

Beating Human Analysts in Nowcasting Corporate Earnings by using Publicly Available Stock Price and Correlation Features

Michael Kamp Mario Boley Thomas Gärtner

Fraunhofer IAIS
{*firstname.lastname*}@iais.fhg.de

Abstract

Corporate earnings are a crucial indicator for investment and business valuation. Despite their importance and the fact that classic econometric approaches fail to match analyst forecasts by orders of magnitude, the automatic prediction of corporate earnings from public data is not in the focus of current machine learning research. In this paper, we present for the first time a fully automatized machine learning method for earnings prediction that at the same time a) only relies on publicly available data and b) can outperform human analysts. The latter is shown empirically in an experiment involving all S&P 100 companies in a test period from 2008 to 2012. The approach employs a simple linear regression model based on a novel feature space of stock market prices and their pairwise correlations. With this work we follow the recent trend of nowcasting, i.e., of creating accurate contemporary forecasts of undisclosed target values based on publicly observable proxy variables.¹

1 Introduction

Corporate earnings are a crucial signal for investment and business valuation as they are a key indicator of a company's success and the development of its equity [17, 19, 22]. They are, however, only published at the end of fixed accounting periods and are confidential up to this point. Hence, there is a great interest in accurate estimations of this value based on publicly available data. While human analysts provide such estimations with reasonable accuracy [3], automatized methods based on classic econometric and statistical approaches fail to reach the quality of human experts by orders of magnitude [9]. Despite this gap, the earnings prediction problem is not in the focus of modern machine learning research. There are exceptions, which, however, use undisclosed variables for prediction (e.g., [7, 26]) or

focus on different objectives such as earning surprises or direct stock price prediction (e.g., [11, 15, 18, 21, 25]).

In this paper, we present for the first time a fully automatized machine learning method for earnings prediction based on completely publicly available data that can outperform human analysts. This is shown empirically in an experiment with the S&P 100 companies in a test period from 2008 to 2012. In this period a total of 2000 earnings predictions have to be provided, for which the proposed method outperforms the predictions of human experts on average as well as on the majority of individual stocks and points in time.

In figure 1 we illustrate the problem and our results for two exemplary companies, Amazon.com Inc. and Time Warner Inc. The earnings of the companies (True) are the cumulated earnings per share for each quarter as published at the end of the corresponding quarter. The goal is to nowcast, or predict these earnings before the actual publication.

For this purpose, finance analysts generate such forecasts using their individual methodologies, which are combined to an Analyst Consensus forecast. These Analyst Consensus forecasts are provided shortly after the beginning of each quarter—in our dataset on average 9.4 days after the beginning of the quarter—and are unavailable before publication. With our proposed method, nowcasts based on daily stock prices are available on each day. The data required to employ the method are the publicly available stock price time series of the target company along with the price series of as many as possible other reference companies.

In particular, the approach employs a simple linear regression model based on a novel feature space of stock market prices and their pairwise correlations. The rationale for this feature space is that the stock price of a company is a good proxy for its earnings [2, 24]. Furthermore, the earnings of a company depend on other market participants, e.g., its competitors or component suppliers. These rather durable interrelationships can be modeled using linear weights. However, stock

¹A preliminary version of this paper has been published at the workshop on Domain Driven Data Mining (DDDM 2013)

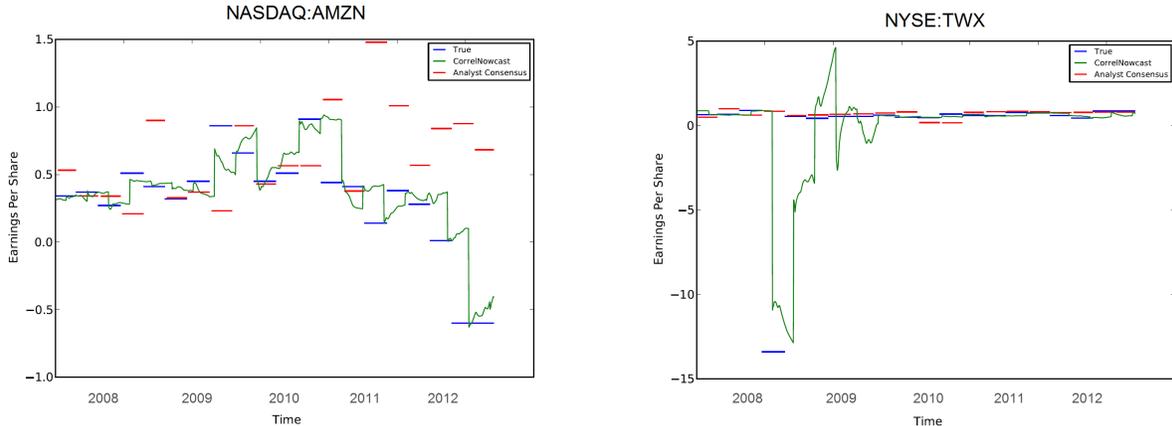


Figure 1: Quarterly earnings per share (True) for Amazon.com Inc. (left) and Time Warner Inc. (right), together with the Analyst Consensus forecasts and the nowcasts generated with the method proposed in this paper (CorrelNowcast). The experiment covers the time from 2008 to 2012.

