2013-01-06

Stock Market Prediction Without Sentiment Analysis: Using a Web-Traffic Based Classifier and User-Level Analysis

Pierpaolo Dondio
Technological University Dublin, pierpaolo.dondio@tudublin.ie

Recommended Citation

Dondio, P. (2013). Stock Market Prediction without Sentiment Analysis: Using a Web-traffic based Classifier and User-level Analysis. *46th Hawaii International Conference on System Sciences*, 7-10 Jan. 2013, Wailea, HI, USA. doi:10.1109/HICSS.2013.498

2013 46th Hawaii International Conference on System Sciences

Stock Market Prediction without Sentiment Analysis: Using a Web-traffic based Classifier and User-level Analysis

Pierpaolo Dondio
Dublin Institute of Technology & Edge Labs Limited
pierpaolo.dondio@dit.ie

Abstract

This paper provides further evidence on the predictive power of online community traffic with regard to stock prices. Using the largest dataset to date, spanning 8 years and almost the complete set of SP500 stocks, we train a classifier using a set of features entirely extracted from web-traffic data of financial online communities. The classifier is shown to outperform the predictive power of a baseline classifier solely based on price time-series, and to have similar performances as the classifier built considering price and traffic features together. The best predictive performances are achieved when information about stock capitalization is coupled with long-term and mid-term web traffic levels. In the second part of the paper we show how there exists a group of users whose traffic patterns constantly outperform the other users in predictive capacity. The findings set interesting future works in the definition of novel market indicators for market analysis.

1. Introduction

Since their inception, online communities about finance have received a growing attention as a valid source of market analysis, and they have gradually gained credibility.

Despite this clear trend, evidence regarding the predictive value of financial social media is not definitive. In one of the earliest papers by Antwellier [1] the author concludes how the impact of the message board is statistically but not economically significant, while more recent results speak about accuracy in the range of 70-80%.

This paper contributes to the debate about whether online communities have predictive market ability.

This work follows our previous work [2], further advancing the application of data mining techniques to raw traffic analysis and introducing user-level analysis. We propose an evaluation using the more extensive experimental data to date, in terms of time span - 8 years - and number of stocks - about 478.

We identified 3 major techniques and 3 levels of analyzing social media content for market predictions.

The first source is the unstructured stream of web-traffic produced by the community. In its essential model, it is a stream of messages (post, twits) tagged with three dimensions: user, time, stock associated.

The second source of information is represented by text-based features, typically an indicator of the sentiment expressed. The previous literature is dominated by such approach. Market prediction models are based on a sentiment index that gives the daily raw traffic a positive/negative direction. Nevertheless, text features are not limited to sentiment. Bollen [9] experiments with 7 text-based features, encompassing things such as *calm*.

Third, other features come from behavioural/social information rather than text, such as reputation of the individual in the community, his profile, friends, the way he interacts with other members of communities – such as analysis of quotes, discussions opened - and so forth. Given these 3 sources of features, it is possible to aggregate them at user-level (where each user is considered to have a different impact on the overall index), at community-level (where indexes are generated by considering all users the same) and at multi-community level (where analyses are aggregation of individual community indexes and predictions).

As shown in Table 1, this study concerns the investigation of web traffic quantitative data, at community and user level, a more unexplored and complementary research to usual text-based analysis. We first pose the following research questions:

1. *Can patterns of raw traffic predict market? If so, under which conditions?*

2. *Are there better users than others? - i.e. Are there users that constantly outperform or underperform their peers?*

1530-1605/12 $26.00 © 2012 IEEE
DOI 10.1109/HICSS.2013.498

Table 1. Methods of analysis. The scope of this paper is delimited by the black thick border.

	User-level (each user is treated differently)	**Community-Level** (indicators for each communities)	**Multi-Community Level** (Aggregation of many community indicators)
Unstructured Web Traffic (stream of users' messages about stocks)		Some features in Antweiler	Never Performed
Text-based Featured (Sentiment, mood, topic, tags)	Gu, B. Cook, Bollen	Classical Sentiment Analysis applications	Never Performed
Behavioural and Social Information (Reputation, popularity, users' profile, acquaintances, users' past performance..)	Vivek Sehgal, U. Spiegel Munmun De Choudhury		Never Performed

The answer to the first question seems an obvious no. Unqualified traffic is too noisy and, more importantly, it has no direction in terms of the positive/negative sentiment. How can we predict something we do not understand?

Apart from the fact that the question has never been fully answered, and a study such as ours should start from a baseline indicator, there are more interesting considerations that justify the question as a valid research question. In the paper we dedicate a section to the analysis of some hypotheses. We stress how, even if there is evidence that high traffic could approximate positive sentiment, this study does not require the hypothesis to be true; here we study whether patterns of traffic have some predictive behaviour rather than if traffic can approximate users' sentiment.

The second question calls for a user-level analysis. We wonder if there are users whose patterns of traffic help to increase the predictive capacity. Even if the general traffic could have little or no predictive capacity, we wonder if the hypothesis is satisfied for a subset of users that seem to outperform/underperform the others with predictable regularity. The hypothesis of the existence of such set is valid. Market efficiency might still be valid for the whole community of traders, but not in specific subsets of it. The user-level analysis is again performed using raw traffic data and market prices.

Therefore the contributions of our paper are the effort to produce an answer to the 2 questions above. In doing so, we also contribute with the largest dataset, filling a gap in previous experimentations where either the time span or the stock set was extremely small.

Finally, we note how the dataset underlying this paper proposes a minimal set of features that can be applied to a large set of information-sharing applications, such as message boards, online *fora* and twitter-like applications, making its result present. We do not consider twitter in our study because our study aims to have a dataset spanning from 8 years of data, making twitter too recent. However, the techniques used in this study can be applied to twitter-like applications without modifications.

