbovespa.com.br/en/produtos/indices/indices-amplos/indice-ibovespa-ibovespa-portfolio-composition.htm.

Table 2 Number of samples for training and test for each stock and publications in D_{-1} and returns for D_0, D_{+5} and D_{+21}

Stock	Training						Test						AVG
	Negative samples			Positive samples			Negative samples			Positive samples			Pos.
	0 d	5 d	21 d	0 d	5 d	21 d	0 d	5 d	21 d	0 d	5 d	21 d	(%)
SUZB3	49	54	66	31	26	14	21	30	38	17	8	0	26
CVCB3	152	187	220	129	94	61	74	86	108	64	52	30	34
MGLU3	178	204	238	103	77	43	67	72	97	71	66	41	35
FIBR3	139	169	187	142	112	94	82	94	105	56	44	33	37
RENT3	154	180	185	127	101	96	70	82	108	68	56	30	38
TIMP3	158	175	191	123	106	90	75	79	96	63	59	42	39
SANB11	153	166	182	128	115	99	75	81	104	63	57	34	39
VALE3	140	153	161	141	128	120	79	89	109	59	49	29	40
PETR3	133	152	148	148	129	133	82	94	96	56	44	42	41
BRAP4	140	165	166	141	116	115	81	76	93	57	62	45	42
PETR4	140	153	157	141	128	124	74	94	91	64	44	47	42
EQTL3	147	162	168	134	119	113	76	81	87	62	57	51	42
RAIL3	156	179	177	125	102	104	75	76	74	63	62	64	42
ABEV3	155	162	178	126	119	103	75	73	85	63	65	53	43
SBSP3	148	169	160	133	112	121	72	81	86	66	57	52	43
BBAS3	157	170	189	124	111	92	75	76	66	63	62	72	43
ESTC3	159	166	169	122	115	112	78	66	82	60	72	56	43
USIM5	136	168	176	145	113	105	72	87	73	66	51	65	44
B3SA3	159	175	187	122	106	94	76	70	65	62	68	73	44
GOLL4	137	164	187	144	117	94	67	62	98	71	76	40	44
GOAU4	146	150	169	135	131	112	69	84	85	69	54	53	44
ITUB4	162	174	190	119	107	91	71	72	65	67	66	73	44
GGBR4	139	151	161	142	130	120	72	87	85	66	51	53	44
BTOW3	147	163	164	134	118	117	72	75	85	66	63	53	44
HYPE3	142	156	177	139	125	104	77	74	80	61	64	58	44
ITSA4	153	173	177	128	108	104	64	76	76	74	62	62	44
BBDC4	159	185	186	122	96	95	70	67	64	68	71	74	44
LREN3	159	173	207	122	108	74	71	73	48	67	65	90	45
VVAR3	156	175	195	125	106	86	62	59	73	76	79	65	45
BRKM5	136	165	188	145	116	93	66	71	75	72	67	63	45
WEGE3	152	149	192	129	132	89	60	79	70	78	59	68	46
SMLS3	155	170	193	126	111	88	70	70	54	68	68	84	46
BBDC3	162	169	175	119	112	106	71	63	60	67	75	78	47
CYRE3	143	167	169	138	114	112	70	69	67	68	69	71	47
CSAN3	151	165	142	130	116	139	74	65	75	64	73	63	47
NATU3	152	153	165	129	128	116	61	72	75	77	66	63	47
IGTA3	148	170	195	133	111	86	69	67	48	69	71	90	47
PCAR4	138	169	188	143	112	93	67	62	59	71	76	79	48
MRVE3	138	146	156	143	135	125	68	68	75	70	70	63	48

Table 2 (continued)

Stock	Training						Test						AVG
	Negative samples			Positive samples			Negative samples			Positive samples			Pos.
	0 d	5 d	21 d	0 d	5 d	21 d	0 d	5 d	21 d	0 d	5 d	21 d	(%)
ECOR3	146	175	197	135	106	84	65	58	48	73	80	90	49
FLRY3	146	158	183	135	123	98	65	70	51	73	68	87	49
JBSS3	136	149	142	145	132	139	66	70	77	72	68	61	49
UGPA3	145	141	154	136	140	127	77	71	58	61	67	80	49
LAME4	128	150	158	153	131	123	72	64	70	66	74	68	49
EMBR3	137	154	161	144	127	120	61	70	66	77	68	72	49
RADL3	137	157	169	144	124	112	69	57	64	69	81	74	50
VIVT4	153	162	144	128	119	137	70	64	54	68	74	84	50
QUAL3	160	174	200	121	107	81	68	45	24	70	93	114	52
MULT3	161	163	180	120	118	101	67	56	28	71	82	110	52
MRFG3	148	153	164	133	128	117	62	49	58	76	89	80	52
EGIE3	126	139	134	155	142	147	63	61	76	75	77	62	52
BBSE3	135	140	152	146	141	129	68	63	52	70	75	86	53
CMIG4	129	137	126	152	144	155	68	67	65	70	71	73	53
ENBR3	146	152	171	135	129	110	64	57	38	74	81	100	53
CSNA3	130	137	155	151	144	126	62	70	50	76	68	88	53
CPLE6	134	141	146	147	140	135	64	57	59	74	81	79	53
CIEL3	132	121	91	149	160	190	68	64	70	70	74	68	55
ELET6	131	127	134	150	154	147	61	62	50	77	76	88	56
KROT3	145	158	180	136	123	101	63	43	18	75	95	120	56
ELET3	127	113	132	154	168	149	62	63	52	76	75	86	57
BRML3	137	168	162	144	113	119	60	45	25	78	93	113	57
CCRO3	145	145	139	136	136	142	60	48	25	78	90	113	58
BRFS3	133	129	124	148	152	157	57	37	21	81	101	117	63

Fig. 3. This approach allows the parameters to be cross-validated across different samples of data while still respecting the original order of the records.

The dataset has been split between training and testing, keeping the 2/3 initial portion for training and the most recent 1/3 part for tests. As shown in Fig. 3, the training part is also subdivided for two validation rounds in the experiments. The classifier training was performed in batch mode, where the model is trained only once and applied to predict new samples.

Dates comprehended between 2016-08-01 and 2017-09-12 were used for training, while the period between 2017-09-13 and 2018-05-07 has been used for tests, corresponding to 281 and 138 records, respectively (more details in Table 2). The only exception is for the stock SUZB3, which has 118 samples for training and 38 for tests due to shorter time-series data.