prices can behave independently from the companies’ performance, e.g., because of speculation or transient trends [16]. In our model, the relation between the target company’s earnings from the stock price of another company is credible if their stock prices are congruent, which is measured by the proposed correlation features. The significance of this feature augmentation is also investigated more rigorously on a further extended experiment involving 5200 earnings forecasts for 200 companies. Prediction methods that only use basic price features are substantially outperformed by our price and correlation based method.

With this work we follow the recent trend of nowcasting [1, 5, 8], i.e., of creating accurate contemporary forecasts of undisclosed target values based on publicly observable proxy variables. The idea is that, while the future of complex systems is hard to predict, their present can be “predicted” reasonably well using the massive amounts of data available nowadays. In this paper, the aggregated earnings of a company in a reporting period are the undisclosed target values that are only published at the end of each business quarter. The stock prices act as observable proxy variables. Note that, while our task of nowcasting the aggregated earnings implies not only estimating the current state but also a mild true forecasting component, this is also true for other successful nowcasting applications [12, 13]. In particular, towards the end of an accounting period the earnings prediction problem approaches a pure nowcasting setting.

In the remainder of the paper, in section 2, we define the correlation feature space as well as the proposed nowcasting method. In section 3, we describe the

experimental setup followed by the presentation and discussion of the empirical results. Finally, in section 4, we summarize our method and the empirical evaluation and conclude by giving an outlook on future work.

2 Earnings Forecasts

This section describes the correlation feature space, formalizes the task of earnings prediction from public stock prices and presents the employed prediction method.

2.1 Preliminaries In the following, we always consider a dedicated target stock s^* from a fixed set of stocks S . For this set of stocks the **price time series** $\mathbf{p} : \mathbb{N} \rightarrow \mathbb{R}^S$ contains the stock prices of all stocks for each point in time, i.e., $\mathbf{p}_s(t)$ is the price of stock $s \in S$ at time $t \in \mathbb{N}$. For simplicity, we assume a stock price exists for each point in time. In practice, trading days can be mapped to this price time series by either only considering trading days as valid points in time or by keeping the stock price constant if no price update exists.

Furthermore, we define the set of **earnings announcements** $T_e \subset \mathbb{N}$ for the target stock s^* , i.e., at each point in time $t \in T_e$, new earnings have been published. For a point in time $t \in \mathbb{N}$, we can now define $t_{\text{next}} = \min\{t' \in T_e : t' > t\}$ and $t_{\text{last}} = \max\{t' \in T_e : t' < t\}$, i.e., the points in time of the next earnings announcement after and the last earnings announcement before t . The **earnings time series** $\mathbf{e} : \mathbb{N} \rightarrow \mathbb{R}$ is a piecewise constant time series with $\mathbf{e}(t) = \mathbf{e}(t_{\text{next}})$ for all $t \in \mathbb{N}$. That is, $\mathbf{e}_{s^*}(t)$ contains the potentially unknown earnings of stock $s^* \in S$ announced at time t_{next} .

In this paper, price updates correspond to daily closing prices of stocks, though the method can be straight-forwardly adapted to any finer time resolution. As a measure for the earnings of a company listed on the stock market, the **earnings per share** (EPS) are used, i.e., the accumulated earnings of the company in an accounting interval divided by the number of stock shares the company emitted. The earnings per share are published quarterly in the company’s income statement or its earnings announcement.

In practice, prices and earnings are only available for a finite timespan. We represent this fact in our notation as time windows. Given a time series $f : \mathbb{N} \rightarrow \mathbb{R}$ and a **time interval** $[i, j]$ with $i, j \in \mathbb{N}$ and $i < j$, the **time window** from i to j is the finite time series

$$f[i, j]: \{1, \dots, j - i\} \rightarrow \mathbb{R}$$

defined by

$$f[i, j](t) = f(i + t) .$$

Using the definitions above, the problem tackled in this paper can be formulated as follows.

DEFINITION 1. *Given a point in time $t \in \mathbb{N}$, a set of stocks S , a designated target stock $s^* \in S$, the stock prices $\mathbf{p}[1, t]$ of all stocks $s \in S$ until t , as well as the earnings $\mathbf{e}[1, t_{last}]$ until the last earnings announcement, **predict** the next earnings $\mathbf{e}_{s^*}(t_{next})$ of target stock s^* .*

2.2 Price and Correlation Feature Space We now motivate and describe our proposed correlation feature space that is derived from publicly available stock prices. The strong relationship between stock prices and earnings is well studied [22] and used for investment; the price-earnings ratio for example is an important indicator for stock valuation. At the same time, information related to the companies earnings are priced in the stock rate as soon as investors learn of it. Thus, we conclude that stock prices are a good proxy for current corporate earnings.