The paper is organized as follows: in section 2 we discuss why the hypothesis of raw traffic could be reasonable, in section 3 we describe how we defined our classifiers, that are evaluated in section 4; in section 5 we describe our user-level analysis. Section 6 presents an extensive review of related works to date before ending with our conclusions.

2. Using Users Traffic as Predictor

In this section we discuss a few reasons why it is worthy to investigate the predictive power of raw quantitative traffic, as we performed in [2]. We also describe why it is interesting to consider individual users' traffic. The main idea is that users distribute their activity with a purpose and traffic could act as a proxy, an approximation or even a substitute for users' sentiment. Recent works seem to back the validity of the hypothesis. We stress how our analysis does not rely or is based on these considerations, even if they are indeed useful to better interpret the results obtained. Our research question is not whatever raw traffic approximates positive sentiment, but rather if levels of raw traffic predict market movements.

2.1 Direct evidence collected via surveys

We conducted a survey on the website FinanzaOnline.it [4] - the largest Italian online community with about 120 000 registered users and 15 million posts. Our aim was to better understand the relationship between users' raw activity in the community and stocks. We asked the following:

Q1: If you write on a stock board, do you hold the stock? If not, why are you writing there?

Q2: Do you still write about stocks you have sold?

We collected about 350 answers. The results show how 78.7% of users replied yes to the first question, adding as most frequent comment that, if they are writing on a stock they do not hold, the majority of time it is because they are considering buying it.

Users also replied how the activity fades after the stock is sold. The large majority of users – about 90% - has a *long* position (i.e. betting on the stock to raise its price), the sentiment results strongly positively biased and the expressions of negative sentiment are usually limited in number and duration.

The results of the survey allow us to believe that users on the online community behave with some regularity – a necessary condition for making predictions. We could hypothesize that, on average:

1. a user writes about stocks (1) he is interested in, (2) he is keen to buy in the next future, or (3) he holds.
2. The large majority of users writing about a stock has a *long* position on that stock.
3. Users tend to distribute their finite effort purposely. They do not spend time on stocks they are not interested for a stable amount of time.
4. Users' activity gradually fades once the stock is sold and new stocks gain activity.

All the above is valid for raw traffic data, without analyzing text and sentiment of users' contributions but only considering *when*, *where* and *how much* users contributed. Our key question is therefore the following: is this kind of *association* between users' raw activity and stocks enough to make market predictions? Is it enough to identify specific group of users?

2.2 Absence of sentiment

Another reason to consider raw traffic data is that the large majority of messages are out-of-topic, containing no sentiment at all. It is common that users never or rarely publicly express their sentiment. However, it is a reasonable hypothesis that the presence of such users' messages about a specific stock at a specific time and market condition is not random.

2.3 Positive bias and technical reasons

There is evidence over a strong positively-biased sentiment populating financial on-line communities (Zhang et al. [8]), confirmed by our survey as well. This allows us to presume that traffic could be a proxy for at least positive sentiment. Messages on average are strongly over-bullish. This suggest that the predictive value of web-traffic, if any, could result asymmetric, i.e. effective in one direction only, either buy or sell.

Partially, the above observations find a confirmation in the work by Bollen [3]. Bollen reports that it is not the positive/negative sentiment that predicts the market, but actually one particular mood extracted by the text that he calls "calm". A reasonable hypothesis is that calm is a concept that can be also effectively identified by patterns of traffic as well. The work by [5] provides further evidence about making good predictions without sentiment. Using a limited dataset of 4 stocks, the author concludes how market movements can be predicted with an 80% accuracy by relying on non-textual blogs dynamics such as increase in blog comments, average response time, quotations, length of comments.

3. Building a Traffic-based Rule Classifier

In this section we describe a classifier for predicting mid-term stock price movements based on web traffic features and historical price series. The aim of the experimentation is three-fold. First, we aim to provide positive evidence on the predictive ability of web-traffic; second, we show that our classifier, based only on web-traffic features (referred as the traffic classifier), outperforms in predictive power a classifier solely based on historical prices (referred as the price classifier). Third, we test if a classifier containing both price and traffic-related features (referred as the complete classifier) exhibits higher performance than the other two classifiers.

We perform our classification tasks using rules extracted from a *J48* decision tree. We remind how a decision tree can be converted into rules, one for each path from the root to each leaf of the tree. The size of the leaf represents the support (or coverage) of the rule - i.e. the number of occurrences of the rules in the dataset, in our context equal to the number of trading days in which the rule is applicable - while the number of objects positively classified divided by the size of the leaf represents the accuracy of the rule.

Using the rules extracted from the decision tree, we study the quality of the predictions varying the level of accuracy and support that a rule must have to be included into the classifier. By varying these two parameters, the set of rules of each classifier decreases and consequently the number of classifiable objects. Diminishing the classifiable set does not represent a serious problem in the context of our task. In a real trading strategy *precision* is usually more *important* than recall (or at least there is a reasonable case why it should be). A trading strategy is not required to provide a prediction at every interval, but that - given that the number of prediction is above a certain required number - the predictions be highly accurate. This is also justified by the existence of commissions at every transaction.

3.1 Dataset

Our dataset is composed by a stream M of meta-data about messages posted on *Yahoo! Finance*. M is a sequence of tuples (u, s, t) associated to each message, where $u \in U$ is the user author of the message, $s \in S$ is the stock the message refers to, t is the time of message creation. We collected about 26 millions tuples from Yahoo! Finance, spanning 8 years and 478 out of 500 stocks of the US SP500 index. The stream M identifies a 3-dimensional space with dimensions stocks (S), users (U) and time (T). The time dimension T is discretized by choosing an interval of time ΔT. In our simulation ΔT is always equal to one day, meaning that we do not study intraday trading.

Distinct to the stream M is a function $\mathrm{P(s,t)}: \mathrm{S} \times \mathrm{T} \rightarrow \mathbb{R}^{+}$ that associates the stock closing price to each stock and day. We use the closing price adjusted for dividends and share splits, using Bloomberg as a source.

