Predicting Surprises to GDP: A Comparison of Econometric and Machine Learning Techniques

By

Ved Rajkumar

B.A. Statistical Science
Cornell University, 2015

SUBMITTED TO THE MIT SLOAN SCHOOL OF MANAGEMENT IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF FINANCE
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

FEBRUARY 2017

Signature of Author: Signature redacted

MIT Sloan School of Management
January 20, 2017

Certified by: Signature redacted

Roberto Rigobon
Society of Sloan Fellows Professor of Management
Thesis Supervisor

Accepted by: Signature redacted

Heidi Pickett
Program Director, MIT Sloan Master of Finance Program
MIT Sloan School of Management

Predicting Surprises to GDP: A Comparison of Econometric and Machine Learning Techniques

By

Ved Rajkumar

Submitted to MIT Sloan School of Management
on January 20, 2017 in Partial Fulfillment of the
requirements for the Degree of Master of Finance.

Abstract

This study takes its inspiration from the practice of nowcasting, which involves making short horizon forecasts of specific data items, typically GDP growth in the context of economics. We alter this approach by targeting surprises to GDP growth, where the expectation is defined as the consensus estimate of economists and a surprise is a deviation of the realized value from the expectation. We seek to determine if surprises are predictable at a better than random rate through the use of four statistical techniques: OLS, logit, random forest, and neural network. In addition to evaluating predictability we also seek to compare the four techniques, the former two of which are common in econometric literature and the latter two of which are machine learning algorithms most commonly seen in engineering settings. We find that the neural network technique predicts surprises at an encouraging rate, and while the results are not overwhelmingly positive they do suggest that the model may identify relationships in the data that elude the consensus.

Thesis Supervisor: Roberto Rigobon
Title: Society of Sloan Fellows Professor of Management

Introduction

The state of the economy is of interest to policy makers, businesses, and financial market participants in order to inform decisions about the future, yet official information on aggregate economic growth, for instance, is typically available only a month after the period of interest. It is reasonable, therefore, that expectations of growth are valuable to policy decisions and investments before the actual growth announcement. There has recently been a keen interest in the academic and professional world over the last ten years to assess the state of the economy in real time, or to 'nowcast.' This study takes its inspiration from nowcasting but focuses on the related but different topic of forecasting surprises to GDP growth expectations rather than the level of growth. GDP is a common target of prediction in the literature simply because it is a key indicator of economic activity and it is released at a low frequency, but it is simply an example of a data item to be forecasted. This study aims to first identify if surprises to GDP growth are predictable. Predicting positive or negative surprises could have significant implications for financial market participants, as a robust prediction could provide an investor with information not priced in to the market before the official data announcement.

We use four different techniques to predict surprises and seek to evaluate and compare the techniques' predictive ability. The techniques are ordinary least squares and logistic regression, which are standard to econometric practice, and random forest and neural network, techniques commonly used in the emerging field of machine learning. Another goal of this study is to introduce these non-traditional techniques to the context of economics. The field of econometrics has been primarily concerned with understanding causality in order to determine policy. Here, we focus simply on prediction, and while they may be difficult to interpret, it is possible that algorithms that allow for more non-linear relationships can better predict outcomes, particularly if the outcomes are shocks as in the case of a surprise. We aim to assess each technique in its ability to predict surprises and to compare efficacy across techniques, given their unique characteristics. We find that there is some dispersion in predictive ability across the four techniques, suggesting that surprises to economic growth may be predictable, and that there exist significant non-linear relationships between surprises and more granular economic data. While the results are not overwhelming, the neural network's out of sample performance for trinomial predictions is encouraging and worthy of further exploration. The rest of the paper will describe

nowcasting and some established methodologies, the data used in our study and its treatment, our methodology and the four prediction models, and finally the results and discussion.

Nowcasting

The practice of nowcasting originates from meteorology. According to the Meteorological Office of the United Kingdom, Admiral Robert FitzRoy was the first to produce forecasts of inclement weather for the office in the 1860's. His methodology involved gathering storm reports from certain coastal areas and relaying those reports to areas that were likely downwind, with the short-term forecast that the storm would soon be upon the downwind port. While this method is perhaps the simplest form of nowcasting, it fits the criteria for the practice. Current information was gathered in order to make a short horizon prediction about a specific target value. In the 1980's, Professor Keith Browning, who also worked for the UK's Met Office, created the term nowcasting. He used the term to describe a process similar to Admiral FitzRoy's, the process of analyzing the information from current radar images to make a short-horizon forecast of rainfall.[1]

Today, the Met Office and other agencies that monitor the weather use nowcasting to quickly predict movements in or the presence of temperature, wind, snow, and fog. Forecasts are updated whenever a new observation is made, a key element of nowcasting, so that predictions reflect all relevant information available. As of 2011, the office reports that nowcasts of rainfall can be relevant for the next three to four hours in the winter and for one to two hours in the summer. Ideally, nowcasts will be relevant for extended periods while the nowcasts themselves are updated instantly, in order to provide predictions of immediate weather patterns. The Met Office currently uses a Short Term Ensemble Prediction System (STEPS) to nowcast rainfall. Developed with the Australian Bureau of Meteorology, the system makes predictions bucketed by size of rainfall, as large events can be nowcast for longer time horizons than small events. Small rainfall events are modeled by introducing randomness with specific characteristics into the system, so that the ensemble is a combination of more precise nowcasts of large rainfall events with bands of uncertainty determined by the modeled small rainfall events. These levels

1. "Nowcasting," Met Office, February 14, 2011, accessed December 6, 2016, http://www.metoffice.gov.uk/learning/science/hours-ahead/nowcasting.

and bands of uncertainty are important in order to predict extreme rain events that could lead to flooding.[2]

In the context of economics, the need for nowcasting is of course man made but features many similar characteristics of using currently available information to make short-horizon, in many cases daily, predictions that will inform decision making. While any data item can be nowcast, GDP is perhaps the most compelling because it is one of the least frequently released yet one of the most telling of the state of the economy. The value of nowcasting GDP accurately lies in knowing and being able to track the state of the macroeconomy without having to wait for a release once a quarter. This allows for more informed and timely decision making in between releases, and it can allow for policy makers to intervene with more confidence than they would have without the nowcast in addition to providing the ability to track the effect of an intervention, ideally in real time. The major challenges in the practice of nowcasting, however, are that the target value, GDP for instance, is often released infrequently, providing few points to train a model on, and that the dimensionality of the model is a constraint. With few data points for the target, estimating precise parameters can be difficult and those few target points combined with the large number of possible explanatory variables leaves the model with few degrees of freedom. The next subsections discuss prior explorations into nowcasting and attempts at overcoming these challenges.

Giannone, Reichlin, and Small (2008)

In amongst the initial ventures into nowcasting the macroeconomy in 2008, Domenico Giannone, Lucrezia Riechlin, and David Small nowcast inflation and real GDP and establish a framework for assessing updates to forecasts within the same month. Their purpose is to answer three questions: does a larger data set and more information lead to more accurate predictions, which types of data add to accuracy of forecasts on the margin, and for a given data type, is it the timeliness or the quality of data releases that contributes towards better nowcasts?

In order to tackle these questions, the authors use a dynamic factor model suggested in Doz, Giannone, and Reichlin (2005). The parameters of the model are estimated by using principal component analysis, and a Kalman filter is applied to extract the signal, or news, content of a data release from the noise associated with series specific errors. The model seeks to

2. Ibid.

capture a common component of all economic data, or rather, the state of the economy. The change of the common component based on autoregressive lagged relationships represents the business cycle. The authors use the noise to signal ratio to rank the impact of each data type.