Because the earnings of a company are related to other market participants [4], e.g., component suppliers or competitors, conclusions can be drawn from the performance of those participants and vice versa. For example, prospering business for Volkswagen leads to a higher demand of parts, implying higher sales for its suppliers Schaeffler and Continental. Similarly, higher earnings for Sharp tends to imply higher earnings of Foxconn, for which Sharp is a major supplier. This again inclines to imply higher earnings of Apple, for which Foxconn is a major supplier. By using the stock prices of related companies as proxy for their prosperity, the mutual influence between companies can be incorporated in the feature space.

This steady relationship between the stock prices of a set of companies and the earnings of a dedicated target company can expressed using a linear model. Assuming a normally distributed noise on stock prices, a regularized least squares, or ridge regression is the maximum likelihood estimator and thus a suitable approach.

To increase the robustness of the estimations, we propose to measure prices according to different temporal resolutions. It is a standard in fundamental as well as in technical investment to not solely rely on the momentary picture of the last available closing price. Since stock prices are highly volatile, one also considers smoothed prices given by moving averages of different time window lengths. Formally, let $l \in \mathbb{N}$ be some interval length. Denoting by

$$\bar{f} = \sum_{t=1}^{|f|} f(t)/|f|$$

the **average** value of a finite time series f , this gives rise to the smoothed time series of **moving average prices**

$$(2.1) \quad a_s^l(t) = \overline{\mathbf{p}_s[t - l, t]}$$

for all stocks $s \in S$.

However, stock prices are susceptible to speculation, trends or temporary effects of high impact that are not related with the company’s performance. By also considering the correlation between stock prices, such temporary effects are expressed as a change in correlation. By weighting the stock prices with the current correlation coefficient between the related and the target company’s stock price, the influence of the stock price of market participants can be temporarily suspended or reversed if the correlation changes.

For example, the short-selling of Volkswagen stocks by investors in 2008 together with the attempt of Porsche to take over Volkswagen at the same time lead to a spectacular increase in Volkswagen’s stock price, even though the automobile sector was performing poorly at that time. Using the price of Volkswagen shares to estimate the earnings of its component supplier Continental would fail in that moment. The 50 days correlation between the stock prices of Volkswagen and Continental decreased significantly in this period but returned to their usual value of around 0.5 shortly after.

Accordingly, we are interested in the correlation between prices of two stocks for a fixed time window. The **sample Pearson correlation coefficient** $\text{cor}(f, g)$ between two finite time series f, g with $|f| = |g|$ is given by

$$\text{cor}(f, g) = \text{cov}(f, g)/(\text{std}(f)\text{std}(g)) ,$$

where $\text{cov}(f, g)$ denotes the sample co-variance and $\text{std}(f)$ the sample standard deviation. As for the stock prices, given an interval length $l \in \mathbb{N}$, we define the **moving correlation** between two stocks $s, u \in S$ as

$$(2.2) \quad c_{s,u}^l(t) = \text{cor}(\mathbf{p}_s[t-l, t], \mathbf{p}_u[t-l, t]) .$$

Following our described intuition, we want to represent the earnings of a target company with stock $s \in S$ at time $t \in \mathbb{N}$ by its stock price as well as by the prices of all other stocks in the market weighted by their correlation with s . Also, we want to provide this information with respect to different time resolutions in order to capture short term and long term relations at the same time. In accordance with economic practice (see, e.g., [10]), for each moment in time t we consider a **short-term**, a **mid-term**, and a **long-term time window**, looking back 11, 50, and 200 days, respectively.

Putting everything together, we can define the **feature representation** $\varphi_s: \mathbb{N} \rightarrow \mathbb{R}^d$ for a stock $s \in S$ as

$$\varphi_s(\cdot) = \circ_{u \in S} (p_u(\cdot), a_u^{11}(\cdot)c_{u,s}^{11}(\cdot), a_u^{50}(\cdot)c_{u,s}^{50}(\cdot), a_u^{200}(\cdot)c_{u,s}^{200}(\cdot))$$

with definitions of moving average prices and moving correlations as given in equations (2.1) and (2.2). Here, the symbol \circ denotes the "concatenation" of features.

Note that weighting the average prices of each stock with its correlation to the target stock is not a linear scaling but a proper non-linear augmentation of the features, because the correlation is a non-linear function in both time series.