By partitioning the stream M we can isolate data regarding a single stock or user in a particular interval of time. For the remaining of this work we need to define the following time series:

$\mathrm{N_{u,s}(t)}$ = n. of messages of user u on stock s at day t

$\mathrm{N_s(t)}$ = n. of messages by all users on stock s at day t

$\mathrm{N_u(t)}$ = n. of messages by user u at day t (on any stock)

We also define $\mathrm{Z_{s,d}}$, that is $N_s(t)$ normalized with a standard score obtained using an average and a standard deviation computed over a time-window of d days before. We call d the memory size. Therefore:

$$\mathrm{Z_{s,d}(t)} = \frac{\mathrm{N_s(t)} - \mu_{\mathrm{N_s(t-d,t)}}}{\sigma_{\mathrm{N_s(t-d,t)}}} \qquad (1)$$

3.2 Preprocessing

Starting from the stream M, we generate a total of 552,016 records, each of them representing daily data for a specific stock. For each stock and day, we only used the following data: the number of messages on the stock that day and the closing price. The dataset contains 478 different stocks.

3.2.1 Labeling Classes. We seek to predict the mid-term trend of the stock price. Rather than predicting the daily return of the following day, we predict if the stock price will rise or fall by a fixed percentage g.

For each stock s we marked each trading day t as positive or negative according to which of the following events happened first: (1) the stock price raises more than a fixed percentage g or (2) the stock prices falls further than g. Therefore each trading day is labeled with a binary value representing whether the upper target price was reached or not. We performed experiments with a 10% fixed symmetric target price. Over the entire dataset, 53.56% of trading days were labeled positive (price rose), and 46.44% negative.

3.2.2 Training Set Splitting. The dataset was split into training and test set as follows: test contains all the data of the most recent year (from May 2011 to May 2012), while all the rest is training test.

Due to the fact that the *Yahoo! Finance message board* allows to access only a limited fixed amount of historical messages per stock, the most frequent stocks have less time span than the others. Therefore we requested a stock to have at least a full year history in the training dataset in order to be used in the classification. We wanted to avoid the situation in which few stocks skew the distribution and alter the testing dataset, since these stocks exhibit very high level of traffic and, due to their limited available history, they exhibits usually higher performance than the average would. The dataset results composed by 511,057 records, and 409 stocks. The training set therefore covers few market cycles: stable bullish (up to 2007), crisis (2008-2009), a violent rally followed by period of high volatility (2010-2011).

3.2.3 Features. Our features are classified into three macro-areas: price data, company data and web-traffic data.

Regarding company data features, we introduced the capitalization of a stock in May 2011.

Regarding price-related features, we consider the return of the stock at different points in time: current day, previous day, previous week (5-day price) and previous month (20-day). Therefore we have the 4 features r_1, r_2, r_5, r_{20}.

Traffic features are derived from the time series of messages for each stock.

The features are divided in raw and z-score data (computed as shown in formula 1), since we make the hypothesis that both of them could contain interesting information. Raw data are taken directly from Ns, and they express absolute levels of traffic (as measured by number of messages), while z-score expresses normalized and re-scaled levels.

We define the following features: rw is the number of messages on a stock that day, i.e. $\mathrm{N_s(t)}$, $\mathrm{rw_0}$ is the value of the previous day, while $\mathrm{rw_{20}}$, $\mathrm{rw_\infty}$ are the value of the 1-month and long-term (since the first message registered) moving average of $N_s(t)$ up to the current day.

We also consider the following z-score features: $\mathrm{z_0}$ (z-score for the current day), $\mathrm{z_2}$ (previous day), $\mathrm{z_5}$ (previous week), $\mathrm{z_{20}}$ (last month) and $\mathrm{z_\infty}$ (long-term since the first available message for that stock). The dataset does not contain null data. Only valid trading days (for which a closing price exists for the stock) are used.

Table 2 – Features for each stock

1	Cap	Stock Capitalization
2-5	r_1, r_2, r_5, r_{20}	Return of the current day, previous day, previous week and previous month
6-9	$rw_0, rw_1, rw_{20}, rw_\infty$	Raw Traffic $N_s(t)$, i.e. the number of daily messages for the current day, previous day, 1-month moving average and long-term moving average
10-14	$z_0, z_1, z_5, z_{20}, z_\infty$	z-score of $\mathrm{N_s(t)}$ for the current day, previous day, previous week, previous month

3.2.4 Discretization. Since we use a J48 decision-tree algorithm, we need to discretize our features. Our discretization is unsupervised, equal frequency binning.

We discretize the raw traffic level by separating some special classes. We create a class for zero messages while the rest of data were discretized with equal frequency bins. The stock capitalization feature was discretized in 5 bins

representing *small* (S), *medium/small* (MS), *medium* (M), *medium*/*big* (MB) and *big* (B) caps.

The 409 stocks filtered for the experiment result divided in each capitalization bin as showed in table 3.

Table 3 – Stocks by Capitalization

Cap	Number of Stocks	% of Positive Cases
1	44	0.625
2	89	0.547
3	94	0.552
4	77	0.554
5	94	0.469

Note how the dataset contains less big stocks since they are the ones more excluded by our minimum requirement (a full year of training data).