According to their results, the authors find that information within each month has a significant effect on the precision of a forecast. They find that the uncertainty around a nowcast decreases uniformly throughout a quarter, as more information becomes available and is implemented into the model. On the question of data types that add the most information to nowcasts of real GDP growth regardless of size, the authors find that the New Residential Construction Release and Philadelphia Business Outlook Survey have the most impact in terms of noise to signal ratio. The authors also find that labor and wage data are important for nowcasting real GDP growth but not as important as the survey data. Finally, the authors condition on timeliness to measure the quality of data and find that hard data, or reports of ex-post economic performance, become important relative to soft data, or ex-ante expectations of the future.[3]

Banbura, Giannone, Modugno, and Reichlin (2013)

In 2013, Marta Banbura, Domenico Giannone, Michele Modugno, and Lucrezia Reichlin of the United States Federal Reserve and the European Central Bank survey a number of nowcasting methodologies previously published in the academic literature, analyzing their approaches and discussing their results. The authors also propose a new nowcasting technique that takes into account daily data, including financial data, factoring in the effect of timeliness and reporting lags to gauge the marginal impact of data on model forecasts. Finally, the authors create a daily index of economic activity and discuss its ability to map onto GDP as well as the S&P 500.

The authors review frameworks that allow for interpretations pertaining to the way financial market participants and policy makers read data releases. The key features include watching many data items, forming expectations of economic activity based on the data, and revising expectations when there are surprises. Most models reviewed are dynamic factor models, similar to that presented in Giannone, Reichlin, and Small (2008), but vector

3. Domenico Giannone, Lucrezia Riechlin, and David Small, "Nowcasting: The Real-Time Informational Content of Macroeconomic Data," Journal of Monetary Economics May 22, 2008.

autoregressive models with mixed frequency structures are also included. Single equations, such as bridge equations and MIDAS equations, are reviewed as well but with the caveat that they do not account for the effect of updates to nowcasts as data becomes available within a period.

The results of the studies suggest a few overall nowcasting takeaways. First is that short horizon forecasts add value over naïve assumptions of constant growth while longer term forecasts do not. The authors also report that the statistical techniques perform as well as more subjective forecasts by institutions. Similarly to the 2008 paper in the previous subsection, nowcasts become more accurate as the time period goes on between GDP releases and more data is incorporated into the model. Finally, the authors report that both timeliness and quality of data matter. In terms of quality, soft, survey information is found to be important for forecast accuracy.

The authors suggest a daily dynamic factor model to nowcast US GDP. They emphasize that the daily model is useful for incorporating financial market data and that while others have studied the effect of financial information on macroeconomic variables, this nowcasting model is ideal in order to evaluate the contribution of financial data's timeliness. As in the review of other methodologies, the authors find that surveys are important to forecast precision and that financial variables are not. More data, released throughout the quarter, is again useful in producing more accurate forecasts. The economic index projection onto GDP and the S&P 500 shows that the nowcasting technique explains a large proportion of GDP variation but not much daily variation in the stock market. Stock market dynamics over longer time horizons, however, can be linked to the economic index and therefore macroeconomic activity.[4]

Atlanta and New York Federal Reserve Bank Nowcasts

Some of the local Federal Reserve Banks in the United States produce GDP nowcasts that can inform decisions made by the Federal Open Market Committee regarding policies on setting short-term interest rates. The forecasts are publicly available and are often cited in financial and business media to inform the public and market participants' expectations for economic growth. In this subsection we will outline the Atlanta and New York Fed's methodologies for nowcasting.

4. Marta Banbura et al., "Now-Casting and the Real Time Data Flow," European Central Bank Working Paper Series 1564 (July 2013).

The Atlanta Federal Reserve Bank produces a nowcast titled GDPNow that is widely followed and cited. Its methodology is similar to that used by the Bureau of Economic Analysis to produce official GDP. The BEA collects data in subcomponents of GDP and aggregates the subcomponents to arrive at a full measure of economic activity. The Atlanta Fed uses a factor model similar to that of Giannone, Reichlin, and Small (2008) to link granular data items to one of thirteen GDP components and then aggregates those subcomponents to produce a nowcast. When data is unavailable, the value itself is forecasted in a technique similar to that used in the aforementioned 2008 paper. The Atlanta Fed authors find that the methodology is slightly inferior to forecasts published by Blue Chip Economic Indicators and that net exports and private inventories are the subcomponents that contribute the most to forecast errors.[5]

The New York Federal Reserve Bank uses a different methodology to nowcast GDP growth. Their approach uses a dynamic factor model and a Kalman filter as in Banbura, Giannone, Modugno, and Reichlin (2013). A variety of data is included with subjects ranging from housing and construction to surveys to prices. The New York Fed approach differs from the Atlanta approach largely in that it takes in all relevant data into the same model, while the Atlanta Fed's GDPNow can be seen as more of an accounting approach with its aggregation of subcomponents.[6]

Beber, Brandt, and Luisi (2014)

Alessandro Beber, Michael Brandt, and Maurisio Luisi propose a simple technique to track economic activity in their paper, "Distilling the Macroeconomic News Flow." Their methodology involves ex-ante groupings of macroeconomic variables into inflation, employment, output, and sentiment. Data is gathered and time stamped according to their actual release date, individual releases are then forward filled to account for days between releases, and each item is z-scored using the data up to the day in question. The first principal component of each group is then taken, following a Newey-West adjustment of the correlation matrix to account for the strong autocorrelations generated by the forward filling procedure. The first principal component weights are then used to aggregate the group's data items, producing a data

5. "GDPNow," Federal Reserve Bank of Atlanta, 2015, accessed December 2016, https://frbatlanta.org/cqer/research/gdpnow.
6. "Nowcasting Report," Federal Reserve Bank of New York, December 2, 2016, accessed December 10, 2016, https://www.newyorkfed.org/medialibrary/media/research/policy/nowcast/nowcast_2016_1202.pdf.

point for the index on that day. In their treatment, the employment and output groups are combined to represent total output, and the sentiment index is residualized against the total output index such that the innovations represent the part of forward looking sentiment independent of ex-post realized output. Bloomberg economist surveys are also used to create a measure of uncertainty around each index. The standard deviation of forecasts across economists is gathered for each data item, and the same principal component weights derived from the raw data are applied to the measures of economist disagreement. The authors show that the sentiment index leads the output index and predicts turning most points in the business cycle, defined by US GDP. While the methodology is not designed to specifically predict GDP growth numbers, the concept is similar to nowcasting in that the technique tracks economic activity on a daily basis, as new data is released and incorporated into updated estimates for each economic category.[7]

Data

This study uses only economic data to predict an economic outcome. While it is certainly possible to incorporate non-economic data, such as financial market data, our goal is to use more granular indicators related to economic growth to predict the surprise to the aggregate growth announcement. We compile the economic indicators used in "Distilling the Macroeconomic News Flow" by Beber, Brandt, and Luisi for the United States from Bloomberg. There are forty-three indicators that are categorized by the economic factors inflation, employment, output, and sentiment (also referred to as anticipated). Inflation indicators include the Consumer Price Index of Urban Consumers and Personal Consumption Expenditure excluding Food and Energy. Employment includes Unemployment and Nonfarm Payrolls, Durable Goods New Orders and Industrial Production are under output, and the University of Michigan Survey of Consumer Confidence and the ISM Milwaukee Purchasers Manufacturing Index are in the sentiment category. The target value is surprises to annualized quarter over quarter growth in GDP measured in 2009 dollars, where the expectation of changes in GDP is defined as the median forecast of the economists surveyed by Bloomberg. A surprise is the difference between the

7. Alessandro Beber, Michael Brandt, and Maurizio Luisi, "Distilling the Macroeconomic News Flow," Journal of Financial Economics 117, no. 3 (September 2015).

median forecast and the initial announcement of real GDP. A positive surprise is coded as a one while a negative surprise is coded as a zero. Instances with no surprise are initially coded as ones. All data, including GDP growth, is seasonally adjusted.

