Journal of Computational Science 19 (2017) 43–56

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Assessing the usefulness of online message board mining in automatic stock prediction systems

Ramiro H. Gálvez [a,*], Agustín Gravano [a,b]

[a] Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Argentina
[b] Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina

ARTICLE INFO

Article history:
Received 23 August 2016
Received in revised form 7 November 2016
Accepted 4 January 2017
Available online 10 January 2017

Keywords:
Stock market
Text mining
Latent semantic analysis
Ridge regression
Random forest

ABSTRACT

We provide evidence of the usefulness of exploiting online text data in stock prediction systems. We do this by mining a popular Argentinian stock message board and empirically answering two questions. First, is there information in the online stock message board useful for predicting stock returns? Second, if useful information is found, is it novel or it is simply a different way of expressing information already available in the past behavior of stock prices?

To address these questions, we build and validate a series of predictive models using state-of-the-art machine learning and topic discovery techniques. Running experiments in which the models are trained with different combinations of features extracted from the past behavior of stock prices, or mined from the online message boards.

Evidence suggests that it is possible to extract predictive information from stock message boards. Furthermore, we find that adding this information improves the performance of classification systems trained solely on technical indicators. Our results suggest that information from online text data is complementary to the one available in the past evolution of stock prices. Additionally, we find that highly predictive features derived from the message board data seem to have an important and relevant semantic content.

1. Introduction

Understanding phenomena which escape the online domain by analyzing people's online interactions is an active and promising area of research. Recent examples of this are [1] or [2]. The former study shows how Google searches can be used to predict flu outbreaks. The latter shows how psychological language on Twitter may be used to predict county-level heart disease mortality. In the present work we follow this line of research by analyzing whether stock message board activity may be mined in order to extract valuable information for predicting future stock returns.

The main objective of this article is to provide evidence of the usefulness of exploiting online text data in automatic stock prediction systems. We do this by mining a popular Argentinian stock message board and empirically answering two questions:

Q_1: Is there information in the online stock message board useful for predicting stock daily returns?

Q_2: If useful information is found, is it novel or it is simply a different way of expressing information already available in the past behavior of stock prices?

The machine learning, decision support systems, expert systems and data mining communities have written a large body of literature aimed at forecasting stock returns. Traditionally, most of these studies place a large degree of importance on the use of advanced and modern forecasting models, making a big emphasis on the need of them being flexible and powerful enough to capture nonlinearities in the evolution of prices. On the other hand, a lesser emphasis has been made on feature extraction and engineering. Regarding feature extraction, the norm has been to train predictive systems on features derived from the past evolution of stock prices (e.g., [3–6]). In particular, a commonly used approach for incorporating the past evolution of stock prices into these system is to feed them the evolution in the values of a selected set of technical indicators as predictive features (e.g., [7,8]). Technical indicators are metrics whose values are derived from the generic price activity of a stock, and are commonly used by traders to predict the future price levels, or simply the general price direction, of a security by looking at their patterns.

* Corresponding author.
E-mail addresses: rgalvez@dc.uba.ar (R.H. Gálvez), gravano@dc.uba.ar (A. Gravano).

http://dx.doi.org/10.1016/j.jocs.2017.01.001
1877-7503/

As stated by [9], attention has lately shifted from sophisticated learning models to better and more diverse sources of information. For example, [10] mine Twitter posts, [11] mine financial events reported in 8-K documents, [12] mine news articles mentioning stocks, [13] mine financial news articles, [14] mine weekly Google trends activity, [15] mine the evolution of Wikipedia usage patterns, and [16] mine news articles comprising corporate announcements.

This article delves further into using novel sources of information to predict stock returns, and even though it is not the first one aimed at studying how online text data can be mined in order to predict stock returns, it has three characteristics that make it novel. First, with the notable exception of [17], most studies simply analyze if it is possible to predict stock returns behavior using online text data. Although this is an interesting question by itself, a critical question for stock market prediction systems design is to check if the new data provides additional information over the one already present in the past evolution of stock prices. Note that, if mining online text data provides the same amount of information as systems which only use as input features the past evolution of stock prices, the effort of mining the online text data has a negative cost-benefit value. We address this inquiry by answering Q_2. Second, as mentioned by [18], most studies model online text data using the plain vanilla bag-of-words model (see [19]). In this work we employ a more sophisticated technique, Latent Semantic Analysis (LSA) [20], which aims at discovering latent topics in the text.[1] Third, as far as we know, this article is the first one to study the case of the Argentinian stock market in the context of stock returns prediction using online text data.

From our results, we find evidence that suggests that it is effectively possible to extract predictive information from stock message boards. Furthermore, we find that adding this information improves the performance of state-of-the-art classification systems trained solely on technical indicators. Although not conclusive, our results suggest that information from online text data is complementary to the one available in the past evolution of stock prices. Additionally, we find that highly predictive features derived from the message board data seem to have an important and relevant semantic content.

The rest of the paper is structured as follows. Section 2 presents and describes the data used during the analysis. Section 3 makes a high-level description of how we address questions Q_1 and Q_2, and later details our methodological decisions. Section 4 presents our main results and Section 5, our conclusions.

2. Data

This section describes in detail the characteristics and sources of our data, as well as the decisions taken in order to select which stocks to analyze over which time periods. Data comes from two sources. First, from public records we retrieve historical daily prices of stocks traded in the main Argentinian stock market. Second, from a popular Argentinian online stock message board we collect a large corpus of written posts. The following subsections describe both sources in detail.

2.1. Daily stock prices

Data of the daily evolution of stock prices in the Buenos Aires Stock Exchange market was retrieved from the webpage of Rava Bursátil S.A., a large Argentinian brokerage firm.[2] We chose to focus on analyzing stocks included in the MERVAL Index calculation. The MERVAL Index is the most important primary index of the Buenos Aires Stock Exchange market. It is a price-weighted index, calculated as the market value of a portfolio of stocks which are selected based on their market share, number of transactions and price. In 2015 the index was calculated using data from twelve stocks. For each of these twelve stocks, we retrieved the evolution in the values of the opening, closing, maximum and minimum daily prices; we also retrieved data on each day traded volume.

The main reason for restricting our analysis to this subset of stocks comes from the fact that, as the Argentinian is not an extensively dynamic market, in order to be able to predict stock returns from online information, we need to concentrate on stocks which have a large amount of transactions, ignoring any stocks for which days may pass without any transaction. By construction, stocks included in the MERVAL Index calculation guarantee a high volume of transactions.

2.2. Stock message board posts

Online text data was collected from the message board of the webpage of Rava Bursátil S.A.[3] This message board is well-known in the Argentinian trading community for being a site in which agents effectively operate on the market and actively exchange opinions. To post on the message board, users must first register and an administrator must approve their registration. All threads are supervised by administrators, who can ban users who violate norms of good behavior. User posts tend to be written in informal language, although there is a fair amount of technical posts in which a more specific, formal language is preferred. Most texts are written in Spanish, but there is a non-negligible number of technical posts written in English or Portuguese, especially in the thread associated to APBR (Petróleo Brasileiro S.A.). The use of emojis is very spread out. Emojis are small images placed in the text area and are usually used to express an idea or emotion. Users are allowed to include emojis from a set of more than fifty different ones.

The message board is composed by threads. One feature that we exploit heavily in this study is that each analyzed stock has an exclusive thread. Fig. 1 shows the structure of the message board main page. Note that each of the stocks analyzed in this study has an individual thread associated — e.g., APBR, PAMP (Pampa Energía S.A.), GGAL (Grupo Financiero Galicia S.A.) and ERAR (Ternium Siderar S.A.). On its own, each thread is a collection of posts written by users. As Fig. 2 shows, posts may contain emojis, as well as quotations from others posts in the same thread. Posts may also contain images and links to external references. Each post also includes metadata indicating its author, date and time.

Our analysis covers the period between 2010-06-01 and 2015-07-31. Although price evolution data from the beginning of 2005 to the present is available, we chose 2010-06-01 as the starting date because, before this time, activity in the message board was too sparse. On the other hand, we chose 2015-07-31 as the ending date because a presidential election process began in Argentina in August, 2015, during which radical structural reforms were promised by the opposition party (who ended up winning the elections), which in turn lead to abnormal market behavior in the subsequent months.

Finally, a considerable amount of user activity is required to extract statistically reliable patterns from their interactions. Therefore, in this work we chose to restrict our analysis to stocks for which its associated thread has more than 20,000 posts during the analyzed period. Table 1 lists the eight stocks from MERVAL that

[1] Note that none of the articles reviewed in [18] uses this technique to model online text data.

[2] http://www.ravaonline.com. Last accessed: 2016-06-01.

[3] http://foro.ravaonline.com. Last accessed: 2016-06-01.

Fig. 1. Main page structure of the analyzed message board.

