Pontifícia Universidade Católica
do Rio de Janeiro

Henrique Fernandes Pires

Essays on nowcasting with high dimensional data

Tese de Doutorado

Dissertation presented to the Programa de Pós–graduação em Economia of PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Economia.

Advisor: Prof. Marcelo Cunha Medeiros

Rio de Janeiro
April 2022

Pontifícia Universidade Católica
do Rio de Janeiro

Henrique Fernandes Pires

Essays on nowcasting with high dimensional data

Dissertation presented to the Programa de Pós–graduação em Economia of PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Economia. Approved by the Examination Committee.

Prof. Marcelo Cunha Medeiros
Advisor
Departamento de Economia – PUC-Rio

Prof. Eduardo Zilberman
Departamento de Economia – PUC-Rio

Prof. Marcelo Fernandes
Departamento de Economia – FGV-EESP

Gabriel Vasconcelos
Departamento de pesquisa quantitativa – Bancom BBM

Prof. André Maranhão
Departamento de Pesquisa – Banco do Brasil S/A

Rio de Janeiro, April the 25th, 2022

Henrique Fernandes Pires

B.A. in Economics, University of Brasilia (UNB), 2016.

Bibliographic data

Fernandes Pires, Henrique

Essays on nowcasting with high dimensional data / Henrique Fernandes Pires; advisor: Marcelo Cunha Medeiros. – Rio de janeiro: PUC-Rio, Departamento de Economia, 2022.

v., 84 f: il. color. ; 30 cm

Tese (doutorado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Economia.

Inclui bibliografia

1. Economia – Teses. 2. Econometria – Teses. 3. Nowcasting;. 4. Aprendizado de máquina;. 5. Big data;. 6. Previsão;. 7. Modelos de alta dimensão;. 8. COVID-19. I. Cunha Medeiros, Marcelo. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Economia. III. Título.

CDD: 620.11

To Julia and Manuela

Acknowledgments

I thank my dear Julia for being the best partner someone could ever wish and for giving us the most beautiful little girl in the world.

I thank my family, for giving me love and courage throughout my entire life. Without them nothing of this would be possible.

I sincerely thank my advisor Marcelo Medeiros, for he truly paved the way for lessons I had to learn, far beyond the academic perspective.

I thank PUC-Rio and the Economics Department, specially Prof. Marcio Garcia and Gustavo Gonzaga for making my wish to study at MIT come true.

Finally, but not less important, the support from CAPES and CNPQ is gratefully acknowledged. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Abstract

Fernandes Pires, Henrique; Cunha Medeiros, Marcelo (Advisor). **Essays on nowcasting with high dimensional data**. Rio de Janeiro, 2022. 84p. Tese de doutorado – Departamento de Economia, Pontifícia Universidade Católica do Rio de Janeiro.

Nowcasting in economics is the prediction of the present, the recent past or even the prediction of the very near future of a certain indicator. Generally, a nowcast model is useful when the value of a target variable is released with a significant delay with respect to its reference period and/or when its value gets notably revised over time and stabilizes only after a while. In this thesis, we develop and analyze several Nowcasting methods using high-dimensional (big) data in different contexts: from the forecasting of economic series to the nowcast of COVID-19. In one of our studies, we compare the performance of different Machine Learning algorithms with more naive models in predicting many economic variables in real-time and we show that, most of the time, Machine Learning beats benchmark models. Then, in the rest of our exercises, we combine several nowcasting techniques with a big dataset (including high-frequency variables, such as Google Trends) in order to track the pandemic in Brazil, showing that we were able to nowcast the true numbers of deaths and cases way before they got available to everyone.

Keywords

Nowcasting; Machine Learning; Big Data; Forecasting; High-Dimensional models; COVID-19

Resumo

Fernandes Pires, Henrique; Cunha Medeiros, Marcelo. **Ensaios sobre Nowcasting com dados em alta dimensão**. Rio de Janeiro, 2022. 84p. Tese de Doutorado – Departamento de Economia, Pontifícia Universidade Católica do Rio de Janeiro.