Bloomberg Consensus

Bloomberg makes available what it calls a consensus estimate for many data items that are released. These include GDP, many of the other economic data items included in this study, as well as company fundamentals like earnings and dividends. To arrive at the consensus for GDP growth, Bloomberg surveys fifty economists for their forecasts of GDP growth for the upcoming announcement. These economists are professionals who typically work at banks, funds, or economic consultancies. While they may not be able to perfectly forecast GDP, these economist forecasts represent some of the best estimates in the private sector. Considering that both their reputation and professional careers depend on the forecasts they make, it is reasonable to assume that the forecasts are made with a fair amount of care and study. The Bloomberg survey results are typically released about a week before the actual data announcement, and the results are often widely published in financial and business media, likely shaping the expectations of the audience. The median forecast from the survey is typically referred to as the consensus of the economists prior to the announcement, and we use this figure as a proxy for GDP growth expectations.[8]

Releases and Frequency

We choose to work only with initial release data rather than revised data in order to have a more realistic sense of the information available at each point in time. As revisions occur about a month after an initial economic data release, and as there are multiple revisions to the same data item, using final revised data as an indication of the state of the economy in the period referenced can be misleading. Particularly when making predictions about the future state of the economy, using only data available at the time of the prediction is critical to replicating a real-time scenario. In this spirit, we use only initial release data, and additionally we exclude indicators that are significantly revised historically in order to minimize the impact revisions may have had on the analysis. This cuts the data set to seventeen variables, including the target.

8. Bloomberg LP, accessed September, 2016.

The frequency of data releases differs across indicators, ranging from quarterly to weekly. GDP is the least frequently released data item, at a quarterly rate, and therefore the more frequently reported variables are adjusted to reflect quarterly percent changes to match the format of and time period referenced by GDP growth. The important distinction between a stock and a flow variable is highlighted at this stage. The Consumer Price Index, for example, is a stock variable. The index represents the relative price of a basket of goods over time and its next release is the change in the index over the last month plus the previous level of the index. Durable Goods New Orders, on the other hand, is a flow. This variable measures the value of durable goods ordered during the period it is referencing, not as it accumulates over time. While there may be a statistical relationship over time, each data point of this indicator is reported separately from the others. The distinction between stocks and flows becomes particularly relevant when calculating rates of change, especially over periods longer than the release frequency of the variable. If a flow variable is reported monthly and we are calculating the change in the variable over the first quarter of a year, we may typically take the difference between the end of December and end of March numbers. This calculation, however, would ignore the flows that occurred in January and February. It is possible that in those two months, there were zero new orders, for instance, making the first quarter new orders much lower than the orders made in the last quarter of the previous, normal year. This reality, however, would not have been picked up by the usual differencing method. Therefore, for all stock variables, we take the normal quarterly difference, and for all flow variables, we convert the flows to stocks by aggregating the flows over the sample period and then take the quarterly difference. This leaves all data items in a consistent, quarterly percent change. All data is collected from December 1996 to June 2016, as Bloomberg economist forecasts for GDP are available only from 1996 onwards.

Methodology

Surprises versus Growth

Perhaps the most significant difference between this study and the typical nowcasting practice is the difference in the target value. We aim to predict a surprise to GDP growth, while most nowcasting methodologies are concerned with predicting the level of GDP growth.

Although both have their respective uses, the task of predicting a surprise is more daunting than predicting a level in terms of achieving accuracy. To illustrate this point, let us consider the example of a random walk, which can be summarized by the following expression.

$$dGDP_t = \mu dt + \sigma dB_t$$

Here, we assume that GDP follows a Brownian motion path where μ represents the trend and σ represents the volatility of the path. dB incorporates a stochastic element in the path, where realizations of randomness are scaled by sigma such that the distribution of steps in the path is known but the realizations are random. Assuming that the random element is unpredictable and centered around zero, the best estimate of the next period's GDP would be last period's GDP plus the trend. The surprise, on the other hand, is the realization of the random element scaled by volatility. In this illustrative example, predicting the level within a reasonable range is relatively easy, but consistently predicting the direction of the surprise is extremely difficult. In exploring the predictability of surprises to GDP growth, we are addressing the difficult task of finding a systematic or nonrandom error made by the consensus.

Frequency of Predictions

Another difference between this study and the typical nowcasting exercise is the difference in frequency of predictions. Part of the appeal of nowcasting is to have constant predictions to track the state of the economy in real time. An oversight when it comes to reviewing the results of these models, those that predict GDP specifically, is that the model can only be trained on quarterly data because GDP is only released at a quarterly rate. Therefore, even if the predictor variables have data available daily, the model can only estimate its parameters or train its classification rules targeting those days that GDP is announced. Once those specific days have been predicted, the model is simply fitting values using parameters that have been estimated and the daily predictor values to produce daily "predictions." Typically, this results in predictions that essentially interpolate between the quarterly announcement day predictions.

This interpolation is not necessarily useful for our purposes of predicting surprises to GDP growth. Our focus is only on the announcement dates, as the surprise will only be realized in one way or the other at the time of the announcement. Moreover, daily predictions do not fit our purposes from a practical point of view, as it seems only reasonable to be interested in

predicting a surprise once the consensus estimate is released, which typically occurs just a week before the actual announcement. While the daily nature of nowcasting GDP growth levels may provide some additional information to policy makers or investors to shape their view of the economy, it does not provide additional value for the purposes of predicting surprises.

Aggregation

In order to conserve degrees of freedom, the individual indicators are aggregated according to economic category. As the quarterly release of GDP leaves just seventy-five dates upon which to train and test the model, we aim to preserve degrees of freedom by following an aggregation method similar to that proposed in Beber et al. The indicators are organized into the subcategories described above: inflation, employment, output, and sentiment (anticipated). The data is normalized on a telescoping basis by dividing by the standard deviation of the sample until the period in question. The first principal component is taken for each subcategory, and the product of the principal component weights are taken with their respective data points to create the category index value for the quarter. One of the pitfalls of principal component analysis is that estimates of the relationship between variables can be unstable over time. To address this concern, the first estimate of the principal component weights uses the first twelve quarters of data. This is done in order to avoid very noisy estimates of correlations from periods two to eleven. After taking the first estimate with twelve data points, we aim to further prevent instability by restricting the sign of the weight with the greatest magnitude to stay the same as its previous sign. This is done in order to minimize the chance that a single estimate can distort the index time series.

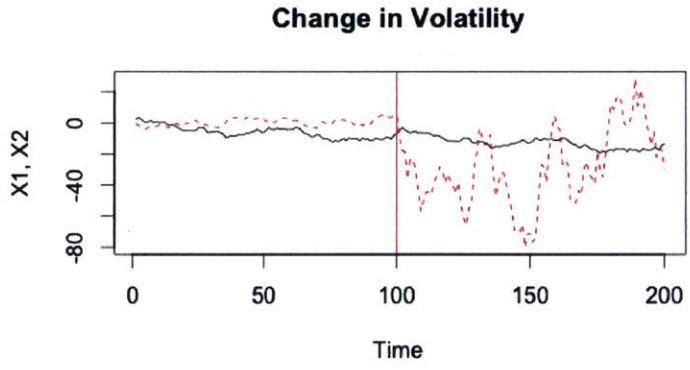
Data Snooping

Care is taken to avoid data snooping when estimating the relationship between the individual variables that make up a subcategory. Principal components for a specific quarter are estimated using the full sample only up to, and not beyond, the quarter for which the index value is being generated. It is important to distinguish this practice from using the full sample to estimate weights and applying the same weights across time, especially for the purpose of prediction. While it may be the case that the full sample weights capture a closer version of the true relationship between the variables over time, the information captured in the full sample

weights was not available in any period prior to the last. Therefore, applying those weights to prior data and training a model with an index formulated this way will not replicate a realistic point-in-time prediction and may lead to misleading conclusions.

This form of data snooping is a common mistake when principal components are involved. To illustrate the importance of following our procedure to avoid data snooping, we detail the following examples. The first illustrates a break in the volatility of a series, and the second illustrates a break in the trend of a series. We start with two time series that follow a path determined by draws from a normal distribution with zero trend and a volatility of one. There are two hundred total time periods. After one hundred time steps, the second series experiences a shock and its volatility jumps up to a new volatility for the rest of the sample, still with zero trend. The other series remains with zero trend and volatility of one. The chart below shows the two series, X1 and X2, where X2 jumps to five standard deviations volatility after time period one hundred.