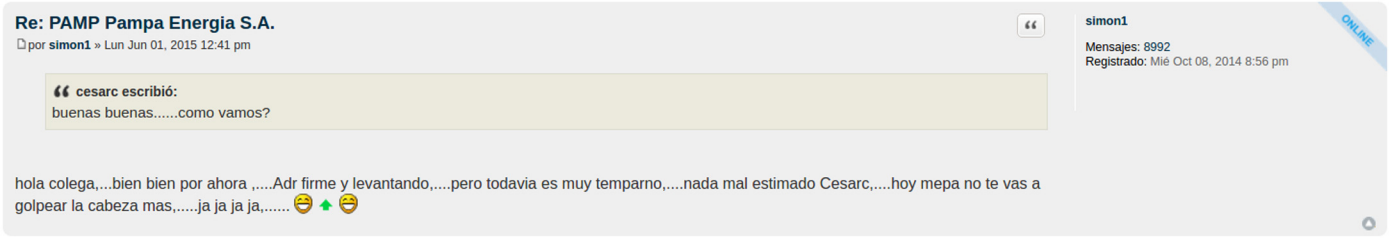

Fig. 2. Example of a post.

satisfy this condition and were considered in our analysis, detailing in which industry and sector each one operates. It is very important to note that, by restricting our analysis to stocks used in the calculation of the MERVAL index and stocks which have at least 20,000 posts in their threads, our study focuses on stocks with a high number of transactions and for which there is a considerable amount of debate in their respective threads.

Fig. 3 summarizes stock message board activity for our sample of stocks. From panel "a" we observe that the number of posts varies greatly across threads. However, panel "b" shows that the number of unique authors in each thread does not vary as much. This indicates that in threads with a higher number of posts, users tend to write more posts on average, probably replying to one another and behaving more like a community. Panel "c" shows that the distribution of the number of written posts by author has a long tail, as there is a considerable mass of users with more than 100 posts or even 1000 posts. Panel "d" plots the evolution of the number of monthly posts for all considered threads as a whole. A pattern which stands out from this panel is the steady activity rise, especially since the year 2013. In order to see if this pattern is stable across threads, Fig. 4 plots the evolution of the monthly number of post by thread. This figure is illustrative as it shows that, although the aggregated activity rose during the analyzed period, the rise was not even across threads; instead, at the thread level, we observe bursts of activity for short periods of time. Lastly, panels "e" and "f" show the distribution of posts across days of the week and through time of the day (for the latter we considered bins of 15 min). Both panels suggest that users are specially active when the market is operating. Note the sharp fall in the number of posts on weekends. Also, taking into account that the stock market is open from 10:00 to 17:00, most posts were written when the market was open, with peaks around opening and closing time.[4]

Table 1
Companies being analyzed and details on their industry and sector.

Ticker	Company name	Sector	Industry
APBR	Petróleo Brasileiro S.A.	Basic Materials	Major Integrated Oil & Gas
COME	Sociedad Comercial del Plata S.A.	Conglomerates	Conglomerates
EDN	Edenor S.A.	Utilities	Electric Utilities
ERAR	Ternium Siderar S.A.	Basic Materials	Steel & Iron
GGAL	Grupo Financiero Galicia S.A.	Financial	Money Center Banks
PAMP	Pampa Energía S.A.	Utilities	Electric Utilities
TS	Tenaris S.A.	Basic Materials	Steel & Iron
YPFD	YPF S.A.	Basic Materials	Major Integrated Oil & Gas

Source: Yahoo! Finance.

[4] The sharp fall in activity exactly after midnight is due to the fact that posting in the message board is not possible from midnight to 8:00.

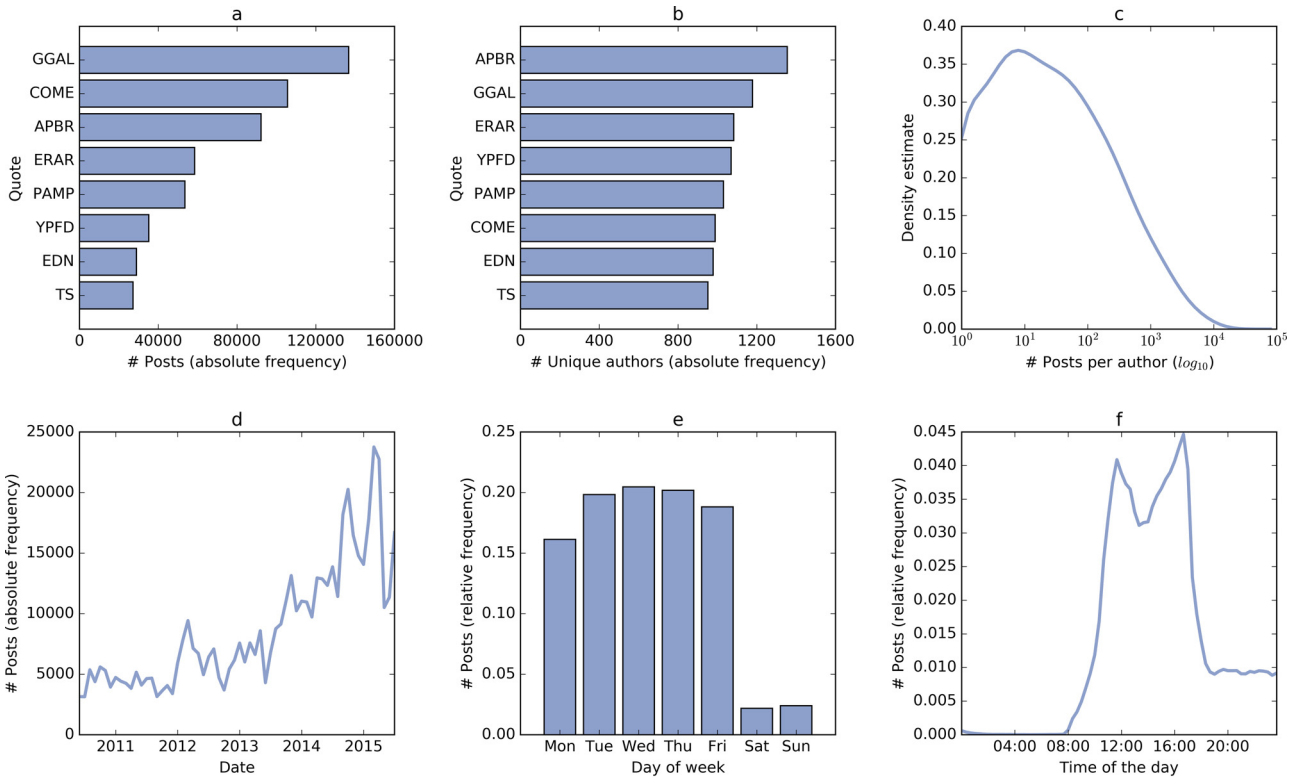

Fig. 3. Message board activity.

3. Methods

In order to answer Q_1 and Q_2 we propose two experiments. In the first one, which aims at answering Q_1, we train, validate and test systems which predict stock returns solely on data coming from the online message board. We then compare the performance achieved by these systems to that obtained by two baseline models that rely only on past stock price data. If the proposed systems behave at least as well as the baseline models do, we would conclude that effectively it is possible to extract information with predictive content from the online message board.

In the second experiment, which aims at answering Q_2, we first replicate competitive models developed in the machine learning community to predict the direction of stock returns. These models use technical indicators as input features. Subsequently, we expand their feature spaces with data from the online message board. If adding message board data to these systems improves the performance, this would evidence that online message board data complements the information captured by technical indicators about the past behavior of prices.

Forecasting stock returns from past prices, and to a greater extend from unstructured data, involves making a large number of methodological decisions for which there is no clear consensus. The remaining of this section summarizes and justifies the main methodological decisions that were taken to arrive to our results.

3.1. Text pre-processing

As mentioned above, each post may contain images, quotations to other posts, links to external references, and emojis. In the rest of our analysis, we focus on the original text and emojis included in each post, ignoring all the other elements. We include posts from all authors, and in no way the authors' behavior or reputation is directly considered in our systems. For the case of emojis, since these are presumed to contain important information on sentiment polarity (see [21]), we treat them as regular words and add them to the message board vocabulary. For doing this, to each post containing emojis we add new tokens consisting of the names of the emojis they contain with the prefix "emoji_"; for example, in a post were the emoji "Laughing" appears twice, the token "emoji_Laughing" is added two times to its main text.

For each post in a thread we apply the following procedures. We tokenize its text using NLTK's sentence and word tokenizers [22], which converts the text into a list of tokens. Then, for each token, we convert any uppercase character into its corresponding lowercase character and replace every non-ASCII character with its closest ASCII character.[5] Additionally, authors in this stock message board tend to informally emphasize some words by repeating some of their characters (e.g., "*hooolaaaa*" instead of "*hola*"). Thus, taking into account that repeating characters is pretty uncommon in Spanish (except for "c", "l" and "r", which are commonly repeated twice), when we find an alphabetic character repeated more than once (or more than twice in the case of "c", "l" and "r") we simply delete any extra repetitions.[6]

Finally, we ignore all tokens which are stopwords (according to NLTK's list of Spanish stopwords); filter out all tokens for which the first character is a non-alphanumeric one; filter out all tokens which

[5] For doing this replacement we use the 'unidecode' library available for Python, see: https://pypi.python.org/pypi/Unidecode. Last accessed: 2016-06-01.

[6] We acknowledge that character repetition is commonly used to emphasize the meaning of a word (i.e., "*hiiiiggggh*" meaning higher than "*high*"), and that removing character repetition could translate into considering two tokens with different degrees of meaning as the same one. But, as there are multiple ways in which character repetition may be used in a single token, the exact same use of the character repetition for a token can be rare. This would translate into having a large number of semantically related low support tokens, many with high risk of being removed by following filtering procedures. To avoid losing these potentially predictive tokens, we opt to remove character repetition as described.