2.3 Prediction Method We now describe our proposed approach to nowcasting earnings per share using the feature representation described above. Given a target stock s^* from a set of stocks S and the definition of the feature representation φ , a point in time t naturally divides the financial data stream into a training and a prediction window. For a point in time $t \in \mathbb{N}$ and the corresponding last earnings announcement t_{last} , the **training set** E for target stock s^* is defined as

$$E = \{(\varphi_{s^*}(t'), e_{s^*}(t')) \mid t' \leq t_{\text{last}}\} .$$

Moreover, we define the **prediction window** P corresponding to t as

$$P = \{\varphi_{s^*}(t') \mid t_{\text{last}} < t' \leq t_{\text{next}}\} .$$

In order to prevent susceptibility to concept drifts, it is common practice to use only the last W data points instead of the entire available data for training. That is,

$$E_W = \{(\varphi_{s^*}(t'), e_{s^*}(t')) \mid t_{\text{last}} - W < t' \leq t_{\text{last}}\} .$$

Using the above definitions of training set and prediction window, any regression technique can be employed to generate earnings forecasts. We propose using ridge regression [14] to construct a linear model in the correlation feature space, which is a simple and fast method that does not modify the explicitly constructed feature space. For a training set E , the ridge regression model is defined as $w^* \in \mathbb{R}^d$ solving

$$\min_{w \in \mathbb{R}^d} \sum_{(\varphi, e) \in E} |w^\top \varphi - e|^2 + \nu \|w\|_2^2$$

with some positive regularization parameter $\nu \in \mathbb{R}_+$.

In practice, predictions are made for each price update without updating the model. Only at the time of a new earnings announcement, the stored training set E_W is updated and the model can be recomputed. This scenario can be viewed as online learning with delayed update. We tackle the problem of delayed update following the straight-forward approach of [20]. Examples for which the label is yet unknown are stored in a buffer. As soon as their label is revealed, i.e., the earnings are announced, the examples are presented to the learner in order of their arrival together with their now known label. Because the intervals between earnings announcement are fixed, this method only adds a constant factor to the space complexity of the algorithm.

For each point in time $t \in \mathbb{N}$, a standard online regression algorithm estimates $\mathbf{e}_{s^*}(t_{\text{next}})$, which is constant for the entire prediction window. Hence, the variance of the estimates can be reduced by averaging all estimates so far. Thereby, the proposed algorithm potentially improves its prediction quality with every element from the data stream. If the new element is an earnings update, a new training set is constructed and the model is updated. If the new element is a price update, a new prediction is generated that adds to the pool of predictions from which the new earnings nowcast is calculated as the average of all predictions in the pool.

The corresponding algorithm, called CorrelNowcast, is presented in algorithm 1. Given a target stock $s^* \in S$, a training window size W and a regularization parameter ν , the algorithm loops over all points in time t (line 3). Using the new prices $\mathbf{p}(t)$, the feature representation $\varphi_{s^*}(t)$ for target stock s^* is constructed and stored in the prediction window P (line 4). Then an earnings prediction for that point in time is calculated using the linear model w^* and stored in the predictions set Q (line 5). The final earnings nowcast is generated as the average of all earnings predictions in the current prediction set (line 6).

In case of an earnings update at time t (line 7), the

training set needs to be updated. Therefore, all elements for which no corresponding earnings have been available yet, i.e., all elements in the prediction window (line 8), are assigned to the current earnings update $\mathbf{e}(t)$. Each thereby obtained tuple of feature representation and earnings value is added to the training set (line 9). After that, the prediction window and set are cleared (line 11).

If by the previous step the training set E_W has been extended beyond its window size W (line 12), the first $|E_W| - W$ elements are removed from the training set (line 13). With the updated training set E_W , a new linear model w^* is calculated (line 15).

Algorithm 1: CorrelNowcast

input : target stock $s^* \in \mathcal{S}$, training window size W , regularization parameter ν
output: earnings forecasts

- 1 initialize $E_W, P, Q \leftarrow \emptyset$
- 2 initialize $w^* \in \mathbb{R}^{|d|} \leftarrow (0, \dots, 0)$
- 3 **foreach** *point in time* t **do**
- 4 $P \leftarrow P \cup \{\varphi_{s^*}(t)\}$
- 5 $Q \leftarrow Q \cup \{\varphi_{s^*}(t)^\top w^*\}$
- 6 **predict** $\text{avg}(Q)$
- 7 **if** $t \in T_e$ **then**
- 8 **foreach** $\varphi \in P$ **do**
- 9 $E_W \leftarrow E_W \cup \{(\varphi, \mathbf{e}_{s^*}(t))\}$
- 10 **end**
- 11 $P, Q \leftarrow \emptyset$
- 12 **if** $|E_W| > W$ **then**
- 13 remove first $|E_W| - W$ elements from E_W
- 14 **end**
- 15 $w^* \leftarrow \arg \min_{w \in \mathbb{R}^d} \sum_{(\varphi, e) \in E_W} |w^\top \varphi - e|^2 + \nu \|w\|_2^2$
- 16 **end**
- 17 **end**

A regularized least squares, or ridge regression can also be transformed into a full online algorithm by introducing a training example matrix A and a prediction vector b [27]. Both, matrix and vector, are incrementally updated with each training step. The weight vector, i.e., the linear model, is given implicitly by $b^\top A^{-1}$. For each step, this online ridge regression minimizes the least squared error as well as the norm of the weight vector. The online ridge regression approach can be adapted to a delayed reward scenario similar to the proposed algorithm. Because of several matrix-vector multiplications for each example, this method

is significantly slower than a batched ridge regression. In the case of earnings predictions, where long delays between new labels make learning only necessary after a considerable amount of examples already arrived, the faster CorrelNowcasting algorithm with a batched learning phase is preferable.