Figure 1

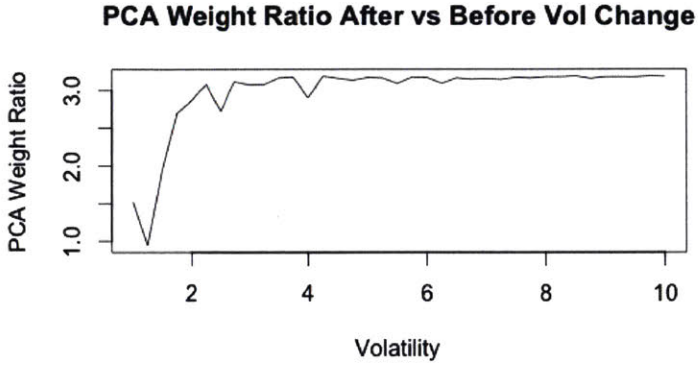


Source: Author's Calculations

We take the first principal component of the pair of series over two samples: up to the hundredth time step and over the full sample. Our purpose is to compare the two weights generated for the changing series, using data before and after the change, to see if there is a significant difference in weights that would be applied to data to aggregate the two series. We also aim to illustrate the sensitivity of the principal component weight to the change in volatility from the one standard deviation benchmark. To do so, we shift the volatility in the period after time one hundred starting at one standard deviation and increasing by increments of 0.25 up to

10 standard deviations. The ratio of the magnitudes of the half and full sample PCA weights are reported and shown in the chart below for comparison.

Figure 2

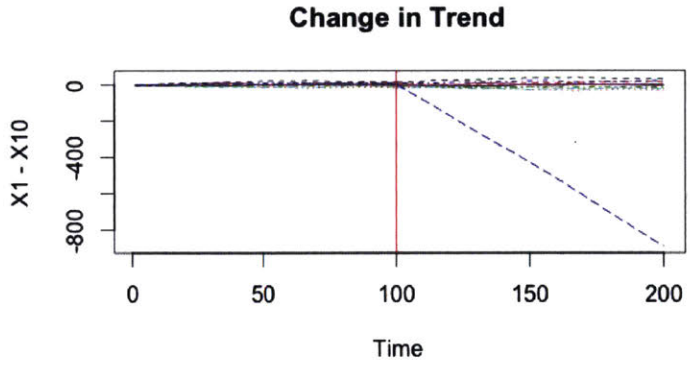


Source: Author's Calculations

While the random nature of the series path does create some deviations, we see a fairly clear relationship here between change in volatility and principal component weight ratio. As the volatility increases in the subsample after the change, the ratio of weights increases. In fact, if the volatility doubles from the benchmark it appears that from that point and higher the ratio converges to about three times the original weight. Clearly, weights calculated using information from the full sample differ from those calculated up to the point in time. This result is what we expect, as principal component analysis is attempting to explain as much variance as possible in the series analyzed. If one series suddenly experiences an increase in volatility, that series will be weighted more heavily in order to account for the overall variance experienced in the group of series.

We observe similar results in the trend change example. This time we start with ten series, X1 through X10. All ten series start as paths of the same benchmark, draws from a normal distribution with zero mean and volatility of one. This time, at period one hundred, X10 experiences a change that causes only its trend to move from zero to negative two. This shock can be interpreted as similar to a crisis expressed with significant negative change in only one series. The series are plotted below.

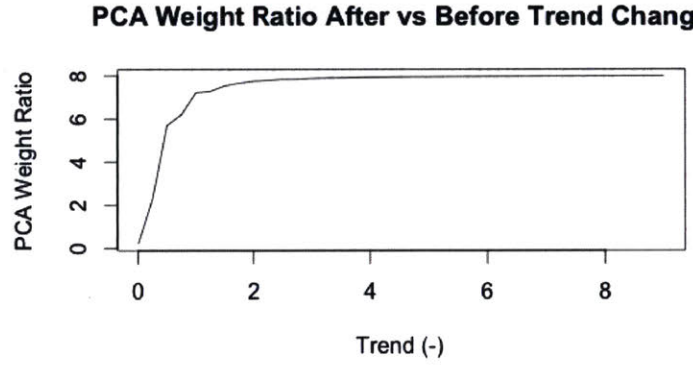
Figure 3



Source: Author's Calculations

We perform the same principal component analysis, capturing weights using data up to the one hundred time mark and using the full sample of data. We again compare the magnitude of the full and half sample weight on X10, the variable experiencing change, to illustrate the potential change in weights used in aggregation. The post change trend is varied in increments of -0.25 from 0 to -9 to record the sensitivity of the PCA weight on the change variable.

Figure 4



Source: Author's Calculations

As the trend diverges more negatively from the standard of zero, the PCA weight on X10 increases in magnitude. We see a similar relationship here to the volatility example, with the ratio converging rather quickly to eight times in this case. It should also be noted that while the weight on X10 is increasing, it necessarily means that the weights associated with the other series are decreasing in magnitude. The sum of squares of PCA weights add to one, so an

increase in magnitude of at least one weight means the others must compensate by decreasing in magnitude.

While it is certainly possible that a change in trend can be interpreted as a change in volatility, we illustrate both examples so as to be thorough in showing the potential effects of using varying samples in PCA weight estimation. If any sort of change occurs from one period to another, which is highly likely, using weights estimated from future data will not provide an accurate representation of weights that would have been known on that day, point-in-time. This inaccurate representation of point-in-time knowledge will likely make in sample predictions deceptively accurate and will therefore provide results that can be interpreted overly optimistically.

Models

With four economic indices constructed to summarize ex-post behavior and forward looking sentiment in the economy, we now use this information to contemporaneously predict surprises to aggregate growth. We assume that the more frequently reported indicators are available before the initial release of GDP. This assumption does not necessarily always hold, but it is generally fair considering that GDP has the longest lag from reference period to announcement date, about one month, of any of the variables. This study examines and compares four different techniques in their ability to predict surprises to GDP: ordinary least squares, logistic regression, random forest, and neural network. The following sections will describe each technique and the way the data is utilized in each analysis. All tests leave twenty quarters to evaluate out of sample performance, and each model is retrained every quarter before making a prediction of the next period's surprise.

Ordinary Least Squares

Ordinary least squares is known as a standard linear regression model and is commonly used in econometric analysis. It estimates a linear relationship between independent, explanatory variables and a dependent, outcome variable. The parameters of the linear model are estimated such that they minimize the in sample mean squared error. A typical model can be denoted as follows:

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \varepsilon_i$$

The objective function to be minimized, the mean squared error of the in sample data, is then written as denoted below, where b_0, b_1, and b_2 are estimates of the true model's betas as seen above.

$$MSE = \sum_i (y_i - (b_0 + b_1 x_{1i} + b_2 x_{2i}))^2$$

Given the linear structure imposed by the model, an interpretable and useful estimation relies on a few assumptions about the true nature of the variables included in the regression. First is the issue of specification. The explanatory variables included in the model must be independent of the residual in order for their coefficients to be unbiased and for the model to reflect a true relationship. The matrix of variables must also have full rank. When variables are multicolinear, or highly correlated with at least one other variable, the interpretation of the model can be compromised and the estimates can be unstable. Finally, OLS assumes homoscedastic errors with finite variance. The errors of the model should have mean zero and should be randomly distributed around the mean. A pattern or clustering in the residuals would suggest that this assumption does not hold.[9]

The advantage of ordinary least squares is that, under the above assumptions, its estimates are unbiased and consistent. A drawback of the linear model is that it assumes and imposes a relatively strict structure of the relationships in the data. In this analysis, we run the binary surprise variable against the four economic indices, inflation, employment, output, and sentiment, as well as a one period lag of each index to account for potential lagged relationships. The binary nature of the dependent variable means that the coefficients corresponding to each economic index can be interpreted as the additional probability of a positive surprise given a one unit increase in the index. OLS can produce predictions of probabilities that are outside of the range of zero to one, and therefore each prediction is rounded to the closest binary outcome. Given the relatively high correlation between output and sentiment, we expect some colinearity issues with regards to those variables. Otherwise, the main issue to be addressed is whether or

9. Roberto Rigobon. "Linear Regression." (lecture, Metrics for Managers MIT Sloan, Cambridge, MA, September, 2016).

not surprises follow a linear structure in the data, and this can be assessed when interpreting the results of the predictive test.[10]

Logistic Regression

Logistic regression is similar to linear regression, except that for the relevant case of the binary dependent variable, it produces interpretable probabilities that are bounded from zero to one. Additionally, the function, shown below, assumes diminishing returns to changes in explanatory variables as probabilities get closer to zero or one. The same increase in the independent variable will result in a larger change in the output when the output is closer to one-half than when it is closer to the extremes of zero or one. This is achieved by modifying the log of the linear function with the logit transformation, $\log\left(\frac{p}{1-p}\right)$.[11]

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 x$$

Which can be rewritten as:

$$p(x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}} = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$$

We use the same eight explanatory variables, the inflation, employment, output, and sentiment indices and their one period lags to predict GDP surprises. With the predictions bounded by zero and one, predictions above one-half are rounded up to positive surprises and those below one-half are rounded down to negative surprises.