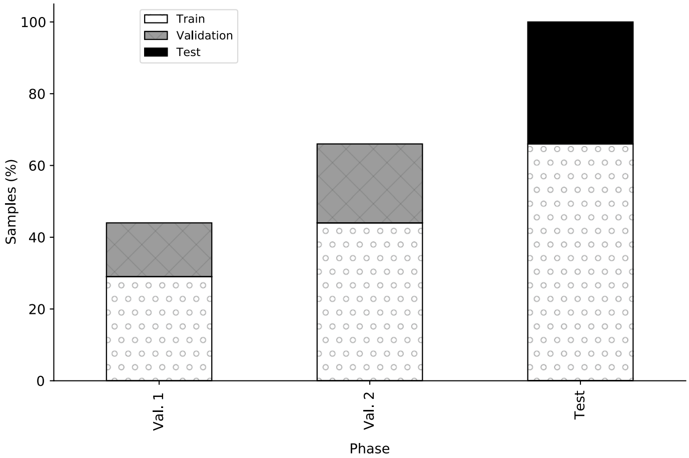

Fig. 3 Sequence of data splits during validation and test phases. The first bar, "Val 1", illustrates the first training-validation round. The second bar, "Val 2", illustrates the second training-validation round. The third bar,"Test", shows the percent of the dataset covered on the final training-test round

4.5 Traditional Trading Strategies and Machine Learning Methods

We compared the traditional strategies Buy & Hold (B&H) and exponential moving Averages (EMA) to results obtained by well-known machine learning methods in similar researches: Gaussian Naïve Bayes (G-NB) (McCallum and Nigam 1998), Multilayer Perceptron neural network (MLP) (Mitchell 1997), Support Vector Machines with linear (SVM-L), polynomial (SVM-P) and radial basis functions (SVM-R) kernels (Cortes and Vapnik 1995), K-Nearest Neighbor (KNN) (Cover and Hart 1967), Logistic Regression (LR) (Abu-Mostafa et al. 2012), Decision Trees (DT) (Mitchell 1997), Random Forests (RF) (Breiman 2001), Gradient Boosting (GB) (Hastie et al. 2009) and XGBoost (XGB) (Chen and Guestrin 2016). These methods are widely used as baseline for text classification. We used the implementations from the *scikit-learn* library (Pedregosa et al. 2011) for all the models except for XGB, for which the *XGBoost* library is applied.

4.5.1 Buy & Hold and Exponential Moving Averages

The B&H strategy involves buying an asset and holding it for a certain period without negotiated it. Investors who use this technique have positive prospects for the long-term.

The use of moving averages tries to detect anomalies in the trend of negotiated prices. To do so, we introduced an observation window of n days before the target date, and then average prices are calculated. For the specific case of exponential moving averages, higher weights are attributed to the latest prices according to Eq. 5.

$$m[t] = (m[t] - m[t-1]) \times \left(\frac{2}{n+1}\right) + m[t-1] \tag{5}$$

In Eq. 5, $m[t]$ represents the tth moving average, composed by the previous average $m[t-1]$ summed to its difference to the most recent price, discounted by the factor $2/n + 1$, where n is the observation window size. The effect of EMA is to smooth the price series so that atypical variations stand out in relation to the smoothed series.

4.5.2 Support Vector Machines

SVM is a technique whose approach consists of maximizing the class separation margin through auxiliary (or support) vectors (Cherkassky and Mulier 2007). One of the main advantages of SVM, however, is the possibility of using kernel functions for handling non-linearly separable data. The kernel functions project the original data into a larger attribute space and, when the kernel type is well-chosen, making the decision boundary of the classes more evident. This process is called nonlinear transformation (Abu-Mostafa et al. 2012; Cherkassky and Mulier 2007). The most common types of kernel functions are linear, polynomial, and radial based functions (RBF).

4.5.3 Naïve Bayes

Each attribute is weighted based on conditional probabilities according to the Bayes theorem (Bird et al. 2009; Hotho et al. 2005). The individual *apriori* probabilities of each attribute are estimated by observing the frequencies in the training records (Bird et al. 2009). Given an example for a diagnostic system (binary classification), Eq. 6 shows the probability of the record belonging to the positive class C, given the values of its attributes represented by the vector $\mathbf{x} = \langle x_1, \dots, x_n \rangle$.

$$p(C|x_1, \dots, x_n) = p(C) \prod_{i=1}^{n} p(x_i|C) \tag{6}$$

The calculation of the *apriori* probabilities, however, is conditional on the type of attribute domain. For binary values (One Hot Encoding, for example), the Bernoulli model can be used. For numerical values belonging to the set of natural numbers, the Multinomial model is suitable; for continuous numerical values, the Gaussian model is necessary.

4.5.4 Decision Tree, Random Forest, Gradient Boosting and XGBoost

The DT algorithm works based on the mapping of input values to rules that, when chained in the form of paths (or branches), lead to the decision between the target classes of a classification problem (Mitchell 1997). The paths that lead to classification can be seen as boolean expressions. Hence, for the disjoint set of rules $R = \{r_1, r_2, \dots, r_n\}$ of a path, the assignment of a class C_k is given by the expression

$r_1 \wedge r_2 \wedge \cdots \wedge r_n \Rightarrow C_k$. The same class can be assigned by several different paths, generating more complex boolean expressions (Mitchell 1997).

Decision trees are susceptible to under-fitting when too simple (few rules, low depth), and overfitting when too complex, making it an algorithm that is difficult to apply for more advanced problems. Random Forests seek to reduce the variance of the classifiers by the composition of a committee of different simple trees, determining the final result by voting (Couronné et al. 2018).

RF is a bagging type of ensemble method, where each underlying tree is trained independently. The Gradient Boost algorithm, on the other hand, uses a boosting strategy to reduce error, sequentially training trees based on errors evaluated on the previous ones (Hastie et al. 2009). A more recent version of GB is XGBoost, which provides enhancements for dealing with sparse data and regularization capabilities (Chen and Guestrin 2016).