3 Experiments

In this section, we present two experiments on publicly available daily stock price and quarterly earnings data². In total, the experiments involve predicting 5200 earnings updates of 200 US stocks. In the first experiment, we show that the proposed method can outperform human analysts and in the second experiment we show that correlation features are significant to the proposed method. For reproducibility of results, data will be made available on our website³.

3.1 Comparison with Human Analysts We compare CorrelNowcast with human analysts on the Standard & Poor’s 100 index in the time from 2008 to 2012. Analyst consensus forecasts aggregated by Zacks Investment Research—main data provider, e.g., for Bloomberg—for this period have been obtained from bloomberg.com. As additional baselines we employ the classic econometric **ARMA** method (details can be found in the appendix A) and a trivial **constant** method for assessing the difficulty of the problem. This method simply uses the last available earnings as prediction value. Optimal parameter settings for each method are obtained on an independent dataset of 100 randomly selected stocks in the time from 2004 to 2006. For CorrelNowcast, the regularization parameter ν and the prediction window W have been found by a grid search with $\nu \in \{10, 100, 1000, 3000, 5000, 10000\}$ and $W \in \{11, 50, 125, 200, 250, 350, 500, 750\}$.

The results of the experiment are listed in table 1. We provide the mean relative error (MRE), i.e., the absolute error relative to the true value, an error measure related to investment. The reporter errors are average values over a) all target stocks from S&P 100 and over b) all prediction days starting from the day of analyst forecast until the earnings are published, i.e., the true label is revealed. Additionally, since CorrelNowcast provides increasingly refined predictions with each day (and is also defined prior to analyst forecasts), we provide the average MRE over all stocks on the first and last days of each quarter and on the exact day on which the analyst forecast is published.

²Prices from Google Finance (www.google.com/finance); earnings from YCharts (www.ycharts.com).

³<http://www-kd.iai.uni-bonn.de>

	MRE				Wins Overall				Wins per Stock			
	CN	A	AR	C	CN	A	AR	C	CN	A	AR	C
last day	0.85	1.24	1.99	2.09	0.48	0.23	0.22	0.07	0.47	0.22	0.25	0.06
analyst day	1.20	1.24	1.87	2.09	0.40	0.32	0.16	0.12	0.41	0.32	0.16	0.11
first day	1.29	-	1.85	2.09	0.51	-	0.31	0.18	0.59	-	0.34	0.07
all days	0.97	1.24	1.93	2.09	0.45	0.25	0.21	0.09	0.47	0.22	0.25	0.06

Table 1: Comparison of the proposed method CorrelNowcast (CN) with analyst’s consensus forecasts (A) on the SP100 stocks from 2008 to 2012 as well as baselines ARMA (AR) and constant (C). Results are averaged over all stocks and all points in time (all days) as well as on particular days in each quarter (first day, last day and day on which the analyst forecast is published).

Please note that on the first day of each quarter, an analyst forecast is not available.

We can observe that CorrelNowcast outperforms the analyst’s forecasts in terms of MRE on the whole quarter, as well as on the day of analyst’s forecasts and the last day of the quarter. In addition, CorrelNowcast is only 4% worse in terms of MRE, when earnings are predicted at the first day of the quarter—on avg. 9.4 days before analyst forecasts are even available.

To give more detailed insights in the results we also computed, which method had the smallest relative error for each day and each stock (Wins Overall), again for all days as well as on particular days. Again, CorrelNowcast outperforms the Analyst Consensus forecasts as well as the baselines on all days as well as on particular ones. Moreover, when directly comparing CorrelNowcast with the analysts—disregarding the baselines—then CorrelNowcast wins on 63% of all points in time and all stocks. To evaluate the performance per stock we computed which method had the smallest average relative error per stock, i.e., for each stock the relative errors for each point in time are averaged. Again, CorrelNowcast outperforms the analyst’s forecasts as well as the baselines. Furthermore, when directly comparing CorrelNowcast and the analyst’s forecasts, our method wins on 73% of stocks.

Confirming previous research, the ARMA model, as well as the constant baseline are outperformed by both methods. In figure 2 we provide a detailed view on the experiments by comparing CorrelNowcast and Analyst Consensus forecasts in terms of their relative errors. Each point in the scatter plot on the left side represents the ratio of relative errors of the predictions for a specific point in time. If a point lies above the diagonal, CorrelNowcast outperforms the Analyst Consensus forecasts and vice versa for points below the diagonal. In this plot, 62.9% of points lie above

the diagonal, i.e., CorrelNowcast outperforms Analyst Consensus forecasts in over 60% of all points in time and on all stocks. Furthermore, when only considering points representing the average over all points in time per stock, CorrelNowcast outperforms the analysts on 72.5% of all stocks. A separate plot of this average error is provided on the right side of 2.