Random Forest

A random forest is a compilation of many decision trees. Here we are using a classification style decision tree in order to predict a binary outcome variable, rather than a continuous number. These two styles of decision trees work very similarly in that they partition the data into two groups at each decision point. At each node, there is a yes or no decision. For example, is x greater than 5, yes or no? Then the data is partitioned according to the answer. The explanatory variable that can account for the greatest separation of the data is selected first and

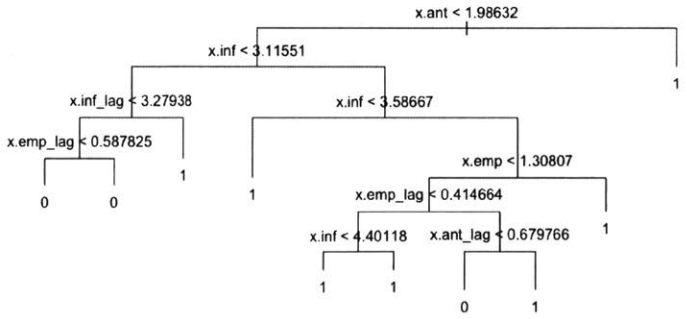
10. Cosma Shalizi, Advanced Data Analysis from an Elementary Point of View (223-236: Cambridge University Press, 2015).
11. Ibid.

the data is then partitioned further and further along additional explanatory variables. The mean value of the separated bucket of data is the model's prediction for that smaller bucket. A decision tree with too many partitions is prone to overfitting, and this could cause the model to perform poorly when it comes to out of sample predictions as it has been trained too closely to the in sample data. Therefore, a limit to the number of variables and decision nodes is wise when out of sample prediction is of primary interest.

The random forest methodology seeks to overcome the issue of overfitting without pruning the tree, or limiting the number of partitions allowed, by building many trees for multiple, random subsets of data. The results of the trees are then averaged in order to reduce the variance of the prediction. Additionally, the random forest chooses the variable according to which to split the data on at each node from a random subsample of the variables. Therefore, the same variables are not available at the same nodes of each tree. With enough random trees, overfitting the in-sample data is ideally not an issue.[12]

Again, we use the same four economic indices and their lags, this time to classify rather than to regress against surprises to GDP. Five hundred trees are utilized for the estimation, and the results of those trees are averaged to produce a prediction for that quarter. The output of this model is simply a one or a zero. Below is a presentation of a sample tree from a forest that is run in this analysis. At each node the name of the variable to split on is shown along with the splitting criteria.

Figure 5



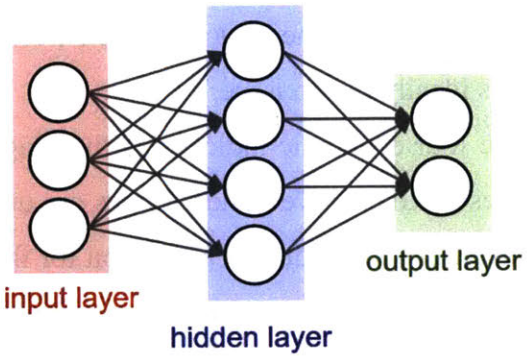
Source: Author's Calculations

12. Andrew Tiffin, "Seeing in the Dark: A Machine-Learning Approach to Nowcasting in Lebanon," International Monetary Fund March 2016.

Neural Network

Neural networks are algorithms designed to emulate neurons in the human brain. Nodes are arranged in layers, as shown below, that recreate a simple version of sets of neurons in the brain. Signals are passed in our brains by being input through dendrites, processed in cell bodies, output of the cells through an axon, and then passed along to the next set of neurons through synapses. In the algorithmic approximation of this complex system, the input layer represents the dendrites as they receive the initial signals without modification, the arrows from an input node to a hidden layer node represent synapses and dendrites of the next layer of cell bodies, and the arrows going to the output layer represent axons, which contain the final signal.

Figure 6



Source: Stanford CS231n Course Materials[13]

As in nature, nodes in a middle, or hidden, layer require that an input signal reach a certain threshold before being sent along to the next layer or being output. In the neural network algorithm, input signals are weighted along each path from the input layer to the hidden layer, mimicking the interaction between synapses and dendrites of the hidden layer, and the weighted signals received from all input nodes are then summed in each node of the hidden layer. If the summed signal reaches the threshold, it is sent along to the next layer. Most commonly in algorithms, however, activation functions are used rather than hard cutoffs. A few examples of activation functions include the sigmoid and tanh functions, both of which are similar to the logit function in that they are non-linear, bound their outputs, and exhibit diminishing returns at the extremes.

13. "CS231n Convolutional Neural Networks for Visual Recognition," accessed October 2016, http://cs231n.github.io/neural-networks-1/.

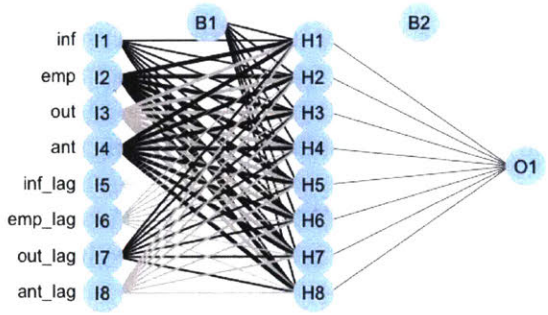
After determining the number of hidden layers and nodes per layer desired in the network, the main remaining parameters to be determined are the weights on each connection from node to node. Weights are set at a predetermined or random starting point, and from there the model "learns" from the data. As the network processes data for each quarter, the output is judged against the known outcome for that quarter. The error, or the difference between the model output and the known outcome, is then backpropagated across each node connection, and each connection's contribution to the error is identified with regard to the gradient of the loss function. The weights are then adjusted accordingly to reduce the error, or loss, that would have occurred, but with a constraint such that the model does not perfectly fit the sample data. With each subset of data that is passed through the network, the model ideally fits the true nature of the data more closely, thus learning the relationships of the inputs with one another and the outputs over time.[14]

As with the previous models, the same economic indices and their one-quarter lags are used as inputs. We choose one hidden layer with eight nodes to match the number of inputs, and we set the starting weights to 0.01 at each connection to lessen the instability in outcomes over various iterations. The maximum iterations are increased to two thousand and the fit criterion is set at 1×10^{-5} in order to help ensure consistent convergence of the model. Finally, while it is possible to constrict the ways in which nodes can connect to one another, we allow all input nodes to send signals to each node in the hidden layer. The output for this model is a categorical variable, so the output is either zero or one.

Below is a visualization of the neural network run in the analysis. The eight variables used as inputs are shown as the initial nodes. Each input node is connected to each of the eight nodes in the hidden layer, with no skip-layer connections, and each hidden node is connected to the final output node, where the final signal and prediction is processed and made.

14. Ibid.

Figure 7



Source: Author's Calculations

Results

Having trained and then tested the models over the last twenty quarters of the data, retraining at each period, we compile the results in confusion matrices to summarize the hits and misses in predicting surprises. The matrices are presented below with results for each method. Prediction refers to the type of surprise predicted by the model, reference refers to the actual outcome, and the numbers in the matrices are the number of hits or misses by surprise type. In sample and out of sample results are presented for comparison, where in sample results are those from the model trained on the first out of sample period.