4.5.5 Logistic Regression and Multilayer Perceptron

Based on the attributes provided in the input, the logistic model classifier takes advantage of a sigmoid function to produce a value from 0 to 1, which provides the prediction of a binary class (Abu-Mostafa et al. 2012) when discretized by a division threshold. The dependent attribute is modeled as a function of one or more independent variables. Each input attribute, which must be numeric, is assigned a coefficient, such that the result is given by the weighted sum of these applied to the sigmoid (Couronné et al. 2018) function. The logistic function activates the logistic regression, which is provided by Eq. 7, where $\vec{X}$ is the vector of inputs, $\vec{\beta}$ is the vector of weights, and α is the bias. The activation is then converted to a prediction by applying a threshold function, as in Eq. 8.

$$activation = \frac{1}{1 + \exp^{-(\vec{X}\vec{\beta}+\alpha)}} \tag{7}$$

$$prediction = threshold\;(activation) \tag{8}$$

In the MLP, each neuron, also called perceptron, is composed of a linear weighted sum followed by an activation function, just like the Logistic Regression. Still, several neurons are grouped to form an artificial neural network. Each neuron receives impulses from the entire previous layer and propagates a new impulse based on its activation function. The neurons are calibrated by establishing weights for the stimulus of the previous layers. These are adjusted by iterative methods such as Backpropagation (Mitchell 1997).

4.5.6 K-Nearest Neighbor

KNN is an algorithm whose operation is based on the projection of the records in a vector space, where each attribute is represented by a vertex of a N-Dimensional plane. After projecting the sample data (training set), when trying to predict new records, they are mapped to the same space, recovering the K nearest neighbors, with

Table 3 Evaluation metrics and their formulas

Metric	Formula
Recall	$Recall = \frac{TP}{TP+FN}$
Precision	$Precision = \frac{TP}{TP+FP}$
F1 or F-score	$F1 = 2 * \frac{Precision*Recall}{Precision+Recall}$
Return factor	$R_F(\Delta_t) = \prod_{i=1}^{n} \left(\frac{Price_{close}}{Price_{open}} \right)$
Loss factor	$L_F(\Delta_t) = \prod_{i=1}^{n} \min \left(\frac{Price_{close}}{Price_{open}}, 1 \right)$
Loss (%)	$L_{\%}(\Delta_t) = 100 \times \frac{L_F(\Delta_t)}{L_T(\Delta_t)}$

the aid of a predefined distance calculation method. Based on the class assigned to most neighbors during training, the new record has its class defined in the test phase (Cherkassky and Mulier 2007).

4.6 Evaluation Metrics

To compare the results, we employed the well-known precision, recall, and F-score. Financial metrics were also calculated to estimate the viability of using the classifiers as trading strategies. Table 3 shows the metrics used and their formulas.

In the Recall and Precision formulas, TP means True Positives, and FN means False Negatives. The return factor is the product of the strategy daily returns for each of the n days. The loss factor is the product of strategy losses on n days; when returns are positive, the loss factor evaluates to 1. The loss percent was used to compare predictive models trained for different stocks, reducing the analysis bias. $L_T(\Delta_t)$ represents all losses during the period, be it realized or not. Thus, classifiers with higher falls in the period are not penalized against the others.

In finance, investors holding a security have their position called "long" on the given stock, traders who do not have the security in their portfolio have their position called "short". When making predictions for test data, the classifier suggests the days where price drops are expected. For those days, short positions are not considered on the accrual. Conversely, for valuation and comparison, all days, where the prediction does not indicate price falls, are considered as long positions so that its returns are regarded during the accumulation.

In Sect. 5.3, the experiment compares predictions for a single day, week, and monthly returns, in order to analyze how fast the market reacts to the news. For these cases, it is assumed that the positive (fall) predictions that occurred earlier overlap the forecasts on the future window. This behavior is exemplified in Table 4, where "Date" indicates the decision date (D_0), "Return Factor" indicates the multiplicative return calculated between the market open and close, "Forecast" indicates the prediction of the algorithm for the given day and "Inv. Position" stands for Investor Position, which is based on the current and previous D_{-k} predictions.

In the example presented in Table 4, a 5-day return window is considered. On September 19, the "L" position is maintained according to the forecast of the fall

Table 4 Example predictions when $p = 5$. “L” stands for long position and “S” for short position

Date	Return factor	Forecast	Inv. position
2017-09-19	0.98	0	L
2017-09-20	0.97	1	S
2017-09-21	0.99	0	S
2017-09-22	1.01	1	S
2017-09-25	0.95	1	S
2017-09-26	0.97	0	S
2017-09-27	1.03	0	S
2017-09-28	1.00	0	S
2017-09-29	1.04	0	S
2017-10-02	1.03	0	L
2017-10-02	1.01	0	L

for the same day, as there were no previous indications. For September 21st, 26, 27, 28, and 29, the short position “S” is assumed because there are previous forecasts within the publication window, indicating that the position should be maintained. The financial return factor of the classifier shown in Table 4 is given by $(0.98 * 1.03 * 1.01) = 1.019$, while the loss is $(1 - 0.98) = 0.02$, as the only unanticipated loss was in September 19.

5 Experiments

Four experiments were conducted. The first investigates the ideal percentage of attribute space reduction. The second applies both weight functions to explore the best option for the publication window. The third explores the return window performance, and the fourth carries out a comparison among the algorithms. The first two experiments are intended to explore the best scope of our method, while the latter two focus on optimizing the results of the classifiers.

5.1 What Percentage of Selected Features Achieves the Best Results?

The length of the text and the variety of views expressed in the documents are challenging for analysis, as the same news can have a positive impact on one stock, but negative on another. Moreover, the proposed scope is multi-document. Thus, the amount of terms and points of view expressed is even more considerable.

Feature selection has been applied to reduce the dimensionality of the data and to select, from a generic vocabulary, the most relevant terms for each security. The method used is based on filter, applying the F1 metric as the ranking criteria.

The original full attribute space has 327,623 different terms. An experiment has been performed to evaluate which subset of the vocabulary works better for the problem. We examined 100%, 75%, 35%, 10%, 5%, and 2.5% of attributes. We have used the return window $p = 0$ and for each algorithm the default parameters