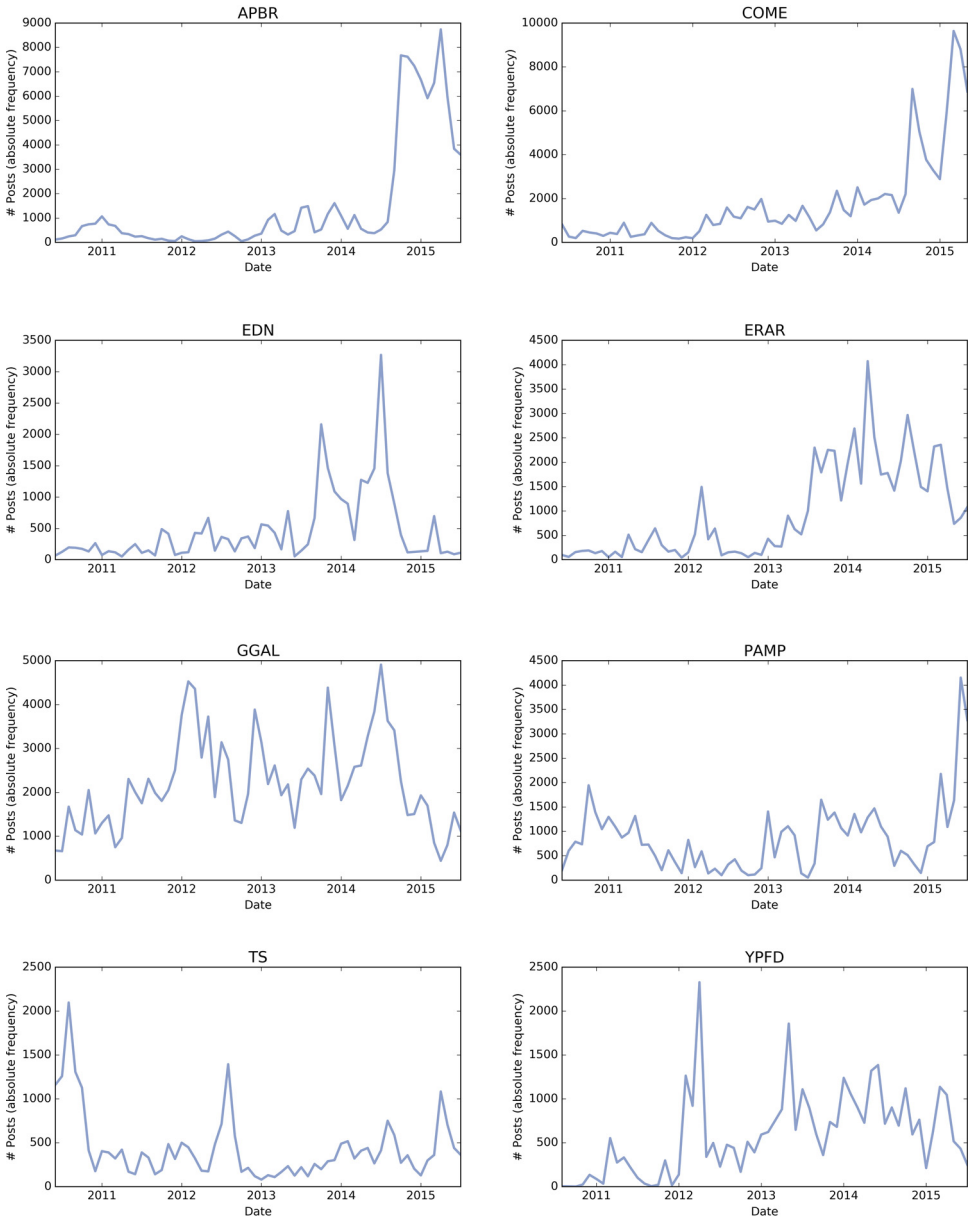

Fig. 4. Message board activity by thread.

appear five or fewer times in the training data[7]; and replace all tokens which we identify as numbers for the special token "*_NUM_*".

Table 2 shows the distribution of tokens across threads for the whole period under analysis after all pre-processing has been completed. It contains details on the number of unique tokens per thread (including emojis), the total number of tokens in each thread (including emojis) and the total number of emojis in each thread.

3.2. Dimensionality reduction and topic discovery strategy

Once the text of each thread is expressed as lists of filtered and cleaned tokens, we model it using a modified version of the bag-of-words model. We first construct a matrix A where each row i represents a day (higher values of i correspond to days closer to the present), each column j represents a token of the thread's vocabulary (B) and the value of each element a_{ij} equals the number of times the word associated to column j was used on the day associated to row i in the thread under analysis.

Given that the message board activity grew in size as time evolved (see panel "d" of Fig. 3) and that there is a presence of activity spikes in the threads of different stocks (see Fig. 4), values

Table 2
Distribution of tokens across threads.

Quote	# Unique tokens	# Tokens	# Emojis
APBR	20,250	1,237,671	32,896
COME	18,986	1,370,022	56,456
EDN	9047	387,956	15,929
ERAR	16,277	935,852	25,507
GGAL	20,138	1,559,695	61,269
PAMP	12,420	644,404	37,568
TS	9985	481,511	15,291
YPFD	12,005	512,068	12,340

[7] Note that this is done dynamically based on the training data for each model and that, as it will be described in Section 3.3, training data varies following a growing window scheme in our experiments.

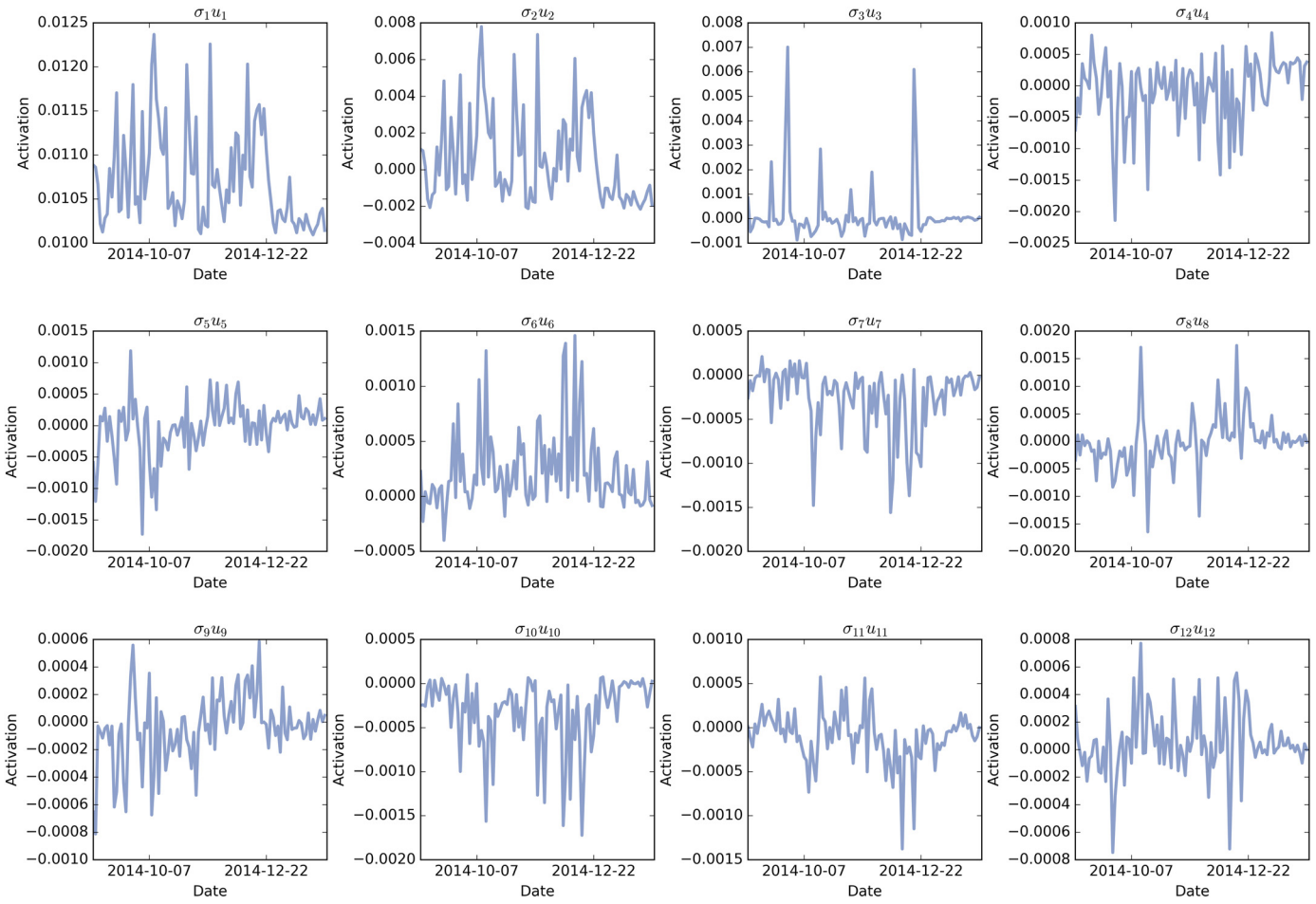

Fig. 5. Evolution of the first twelve detected topics for the period going from 2014-09-03 to 2015-02-02 in the YPFD thread.

of a_{ij} may grow or experience burst of activity as i gets larger. To better handle this, we construct A', a new version of A with two modifications. First, element a'_{ij} is now related to the proportion of times token j was used on day i relative to the total number of tokens which were used on day i. Second, given that on days with less activity the impact of an additional token is greater than for days with greater activity,[8] we smooth these relative frequencies by row using the Laplace smoothing method with $\alpha = 1$, commonly known as the add-one smoothing method (see [19]). Summing up, being $|B|$ the size of the vocabulary, the value of each element a'_{ij} is now calculated as follows:

$$a'_{ij} = \frac{a_{ij} + 1}{\sum_{k=1}^{|B|} a_{ik} + |B|} \tag{1}$$

Note that matrix A' is expected to have many more columns than rows, and thus would normally be regarded as unsuitable for training machine learning models. For this reason, it is necessary to conduct some sort of dimensionality reduction, and we chose Singular Value Decomposition (SVD) of matrix A' for this purpose. It should be noted that, as stated in [23], combining some transformed form of the bag-of-words model with SVD is known as Latent Semantic Analysis (LSA).