Binomial Outcome with No Surprise as 1

We begin with our original treatment of the target values, which is to code surprises to GDP as zeros for negative surprises and ones for positive surprises, coding instances where GDP came in line with the consensus as a one. We present both the confusion matrices and a hit or miss table with hit and miss rates. The rate we use for evaluation is simply the number of hits or misses divided by the total number of predictions made in the out of sample test, in this case twenty.

Table 1

Binomial Outcome Out of Sample (No Surp as 1)						
OLS	Reference		**OLS**			
Prediction	0	1		Hit	Miss	
0	0	0	Number	11	9	
1	9	11	Rate	0.55	0.45	
Logit	Reference		**Logit**			
Prediction	0	1		Hit	Miss	
0	0	0	Number	11	9	
1	9	11	Rate	0.55	0.45	
RF	Reference		**RF**			
Prediction	0	1		Hit	Miss	
0	3	3	Number	11	9	
1	6	8	Rate	0.55	0.45	
NN	Reference		**NN**			
Prediction	0	1		Hit	Miss	
0	4	5	Number	10	10	
1	5	6	Rate	0.50	0.50	

Table 2

Binomial Outcome In Sample (No Surp as 1)						
OLS	Reference		**OLS**			
Prediction	0	1		Hit	Miss	
0	6	5	Number	32	15	
1	10	26	Rate	0.68	0.32	
Logit	Reference		**Logit**			
Prediction	0	1		Hit	Miss	
0	6	6	Number	31	16	
1	10	25	Rate	0.66	0.34	
RF	Reference		**RF**			
Prediction	0	1		Hit	Miss	
0	1	4	Number	28	19	
1	15	27	Rate	0.60	0.40	
NN	Reference		**NN**			
Prediction	0	1		Hit	Miss	
0	16	14	Number	33	14	
1	0	17	Rate	0.70	0.30	

As seen in the hit or miss section of the out of sample results, the four methodologies perform similarly in the out of sample test, according to the simple rate calculation, with the neural network slightly underperforming, at a fifty percent hit rate. A closer look at the results, however, suggests that there are indeed some differences between the more linear OLS and logistic regressions and the machine learning techniques that allow for nonlinearities and interactions. While both OLS and logistic regression match the random forest's fifty-five percent

hit rate, the regression models did so by only predicting positive surprises. Out of the eleven positive surprises in the twenty out of sample quarters, the regression models get all eleven correct. They fail, however, to predict any negative surprises, missing all nine. The random forest, which achieves the same hit rate, predicts eight out of eleven positive surprises and three negative surprises. In distinguishing between the random forest and regression methods, it appears that the random forest produces more robust predictions. It is noteworthy that here the linear regression and logistic regression do not distinguish themselves from one another.

The in sample results, on the other hand, suggest that the neural network had the best fit, followed by OLS, logistic regression, and finally the random forest. In contrast to the out of sample results, the random forest predicts almost all positive surprises, in forty-two out of forty-seven quarters, while the econometric models are more modest in their positive surprise predictions with thirty-two and thirty-one out of the forty-seven possible. The fact that the random forest had the worst initial in sample but arguably the best out of sample performance goes to show that in sample fit cannot solely be used to judge the ability of a model to make predictions.

Overall, an out of sample performance of anything better than a random prediction, a fifty percent hit rate, is a significant result. As stated previously, the task of predicting a surprise is a difficult one, as the aim is to find a systematic signal in a process subject to, by nature, a high degree of randomness.

Binomial Outcome with No Surprise as 0

It is possible that our treatment of no surprises, when GDP growth comes in at the consensus estimate, may have a significant impact on the way models are trained and, most importantly, their out of sample predictions. As there are thirteen instances of the consensus estimate being correct, when there is no surprise, in the training set, it does seem plausible that assigning them to either the positive or negative surprise groups, which are almost even in the training set, may sway the models to over emphasize one type of surprise over the other. To test this conjecture, we run the analysis and the out of sample tests with no surprises coded as zeros, along with negative surprises, instead of as ones as done originally. The in and out of sample results are presented below.

Table 3

Binomial Outcome Out of Sample (No Surp as 0)					
OLS	Reference		**OLS**		
Prediction	0	1		Hit	Miss
0	6	3	Number	10	10
1	7	4	Rate	0.50	0.50
Logit	Reference		**Logit**		
Prediction	0	1		Hit	Miss
0	4	3	Number	8	12
1	9	4	Rate	0.40	0.60
RF	Reference		**RF**		
Prediction	0	1		Hit	Miss
0	12	7	Number	12	8
1	1	0	Rate	0.60	0.40
NN	Reference		**NN**		
Prediction	0	1		Hit	Miss
0	7	6	Number	8	12
1	6	1	Rate	0.40	0.60

Table 4

Binomial Outcome In Sample (No Surp as 0)					
OLS	Reference		**OLS**		
Prediction	0	1		Hit	Miss
0	26	8	Number	36	11
1	3	10	Rate	0.77	0.23
Logit	Reference		**Logit**		
Prediction	0	1		Hit	Miss
0	24	8	Number	34	13
1	5	10	Rate	0.72	0.28
RF	Reference		**RF**		
Prediction	0	1		Hit	Miss
0	22	10	Number	30	17
1	7	8	Rate	0.64	0.36
NN	Reference		**NN**		
Prediction	0	1		Hit	Miss
0	22	0	Number	40	7
1	7	18	Rate	0.85	0.15

With this specification of no surprises as zeros, all models perform worse according to hit rate in the out of sample tests, except the random forest which improves its hit rate by five percent. Of note as well, however, is that the random forest makes predictions with this specification in a similar fashion to the econometric models in the original specification of no surprises, predicting a negative surprise in nineteen out of twenty possible quarters. The OLS and logistic models, in this instance of the test switch to predicting only zeros. Their predictions are

more balanced, but neither succeeds in exceeding the fifty percent benchmark. The OLS matches the fifty percent threshold, while the logistic regression performs poorly with a forty percent hit rate. The neural network has the worst performance out of the four models in both specifications of the binomial outcome tests.

The in sample results are rather surprising as well, as all samples show a better fit within the training set in this specification of the outcomes than in the previous analysis. Again, the neural network shows the best fit with an eighty five percent hit rate. As with the previous specification, OLS ranks second in sample, followed by logistic regression. Random forest has the lowest in sample fit of sixty-four percent, slightly higher than its previous in sample fit of sixty percent. Once again, the random forest is the least closely fit model in the training set while achieving the best out of sample performance.

Given that the performance of three of the four models changed dramatically depending on how no surprises were defined, and that the random forest's strong performance was not robust in the second specification, given its over-prediction of negative surprises, we conclude that the treatment of no surprise outcomes does significantly alter the models' interpretation of the training data and predictions.

Trinomial Outcome

Seeing as the models' predictions vary dramatically based on our treatment of no surprises and because our goal in this study is to assess the four models' ability to predict instances when GDP growth will not be reported in line with expectations, we conclude that the binomial outcome model is not ideal for our purpose. Thus, we leave the no surprise outcomes as they are and train and test a model with three outcomes, negative one for a negative surprise, zero for no surprise, and one for a positive surprise. We again use the same four models, with the logistic regression modified to handle a trinomial outcome variable rather than a binomial outcome. Holding the same twenty quarters out of sample, we train, retrain, and test the models on their out of sample predictions. Both the in and out of sample results of the trinomial tests are presented.

Given the three outcomes of the test, the confusion matrix takes on a three by three shape rather than the two by two case of the binomial results. Our interpretation of the results changes slightly as well. Whereas previously, in the binomial case, we defined hits as the diagonal with

the top left and bottom right cells and misses as the off diagonal cells, the trinomial results have correct predictions along the same left to right diagonal, and incorrect predictions in all other cells. For our purpose, however, we define hits as the top left and bottom right cells and misses as the bottom left and top right cells. We are primarily interested in what we call "big" hits and "big" misses. Or, put differently, cases when the realized surprise is the same or the opposite of the prediction and non-zero.