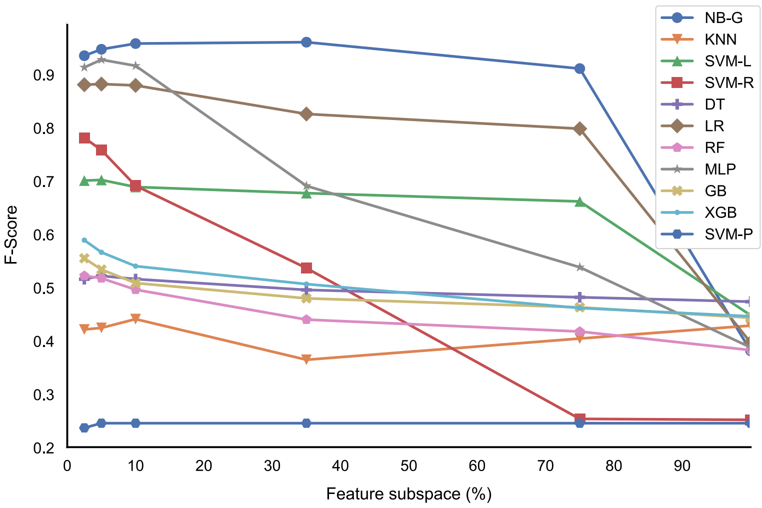

Fig. 4 F-Score average for each feature space subset

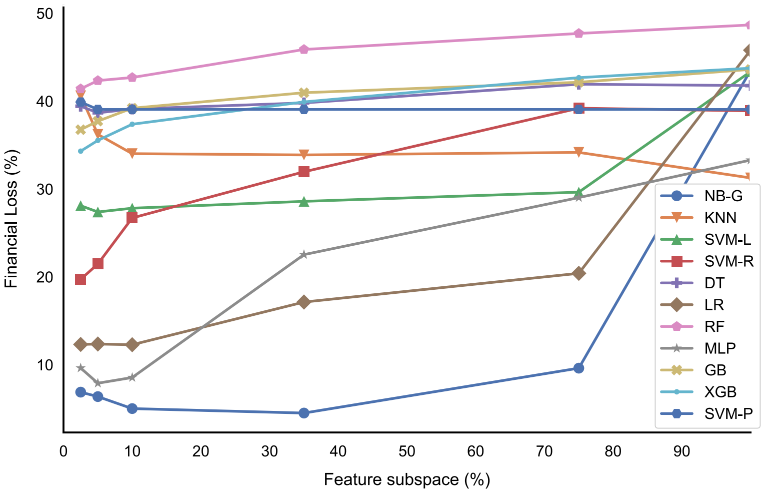

Fig. 5 Loss Percent average for each feature space subset

from *Scikit-learn* and *XGBoost* libraries. The resulting F-Scores are shown in Fig. 4, while the financial metric Loss Percent is given in Fig. 5. The values are the average among all stocks.

From Figs. 4 and 5, it is possible to identify that, for reductions up to 10% of attribute space, not only the performance of the classifiers is maintained but, in most cases, there are significant gains compared to greater feature subspaces. From 2.5% to 10%, the performance varies depending on the classification algorithm, with a higher variation for KNN and SVM-R, in opposite directions.

Table 5 Function w applied to the news' vector representation

ID	k	Equation
TF-IDF-1	1	Equation 9, where d are news from previous day
TF-IDF-5	5	Equation 9, where d are news from the past 5 days
TF-IDF-21	21	Equation 9, where d are news from the past 21 days
EW-TF-IDF-5	5	$tfidf(t_j, d) = \sum_{i=0}^{4} tfidf(t_j, d_i) \times w_2(5, i)$
EW-TF-IDF-21	21	$tfidf(t_j, d) = \sum_{i=0}^{20} tfidf(t_j, d_i) \times w_2(21, i)$

The generic character of the documents can justify the results because of the original data. The words are given to the classifier without previous analysis, without judging the relevance of the news. By applying a filter to the labeled series, less relevant terms and noisy data are discarded, possibly making the decision boundaries less complex.

In terms of computational effort, considering 10% of the original attributes with positive results (therefore, a reduction of 90% in the original vocabulary) indicates the possibility of simplifying the tests. As the majority of the algorithms showed better or similar results on the 2.5% subset compared to 10%, on further experiments, the 2.5% subset has been chosen. The choice contributes to require less computational effort and provides faster executions.

5.2 Is There Any Difference Between the Results Obtained from w_1 and w_2?

The question on whether the publications from the day before are sufficient to predict subsequent price movements, or how many previous days should be considered, and if they should be equally or temporal proportionally weighted are points to be clarified in this experiment. The experiment seeks to determine which function best reflects the news relevance for the investor's decision making.

To compute the TF-IDF frequency of a token t_j in a document d, we use Eq. 9:

$$tfidf(t_j, d) = log(1 + TF(t_j, d)) \times log\left(\frac{|D| + 1}{DF_{t_j} + 1}\right) \tag{9}$$

where d represents the set of documents to be considered, $TF(t_j, d)$ is the term frequency of t_j evaluated on d, $|D|$ is the total number of documents in d and DF_{t_j} is the number of documents in d with occurrences of t_j.

Both weight functions ($w_1(k, i)$ and $w_2(k, i)$) were applied. The naming convention and respective formulas are defined in Table 5. TF-IDF-1, TF-IDF-5 and TF-IDF-21 are computed using w_1 in Eq. 1. Hence, the frequencies are computed as in Eq. 9. For TF-IDF-1 frequencies, d is the set of all news published in the previous day. For TF-IDF-5 and TF-IDF-21, d is the set of news published from $-k$ days before till the previous day.

For EW-TF-IDF-5 and EW-TF-IDF-21 frequencies are computed daily according to Eq. 9 and then, for each day, the TF-IDF values from $-k$ to -1 are calculated

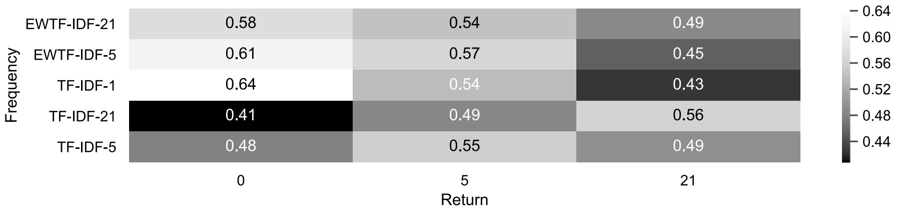

Fig. 6 Average F1 among classifiers trained under different representations, with and without news replication. Light tones indicate higher performance, dark tones worse

according to the weighting criteria in Eq. 2, where the more recent the date, the higher the weight.

The tests were performed using all algorithms over the return windows $p \in \{0, 5, 21\}$. The attribute space was previously filtered to 2.5% of the original (see Sect. 5.1). Results are illustrated in Fig. 6.

From the heatmap presented in Fig. 6, it is possible to notice that publications from the previous day are the most impactful for decisions on $p = 0$, with TF-IDF-1 holding the best overall results. For $p = 5$ return window, the EWTF-IDF-5 showed the best results, while for $p = 21$, the dataset TF-IDF-21 reveals significant improvement compared to the others.