SVD is a matrix factorization algorithm which decomposes a given matrix as the product of three other matrices with useful properties. Specifically, for the case of A', with I rows and $|B|$ columns, SVD will factorize it as the product of three matrices as follows:

$$A' = U \Sigma V^T, \tag{2}$$

where $U \in \mathbb{R}^{I \times I}$ is orthogonal, $V \in \mathbb{R}^{|B| \times |B|}$ is orthogonal, and $\Sigma \in \mathbb{R}^{I \times |B|}$ is a diagonal matrix which contains the singular values of A' in its diagonal, sorted in descending order (we call these elements σ_i). If we consider only the first h columns of U (we call this matrix U_h), the first h columns of V (we call this matrix V_h), and a submatrix formed by the first h rows and h columns of Σ (we call this matrix Σ_h), then the resulting decomposition is called Truncated Singular Values Decomposition (TSVD). Most relevant to our analysis is that, in the context of LSA, it is presumed that U_h indicates the presence of different latent topics across documents (or days in our case) and V_h indicates the association of tokens to topics. To better visualize these concepts, Fig. 5 plots for the thread of YPFD the daily activation of the first twelve detected topics for the period going from 2014-09-03 to 2015-02-02.

Taking all of this into account, if matrix A' is constructed on training data and TSVD is applied to it, the values of matrix $U_h \Sigma_h \in \mathbb{R}^{I \times h}$ can be used as input features for training machine learning models. This means that, when training models using data from the online message board, each observation will include the daily activation levels of the different latent topics as input features. Also note that, to make predictions, one can construct in a similar way the modified version of the bag-of-words model for the testing data

[8] For example, compare a day on which only the token *"hola"* was used once, to another day on which *"hola"* was also used once, but other 999 tokens were also used. In the first case the relative frequency of the element corresponding to *"hola"* for that day will we equal to 1, whereas in the second it will be equal to 0.001.

(let us call this matrix A'_{ts}) and calculate the input for the models to predict at testing as $A'_{ts}V_h$. Finally, the value of h that maximizes a system trained on $U_h\Sigma_h$ depends entirely on the data. For this reason, a sound experimental setup is necessary for selecting the best values of this and other hyperparameters.

3.3. Experimental pipeline

In this work we predict variables which have a time-series structure (i.e., the order of the data matters), meaning that traditional methods for validating machine learning models, such as cross-validation or bootstrapping, may be misleading. If used naively, such methods break the temporal structure of the data by allowing models to be trained on data which may be newer than some fraction of the data used for validating or testing it. In consequence, systems trained on data that is not supposed to be available at training time will fail dramatically when deployed. For this reason, as recommended in [18], we use a growing windows scheme to validate and test our models (see [24]). Specifically, if the data is contained in matrix $X \in \mathbb{R}^{I\times|B|}$, where each row represents an observation and each column a feature (for simplicity, we assume that one column contains the variable to be predicted), the scheme works as follows:

1. Divide X into two matrices, X_1 and X_2, such that X_1 contains the first n_1 rows of X, and X_2 contains the last n_2 rows of X ($n_1 + n_2 = I$).
2. Train the learning model using data from X_1, and make predictions for the first s rows of X_2 (with $s < n_2$).
3. Remove the first s rows from X_2 and append them to X_1 (where $n_1 \leftarrow n_1 + s$ and $n_2 \leftarrow n_2 - s$).

Steps 2 and 3 are repeated until X_2 holds no more observations. These steps are illustrated in Fig. 6. Once the process stops, performance metrics may be calculated by comparing the predictions obtained for each observation in X_2 to their actual values. We divide the data for each stock into three sets: Training, Validation and Testing sets, as depicted in Fig. 6. Observations are always sorted from older to newer, such that the Validation (Testing) set contains newer observations than the Training (Validation) set. To optimize the model hyperparameters, we train our models on the Training set, and estimate the out-of-sample performance on the Validation set, always using the growing windows setup as explained in the previous paragraph. Next, we combine the Training and Validation sets into one dataset, and use it to train our systems, which we subsequently test on the Testing set, again using the growing windows scheme. In all cases we set s equal to 20 (roughly a calendar month) and the initial value of n_2 equal to 120 (roughly half a year).

3.4. Forecasting daily returns using data from the online message board posts

To answer Q_1, we train models which aim to predict the daily return a particular stock will have on a given day t by only using data collected from its message board thread. We do this by training machine learning models which use as predictive features the daily activation of the first h topics detected on its thread training data (calculated as explained in Section 3.2). Concretely, we predict the daily return of a stock expressed as percentage, which is defined as

$$r_t = 100\left(\frac{\bar{P}_{t+1}}{\bar{P}_t} - 1\right), \tag{3}$$

where $\bar{P}_t$ is the average of the opening, closing, maximum and minimum prices at day t. It is important to note that, given that the market closes at 17:00, for predicting r_t and its direction, we only consider posts written up to 17:00 of day t. Any post written after this hour is considered to belong to the following calendar day.

In this experiment we choose **ridge regression** as the learning algorithm [25]. Ridge regression is quite similar to ordinary least squares linear regression, with the advantage that it avoids overfitting the training data by penalizing high coefficients. This penalization grows with the value of hyperparameter λ, with $\lambda = 0$ meaning no penalization. Formally, this technique aims at finding the values of b and β_i which, given the training data, minimize the following expression,

$$J(b, \beta_1, \beta_2, \ldots, \beta_h) = \sum_{t=1}^{n_1}\left(b + \sum_{i=1}^{h}\beta_i(U_h\Sigma_h)_{t,i} - r_t\right)^2 + \lambda\sum_{i=1}^{h}\beta_i^2, \tag{4}$$

where $U_h\Sigma_h$ is a matrix containing the activation in the first h detected topics in the stock's thread (obtained as explained in Section 3.2).

Before training each model, all input variables are centered by removing the mean and scaled to unit variance. Both mean and standard deviation are always estimated from the training data only.

We evaluate the performance of our models on the out-of-sample predictions using two standard metrics, the **root mean square error** of the prediction (RMSE)[9] and **Pearson's product-moment correlation coefficient** (ρ)[10] obtained by comparing the predictions to the real values. In the case of ρ we are also able to estimate its statistical significance.

For each stock, we search for the best hyperparameters using a grid-search approach [26] on λ and h, with $\lambda \in \{0, 0.001, 0.01, 0.1, 0.25, 0.50, 0.75, 1, 1.25, 1.5, 1.75, 2, 2.25, 2.5, 2.75, 3\}$ and $h \in \{10, 20, 30, 40, 50\}$. We select the hyperparameters for which the RMSE is minimized in the validation phase.

In order to contextualize the results of the proposed models, we compare them against two baseline models. The first baseline model predicts the return on day $t+1$ as the observed return on day t (i.e., $\hat{r}_t = r_{t-1}$). We refer to this model as the **Lagged Return** model. The second baseline model predicts the return at day $t+1$ as the average return in the training period (i.e., $\hat{r}_t = \sum_{k=1}^{n_1} r_k/n_1$). We call this the **Training Average** model.[11] Note that even though both of these models are quite simple, they are believed to be competitive for variables with structural breaks as the one which is being forecasted (see [27]).

3.5. Forecasting daily returns using data from the online message board posts and technical indicators

To address Q_2, we train competitive models based on technical indicators and analyze how their performance vary when data from the online message board is added to them. As mentioned above, technical indicators are metrics derived from generic stock price activity. They are commonly used by traders to predict future price levels of a security, or simply its general price direction, by looking at past patterns. In our analysis we use the same technical indicators presented in [8], as listed in Table 3 along with short descriptions and relevant references. To illustrate, Fig. 7 plots the evolution of

[9] The RMSE is equal the square root of the average of the square of all errors. Lower values indicate better predictive performance.

[10] The ρ coefficient is a measure of linear dependence between two variables, giving values between +1 and −1 inclusive, where 1 reflects total positive linear correlation, 0 no linear correlation, and −1 total negative linear correlation. Higher values indicate better predictive performance.

[11] Note that predictions of these systems tend to vary slightly as the training data window grows, as explained in Section 3.3.