The following is an example from a financial markets context to illustrate the choice in model evaluation. We assume that if we predict a positive surprise, financial markets are not expecting aggregate economic growth to be as high as it will be. Therefore, we extend this logic to assume that the price seen in the market at the time of our prediction does not incorporate the information that we have based on our prediction, as the market should reflect information about the expectation of GDP, among other things, which we are using the Bloomberg consensus as a proxy for. In other words, the market has not priced in the surprise we have predicted. If our model can predict surprises to GDP at a rate better than a random coin flip, we conclude that our model provides an advantage over the market, and we buy a financial asset expected to increase in value when growth is high, at its current price. If GDP does in fact surprise to the upside, we would expect the asset's price to increase, and we will collect a positive return from the trade. This is a big hit. In an idealized world of no trading cost or other market frictions, if we follow the signal of our predictive model and buy the asset expecting a positive surprise but GDP growth is announced with no surprise, then we expect prices to not change based on this news, as the market price should have already reflected that public consensus information. In this ideal case, we should be able to exit our position with no gain and no loss. Thus, if we enter a trade and the eventual outcome is no surprise, we are not too concerned from the portfolio's perspective, as we did not make a loss. This case is a miss of little or no consequence. If on the other hand we have gone long on the asset and GDP is announced with a negative surprise, we would expect the asset price to fall and our trade to result in a loss. This is a big miss as we entered a position expecting the opposite of the realized outcome and our trade resulted in a loss.

Similar but opposite logic holds for a negative surprise prediction. With a negative surprise predicted, we would enter into a short position on the same financial asset. In the case of actual GDP growth surprising to the downside, we expect to make a positive return, a big hit. We expect a negative return if growth surprises to the upside, a big miss, and, assuming little to no

short interest, we expect zero return for growth reported at the consensus estimate. In the case of no surprise predicted, we would stay out of the market. Arguably, there is an opportunity cost associated with missing opportunities for positive returns and we would be incurring that cost by not making a trade, but for our purposes, we assume that zero return for zero risk taken is an acceptable outcome.

Table 5

Trinomial Outcome Out of Sample

OLS	Reference			**OLS**		
Prediction	-1	0	1		Hit	Miss
-1	0	0	0	Number	2	2
0	7	2	5	Rate	0.10	0.10
1	2	2	2	Score	-0.2	

Logit	Reference			**Logit**		
Prediction	-1	0	1		Hit	Miss
-1	3	0	3	Number	7	9
0	0	0	0	Rate	0.35	0.45
1	6	4	4	Score	-2.9	

RF	Reference			**RF**		
Prediction	-1	0	1		Hit	Miss
-1	6	1	5	Number	8	8
0	0	1	0	Rate	0.40	0.40
1	3	2	2	Score	-0.8	

NN	Reference			**NN**		
Prediction	-1	0	1		Hit	Miss
-1	2	1	2	Number	7	5
0	4	1	0	Rate	0.35	0.25
1	3	2	5	Score	1.5	

Table 6

Trinomial Outcome In Sample

OLS	Reference			**OLS**		
Prediction	-1	0	1		Hit	Miss
-1	3	2	0	Number	8	0
0	13	8	13	Rate	0.17	0.00
1	0	3	5	Score	8	

Logit	Reference			**Logit**		
Prediction	-1	0	1		Hit	Miss
-1	11	7	2	Number	24	6
0	1	3	3	Rate	0.51	0.13
1	4	3	13	Score	17.4	

RF	Reference			**RF**		
Prediction	-1	0	1		Hit	Miss
-1	4	6	5	Number	13	11
0	6	1	4	Rate	0.28	0.23
1	6	6	9	Score	0.9	

NN	Reference			**NN**		
Prediction	-1	0	1		Hit	Miss
-1	16	13	15	Number	19	15
0	0	0	0	Rate	0.40	0.32
1	0	0	3	Score	2.5	

The trinomial test results account for the presence of no surprise outcomes in both the training and test sets and therefore provide a better indication of whether surprises are predictable at all and which models may be better suited for prediction than others. In the trinomial case, we deem a hit or miss rate of respectively above or below about twenty-two percent as better than random. This is because we focus on only the "big" hits or misses and not the zero surprise predictions or outcomes. Therefore, rather than a one-half chance of making a hit randomly, as in the binomial case, we now have a one-third chance of a random hit, including zeros, in the trinomial case. Excluding zeros, we arrive at a two-ninths chance of a random big hit, which is approximately twenty-two percent.

In the out of sample test, none of the models achieve the ideal results of a hit rate above twenty-two percent and a miss rate below the same threshold. Three models, the logistic regression, random forest, and neural network, beat the random threshold on the hit rate, and one model, the OLS, beats the random threshold on the miss rate. The random forest achieves the most hits with eight out of twenty quarters. The neural network and logistic regression follow with seven each and OLS is last in terms of hits with two. The logit model makes the most misses with nine and the random forest follows with eight misses. The neural network makes five misses, a rate of twenty-five percent, and the OLS makes the fewest misses with two.

We use a simple score metric to compare results across models, as predictions are not binomial. We take a weighted difference between hits and misses for each model with hits weighted with 1 and misses weighted with -1.1. We overweight misses on the negative side because we assume, again from a portfolio perspective, that misses and drawdowns in portfolio value caused by negative returns result in more negative utility than the positive utility that results from positive returns. Given that we assume a concave utility function, models that make fewer total misses will be penalized less than those with the same net hits and misses but with more total misses.

Using the above scoring method, the neural network performs the best with a score of 1.5, OLS ranks second, scoring -0.2, random forest is third with a score of -0.8, and the trinomial logit ranks last, scoring -2.9. The in sample results once again differ rather dramatically from the out of sample results. In terms of the score metric, the models all perform positively and rank as follows from best to worst for in sample performance: trinomial logit with 17.4, OLS with 8,

neural network with 2.5, and finally random forest with 0.9. Clearly, the in sample performance in the trinomial case cannot be used to proxy for out of sample prediction ability.

Neural network is the only model that has a positive score, suggesting that according to our metric, it is the only model that has an out of sample result that would suggest that predicting surprises to growth is indeed possible, and is therefore also the best model to use for predicting surprises. Again, the neural network's hit rate of thirty-five percent beats the random benchmark of twenty-two percent and its miss rate almost matches the benchmark at twenty-five percent. Both rates suggest that the neural network performs decently well in predicting surprises to growth.

The machine learning techniques' advantage over the more standard econometric models, OLS and logistic regression, is their ability to incorporate non-linear relationships and interactions. Additionally, the form of the model is more flexible for both the random forest and neural network than it is for econometric techniques. This allows the form of the model, which non-linearities and interactions to include for example, to adapt to the data. Machine learning techniques perform better and about on par with the best performing econometric technique, OLS, depending on how the score metric is defined (two hits and two misses can be better or worse than eight hits and eight misses). This suggests that the machine learning advantages of allowing for non-linear relationships and interactions, as well as adaptive forms as data is processed, may help in predicting a surprise. This does seem to coincide with our intuition given the nature of a surprise. The outcome of a surprise is based largely on the consensus estimate going into the announcement. Assuming that this estimate is the expectation of economic growth in the quarter, the surprise is the error term. In attempting to find a systematic pattern in which errors are realized, there must be a systematic pattern in which economists surveyed by Bloomberg misestimate GDP growth. While we do not know the methodology used by each of the fifty economists, it is likely that they use econometric models to make growth forecasts. Thus, it would be expected, under that assumption, that an econometric model would not be able to predict a surprise to an estimate made using an econometric model itself. It is reasonable, and the results suggest, that there may be nonlinear relationships that are poorly estimated by economists because they are not economically intuitive or because their priors about the model form do not allow for them. The difficulty with using a machine learning algorithm is that the output and model's process is not easy to interpret, and is often difficult to map to an economic

intuition. This may dissuade economists from using these models to make forecasts, as they may not be able to explain their intuition to clients or internal users.