In general, the experiment outcome indicates that when predicting longer return windows, having a longer period of news contributes to better results. Regarding the weight strategy, the results suggest that function $w_2(k, i)$ works for short periods, but needs further adjusts for longer return windows since uniform weights performed better.

The representation with the higher average F-Score between the classifiers will be used on each return window on the next experiments, being TF-IDF-1 for $p = 0$, EWTF-IDF-5 for $p = 5$ and TF-IDF-21 for $p = 21$.

5.3 How Fast Does the Market React to the News?

As per Fama (1970), inefficiencies on obtaining and assimilating new information available may result in a delayed response from investors. This experiment evaluates if news published till the day before the decision provides arbitration opportunities for daily (D_0), from market open to close, weekly (D_{+5}), and monthly (D_{+21}) returns. The $p \in \{0, 5, 21\}$ return windows are expressed in business days, respectively associated with the representations TF-IDF-1, EWTF-IDF-5 and TF-IDF-21, as defined on Sect. 5.2.

With this experiment, it is investigated if the news impact on prices is short or long term. The experiment has been performed with feature selection set to 2.5% subspace, by F-Score filter, and binary classification with the threshold at 0%. Different from previous sections, in this evaluation, all algorithms were trained with grid search, with the parameters space as defined in the "Parameters" column of Table 6.

Table 6 Algorithms and parameters for grid search

Algorithm	Parameters
G-NB	Default
DT	Criteria = {Gini, Entropy};
	Balance = {Automatic, None};
	Max leaf nodes = {5, 10, 35, None};
	Min. samples on Leaf = {1, 3, 5}
RF	Criteria = {Gini, Entropy};
	Estimators = {1500};
	Balance = {Automatic, None};
	OOB score = {True};
	Max leaf nodes = {5, 10, 35, None};
	Min. samples on leaf = {1, 3, 5}
GB	Criterion = {Friedman MSE};
	Estimators = {500};
	Learn. Rate = {0.001, 0.01, 0.1};
	Max. depth = {3, 5, 10};
	Max. leaf nodes = {5, 10, 35, None};
	Min. samples on leaf = {1, 3, 5}
XGB	Booster = {gbtree,gblinear,dart};
	Max. delta step = {0,1,5};
	Reg. lambda = {1, 3, 5, 10, 50, 100}
MLP	Shuffle = {Yes, No};
	Alpha = {0.0001, 0.01, 1, 10};
	Epochs = {100};
	Neurons = {100, 50}
SVM	Kernel = {Linear, RBF, Polynomial};
	C = {0.001, 0.01, 0.1,1};
	Degree = {2, 3};
	Gamma = {0.001, 0.01, 0.1, 1, Auto};
	Balance = {Automatic, None}
LR	Solver = {lbfgs};
	C = {0.001, 0.01, 0.1, 1, 10}
KNN	K = {3, 8, 15};
	Weights = {Uniforme, Distance}

The NB-G algorithm has been used with default parameters, with prior probabilities adjusted according to the training examples. L2 regularization is used on SVM and RL through the parameter "C", on MLP through the parameter "Alpha" and on XGB through "Reg. lambda". The algorithms, DT, RF, and SVM, were tested both with automatic "Balance", the built-in model adjusts for handling unbalanced records, and no treatment.

The out-of-bag samples have been enabled on RF with the parameter "OOB score", which allows the algorithm to pre-validate each of the estimators and reduce overfitting. The "Estimators" parameter has been set to the highest value with feasible execution time, being 1500 chosen for RF and 500 for GB.

For MLP, the default Adam optimizer has been selected, using a single hidden layer with either 100 or 50 neurons as in "Neurons", shuffled samples both enabled and disabled, and 100 epochs. The Linear, Polynomial, and RBF kernels were tested for SVM, with the "Gamma" parameter both with preset values and "Auto", which is equivalent to $\frac{1}{N}$, where N is the number of features. For Polynomial kernel the "Degree" parameter has been tested with 2nd and 3rd order functions. The tree-based algorithms DT, RF, and GB use "Max leaf nodes" and "Min. samples on leaf" to control the size of the tree, as well as "Max.depth", used only on GB.

The results for the different return windows, broken down by classification algorithm, are shown in Table 7. The "Return F." column indicates the factor of the invested capital at the end of the test period, while the "Loss (%)" shows the percent of non predicted price falls between all the ones occurred.

It is possible to notice that both F-Score and financial metrics decrease as the return window gets longer. The financial gain for 21-day analysis drops considerably. As an operating strategy, when looking at predictions on a case-by-case basis, long-term analyzes are impacted by long periods under the short position, where earning opportunities are missed.

The experiment is consistent with the results described in Aase and Öztürk (2011) and Azar (2009), in the sense that the best results have been verified on $p = 0$, with a possible explanation as suggested by the study Yu et al. (2013): the longer the return window, the more events accumulate, bringing high randomness in prices. Moreover, our results comply with those presented in Yu et al. (2013). That is, emerging markets tend to be less efficient since returns for $p = 5$ and $p = 21$ could also be predicted, even though to a smaller extent, with F-score between 0.73 and 0.86 by the two best algorithms.

Long-term predictions are less viable for being used as investment strategies, as can be verified on financial metrics. However, they can still contribute to portfolio optimization, avoiding long-term downtrend stocks, and providing risk reduction.

5.4 What is(are) the Best Algorithm(s) to Forecast Losses?

This experiment explores in further detail the results presented in Sect. 5.3, with a special focus on $p = 0$, which showed the best results. Resuming Table 7, the following observations can be made:

1. Regarding losses, all algorithms managed to reduce it, varying from 88% losses reduction for NB-G on a daily return window on the best case, to 40% for GB on a monthly return window on the worst scenario.
2. For return on investment, the most impressive results have been obtained by NB-G, MLP, and LR, reaching an average of more than 200% return on $p = 0$, between 2017-09-13 and 2018-05-07.
3. It can be noted that the combination of high precision and recall lead the top algorithms to anticipate falls properly without assuming a very conservative position.