3.3 Experimental Results

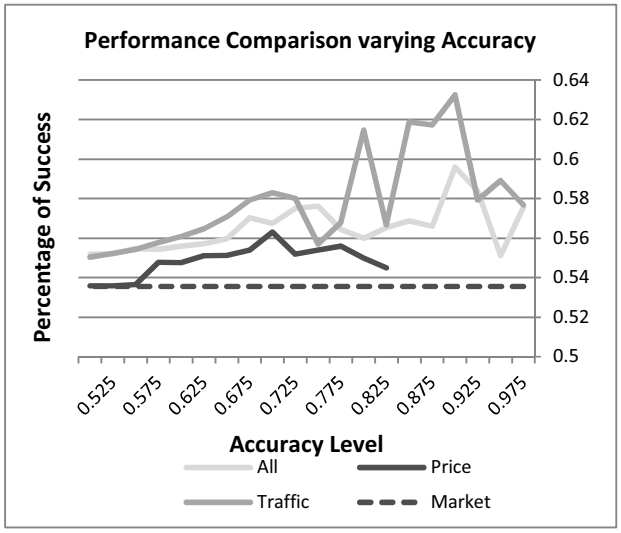
We report experimentations done with a C.35 decision tree, implemented using *Weka* J48 algorithm. For each classifier (*price, traffic* and *complete*) we trained a set of decision trees with various confidence factors and minimum number of objects per nodes. Since the results obtained by the various trees follow similar patterns, we report data for an aggregation of 50 models corresponding to the following parameters: a pruned tree with minimum number of nodes from 5 to 50 (at an interval of 5 nodes) and a confidence factor ranging from 0.05 to 0.25 (with 0.05 interval), used to decide the further splitting of leaves.

Regarding our evaluation criteria, we focus on *precision* rather than *recall*, however looking for a reasonable high number of cases to support a feasible trading strategy.

After growing our tree, we extracted the associated rules, each of them with support R_s and accuracyR_a. When we apply our rules over the test set, we can study the performance of the predictions varying the minimum level of accuracy and support that a rule must have to be used. As a consequence of increasing the minimum $R_a$and R_s, the number of classifiable cases decreases. Therefore we discard experimentation settings in which the total number of classifiable cases becomes statistically too small (i.e. 95% confidence level more than 1% size).

3.3.1 Overall Performance. Graph 1and 2 show the results of the three classifiers in predicting a price increase (Graph 1) and decrease (Graph 2), varying the level of accuracy required. The horizontal line represents the market benchmark, i.e. the proportion of respectively positive/negative cases in all the dataset (equal to the accuracy of a *zero rule* model always suggesting to buy/sell the stock).

Graph 1 shows how all the classifiers are slightly (not significantly) above the market benchmark when we do not set any threshold over the accuracy. Anyway, when the required accuracy increases, the three classifiers show large regions where they diverge significantly from the market benchmark.



Graph 1. Performance predicting a price increase

The *traffic* classifier outperforms the other two: it is always above the *market* benchmark, it is statistically higher from an accuracy of 0.625; it exhibits an increasing trend when the accuracy level is increased, it has the highest accuracy level (63.24%), and it is always outperforming the *complete* classifier (except for one level of accuracy).

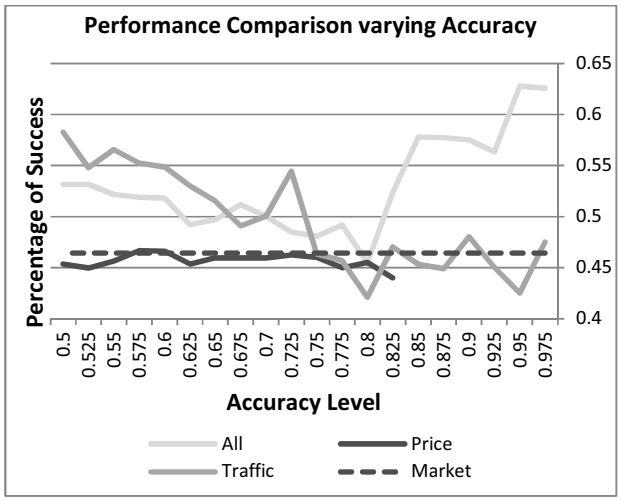
Regarding the *complete* classifier, it outperforms the market benchmark constantly, but it clearly underperforms the *traffic*. Surprisingly, by adding price-related information the classifier deteriorates its performance in predicting positive outcome.

Regarding the *price* classifier, we first notice that we have performance values only up to an accuracy of 0.825, due to the fact that further values restrict the size of the classifiable cases too much. The classifier generates fewer rules with bigger support and lower accuracy. Where we have data, the *price* classifier tends to behave in a similar way as the *all* classifier. However, the absence of rules with high accuracy and support limits their performances that do not go beyond a peak of 57.1% prediction accuracy.

On average, the *traffic* classifier outperforms the market by about 4.8%, increasing the probability of success from 53.5% to 58.3%, while the *complete* classifier increases the probability of 2.9% and the price classifier of about 1.1%.

Regarding the ability to predict a fall in price, the predictors behave in quite different ways. The *price* classifier, where defined, exhibits performances that do not diverge from the *market* benchmark (*zero rule*). The *traffic* classifier outperforms significantly the market when we allow a lower level of accuracy threshold (therefore classifying more cases), while its performance plunges when we increase the accuracy level over 0.675. On the contrary, the *complete* classifier performs much better

when a high level of accuracy threshold is set on the rules, but it also outperforms the *market* - even if with lower degree - for low accuracy level thresholds. The *complete* classifier performs best, while the *traffic* classifier has variable performance. On average, *traffic* outperforms the market benchmark by 3.58%, while adding price to traffic increases the performance on a market benchmark up to 6.7% percentage points (52.57% vs 45.88%, a relative gain of about 10%) with a peak of 25% relative increase for higher accuracy thresholds.