The two machine learning techniques, however, differ in performance as well. While both have the ability to incorporate non-linear relationships and interactions, they vary in how they are trained and therefore use the limited number of data points available to learn to different degrees. Random forests use a splitting function at each node to divide the data and therefore learn at each layer of the tree. Neural networks, on the other hand, have weight parameters that are adjusted, through backpropagation, based on the error associated with each data point in the training set. As the error is propagated back through the network, each weight associated with the error in the network is adjusted based on the gradient of the loss function. This adjustment based on in sample errors is not done in the random forest model, which relies more on bootstrapping the sample set to arrive at robust splitting decisions. In this case of limited training data and eight variables, the random forest can split the data in at most eight ways, and in each tree it limits itself to a subset of those eight at each node. The neural network operates differently and has many more parameters that are tuned, ninety-nine in this case. These aspects of the neural network allow it to perform what is called feature learning, where the model adapts quickly to the form of the data, better than the random forest by adjusting which weights and nodes are emphasized over others to fit the training set more closely, ideally without over fitting. It may be the more flexible and closely fit nature of the neural network that explains its outperformance of both the econometric models and the random forest in the trinomial case.

Conclusion

We build on the practice of nowcasting by extending it to predicting the surprise to GDP growth rather than the level, a more difficult task of identifying and predicting a systematic error. Our methodology deviates in some ways from the more typical approach to nowcasting, primarily because we target the innovation from expectation rather than the expectation of growth itself. We also focus only on predicting at a quarterly rate when consensus estimates are made and surprises are realized. The study involves the testing of four different statistical techniques, two of which are standard in econometrics and the other two of which are machine learning algorithms more commonly used in engineering contexts. Our tests reveal that firstly, the definition and treatment of no surprises matter in terms of training the models. Over

emphasizing positive or negative surprises artificially through no surprises, in the binomial case, changes the performance of the model out of sample significantly. The trinomial case is the appropriate context within which to judge the performance of the models in predicting surprises to GDP growth, and here we find that the neural network is the only model that suggests that predicting surprises may be possible. While the results are not overwhelmingly positive, they do suggest that the form of the neural network is worth further consideration and research in its ability to identify non-linear relationships that might not yet be mainstream in forming consensus expectations. We do not look closely at possible alterations to the neural net as our focus is on comparison across techniques, but an exploration of changes in the inputs and the structure of the hidden layers and node connections could shed further light on the relationships that the neural network appears to be identifying that other models are not. Overall, we conclude based on our results that machine learning algorithms do have a place in the economics context, particularly when it comes to prediction. Predicting surprises to announcements consistently is far from an easy or certain task but it does appear that it is possible given the current regime of consensus forming and with the appropriate tools to take advantage of systematic errors in the consensus.

Bibliography

Banbura, Marta, Domenico Giannone, Michele Modugno, and Lucrezia Reichlin. "Now-Casting and the Real Time Data Flow." European Central Bank Working Paper Series 1564 (July 2013).

Beber, Alessandro, Michael Brandt, and Maurizio Luisi. "Distilling the Macroeconomic News Flow." *Journal of Financial Economics* 117, no. 3 (September 2015): 489–507.

Breiman, Leo. *Random Forests*. January 2001.

"GDPNow." 2015. Accessed December 2016. https://frbatlanta.org/cqer/research/gdpnow.

Giannone, Domenico, Lucrezia Riechlin, and David Small. "Nowcasting: The Real-Time Informational Content of Macroeconomic Data." *Journal of Monetary Economics* May 22, 2008: 665–76.

Gurney, Kevin. *An Introduction to Neural Networks*. London: UCL Press, 2009.

Karpathy, Andrej. "CS231n Convolutional Neural Networks for Visual Recognition." Accessed October 2016. http://cs231n.github.io/neural-networks-1/.

"Nowcasting." February 14, 2011. Accessed December 6, 2016. http://www.metoffice.gov.uk/learning/science/hours-ahead/nowcasting.

"Nowcasting Report." December 2, 2016. Accessed December 10, 2016. https://www.newyorkfed.org/medialibrary/media/research/policy/nowcast/nowcast_2016_1202.pdf.

Raffinot, Thomas. "Can Macroeconomists Get Rich Nowcasting Economic Turning Points?" *SSRN Electronic Journal*. doi:10.2139/ssrn.2545256.

Rigobon, Roberto. "Linear Regression." Lecture for Metrics for Managers, MIT Sloan, September, 2016.

Shalizi, Cosma. *Advanced Data Analysis from an Elementary Point of View*. Cambridge University Press, 2015.

Tiffin, Andrew. "Seeing in the Dark: A Machine-Learning Approach to Nowcasting in Lebanon." *International Monetary Fund* March 2016.

US Macroeconomic Data. December, 1996 - June, 2016, via Bloomberg LP, accessed September, 2016.

Appendix

Data and Sources

Table 7

Category	Country	BB Ticker	Release Name	Units	Freq	Source
Inf	US	CPI INDX Index	CPI Urban Consumers SA	Value	M	Bureau Labor Statistics
Inf	US	CPUPAXFE Index	CPI Urban Consumers Less Food Energy	Value	M	Bureau Labor Statistics
Inf	US	PCE CUR$ Index	Personal Cons. Expenditure Ex Food & Energy Deflator SA	Value	M	Bureau Economic Analysis
Emp	US	NFP T Index	Employees on Nonfarm Payrolls Total SA	Volume	M	Bureau Labor Statistics
Emp	US	USMMMANU Index	Employees on Nonfarm Payrolls Manufacturing Industry	Volume	M	Bureau Labor Statistics
Emp	US	USURTOT Index	Unemployment Rate Total in Labor Force	Rate	M	Bureau Labor Statistics
Out	US	CCOSTOT Index	Federal Reserve Consumer Credit Outstanding Amount Total SA	Volume	M	Federal Reserve
Out	US	IP Index	Industrial Production SA	Volume	M	Federal Reserve
Out	US	DGNOTOT Index	Durable Goods New Orders Industries SA	Volume	M	U.S. Census Bureau
Out	US	DGNOXTRN Index	Durable Goods New Orders Ex Transp.	Volume	M	U.S. Census Bureau
Out	US	PITL Index	Personal Income SAAR	Rate	M	Bureau Economic Analysis
Out	US	PCE CUR$ Index	Personal Consumption Expend. Nominal Dollars	Rate	M	Bureau Economic Analysis
Sen	US	COMFCOMF Index	Bloomberg US Weekly Consumer Comfort Index	Value	W	Bloomberg
Sen	US	CONSSENT Index	University Michigan Survey Consumer Confidence	Value	M	U. of Michigan Survey Research
Sen	US	LEI TOTL Index	Conference Board US Leading Index Ten Econ Indicators	Value	M	Conference Board
Sen	US	MAPMINDX Index	ISM Milwaukee Purchasers Manufacturing Index	Value	M	NAPM - Milwaukee
Target	US	GDP CQOQ Index	GDP US Chained 2009 Dollars QoQ SAAR	Rate	Q	Bureau Economic Analysis

Data Summary

Table 8

	CPI Urban	CPI Core	Personal Consumption	Nonfarm Total
Mean	1.278	4.181	2.007	1.721
Std Dev	1.342	3.341	2.024	3.363
Median	1.241	3.389	1.74	0.838
Min	-4.366	-0.248	-3.712	-3.333
Max	5.17	18.652	15.215	14.204
	Nonfarm Manufacturing	Unemployment	Consumer Credit	IP
Mean	-0.372	0.027	1.674	0.464
Std Dev	1.47	1.173	1.25	1.34
Median	-0.146	-0.293	1.57	0.556
Min	-4.772	-1.591	-1.205	-4.225
Max	4.073	3.846	5.221	4.699
	Durable Goods Indust.	Durable Goods ex Trans.	Personal Income	Personal Consumption
Mean	0.112	0.147	1.561	2.007
Std Dev	1.15	1.098	1.663	2.024
Median	0.025	0.309	1.342	1.74
Min	-3.272	-4.11	-3.476	-3.712
Max	3.242	1.889	5.247	15.215
	BB Cons. Confidence	Umich Cons Confidence	LEI	ISM PMI
Mean	-0.02	0.133	0.27	0.08
Std Dev	1.081	1.465	1.232	0.902
Median	-0.283	0.087	0.463	0.062
Min	-2.744	-3.14	-3.995	-2.036
Max	2.303	5.164	3.536	3.422

Table 9

Surprise Type Count		
-1	0	1
30	17	28

Table 10

Correlations	Inf	Emp	Out	Sent
Inf	1			
Emp	0.273	1		
Out	0.350	0.700	1	
Sent	-0.120	0.141	0.275	1