Table 7 Computational and financial metrics comparison for daily, weekly, and monthly return windows

Return window	Algorithm	Precision	Recall	F1	Return F.	Loss (%)
0	**NB-G**	**0.9506**	**0.9267**	**0.9360**	**2.2579**	**11.1340**
0	MLP	0.9390	0.9087	0.9181	2.1795	13.5336
0	LR	0.9267	0.8679	0.8843	2.0673	19.1129
0	XGB	0.9252	0.8597	0.8773	2.0375	20.2866
0	SVM-L	0.8902	0.8121	0.8232	1.8362	26.3662
0	SVM-R	0.8233	0.7698	0.7627	1.7300	29.0689
0	RF	0.7284	0.6797	0.6765	1.4130	41.5300
0	GB	0.7179	0.6641	0.6705	1.3721	43.5216
0	SVM-P	0.6606	0.7698	0.6670	1.3937	27.1112
0	DT	0.5451	0.5113	0.4908	1.0017	58.7191
0	KNN	0.5563	0.5575	0.4776	1.0073	50.5174
5	**NB-G**	**0.8727**	**0.8808**	**0.8695**	**1.3121**	**17.0703**
5	MLP	0.9116	0.8102	0.8440	1.3586	27.1499
5	LR	0.9159	0.7337	0.7905	1.3195	33.4121
5	SVM-L	0.8563	0.7729	0.7829	1.2773	27.5322
5	XGB	0.8546	0.6670	0.7175	1.2765	37.7506
5	SVM-R	0.7168	0.6881	0.6549	1.1716	34.5835
5	RF	0.7429	0.4460	0.5022	1.1468	53.2948
5	DT	0.5310	0.4690	0.4806	0.9930	32.3220
5	KNN	0.5994	0.4803	0.4413	1.0129	40.1789
5	SVM-P	0.3435	0.5865	0.3963	0.9570	40.9465
5	GB	0.5010	0.2710	0.3106	0.9927	59.5210
21	**NB-G**	**0.8235**	**0.7752**	**0.7851**	**1.0900**	**21.2542**
21	MLP	0.8251	0.6876	0.7302	1.0946	29.8812
21	RF	0.7751	0.6376	0.6549	1.0375	25.4715
21	LR	0.7990	0.5962	0.6427	1.0743	36.1470
21	SVM-L	0.7332	0.6500	0.6375	1.0475	29.0172
21	XGB	0.7679	0.5703	0.6159	1.0532	33.1079
21	SVM-P	0.5332	0.7721	0.5922	1.0017	22.8087
21	SVM-R	0.6319	0.6684	0.5868	1.0219	31.5701
21	DT	0.5694	0.4825	0.4900	0.9978	26.9699
21	KNN	0.5471	0.5014	0.4763	0.9856	33.4516
21	GB	0.6683	0.4024	0.4600	1.0124	39.7147

The values are the average among all stocks

Bold values represent the best scores

To further illustrate points 1 and 3, Fig. 7 presents the financial performance of the $p = 0$ classifiers on the worst-performing stock in the test period: BRFS3. Figure 7 shows that all algorithms can at least mitigate the losses that occurred in the period when Buy & Hold (B&H) strategy lost almost 60% of the invested

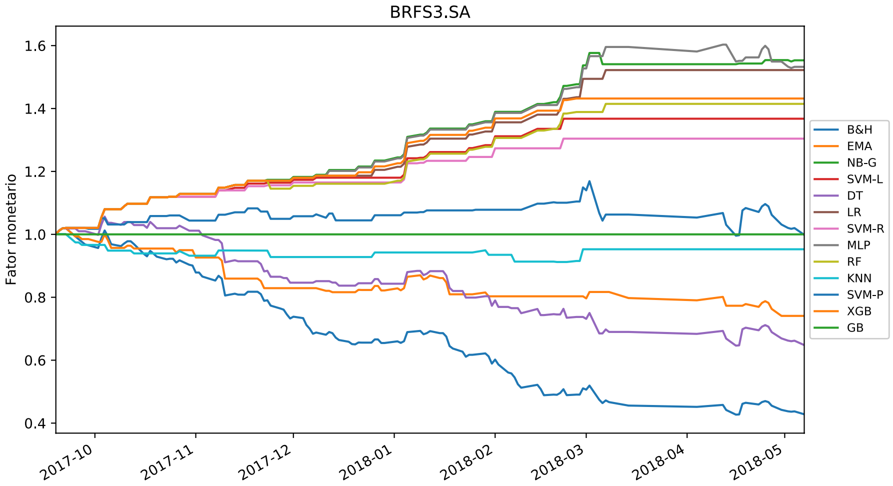

Fig. 7 Classifiers performance for the worst performing stock in the period: BRFS3

capital. It is also noted that, except for the DT algorithm, which may not have been favored by data processing, all other machine learning approaches outperformed the two traditional techniques, Buy & Hold (B&H) and Exponential Moving Averages (EMA).

Figure 7 also indicates that, for the top seven algorithms, falls are anticipated and prevented accurately and almost instantaneously in most cases, unlike the EMA technique, which notoriously suffers delays in sharp falls. Cases can be found between November and December 2017, for example.

It is also interesting to notice that after March 2018, about six months ahead of the latest news used on the models training, some of the top seven models start failing to anticipate price falls, clearly exemplified on MLP. This behavior may suggest considerable changes in the distributions of frequencies for the relevant words used on the model, therefore retraining the models as new publications become available may lead to better results.

Extending the analysis to each of the stocks evaluated, the box plot charts in Fig. 8 presents the distribution of returns and losses, across classifiers, for all the securities. Regardless of the evaluated stock, the results obtained by the classifiers G-NB, MLP, LR, XGB, and SVM-L were always positive in the tested period. Looking at the losses, Fig. 8 shows that NB-G models realized at most 30% of all losses on the test period, which represents a significant reduction. The algorithms MLP, LR, XGB, and SVM-L also showed good results, avoiding at least 50% of the losses for 3/4 of the stocks, but with higher losses for one-third of them.

Table 8 shows the F1 metric for the classifiers along the training, validation and test phases, for different return windows. The gap between training and test scores ranges from 0.00 for NP-G on $p = 0$ to 0.44 for DT on $p = 21$, with medians 0.10, 0.19 and 0.30 for $p = 0$, $p = 5$ and $p = 21$, respectively.