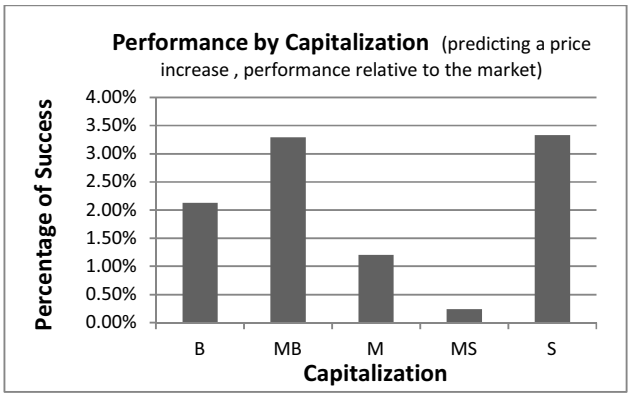
Graph 2. Performance predicting a price fall

In conclusion, our experimentation showed how a classifier built considering both traffic and price-related features outperforms a price-only classifier and the market, while a traffic only classifier outperforms all the other classifiers in predicting price increases.

3.3.2 Performance by Capitalization. We wondered if the size of a company, quantified by its market capitalization, has an impact on the quality of predictions. The analysis has important practical implications: if good performances were limited to smaller caps, a real trading strategy would have limited investment capabilities. We divided the 478 companies in the database in 5 equal frequency bins we labelled *small* (S), *medium*/*small* (MS), *medium* (M), *medium/big* (MB) and *big* (B) caps. This division is not optimal, since the underlying distribution of stocks capitalization is indeed skewed and each bin results populated with quite different companies, but the absence of few extremely big stocks from our dataset removes the major outliers.

Graph 3 shows the results of our analysis. The graph displays the performance of each group of stocks offset by the *market benchmark* (the *zero rule* model) of each group, showed in Table 3. There are substantial variations in the dataset, ranging from above 60% down to 47%.

All the 5 groups outperform their *market benchmark*, and a 95% statistical difference is not satisfied only for *medium/small* caps. Best results are achieved with small and big stocks. The results do not show a linear trend, but from a trading prospective the fact that performances of big capitalization stocks are still statistically significant makes a trading strategy able to absorb large investments.



Graph 3. Performance by capitalization

3.3.3 Rules and importance of factors. We now analyze the importance of each feature in the *traffic* classifier. The information gain of each feature and their presence in the rules with greatest support help to identify their impact on the overall predictions.

The capital of a company and the level of Rw_∞ and Rw_{20} (the long-term average and the monthly moving number of messages) are the most significant factors, followed by z_0 (z-score of the traffic at present day). Raw values are more significant than z-scores, monthly and long term moving averages are more important than current values. The classifier tends to make its predictions mainly by coupling the size of the company with its level of traffic in the mid and long term.

Table 3. The 6 rules with highest support

	Training		**Test**		**Rule body**			
	R_s	R_a	R_s	R_a	Cap	rw_∞	rw_{20}	z_0
1	9.13	70.1	11.8	57.2	MB	3,4,5	>1	
2	5.12	65.2	19.3	61.1	B, MB	6,7		
3	4.42	66.1	5.3	56.2	MB	4	0,1,2	
4	4.20	64.6	2.1	53.9	MB	3	0,1,2	
5	4.01	62.8	3.2	51.94	S	0,1,2	0,1	2,3
6	3.70	59.8	2.91	52.5	M, MS	<7	<5	0,1

The above table 3 shows the 6 rules with the largest support, responsible for about 45% of the classification process. All the 6 rules predict a price increase. For each rule we show: the support and accuracy during the training phase, the support and accuracy during the test phase and each rule's body. We remind that each feature have been discredized into 10 bins, where class *0* represents lowest levels, *5-6* medium values and *9* highest values.

The top rules clearly show how the classifier tends to associate company capital with certain levels of raw traffic.

By looking at the rules, for stocks with capitalization above the average the classifier requires a medium level of long-term traffic and usually a lower level of traffic in the last month. For instance, rules 1-4 apply to big and medium/big stocks and they all require a level of long-term raw traffic around the median (between classes 4 and 7), and a 1-month moving average always below class 2 (from low to very low). The rules could be summarized as follows: *for big or medium/big stock, there is a buy signal when the monthly moving average of the daily number of messages goes below the long-term moving average, and the latter has values around the median*.

For *small/medium* stocks (rules 5-6) the situation is similar, with slightly higher level of long-term traffic and considering current day values (represented by z_0). The rules avoid very high level of traffic indicators, favouring long, mid and short-term low of very low values. Mid-term values are smaller than long-terms ones as in rules 1-4.

In conclusion, traffic seems effective in predicting stock rise when certain levels of traffic are coupled with stock size. Common buy signals are the ones where the mid-term moving average of numbers of messages is lower than the long-term m.a., and it has a medium value. The findings seems to verify our conclusions in [2] that a decreasing but not null level of traffic seems more effective than increasing levels, and a high level of traffic usually has little predictive power.

5. User-Level Analysis

We now focus on the second research question. We aim to investigate if there is a subset of users that significantly and constantly outperforms/underperforms other users. The idea is to compare users based on their level of performance computed using a user-level version of the operator Tr defined in [2]. Tr is a cross-correlation-like coefficient between $G_s(t)$, the daily return for stock s, and S_T, a binary time series derived from $Z_{s,d}(t)$, that is the normalized version of $N_s(t)$ defined in equation 1. S_T is defined as:

$$S_T = \begin{cases} 0 \text{ if } Z_{s,d} & < T \\ 1 \text{ if } Z_{s,d} & \geq T \end{cases} \qquad (2)$$

Therefore S_T filters $Z_{s,d}$ and considers only days with certain levels of traffic. We call Tr the cross-correlation-like coefficient between the time series G_s and S_T:

$$Tr(n, s, d) = (S_T * G_s)(n) = \sum_{t=0}^{\infty} S_T(t) G_s(t - n), n < 0 \quad (3)$$

$$Tr(n, s, d) = (S_T * G_s)(n) = \sum_{t=0}^{\infty} S_T(t + n) G_s(t), n \geq 0$$

Since the value of Tr results in a sum of daily returns $G_s(t)$, Tr quantifies daily performances of a trading strategy based on signals extracted from $N_s(t)$.