In this separate plot we highlighted stocks for which either of the compared method outperforms the other significantly. For two of these stocks we exemplarily plotted the predictions and true values over time in figure 1. For Amazon.com Inc. (NASDAQ:AMZN), CorrelNowcast clearly outperformed the analysts, mainly because the analysts were overestimating the earnings in 2011 and even more in 2012, whereas CorrelNowcast was able to predict the earnings course more accurately. For Time Warner Inc. (NYSE:TWX) on the other hand, Analyst Consensus forecasts have a much lower mean relative error than CorrelNowcast. An explanation for this particular stock is that CorrelNowcast is able to predict the very low earnings at the end of 2008. However, the method overestimated earnings afterwards, when they went back to values close to zero, resulting in large relative errors.

In order to further assess the importance of the correlation feature space, we analyzed the weights of the regression model over time exemplary for Amazon Inc. In figure 3 we have taken six features that have the highest average weight for the entire dataset and plotted their weight development over time. Even though the weights are not absolutely stable over time, two transportation and logistic related companies have the highest average weight. That is, Norfolk Southern, a railway transportation and telecommunication holding and FedEx, a logistics and transportation corporation. With Amazon being one of the largest online shops world wide, this observation confirms the intuitive expectation that the

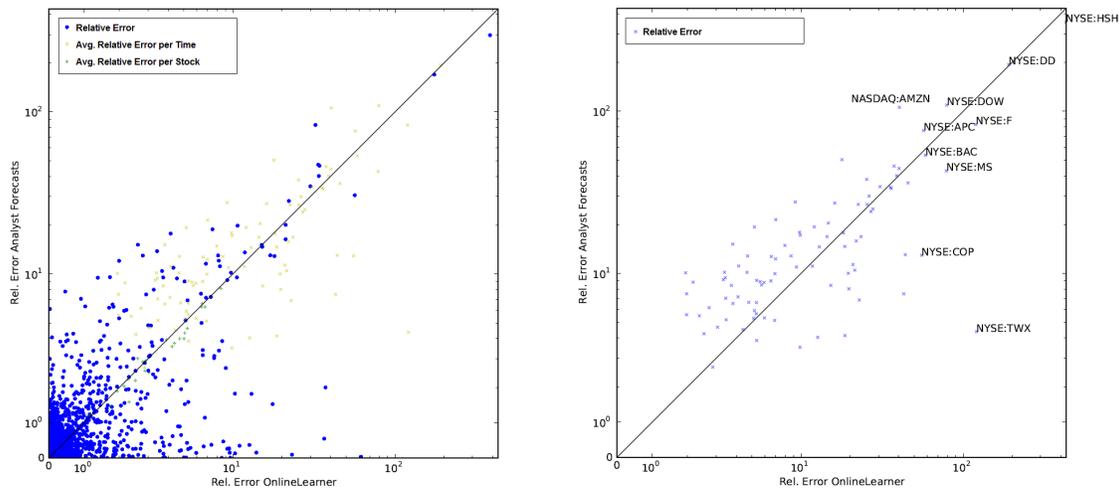


Figure 2: Comparison of the performance of predictions from CorrelNowcast and Analyst Consensus forecasts. The picture on the left provides an overview over the relative errors of CorrelNowcast vs. Analyst in log-space. Each point (blue) in the scatter plot represents a prediction date, its position represents the ratio of the relative errors. Furthermore, the ratio of average relative errors over all stocks per time point (yellow) and over all points in time per stock (green) are depicted. The picture on the right is a detailed view on the ratio of relative errors per stock, averaged over all points in time. Highlighted are stocks for which one of the compared methods is clearly superior. Again, the relative error is plotted in log-space.

weights reflect economic relationships. For Pfizer, a research based biotech company, it can be noted that the company uses Amazon’s Virtual Private Cloud, part of their Web Services (AWS), for their high performance computing needs.

ings from publicly available data that outperform human analysts. It furthermore indicates that the proposed method is able to generate accurate nowcasts even significantly before analyst’s forecasts are published.

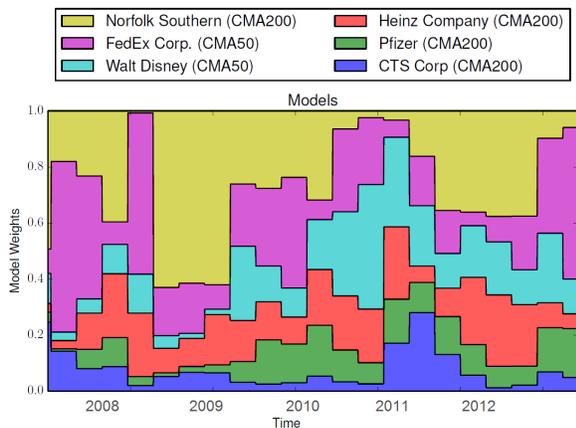


Figure 3: Weights of top 6 features for prediction of Amazon Inc.’s earnings over time. Here, CMA stands for correlation weighted moving average, the succeeding number denotes the time window.