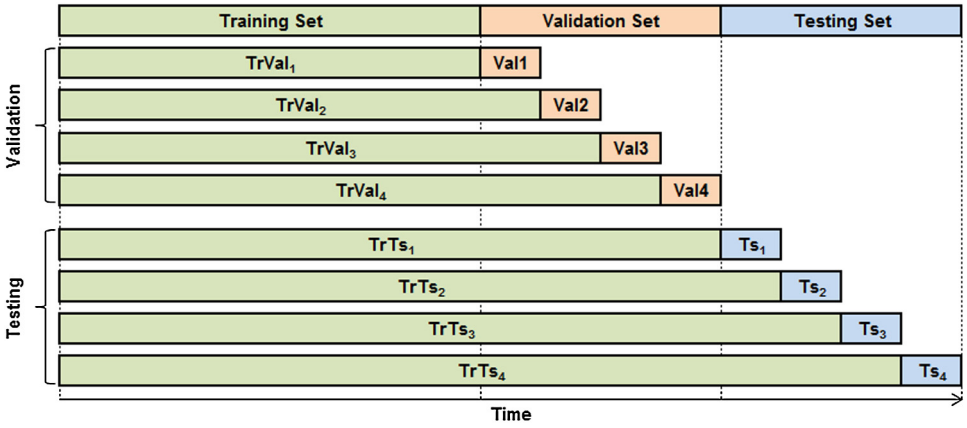

Fig. 6. Diagram of the scheme used for validation and testing.

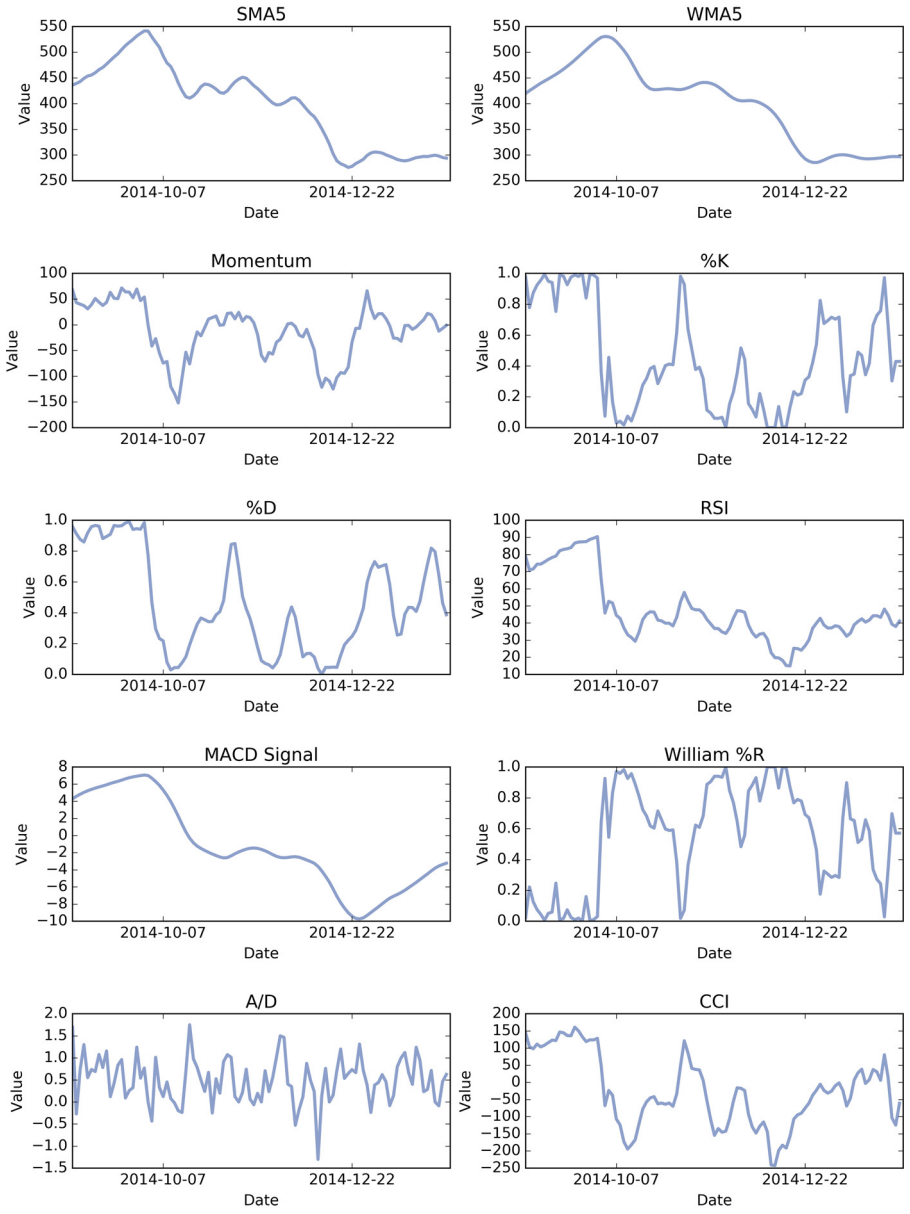

Fig. 7. Evolution of the set of technical indicators used for the period going from 2014-09-03 to 2015-02-02 in the YPFD thread.

Table 3
Technical indicators included as features in our forecasting systems.

Technical indicator	Description	Reference
SMA5 (Simple 5-day moving average)	A trend-following technical indicator obtained by running a 5-day simple moving average over the series which results from averaging each day's opening, closing, maximum and minimum prices.	[28,8,4]
WMA5 (Weighted 5-day moving average)	A trend-following technical indicator obtained by running a 5-day linearly weighted moving average over the series which results from averaging each day's opening, closing, maximum and minimum prices.	[28,8,4]
Momentum	A momentum indicator calculated as the difference between the current day's closing price and the closing price n days before. In our experiments we set $n = 9$.	[8,4,29]
%K	A momentum indicator which focuses on the location of the difference between the closing price and the lowest low price relative to the high-low range over the previous n days. In our experiments we set $n = 14$.	[28,8,4,29]
%D	Simple 3-day moving average of %K.	[28,8,4,29]
RSI (Relative Strength Index)	An impulse indicator that compares the magnitude of recent gains to recent losses in an attempt to determine "overbought" and "oversold" conditions of an asset.	[28,8,4,30,29]
MACD Signal (moving average convergence divergence)	A technical indicator which turns two trend-following indicators, moving averages, into a momentum oscillator by subtracting the longer moving average from the shorter one. It is calculated as the 12-day exponential moving average (EMA) of the closing prices less the 26-day EMA of the closing prices.	[28,8,4,29]
William %R	A momentum indicator which focuses on the location of the difference between the highest high price and the closing price relative to the high-low range over the previous n days. In our experiments we set $n = 14$.	[28,8,4]
A/D (Accumulation/ Distribution)	An impulse indicator calculated as the difference between a day maximum price and the previous day closing price divided the difference between the day maximum and minimum prices.	[8,4]
CCI (Commodity Channel Index)	An oscillator technical indicator which attempts to identify starting and ending trends by relating the current price and the average of price over n periods. In our experiments we set $n = 20$.	[28,8,4]

the daily values of the selected technical indicators for the period that goes from 2014-09-03 to 2015-02-02 for YPFD.

In this particular experiment, rather than forecasting the daily returns of a stock, we follow the traditional machine learning approach of predicting its *direction* — i.e., $rc_t = 1$ if $r_t \geq 0$, and $rc_t = 0$ otherwise. By doing this, the task is reduced to a binary classification. We follow this approach to better replicate models presented in [8], which we used to guide our selection of technical indicators. Additionally, according to [18], predicting stock returns direction is by far the most common approach used in text mining for market prediction.

In this context, when we say that a model is trained using only technical indicators, we mean that it is trained to predict the values of rc_t using a matrix containing the evolution of the ten selected technical indicators as input (we call this matrix $TI \in \mathbb{R}^{n_1 \times 10}$). Similarly, when we say that a model is trained using only message board data, we mean that it is trained using as input a matrix $U_h \Sigma_h$. Finally, when we say that a model is trained on both technical indicators and message board data, we mean that it is trained using as input the augmented matrix obtained by concatenating the columns of TI and $U_h \Sigma_h$ (we call this matrix $TM \in \mathbb{R}^{n_1 \times (10+h)}$).

We use **random forest** as our learning model [25]. Random forest is a classifier well-known for its good performance, its ability to detect nonlinear patterns in the data, its low number of hyperparameters, and its robustness to bad hyperparameter setups. Before training each model, missing values in a feature are replaced by the observed median calculated on training data (missing values arise for a subset of technical indicators over particular time periods).

In this case, we evaluate the performance of our models using two standard measures, the **accuracy**[12] and the **area under the ROC curve** (AUC).[13] We search for the best hyperparameters using the experimental design described in Section 3.3. For each stock, we choose the setup associated to the model with highest accuracy in the validation phase. In this experiment we only modify the number of candidate variables in each node split of random forest (va), doing the search on $va \in \{15, 12, 9, 6, 3\}$. We leave the number of base learners constant at 1000.[14] Finally, given that random forest is regarded as a good algorithm for feature selection, when training this model adding the online message board data we set the number of topics equal to 50, leaving the algorithm to choose which topics, if any, to use.

4. Results

This section summarizes the results drawn from our analysis. Section 4.1 presents the results for question Q_1 using the approach described in Section 3.4. Subsequently, Section 4.2 presents the results for Q_2 with the methods described in Section 3.5.

4.1. Results obtained using only online message board data

Table 4 shows the results obtained for the experiment aimed at answering Q_1. For each stock, it presents the selected performance metrics (RMSE and ρ) obtained in the validation and testing sets by the model trained using the online message board (Message Board Data) and by the baseline models (Training Average and Lagged Return). For each ρ coefficient we also include the corresponding p-value indicating its statistical significance. Although in practice results on testing data are the relevant ones, results obtained in the validation set are also presented in order to better understand the system's behavior.