As the stock market is dynamic, the gap cannot be entirely attributed to model overfitting. Changes in the political and economic scenario could also drastically

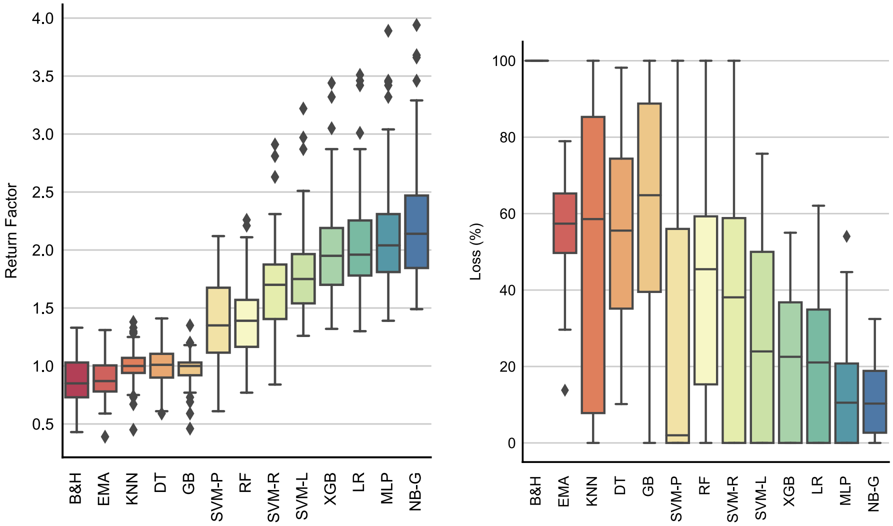

Fig. 8 Distribution of returns and losses for all stocks, classifiers and strategies, when $k = 1$ and $p = 0$

Table 8 F-score obtained by each method for different return windows

Classifier	$(k = 1, p = 0)$			$(k = 5, p = 5)$			$(k = 21, p = 21)$		
	Train	Validation	Test	Train	Validation	Test	Train	Validation	Test
NB-G	0.93	0.98	0.93	0.93	0.93	0.86	0.94	0.70	0.78
MLP	0.99	0.97	0.91	0.99	0.90	0.84	0.99	0.63	0.73
LR	0.96	0.88	0.88	0.98	0.66	0.79	0.99	0.46	0.64
SVM-L	0.97	0.67	0.82	0.98	0.61	0.77	0.99	0.39	0.63
SVM-R	0.86	0.81	0.76	0.84	0.66	0.70	0.77	0.39	0.59
SVM-P	0.76	0.68	0.66	0.73	0.57	0.59	0.86	0.51	0.58
XGB	0.92	0.92	0.87	0.91	0.74	0.80	0.94	0.55	0.61
GB	0.80	0.63	0.67	0.82	0.54	0.43	0.89	0.50	0.46
RF	0.91	0.73	0.67	0.88	0.48	0.55	0.95	0.20	0.65
DT	0.86	0.60	0.49	0.87	0.52	0.49	0.93	0.48	0.49
KNN	0.87	0.47	0.47	0.78	0.31	0.44	0.73	0.23	0.47

impact model performance during tests. The main source of the gap between training and test scores is possibly the use of batch learning instead of online learning. This hypothesis is supported by Fig. 7, as previously pointed on the last two months of the test set, and also by the median F-score gap along with the return windows in Table 8.

KNN, DT, RF, and GB are the classifiers with most decreases in performance from train to test. Using NB-G as a baseline, it indicates that these models are more susceptible to overfitting than the others. For tree-based techniques, both XGB and

RF achieve good results on training, but XGB produces better results on tests, most likely due to L2 regularization.

The best train performances are verified on MLP, LR, and SVM-L, but these models seem to have less generalization capability than NB-G on the test data. One of the intents of the proposed method is to reduce specialist dependence to filter news with relevant events and pre-selecting the pertinent vocabulary. Still, the use of general news and terms may introduce a considerable amount of noise on data, favoring overfitting.

The experiment shows that the most commonly used algorithms cited in Khadjeh Nassirtoussi et al. (2014), Naïve Bayes (NB-G), SVM[3] and ANNs, are between the best choices for forecasting price movements. Furthermore, LR and XGB also achieved good results and are worth considering for similar studies.

In the experiments, we noticed that NB-G, LR, SVM-L, and SVM-P are highly effective, easy to set up, and very fast to train. The MLP and XGB are also very effective but bring complexity on the search for the best parameters since they take up to ten times longer to train (as observed in the experiments' execution logs).

6 Conclusion

Our study focused on the prediction of financial losses using past information from the news. We found a strong relationship between news publications and stock price changes in Brazil, suggesting even short-term arbitrage opportunities. The study shows that it is possible to predict stock price falls using a set of news in Portuguese. We used several experiments to demonstrate that the MLP and NB-G algorithms were able to avert up to 100% of depreciation over the test period, providing returns over 200% on the initial investment for some securities.

Compared to the traditional Buy & Hold and exponential moving averages strategies, machine-learning predictors delivered substantially better results. These results, in most cases, are based on short-term forecasts with $(t-1)$ news achieving a similar performance of studies on the international markets. Besides, to successfully avoid price falls, several algorithms (Gaussian Naïve Bayes, SVM-L, SVM-R, MLP, XGBoost, and Logistic Regression) also led to significant capital gains.

Our experiments also showed that there are significant opportunities for gains based on one-day predictions, considerable gains based on one week-predictions, and fewer gains for one-month forecasts. These results suggest that investors absorb new information with a delay. Our tests also indicate low information efficiency, as prices do not adjust immediately after new data is released.

The provision of a generic news body to the classifier proved to be feasible, mainly when associated with feature selection. We also showed that an attribute space could be reduced to 2.5% of its original size with performance gains when using an F1 filter.

[3] The SVM-L and SVM-R showed good results, while SVM-P didn't.

We found that the use of financial metrics is essential for results evaluation, as accuracy metrics do not necessarily imply financial viability. We concluded that, despite the importance of the recall measure in identifying price falls, the precision of classifiers leads to better financial results by preventing models with conservative positions. In other words, reducing false positives leads not only to gains but, above all, to recover losses sustained as a result of false negatives.

We suggest that future studies use online learning techniques instead of the batch approach. By doing so, we can assure that classifiers will incorporate the most recent information available. Trading costs should also be considered in further analysis since their presence is important to determine whether or not the gains compensate the costs of the training strategy. Additionally, we suggest that our approach should be tested using a larger dataset that includes terms related to the COVID-19 pandemic. Even though we expect that the new flow of textual information significantly changed the distribution of term frequencies (as COVID-19 related terms have gained evidence), we believe that our approach can still be efficient in predicting financial losses on the Brazilian market. Additionally, we also suggest that future studies extend the analysis to the derivative markets.

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

References

Aase, K. G., & Öztürk, P. (2011). *Text mining of news articles for stock price predictions*. Master Thesis, Norwegian University of Science and Technology.