Therefore, we first need to define a time series $Z_u(u, s, t)$ expressing the level of traffic for a specific user u instead of the entire community (as $Z_{s,d}$ was). Once we have defined Z_u, we can build a time series S_{uT} analogous to S_T for each user and proceed to correlate S_{uT} with G_S, as done in the previous section, in order to have a quantification of user performance over time.

A simple choice would be to repeat the analysis of [2] replacing $N_s(t)$ with $N_{u,s}(t)$. Anyway, when the analysis is done at user level, many interesting factors would be hidden by simply considering $N_{u,s}(t)$, such as:

1. the relation with other stocks where user u wrote. It would be interesting to consider how the user distributes its finite daily activity on various stocks s. For example, a value of $N_{u,s}(t) = 3$ has a different meaning if the user wrote only those 3 messages that day or if he wrote 30 messages over many other stocks.
2. the relation with other users: i.e. how user u's traffic differs from other users writing on common stocks. For instance, if a user is increasing his activity on a stock where other users are decreasing it, that is a stronger evidence of user u's interest in stock s.

In order to catch the above two properties, we propose to model Z_u in the following way. We start from $N_{u,s}(t)$ and we consider the portion of daily activity that user u generated on stock s, defining $A(u, s, t)$:

$$A(u, s, t) = \frac{N_{u,s}(t)}{N_u(t)} \qquad (4)$$

$A(u, s, t)$ is a time series telling how much of user u's activity is spent on stock s at day t, and it is therefore an indicator that also considers the activity of the user *outside* stock s. Since we are interested in considering the value of A over time, we normalize $A(u, s, t)$ using its z-score obtaining $Z_{A(u,s,t)}$ using formula (1).

Now $Z_{A(u,s,t)}$ tells us how the portion of activity that user u is dedicating to stock s is historically higher or lower than average.

We also want to add information about other users' traffic. For each stock where user u wrote, we consider the distribution $D_A(s, t)$ of values of A for all the users writing on s that day . Note how D_A is a distribution across users. The position of user u in this distribution tells us if the user is writing more or less than the other users on that stock. We normalize the distribution $D_A(s, t)$ defining $Z_{D_{A(s,t)}}$, that express the level of activity of u on stock s compared to other users that day. Note how, given a stock s, $Z_{D_{A(s,t)}}$ goes across the users' dimension for a fixed day t , while $Z_{A(u,s,t)}$ goes across the time dimension for a fixed user u, catching the two features we wanted to model.

Finally, $Z_u(u, s, t_0)$ for user u at time t_0 for stock s is the average of the two above series:

$$Z_u(u, s, t_0) = \frac{1}{2}\left(Z_{D_{A(s,t_0)}} + Z_{A(u,s,t)}\right) \qquad (5)$$

A user has a high value of Z_u if he:

(1) writes on stock s more than the other users,
(2) writes on stock s more than what he does on the other stocks and
(3) writes more on stock s than its historical average on stock s.

We now treat Z_u as we treated the series $Z_{s,d}$ for the aggregated traffic.

We therefore build a series Z_{u_T} and we generate a value $Tr(u,s,t)$ – using G_S- that quantifies user u performance on stock s at day t. We consider the set $X_u(t)$ of all the stocks where user u wrote at day t, and we define the daily indicator of performance $\alpha_u(t)$ for user u by averaging Tr over all the stocks in $X_u(t)$. Therefore:

$$\alpha_u(t) = \frac{1}{|X_u(t)|} \sum_{s \in X_u(t)} Tr(u,s,t) \qquad (6)$$

We can also aggregate the value of α_u over a given interval $[t_1, t_2]$ obtaining $\alpha_u(t_1, t_2)$.

Using α_u we can analyze if there exists a subset of users whose predictions statistically outperform the market.

A series of problems arise. Since data are sparse and market volatility changed dramatically during our 8 years, it is not possible to compare performance values of users collected in different market trends. In fact, the sparsity of data forces us to extend the period of time to collect enough information on a specific user, but the volatility of the market makes the data collected – and the values of α_u – no more directly comparable among users. We decide to use rank-based values, replacing absolute values of performances with median and percentile scores for each user. We use therefore a relative performance indicator among users.

In order to rank users on a given day, we first compute the daily index of performance α_u for each user. The value of α_u is offset with a market benchmark (i.e. the SP500 daily index value) to remove market conditions. If we are interested in intervals of time including many days, we aggregate the performance values to generate $\alpha_u(t_1, t_2)$. The overall performance score for user u in an interval $[t_1, t_2]$, called $r_u(t_1, t_2)$, is the percentile rank of user u among the distribution of $\alpha_u(t_1, t_2)$ for all the users available in $[t_1, t_2]$.

Another issue is how to handle missing values when users did not generate any activity for some days in $[t_1, t_2]$. If we assign to missing days a user's performance value of zero, that could represent a very high performance in period of falling market and vice-versa. Moreover, we actually do not know if the absence is intentional. We decide to discard periods of no activity into the computation of user's past performance. User's performances are tested only when he generates some activity on some stocks.

5.1 Experimental Analysis

In order to test the presence of a set of users that constantly outperform or underperform the market, we compute for each user a daily level of performance α_u, we aggregate it in weekly indicators and we use it to rank users to generate the performance score r_u. Users were required to have a minimum of 3 days of activity during each week to be considered. Each user has therefore associated a set of weekly ranks $r_u(t)$ where t is the number of the week considered. Our data spans 401 weeks - almost 8 years, so $t \in [0,401]$ where zero is the most recent week. The rank r_u is a number in $[0,1]$ that - as any percentile rank - represents the portion of users that scored less than user u (highest score is therefore 1).

We wonder if knowing that a user constantly outperformed the market in its last m available weeks helps predicting its next future performance.