Table 4 shows that the Message Board Data model achieved a lower RMSE than both baseline models. Paired one-tailed Wilcoxon rank-sum tests reject the null-hypothesis that the RMSE of the Message Board Data model is equal to or greater than those of each of the other models; this happens both in validation ($W = 0$, $p < 0.0039$ for Training Average model; $W = 0$, $p < 0.0039$ for Lagged Return)

[12] Accuracy is equal to the ratio obtained by dividing the number of correct predictions by the number predictions. Higher values indicate better predictive performance.

[13] The AUC is equal to the probability that a classifier will rank a randomly chosen positive instance (in our case $rc_t = 1$) higher than a randomly chosen negative one (in our case $rc_t = 0$). It is calculated as the integral of the receiver operating characteristic (ROC) curve. The ROC curve plots the true positive rate (TPR) of a classifier against its false positive rate (FPR) at various threshold settings (see [26]). Higher values indicate better predictive performance.

[14] Note that, as stated in [25], increasing the number of base learners does not cause the random forest sequence to overfit.

Table 4
Results obtained by models trained on topics derived from the online message board (Message Board Data) and by two baseline models (Training Average and Lagged Return).

Stock	Forecasting system								
	Training Average			Lagged Return			Message Board Data		
	RMSE	ρ	p-Value	RMSE	ρ	p-Value	RMSE	ρ	p-Value
Validation									
APBR	20.417	−0.109	0.236	29.879	0.238	0.009	19.602	0.173	0.059
COME	11.047	−0.241	0.008	11.228	0.482	0.000	9.642	0.331	0.000
EDN	10.366	−0.110	0.233	13.766	0.329	0.000	10.074	0.176	0.055
ERAR	8.952	−0.232	0.011	9.604	0.455	0.000	8.073	0.290	0.001
GGAL	6.456	−0.159	0.082	7.984	0.390	0.000	6.271	0.186	0.042
PAMP	8.633	−0.215	0.018	11.842	0.309	0.001	8.434	0.141	0.125
TS	5.030	−0.127	0.165	5.552	0.420	0.000	4.752	0.313	0.001
YPFD	8.736	−0.098	0.286	10.453	0.397	0.000	8.148	0.258	0.004
Testing									
APBR	11.695	−0.136	0.140	18.023	0.254	0.005	11.226	0.196	0.032
COME	6.817	−0.245	0.007	8.517	0.365	0.000	6.076	0.318	0.000
EDN	6.178	−0.261	0.004	7.683	0.374	0.000	5.798	0.293	0.001
ERAR	3.362	−0.038	0.679	3.452	0.485	0.000	3.224	0.224	0.014
GGAL	4.644	−0.150	0.102	6.157	0.340	0.000	4.480	0.212	0.020
PAMP	5.899	−0.221	0.015	6.544	0.442	0.000	5.340	0.311	0.001
TS	2.880	−0.100	0.277	5.210	0.133	0.146	2.836	0.110	0.232
YPFD	2.512	−0.186	0.042	2.666	0.472	0.000	2.392	0.231	0.011

and in testing ($W = 0$, $p < 0.0039$ for Training Average model; $W = 0$, $p < 0.0039$ for Lagged Return).

Regarding the correlation coefficients, paired and one-tailed Wilcoxon rank-sum tests reject the null-hypothesis that the ρ coefficient of the Message Board Data model is equal or smaller than the obtained ones by the Training Average model ($W = 36$, $p < 0.0039$ in validation and $W = 36$, $p < 0.0039$ in testing). A similar test indicates that the Message Board Data model fails in outperforming the Lagged Return one in terms of correlation coefficient.

Table 4 provides valuable information on the behavior and distribution of errors of the proposed models. First, since the mean value of r_t in our data is close to zero, the Training Average model tends to predict values of r_t also close to zero. This translates into relatively low RMSE values on the one hand, but this inflexibility also leads to low values of ρ on the other (given that the model does not react quickly enough to changes in the behavior of r_t). Second, taking into account that the behavior of r_t in our data shows a positive auto-correlation, the high values of ρ obtained by the Lagged Return model are expectable. This very fact also explains the high values of RMSE — when this model makes a wrong prediction (perhaps because the sign of r_t changed from one day to the other), it misses by much, thus leading to a large increase in the squared error. In contrast, the Message Board Data model manages to have lower RMSE values than the other two, while its predictions still maintain a considerable correlation with the predicted values.

4.1.1. Semantic content of the predictive topics

Given that Table 4 places the Message Board Data model results in a good position relative to the ones obtained by the baseline models, further analysis of what may be driving the Message Board Data model predictions seems relevant. Here we exploit the fact that combining LSA with ridge regression allows us to identify the structure of those topics detected as predictive by the learning models. We do this by taking into account two properties of these techniques. First, as mentioned in Section 3.2, the elements in matrix V_h indicate the degree of association between individual words and topics. And second, the coefficients estimated by ridge regression indicate the level of influence of specific topics in the model predictions. Combining these two ideas, it is possible to inspect the structure of the predictive topics by seeing which values of β_i are high (in absolute value) and then inspect the structure of the columns of matrix V_h associated to those β_i, thus identifying which tokens are activated or deactivated in these topics.

Taking this into account, in Fig. 8 we present **word clouds** for four selected stocks, following the figures introduced in [2]. Each word cloud shows the 30 tokens with the highest influence on the topic associated to the β_i with the greatest absolute value.[15] Tokens in blue indicate that the presence of the token *activates* the topic values (i.e., $V_{h_{j,i}} > 0$ for token j and topic i); tokens in red indicate that the presence of the token *deactivates* the topic values ($V_{h_{j,i}} < 0$); and the size of words corresponds to the absolute value of $V_{h_{j,i}}$.

Fig. 8 is quite illustrative. A first pattern that emerges is that emojis are commonly detected as important features, with the signs one would expect: e.g., "*emoji_Arriba*" (a green arrow pointing up) and "*emoji_Abajo*" (a red arrow pointing down). Another interesting observation is that topics detected as predictive tend to contain tokens related to political or economical aspects of the company associated to the stock. The case of APBR is an example of a topic reflecting political aspects, as the presence of the tokens "*dilma*" (first name of the elected president in Brazil's 2014 presidential elections) and "*aecio*" (first name of its main contender) impact negatively in the activation of the topic. The cases of EDN and YPF are examples of topics reflecting economical aspects. EDN presents tokens such as "*tarifas*" (which refers to electricity rates), "*subsidios*" (which commonly refers to electricity subsidies) and "*gobierno*" (which commonly refers to the national government). YPFD has tokens such as "*produccion*" ("production") and "*chevron*" (a company with which YPF signed a partnership to exploit a large tight oil and shale gas deposit). Finally the case of GGAL stands apart from the rest; it contains tokens such as "*volumen*" ("volume") and "*resistencia*" ("resistance"), both technical financial terms. This suggests that for the case of GGAL, our system manages to detect and give importance to technical discussions in which users share their analysis of the stock behavior.

[15] To generate this figure, we trained our systems on the full training and validation sets, setting all hyperparameters to the values used for obtaining the results shown in Table 4.

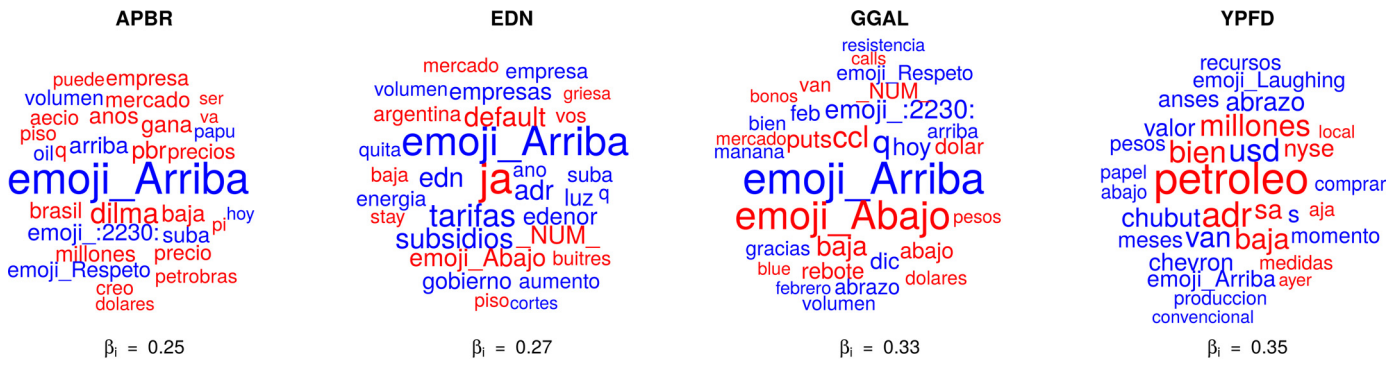

Fig. 8. Influential tokens in the topics detected as predictive.

Table 5
Results obtained when stock returns direction is predicted using models trained on technical indicators and/or topics derived from the online message board.