Abu-Mostafa, Y. S., Magdon-Ismail, M., & Lin, H. T. (2012). *Learning from data*. New York: AMLBook.

Alves, D. S. (2015). *Uso de técnicas de Computação Social para tomada de decisão de compra e venda de ações no mercado brasileiro de bolsa de valores*. Doctorate, Universidade de Brasília.

Azar, P. D. (2009). *Sentiment analysis in financial news*. Bachelor Thesis, Harvard University.

Bernanke, B. S., & Reinhart, V. R. (2004). Conducting monetary policy at very low short-term interest rates. *American Economic Review*, *94*(2), 85–90. https://doi.org/10.1257/0002828041302118.

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python* (1st ed.). Newton: O'Reilly Media, Inc.

Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, *2*(1), 1. https://doi.org/10.1016/J.JOCS.2010.12.007.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Chen, T., & Guestrin, C. (2016). XGBoost: a scalable tree boosting system. In *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). https://doi.org/10.1145/2939672.2939785.

Cherkassky, V., & Mulier, F. (2007). *Learning from data: Concepts, theory and methods* (2nd ed.). New York: John Wiley & Sons, Inc.

Chowdhury, S. G., Routh, S., & Chakrabarti, S. (2014). News analytics and sentiment analysis to predict stock price trends. *International Journal of Computer Science and Information Technologies*, *5*(3), 3595.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

Couronné, R., Probst, P., & Boulesteix, A. L. (2018). Random forest versus logistic regression: A large-scale benchmark experiment. *BMC Bioinformatics*, *19*, 270. https://doi.org/10.1186/s12859-018-2264-5.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 21–27.

de Araújo, J. G., & Marinho, L. B. (2018). Using online economic news to predict trends in Brazilian stock market sectors. In *Proceedings of the 24th Brazilian Symposium on Multimedia and the Web* (pp. 37–44).

de Camargos, M. A., & Barbosa, F. V. (2003). Teoria e evidência da eficiência informacional do mercado de capitais brasileiro. *Caderno de Pesquisas em Administração*, *10*(1), 41.

de Carvalho, V. P. (2018). Previsão de séries temporais no mercado financeiro de ações com o uso de rede neural artificial. Master Thesis, Universidade Presbiteriana Mackenzie.

de Oliveira, F. A., Nobre, C. N., & Zárate, L. E. (2013). Applying Artificial Neural Networks to prediction of stock price and improvement of the directional prediction index–Case study of PETR4, Petrobras, Brazil. *Expert Systems with Applications*, *40*(18), 7596. https://doi.org/10.1016/j.eswa.2013.06.071.

Del Negro, M., Giannone, D., Giannoni, M. P., & Tambalotti, A. (2019). Global trends in interest rates. *Journal of International Economics*, *118*, 248.

Fama, E. F. (1970). Efficient capital markets—A review of theory and empirical work. *The Journal of Finance*, *25*(2), 383. https://doi.org/10.2307/2329297.

Fund, I. M. (2019). International financial statistics. Retrieved October 20, 2019, from https://data.imf.org/regular.aspx?key=61545867.

Haddi, E. (2015). *Sentiment analisys: Text pre-procesing, reader views and cross domains*. Doctorate, Brunel University London.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning* (2nd ed.). Berlin: Springer. https://doi.org/10.1007/b94608.

Hotho, A., Nürnberger, A., & Paaß, G. (2005). A brief survey of text mining. *LDV-Forum*, *20*(1), 19–62.

Khadjeh Nassirtoussi, A., Aghabozorgi, S., Wah, T., & Ngo, D. (2014). Text mining for market prediction: A systematic review. *Expert Systems with Applications*, *41*(16), 7653. https://doi.org/10.1016/j.eswa.2014.06.009.

Kumar, B. S., & Ravi, V. (2016). A survey of the applications of text mining in financial domain. *Knowledge-Based Systems*, *114*, 128. https://doi.org/10.1016/j.knosys.2016.10.003.

Li, X., Xie, H., Song, Y., Zhu, S., Li, Q., & Wang, F. L. (2015). Does summarization help stock prediction? A news impact analysis. *IEEE Intelligent Systems*, *30*(3), 26. https://doi.org/10.1109/MIS.2015.1.

Lopes, T. J. P., Hiratani, G. K. L., & Barth, F. J. (2008). Mineração de opiniões aplicada à análise de investimentos. In *WebMedia '08: Proceedings of the XIV Brazilian Symposium on Multimedia and the Web* (pp. 117–120). https://doi.org/10.1145/1809980.1810012.

Makrehchi, M., Shah, S., & Liao, W. (2013). Stock prediction using event-based sentiment analysis. In *IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)* (Vol. 1, pp. 337–342). https://doi.org/10.1109/WI-IAT.2013.48.

McCallum, A., & Nigam, K. (1998). A comparison of event models for naive bayes text classification. In *Proceedings of the 15th AAAI Workshop on Learning for Text Categorization* (pp. 41–48).

Mitchell, T. M. (1997). *Machine learning* (1st ed.). New York: McGraw-Hill.

Mittal, A., & Goel, A. (2012). Stock prediction using twitter sentiment analysis. In *International Conference on Web Intelligence and Intelligent Agent Technology*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, *12*, 2825–2830.

Rehbein, O. M. (2012). Mineração de texto aplicado à análise de carteira de ações. Bachelor Thesis, Universidade de Santa Cruz do Sul.

Shiller, R. J. (2000). *Irrational Exuberance* (3rd ed.). Princeton: Princeton University Press.

Tversky, A., & Kahneman, D. (1991). Loss aversion in riskless choice: A reference-dependent model. *The Quarterly Journal of Economics*, *106*(4), 1039.

Xing, F. Z., Cambria, E., & Welsch, R. E. (2018). Natural language based financial forecasting: a survey. *Artificial Intelligence Review*, *50*(1), 49. https://doi.org/10.1007/s10462-017-9588-9.

Yim, J., & Mitchell, H. (2005). A comparison of corporate distress prediction models in Brazil. *Nova Economia Belo Horizonte*, *15*(1), 73.

Yu, H., Nartea, G. V., Gan, C., & Yao, L. J. (2013). Predictive ability and profitability of simple technical trading rules: Recent evidence from Southeast Asian stock markets. *International Review of Economics and Finance*, *25*, 356. https://doi.org/10.1016/j.iref.2012.07.016.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.