Since we use percentile ranks, a user outperforms when its rank is at least greater than 0.5 and vice-versa. Moreover, given a value of R in [0,1], a user has a theoretical probability $1 - R$ of having a rank greater than R (for instance, if R = 0.7, theoretically a user has a probability of 0.3 to be in the top 30% users, i.e. having a percentile rank more than 0.7).

We set a memory value m and a rank threshold $T = [0,1]$. We select users that have been outperforming – i.e. their rank r_u was above threshold T – for the last m available weekly performance. This m past performance is simply the last m available for that user, and it can be distributed over any amount of time, consecutive or not. We call the last m available weeks for user u $t_{ua_1}, t_{ua_2}, \dots, t_{ua_m}$

We are interested in computing the following conditional probability $P_u(T, m)$:

$$P_u(T, m) = P\left(r_u(t_{un}) > T \,\middle|\, r_u(t_{ua_1}) > T, \dots, r_u(t_{ua_m}) > T\right)$$

that measures the probability that u will have a score greater than T in the next available week if we know that user u had a performance rank more than T in the past m available weeks $t_{ua_1}, \dots, t_{ua_m}$. If $P_u(T, m)$ is above the theoretical random probability (equal to $1 - T$), this means that user u predictably beats its peers.

Table 4. Users' predictions, probability $P_u(T, m)$

T	Theoretical Probability	Memory (weeks)				
T	1-T	1	2	3	4	5
0.5	0.5	0.66	0.66	0.67	0.67	0.66
0.55	0.45	0.59	0.60	0.61	0.61	0.60
0.6	0.4	0.52	0.52	0.55	0.53	0.51
0.65	0.35	0.45	0.45	0.48	0.47	0.46
0.7	0.3	0.38	0.39	0.43	0.45	0.43
0.75	0.25	0.32	0.33	0.36	0.40	0.34
0.8	0.2	0.25	0.27	0.34	0.37	0.30
0.85	0.15	0.19	0.21	0.28	0.31	0.29
0.9	0.1	0.13	0.16	0.24	0.21	0.00
0.95	0.05	0.06	0.11	0.05	0.00	N/A

We varied the rank threshold T from 0.5 to 0.95 with a 0.05 interval, and the memory size m from 1 to 5 weeks, defining 50 different test scenarios.

We tested using over 70 000 users from Yahoo! Finance obtaining the results displayed in table 3. The table shows the average value of $P_u(T, m)$ for the 70 000 users for a chosen memory size m and a threshold value of T. In all the tests only twice was the probability P_u less than $1 - T$, and in two very extreme cases with very little users satisfying the conditions. The data shows how there are users that constantly outperform the others. For instance, the average probability of a user to have a rank greater than 0.5 if he had a rank greater than 0.5 in the last 3 available weeks is 67%, against the theoretical 50%; while the probability of having

a rank above 0.8 (if he was above that rank in the last 3 available weeks) is 34% against the theoretical 20%.

6. Related Works

This paper investigates the predictive power of online communities' data with respect to financial trading.

The issue has been first extensively by Antweiler and Frank in [1]. The dataset used was 1.5 million posts from Yahoo Finance and RagBull, and the study covered 45 stocks of the Dow Jones Industrial Average. The authors applied text-mining techniques - a trained naive Bayes classifier - to extract a polarity sentiment from users' posts. The authors' key conclusion was the following: the effect of stock messages helps predict market volatility, but the effect on stock return is statistically significant but economically moderate. Disagreement among posted messages is correlated with increasing trading volume.

A recent study has been performed by Spiegel et al. [11] over the effect of rumours over stock return. In their context, rumours are not coming from online communities and they are not user-generated, but rather news, recommendation and indications coming from financial portal such as The Bursa (www.dbursa.com) or trading4living.com. The study concludes how during the event day and the 5 days preceding it the abnormal stock return is positively and statically significant. The dataset was composed by 958 Israeli stocks monitored for 27 months using a set of about 2000 rumours.

The recent work by [3] investigates the predictive power of Twitter's messages. The dataset used consisted of about 10m posts by 2.7M users in the period February-December 2008. The trained system was tested over 1 months period in December 2008 over the closing of the Dow Jones index. The methodology used was as follows: authors extracted from tweets' text 7 indicators of mood using OpinionFinder and GPMOS. Using a Granger causality analysis, authors correlate DJIA values to GPOMs and OF values of the past n days to obtain 83% accuracy. The author reports that calm, other then positve/negatie sentiment better predicts the market.

The work by Cook and Lu [7] follows a similar methodology. Our research, by improving sample selection and removing noise caused by program generated sentiments, finds the bullishness of board messages positively and significantly predict abnormal stock return up to 3 days ahead. More importantly, when taking poster's credibility into account, we find that the board messages' predictive power over stock returns becomes much stronger in terms of both economic magnitude and significance

The dataset used consists of Yahoo Finance messages collected in one year time (2007), applied over a set of 52 shares. The model contains an indicator of the sentiment computed over the tagged 5-level sentiment that Yahoo! users can declare, and they added a novel indicator of users' past performance, based on the sentiment users declared and stock movement t-days after. The test methodology follows Antwellier's panel regression [1].

The work by Gu [9] follows a similar methodology to the one described in [1]. Authors selected Yahoo! Finance messages from April 2005 to April 2006, using the same stock dataset as in [1]. The model encompasses a past-performance indicator based on the tagged sentiment by the users. The author finds that a weighted average recommendation of a stock message board has prediction power over future excessive returns of the stock. The effect is both statistically and economically significant. Interestingly, a simple average recommendation of a stock has no prediction power for future stock movements.

The work by Sehgal [10] is also in the space of user-level analysis. Users' past sentiment is computed by using a Naive Bayesian classifier over a trained set of messages, using as ground-truth for the training the messages containing tagged sentiment. The sentiment is also computed conditionally to the market movements and news announcement. The dataset used is limited to 3 stocks (Apple, Exxon Mobile and Starbucks) and shows about a 70% degree of accuracy for short term prediction, that is augmented by 9% when user trust value is introduced.