In summary, this experiment verifies that the proposed method is capable of nowcasting corporate earn-

	RAND 100 (2007-2012)		S&P 100 (2004-2012)	
	RMSE	MRE	RMSE	MRE
CorrelNowcast	0.59	21.07	0.72	2.11
KernelPriceNowcast	0.69	23.67	0.99	2.47
LinPriceNowcast	0.75	24.23	0.79	2.79
TargetOnly	0.94	26.09	1.19	3.37
ARMA	1.39	32.61	2.24	4.23
Constant	1.63	162.17	1.97	5.54

Table 2: Averaged root mean squared error (RMSE) and mean relative error (MRE) on all S&P 100 target stocks between 2004 and 2012 as well as an additional independent test dataset (RAND 100) of 100 random US stocks between 2007 and 2012.

3.2 Significance of Correlation Features It remains to verify the significance of the design choices involved in the definition of CorrelNowcast—in particular the use of the price correlation features. For this purpose we compare the method against two nowcasting approaches that use a reduced feature space containing only prices and average prices:

$$\varphi'_{s^*}(\cdot) = \circ_{u \in S} (p_u(\cdot), a_u^{11}(\cdot), a_u^{50}(\cdot), a_u^{200}(\cdot)) \quad .$$

The two methods are for once the same linear ridge regression as for CorrelNowcast (**LinPriceNowcast**) and moreover a more expressive kernelized ridge regression (**KernelPriceNowcast**) utilizing a polynomial kernel

$$k(x, y) = (\gamma x^\top y + d)^d \quad .$$

A third baseline (**TargetOnly**) uses linear ridge regression only on the price time series of the target stock, including the smoothed time series. Again all parameters are optimized on the same tuning set as in the first experiment. For testing we now use an extended setup with two datasets: the S&P 100 from 2004-2012 and an independent set of 100 random US stocks from 2007-2012 (**RAND 100**⁴).

The results are listed in table 2. We provide the mean relative error and furthermore the rooted mean squared error. For additional comparison we again also provide results for ARMA and constant. For both error measures, CorrelNowcast substantially outperforms the baselines using only simple price features.

The results show that the correlation features clearly outperform simple price features. Even the more expressive kernel variant cannot lift the simple price features into competitive range. Note that the MRE is significantly smaller on S&P 100 than RAND 100, because the companies listed in S&P 100 are the 100 largest American companies with rather high earnings values, whereas in RAND 100 many stocks with very little earnings are listed so that even a moderate absolute error results in a large relative error.

Altogether this experiment not only confirms the necessity of using price correlation features, but also provides a much broader performance assessment of CorrelNowcast. Besides using more stocks, the included time periods show more variety of the underlying economic environment. It does not only contain data from a stable growth period (2004 to 2006), but also the financial crisis of 2007, the recession from 2008 to 2009 and the recovery period until 2012. Thus, this experiment shows that CorrelNowcast can maintain its good performance on independent data sets and longer time spans.

⁴The names of the stocks in this dataset are provided on our website, together with the data.

4 Conclusion

We presented a fully automatized method for nowcasting corporate earnings using a novel correlation feature space derived from publicly available price data. The proposed method is simple, fast and can be applied to any set of stocks, their prices and earnings. Experiments have shown that these nowcasts outperform analyst’s forecasts and are even competitive when generated significantly before the analyst’s forecasts are published. Besides the implications for the proposed method, these results emphasize the importance and potential capabilities of purely data driven methods for financial data.

With correlation features established as valuable information for earnings prediction, a natural advancement is to employ this feature space with more elaborate machine learning methods in order to further improve the predictive performance.

An interesting direction for follow up research is improving the feature space by including features derived from different data sources, most prominent from financial news [23]. To this extend, a bag of words approach as well as an approach using features derived by sentiment analyses appear promising.

5 Acknowledgments

This research has been supported by the EU FP7-ICT-2013-11 under grant 619491 (FERARI) and by the German Science Foundation (GA 1615/1-1 and GA 1615/2-1).

References

- [1] Knut Are Aastveit and Tørres Trovik. Nowcasting norwegian gdp: The role of asset prices in a small open economy. *Empirical Economics*, 42(1):95–119, 2012.
- [2] J.S. Abarbanell. Do analysts’ earnings forecasts incorporate information in prior stock price changes? *Journal of Accounting and Economics*, 14(2):147–165, 1991.
- [3] John Affleck-Graves, Larry R Davis, and Richard R Mendenhall. Forecasts of earnings per share: Possible sources of analyst superiority and bias. *Contemporary Accounting Research*, 6(2):501–517, 1990.
- [4] Andrew W. Alford. The effect of the set of comparable firms on the accuracy of the price-earnings valuation. *Journal of Accounting Research*, 30(1):94–108.
- [5] Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.
- [6] Lawrence D Brown and Michael S Rozeff. Univariate time-series models of quarterly accounting earnings per share: A proposed model. *Journal of Accounting Research*, pages 179–189, 1979.