Stock	Prediction system							
	Majority Class		Topics		TI		TI + Topics	
	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC
Validation								
APBR	0.492	0.506	0.508	0.503	0.592	0.619	0.625	0.623
COME	0.433	0.500	0.608	0.620	0.708	0.746	0.758	0.778
EDN	0.517	0.500	0.567	0.553	0.700	0.710	0.633	0.653
ERAR	0.583	0.500	0.533	0.553	0.633	0.652	0.700	0.714
GGAL	0.558	0.500	0.567	0.540	0.625	0.638	0.683	0.709
PAMP	0.633	0.500	0.517	0.523	0.608	0.638	0.667	0.690
TS	0.525	0.500	0.608	0.636	0.600	0.634	0.692	0.725
YPFD	0.525	0.500	0.525	0.603	0.658	0.720	0.742	0.793
Testing								
APBR	0.433	0.432	0.533	0.529	0.592	0.631	0.625	0.629
COME	0.467	0.500	0.550	0.500	0.625	0.705	0.658	0.722
EDN	0.417	0.394	0.508	0.511	0.567	0.604	0.600	0.653
ERAR	0.483	0.500	0.500	0.549	0.783	0.816	0.750	0.802
GGAL	0.517	0.500	0.525	0.563	0.642	0.691	0.683	0.749
PAMP	0.508	0.500	0.467	0.467	0.650	0.739	0.717	0.749
TS	0.467	0.500	0.600	0.599	0.500	0.527	0.517	0.538
YPFD	0.517	0.500	0.600	0.677	0.625	0.691	0.733	0.751

4.2. Results obtained combining online message board data and technical indicators

Table 5 summarizes the results aimed at answering Q_2. It presents the selected performance metrics (accuracy and AUC) obtained both in the validation and testing sets for models trained using only technical indicators (TI), only message board data (Topics), and both technical indicators and message board data (TI+Topics). Additionally, we include the performance of a baseline model that predicts for every observation the majority class observed in the training data (Majority Class).[16]

From Table 5 it can be seen that both in terms of accuracy and AUC the Topics model outperforms the Majority Class one, which is consistent with the results presented in Section 4.2 and serves as further evidence for answering Q_1 affirmatively. This table also shows that the TI model outperforms the Topics one, suggesting that, even though models which only use data from the message board posts are able to make reasonable predictions, simple technical indicators constructed on the past behavior of prices seems to be more informative than the message board data by its own.

Although these two last results are interesting by themselves, the key result from Table 5 comes from comparing the performance of the TI+Topics models relative to the TI ones, as this result is linked directly to answering Q_2. This comparison suggests that models that enrich traditional technical indicators with data coming from online message board present a relative boost in performance. Specifically, paired one-tailed Wilcoxon rank-sum tests reject the null hypothesis that the TI+Topics model has an equal or worst performance than the TI one, both in term of accuracy ($W = 30.5$, $p < 0.0430$ in validation; $W = 32$, $p < 0.0273$ in testing) and AUC ($W = 32$, $p < 0.0273$ in validation; $W = 31$, $p < 0.0391$ in testing).

Lastly, to put in context these results, [18] show that systems which rely on online text data to make their predictions achieve accuracy levels between 50% and 70%. Even though we work on different data and use different techniques than the articles reviewed in [18], it should be remarked that the models we propose achieve highly competitive levels of performance.

4.3. Analysis of the predictive importance of technical indicators and message board features

Table 5 suggests that the message board data effectively contributes additional predictive power to that of technical indicators. To better understand this, Fig. 9 plots the importance of individual features, as estimated by the TI+Topics model for each stock. In Random Forests, feature importance may be estimated as the total decrease in node impurity (weighted by the probability of

[16] Note that predictions of these systems may vary as the training data window grows, as explained in Section 3.3.

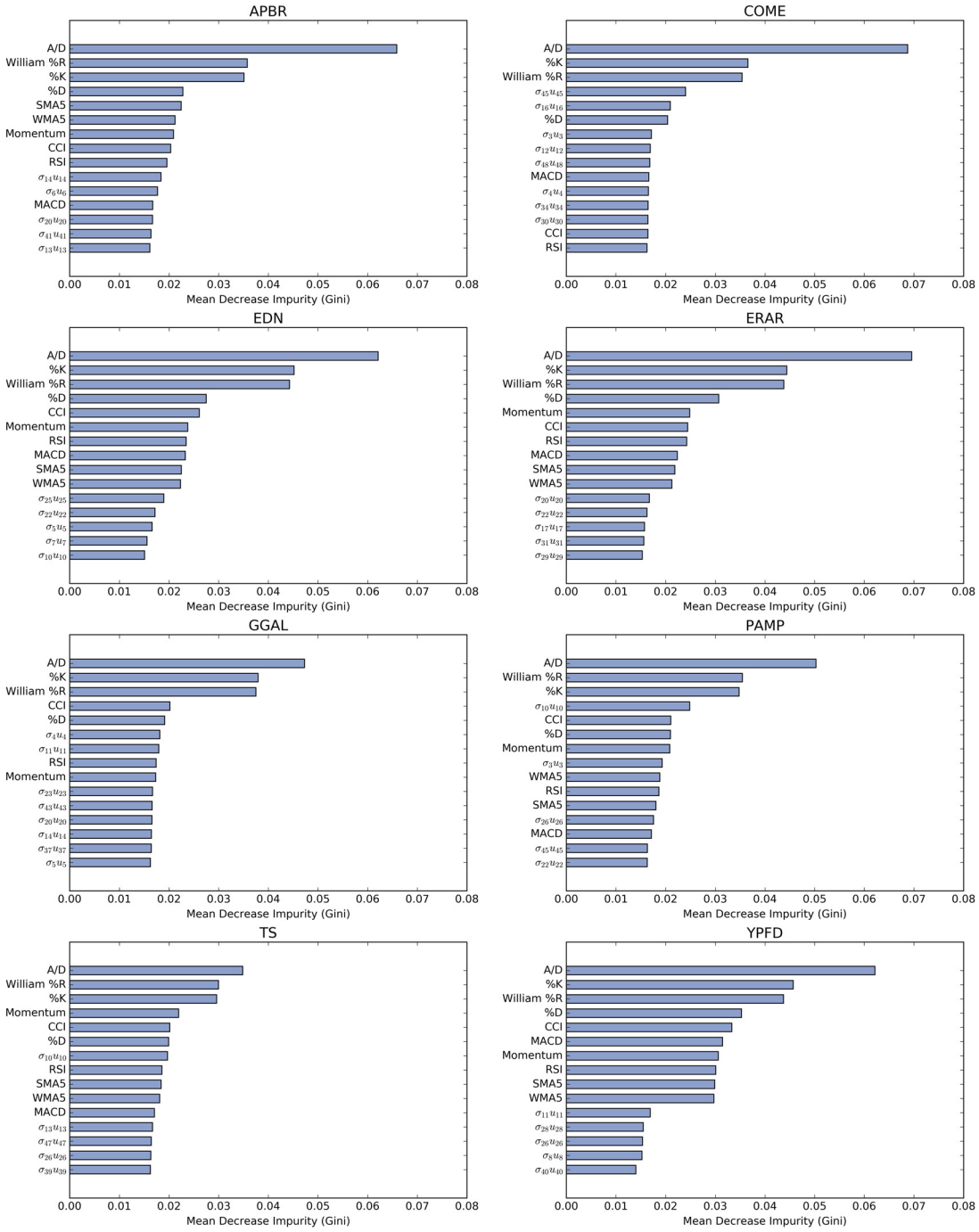

Fig. 9. *Mean Decrease in Impurity* (MDI) of the 15 most predictive features for each stock. Technical indicators are listed according to the abbreviations of Table 3, and message board features are listed as $\sigma_h u_h$.

reaching that node) averaged over all trees in the ensemble [25, see][]. This measure is known as the *Mean Decrease in Impurity* (MDI); the higher the MDI of a feature, the more predictive it is believed to be. Fig. 9 shows the MDI of the 15 features detected as most predictive for each stock. Technical indicators are listed using the abbreviations introduced in Table 3, and message board features are listed as $\sigma_h u_h$, as described above.[17]

A first pattern that stands out from Fig. 9 is that across all stocks the three most predictive features are always technical indicators. Notably, in all stocks the three most predictive features are the same (A/D, William %R and %K). This goes in hand with Table 5, which shows that the TI model outperforms the Topics model. Nonetheless, when observing the estimated predictive importance of the remaining features, technical indicators do *not* consistently outperform message board features. In fact, for five out of eight stocks, at least one message board derived feature outperforms one or more technical indicators. Moreover, for two stocks (COME and PAMP) the fourth most predictive feature was derived from message boards.

[17] To generate this figure, we trained our systems on the full training and validation sets, setting the values of *va* to the ones used for obtaining the results shown in Table 5.

Overall, Fig. 9 suggests that, when training classifiers on technical indicators and message board derived topics, most of the predictions are driven by a small subset of highly predictive technical indicators. Still, taking into account that TI+Topics model surpass the TI model in predictive performance, this figure also points toward the conclusion that message board derived features do *complement* these highly predictive technical indicators.

5. Conclusions

In this work we have explored whether available online data can be exploited to predict the future behavior of stock prices. We have addressed this by empirically studying the case of a popular Argentinian online message board and placed our focus on answering two questions. The first question consisted in determining whether data coming from online message boards can be mined in order to predict the future daily return of a series of stocks. The second question — not extensively studied in the literature, was whether the mined data complements information already available in the past behavior of prices or, alternatively, it is just another source of the same information. To address these questions, we built and validated a series of predictive models using state-of-the-art machine learning techniques. Each model was trained with different combinations of features extracted from the past behavior of stock prices, or mined from the online message boards.