The above three works introduce a user-level analysis to enhance predictions, i.e. users are not considered all the same but they are somehow ranked on the basis of their credibility. The past-performance closed-loop is based on both cases to explicitly tagged sentiment. This source of information is in any case limited.

The work by De Choudhury [5] is of particular interests, since it derives market predictions by analyzing communities' dynamics rather than text. The authors focuses on blogs and they identify a set of dynamic features, such as normalized response time, early and late responses, and activity measurement such as activity loyalist and outliers. Other features are post length, rank - as provided by the blog editor software, number of posts, comments, size of loyal and outliers. These features are then correlated to the market dynamics training a support vector machine with the following results: 78% accuracy in predicting the magnitude of the movement and 87% in the accuracy of the movement after one week (weekly)

Similar works in the area are the ones by Agarwall et al. [6] on the general problem of identifying influential bloggers in a community and the work by U. Zhang [8], that studied the correlation between past-performance of an user and its reputation. The authors provides insight on what constitutes a reputable and respected user, and concludes how reputation derive from a more complex synthesis of various behavioral factors besides its textual contributions, implicitly confirming the validity of non-textual features.

In conclusion, the panorama is dominated by text-mining technique and past-performance indicators based again on explicitly tagged sentiment. Moreover, there is a mixed set of conclusions about the predictive capacity of online communities, ranging from not economically significant to highly significant impact. The study, except one, covers 1-year period or less, and no more than 45 stocks. Only the paper by De Choudhury [5] provides behavioural elements that are then correlated to the stock

market. Table 4 (next page 10) summarizes the dataset and techniques used, comparing them with our work.

Conclusions

In this study we have investigated the predictive power of online communities in respect to stock prices. We used the largest dataset to date, spanning 8 years and almost the complete set of SP500 stocks; we first build a decision-tree classifier using a features set entirely extracted from web-traffic data of financial online communities.

Our experimentation showed how a classifier built considering both traffic and price-related features outperforms a price-only classifier and the market benchmark, while a traffic only classifier outperforms all the other classifiers in predicting price increases, with a gain of 4.2% on average and up to 25% compared to the market benchmark. Traffic-related features seem effective in predicting stock rises when certain levels of traffic are coupled with stock size. The best predictive performances are achieved when information about stock capitalization is coupled with long-term and mid-term web traffic levels.

In the second part of our analysis we have shown how there is a subset of users that constantly outperforms the others. The finding sets the foundation of a promising study into user-level and behavioural models for market predictions. We believe to have provided enough evidence to set the foundation of future works in the development of new market analysis indicators.

7. References

[1] W. Antweiler and M.Z. Frank, 2004. Is all that talk just noise? The information content of internet stock message boards. Journal of Finance, 59: 1259-1295.

[2] P. Dondio, Predicting Stock Market Using Online Communities Raw Traffic, in the proceedings of IEEE/ACM/WI International Conference on Web Intelligence 2012, Macau, China

[3] J. Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. Journal of Computational Science, In press, 2011.

[4] Finanza Online Community, www.finanzaonline.it/forum

[5] De Choudhury M., Hari S., Ajita J., and Dorée Duncan S. Can blog communication dynamics be correlated with stock market activity?. In *Proceedings of the 19th ACM conference on Hypertext and hypermedia* (HT '08). ACM, New York, NY, USA, 55-60.

[6] Agarwal N., Huan Liu, Lei Tang, and Philip S. Yu. 2008. Identifying the influential bloggers in a community. In *Proceedings of the international conference on Web search and web data mining* (WSDM '08). ACM, New York, NY, USA, 207-218.

[7] Cook, D.O. & Lu, X. 2009 Noise, Information, and Rumors: Internet Board Messages Affect Stock Returns. Working Paper.

[8] Zhang, Y. and P.E. Swanson, 2009. Are day traders bias free?-Evidence from internet stock message boards. J. Econ. Finance. DOI: 10.1007/s12197008-9063-1

[9] Gu, B., P. Konana, A. Liu, B. Rajagopalan and J. Ghosh, 2007. Predictive value of stock message board sentiments. Working Paper, University of Texas at Austin.

[10] Sehgal V. and Song C.. 2007. SOPS: Stock Prediction Using Web Sentiment. In *Proceedings of the Seventh IEEE International Conference on Data Mining Workshops* (ICDMW '07), Washington, DC, USA, 21-26.

[11] Spiegel U., T. Tavor, J. Templeman, 2010. "The effects of rumours on financial market efficiency," Applied Economics Letters, Taylor and Francis Journals, vol. 17(15), 1461-1464.

Table 5. Datasets and Techniques

Author	Source	Size	Time	Stocks	Techniques/Features
Antweiler [1]	Yahoo Finance, RagingBull	1.5 million	1 yr	45 DJA	Text-mining, Bayes Classifier
Choudhury [5]	The Bursa Trading4Living	2000 news	2 yrs	958 Israeli stocks	Sentiment of news and experts opinions
Bollen [3]	Twitter	10 million	8 mo.	DJA INDEX	Text-mining, 7 mood indicators extracted from text features
Cook [7]	Yahoo Finance	1 million	1 year	52 US big cap	Sentiment tagged by users + user past performance
Agarwall [6]	/www.engadget.com BLOG	2,469 posts, 41,372 comments	10 mo.	4 big cap	Behavioural features of users posting and commenting Blogs
Our study	**Yahoo! Finance**	**26.28 m**	**8 yrs**	**478**	**Raw traffic, user-level indicator**
Sehgal [10]	Yahoo! Finance	Not Stated	Not Stated	3	Bayes Classifier, users' rank based on past performance
Gu [9]	Yahoo! Finance MB	Not Stated	1 year	45 stocks DJ	Sentiment tagged by users + users' past performance