- [7] Qing Cao and Mark E Parry. Neural network earnings per share forecasting models: A comparison of backward propagation and the genetic algorithm. *Decision Support Systems*, 47(1):32–41, 2009.
- [8] Hyunyoung Choi and Hal Varian. Predicting the present with google trends. *Economic Record*, 88(s1):2–9, 2012.
- [9] Robert Conroy and Robert Harris. Consensus forecasts of corporate earnings: Analysts’ forecasts and time series methods. *Management Science*, 33(6):725–738, 1987.
- [10] Credit Suisse. *Technical Analysis - Explained*. Credit Suisse Group AG.
- [11] Vasant Dhar and Dashin Chou. A comparison of nonlinear methods for predicting earnings surprises and returns. *Neural Networks, IEEE Transactions on*, 12(4):907–921, 2001.
- [12] Michael Ettredge, John Gerdes, and Gilbert Karuga. Using web-based search data to predict macroeconomic statistics. *Communications of the ACM*, 48(11):87–92, 2005.
- [13] Sharad Goel, Jake M Hofman, Sébastien Lahaie, David M Pennock, and Duncan J Watts. Predicting consumer behavior with web search. *Proceedings of the National Academy of Sciences*, 107(41):17486–17490, 2010.
- [14] Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.
- [15] Kyoung-jae Kim. Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1):307–319, 2003.
- [16] R.R. King, V.L. Smith, A.W. Williams, and M. Van Boening. The Robustness of Bubbles and Crashes in Experimental Stock Markets. *Nonlinear Dynamics and Evolutionary Economics*, pages 183–200, 1993.
- [17] Owen Lamont. Earnings and expected returns. *The Journal of Finance*, 53(5):1563–1587, 1998.
- [18] Sam Mahfoud and Ganesh Mani. Financial forecasting using genetic algorithms. *Applied Artificial Intelligence*, 10(6):543–566, 1996.
- [19] Spyros Makridakis. Forecasting: its role and value for planning and strategy. *International Journal of Forecasting*, 12(4):513–537, 1996.
- [20] Chris Mesterharm. On-line learning with delayed label feedback. In *Algorithmic Learning Theory*, pages 399–413. Springer, 2005.
- [21] M-A Mittermayer. Forecasting intraday stock price trends with text mining techniques. In *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*, pages 10–pp. IEEE, 2004.
- [22] James M Patell. Corporate forecasts of earnings per share and stock price behavior: Empirical test. *Journal of Accounting Research*, pages 246–276, 1976.
- [23] Robert P Schumaker and Hsinchun Chen. Textual analysis of stock market prediction using breaking financial news: The azfin text system. *ACM Transactions on Information Systems (TOIS)*, 27(2):12, 2009.
- [24] Richard G Sloan. Do stock prices fully reflect information in accruals and cash flows about future earnings? *Accounting Review*, pages 289–315, 1996.
- [25] CF Tsai and SP Wang. Stock price forecasting by hybrid machine learning techniques. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, volume 1, page 60, 2009.
- [26] Wei Zhang, Qing Cao, and Marc J Schniederjans. Neural network earnings per share forecasting models: a comparative analysis of alternative methods. *Decision Sciences*, 35(2):205–237, 2004.
- [27] Fedor Zhdanov and Vladimir Vovk. Competing with gaussian linear experts. *Transactions of the IRE Professional Group on Audio*, 30(6), 2009.

A ARMA

In our experiments we compare CorrelNowcast to a Box-Jenkins method, i.e., an autoregressive moving average (ARMA) model. The ARMA(p, q) model consists of the sum of an autoregressive component of order p and a moving average component of order q . For a time series $x: \mathbb{N} \rightarrow \mathbb{R}$, the model is defined as

$$x(t) = c + \epsilon_t + \sum_{i=1}^p v_i x(t-i) + \sum_{i=1}^q w_i \epsilon_{t-i} ,$$

with weights $v \in \mathbb{R}^p$ and $w \in \mathbb{R}^q$, an offset constant c and a Gaussian noise terms $\epsilon_t \sim \mathcal{N}(0, \sigma)$.

In our experiment, we apply this approach to the earnings time series as suggested in [6], updating the model whenever a new earnings value is published. For each model update, a grid search over the parameters p and q is performed to find the best fitting ARMA-model. For the grid search, we used $p, q \in \{1, \dots, 5\}$. For each setting of p and q , the parameters v, w, c are fitted to the training time series (containing all known values) of earnings per share values using a least squares regression. For each training set, the best performing model is chosen for prediction.