Regarding the first question, we find that the systems trained only on message board data perform at least as well as two baseline models trained only on data from the past behavior of stock prices. This result goes in hand with previous studies which find evidence suggesting that online text data can be mined in order to train models which effectively predict stock future behavior better than random guessing. More importantly, for our second question we find that the addition of message board data improves the performance of traditional forecast systems trained on the past evolution of prices. This boost in performance suggests that the features mined from online message boards manage to capture relevant information apparently not contained in the technical indicators.

Further inspection of the structure of highly predictive features derived from the online message board data revealed that our trained models not only capture aspects related to the expectation of agents (such as emojis of arrows pointing upwards or downwards), but also that these topics usually contain tokens which can be readily related to the economical and political environment surrounding the analyzed companies.

Overall, having trained competitive stock returns prediction systems and analyzed the impact of adding features derived from message boards, our results further validate the value of novel sources of information in predicting events which might escape the online domain. Given these results, future research, not only in stock prediction systems, should focus on better understanding the links between online data and offline events.

References

[1] J. Ginsberg, M.H. Mohebbi, R.S. Patel, L. Brammer, M.S. Smolinski, L. Brilliant, Detecting influenza epidemics using search engine query data, Nature 457 (7232) (2009) 1012–1014, http://dx.doi.org/10.1038/nature07634.

[2] J.C. Eichstaedt, H.A. Schwartz, M.L. Kern, G. Park, D.R. Labarthe, R.M. Merchant, S. Jha, M. Agrawal, L.A. Dziurzynski, M. Sap, C. Weeg, E.E. Larson, L.H. Ungar, M.E.P. Seligman, Psychological language on twitter predicts county-level heart disease mortality, Psychol. Sci. 26 (2) (2015) 159–169, http://dx.doi.org/10.1177/0956797614557867.

[3] M.G. Yunusoglu, H. Selim, A fuzzy rule based expert system for stock evaluation and portfolio construction: an application to Istanbul stock exchange, in: 2nd International Fuzzy Systems Symposium, Ankara, Turkey, 17–18 November 2011, Expert Syst. Appl. 40 (3) (2013) 908–920, http://dx.doi.org/10.1016/j.eswa.2012.05.047.

[4] J. Patel, S. Shah, P. Thakkar, K. Kotecha, Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques, Expert Syst. Appl. 42 (1) (2015) 259–268, http://dx.doi.org/10.1016/j.eswa.2014.07.040.

[5] E. Guresen, G. Kayakutlu, T.U. Daim, Using artificial neural network models in stock market index prediction, Expert Syst. Appl. 38 (8) (2011) 10389–10397, http://dx.doi.org/10.1016/j.eswa.2011.02.068.

[6] M.A. Boyacioglu, D. Avci, An adaptive network-based fuzzy inference system (ANFIS) for the prediction of stock market return: the case of the Istanbul stock exchange, Expert Syst. Appl. 37 (12) (2010) 7908–7912, http://dx.doi.org/10.1016/j.eswa.2010.04.045.

[7] J. Yao, C.L. Tan, H.-L. Poh, Neural networks for technical analysis: a study on KLCI, Int. J. Theor. Appl. Finance 02 (02) (1999) 221–241, http://dx.doi.org/10.1142/S0219024999000145.

[8] Y. Kara, M.A. Boyacioglu, Ö.K. Baykan, Predicting direction of stock price index movement using artificial neural networks and support vector machines: the sample of the istanbul stock exchange, Expert Syst. Appl. 38 (5) (2011) 5311–5319, http://dx.doi.org/10.1016/j.eswa.2010.10.027.

[9] M. Nardo, M. Petracco-Giudici, M. Naltsidis, Walking down wall street with a tablet: a survey of stock market predictions using the web, J. Econ. Surv. 30 (2) (2016) 356–369, http://dx.doi.org/10.1111/joes.12102.

[10] J. Bollen, H. Mao, X. Zeng, Twitter mood predicts the stock market, J. Comput. Sci. 2 (1) (2011) 1–8, http://dx.doi.org/10.1016/j.jocs.2010.12.007.

[11] H. Lee, M. Surdeanu, B. MacCartney, D. Jurafsky, On the importance of text analysis for stock price prediction, in: Proceedings of LREC 2014, 2014, pp. 1170–1175 http://www.lrec-conf.org/proceedings/lrec2014/pdf/1065_Paper.pdf.

[12] Y. Shynkevich, T. McGinnity, S.A. Coleman, A. Belatreche, Forecasting movements of health-care stock prices based on different categories of news articles using multiple kernel learning, Decis. Support Syst. 85 (2016) 74–83, http://dx.doi.org/10.1016/j.dss.2016.03.001.

[13] R.P. Schumaker, H. Chen, Textual analysis of stock market prediction using breaking financial news: the AZFin text system, ACM Trans. Inf. Syst. 27 (2) (2009), http://dx.doi.org/10.1145/1462198.1462204, 12:1–12:19.

[14] T. Preis, H.S. Moat, H.E. Stanley, Quantifying trading behavior in financial markets using google trends, Sci. Rep. 3 (2013), http://dx.doi.org/10.1038/srep01684.

[15] H.S. Moat, C. Curme, A. Avakian, D.Y. Kenett, H.E. Stanley, T. Preis, Quantifying wikipedia usage patterns before stock market moves, Sci. Rep. 3 (2013), http://dx.doi.org/10.1038/srep01801.

[16] M. Hagenau, M. Liebmann, D. Neumann, Automated news reading: stock price prediction based on financial news using context-capturing features, Decis. Support Syst. 55 (3) (2013) 685–697, http://dx.doi.org/10.1016/j.dss.2013.02.006.

[17] T. Geva, J. Zahavi, Empirical evaluation of an automated intraday stock recommendation system incorporating both market data and textual news, Decis. Support Syst. 57 (2014) 212–223, http://dx.doi.org/10.1016/j.dss.2013.09.013.

[18] A.K. Nassirtoussi, S. Aghabozorgi, T.Y. Wah, D.C.L. Ngo, Text mining for market prediction: a systematic review, Expert Syst. Appl. 41 (16) (2014) 7653–7670, http://dx.doi.org/10.1016/j.eswa.2014.06.009.

[19] D. Jurafsky, J.H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 2nd edition, Prentice Hall PTR, Upper Saddle River, NJ, USA, 2008.

[20] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman, Indexing by latent semantic analysis, J. Am. Soc. Inf. Sci. 41 (6) (1990) 391–407, http://dx.doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9.

[21] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, R. Passonneau, Sentiment analysis of twitter data, in: Proceedings of the Workshop on Languages in Social Media, LSM'11, Association for Computational Linguistics, Stroudsburg, PA, USA, 2011, pp. 30–38 http://dl.acm.org/citation.cfm?id=2021109.2021114.

[22] S. Bird, E. Klein, E. Loper, Natural Language Processing with Python, O'Reilly Media, Inc., 2009.

[23] S.T. Dumais, Latent Semantic Analysis, Annu. Rev. Inf. Sci. Technol. 38 (1) (2004) 188–230, http://dx.doi.org/10.1002/aris.1440380105.

[24] L. Torgo, Data Mining with R, Learning with Case Studies, Chapman and Hall/CRC, 2010.

[25] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer Series in Statistics, 2nd edition, Springer New York Inc., New York, NY, USA, 2001.

[26] E. Alpaydin, Introduction to Machine Learning, 2nd edition, MIT Press, 2009.

[27] M.P. Clements, D.F. Hendry, Forecasting Non-stationary Economic Time Series, MIT Press, 2001.

[28] J. Ulrich, TTR: Technical Trading Rules, R Package Version 0.23-0, 2015 http://CRAN.R-project.org/package=TTR.

[29] R. Rosillo, D. de la Fuente, J.A.L. Brugos, Technical analysis and the Spanish stock exchange: testing the RSI, MACD, momentum and stochastic rules using Spanish market companies, Appl. Econ. 45 (12) (2013) 1541–1550, http://dx.doi.org/10.1080/00036846.2011.631894.

[30] G. Armano, M. Marchesi, A. Murru, A hybrid genetic-neural architecture for stock indexes forecasting, Inf. Sci. 170 (1) (2005) 3–33, http://dx.doi.org/10.

1016/j.ins.2003.03.023, Computational Intelligence in Economics and Finance.

Ramiro H. Gálvez is a Ph.D. Student in Computer Science in the Computer Science Department at the University of Buenos Aires. He received a M.Sc. in Data Mining & Knowledge Discovery from the University of Buenos Aires in 2016, a M.Sc. in Development Economics from the Carlos III University of Madrid in 2008 and his *Licenciatura* from the National University of Córdoba (Argentina) in 2007. He is also member of the Speech Processing Group and the Applied Artificial Intelligence Lab at the CS Department at the University of Buenos Aires. His research interests include predictive analytics, machine learning and pattern recognition.

Agustín Gravano received his Ph.D. in Computer Science from Columbia University in 2009 and his *Licenciatura* in Computer Science at the CS Department at the University of Buenos Aires in 2001. He has been a Professor in the Computer Science Department at the University of Buenos Aires since 2011, and Researcher at CONICET (National Research Council) since 2010. He is also member of the Speech Processing Group and the Applied Artificial Intelligence Lab. His main research topic consists in understanding and modeling the extraordinary degree of coordination exhibited by human beings while holding a conversation, both at a temporal level and along other dimensions of speech. The ultimate goal is to include this knowledge into spoken dialogue systems, aiming at improving their naturalness.