Em economia, Nowcasting é a previsão do presente, do passado recente ou mesmo a previsão do futuro muito próximo de um determinado indicador. Geralmente, um modelo nowcast é útil quando o valor de uma variável de interesse é disponibilizado com um atraso significativo em relação ao seu período de referência e/ou sua realização inicial é notavelmente revisada ao longo do tempo, se estabilizando somente após um tempo. Nesta tese, desenvolvemos e analisamos vários métodos de Nowcasting usando dados de alta dimensão (big data) em diferentes contextos: desde a previsão de séries econômicas até o nowcast de óbitos pela COVID-19. Em um de nossos estudos, comparamos o desempenho de diferentes algoritmos de Machine Learning com modelos mais *naive* na previsão de muitas variáveis econômicas em tempo real e mostramos que, na maioria das vezes, o Machine Learning supera os modelos de benchmark. Já no restante dos nossos exercícios, combinamos várias técnicas de nowcasting com um grande conjunto de dados (incluindo variáveis de alta frequência, como o Google Trends) para rastrear a pandemia no Brasil, mostrando que fomos capazes de antecipar os números reais de mortes e casos muito antes de estarem disponíveis oficialmente para todos.

Palavras-chave

Nowcasting; Aprendizado de máquina; Big data; Previsão; Modelos de alta dimensão; COVID-19

Table of contents

List of figures

List of tables

List of Abreviations

ML – *Machine Learning*
HD – *High-dimension*
BT – *Backtest*
GT – *Google Trends*
GDP – *Gross Domestic Product*
LASSO – *Least Absolute Shrinkage Operator*
RF – *Random Forest*
CSR – Complete Subset regression
NN – *Neural Nets*
CLM – *Chain Ladder Model*

1
Nowcasting constantly revised variables: an application to COVID-19 in Brazil.

1.1
Introduction

As the name suggests, Nowcasting is the forecasting of the present, the very near future or even the very recent past state of a variable of interest. It has originated in meteorology, but it has become popular in several fields in the past few decades. In economics, even though it wasn't named Nowcasting yet, it has been around at least since Burns and Mitchell (1956). However, the economics literature that formalizes and treats this class of problems started approximately 20 years ago with Stock and Watson (2002), then Evans (2005) and Banbura et al. (2010b). Since then, Nowcasting models have been widely applied in economics. For instance, several Central Banks use different Nowcasting techniques to monitor the state of the economy in real-time as a proxy while official measures are not due. (Eraslan and Götz (2020), Adam et al. (2021), Lewis et al. (2020) are a few examples).

As Banbura et al. (2010b) argue, while weather forecasters know weather conditions today and only have to predict future weather, economists sometimes have to forecast the present and even the recent past. This is due to the fact that many official economic measures are not timely due to the difficulty in collecting information. For example, the Gross Domestic Product (GDP) of most countries is only determined after a long delay since the end of each quarter of reference. While the official number is not yet released, by applying Nowcast models one can exploit information from a large quantity of data series (at different frequencies and with different publication lags) in order to gather information about the variable of interest.

However, this report delay of a variable of interest aforementioned is only one reason why Nowcasting models are so important sometimes. It could also be that the variable is available almost in real-time, but that it is subject to constant revisions over time. In this case, even if the variable is timely, one can't fully rely on the initial realized values, because as time goes by they will

keep getting updated until it saturates after a while.

It is specifically the case of COVID-19 new daily deaths in Brazil, which we will start discussing in section 2. In section 3, we detail all datasets used in this article to motivate the need for a Nowcasting model and the ones actually used in the model proposed. Then, in section 4, we describe the methodology proposed for a Nowcasting model when the dependent variable is constantly subject to revisions and we detail how we have been applying it to COVID-19 deaths in Brazil. Section 5 is the results section, in which we show the results of our methodology for Brazil and some states. Finally, we conclude in section 5.

1.2 Why Nowcasting COVID-19 deaths in Brazil matters?

During the pandemic, many studies attempted to predict the pattern of COVID-19 in several regions with all sort of models, including both epidemiological and statistical ones; see, for example, Medeiros et al. (2022), Bertozzia et al. (2020), Anastassopoulou et al. (2020), Carneiro et al. (2020), and Matjaž et al. (2020a). In most countries, the forecasting of the following few weeks was of utmost importance, so that policy makers and health authorities could prepare the system to absorb potential new waves of the virus.

However, in Brazil, as we will argue in this paper, one can't even trust the numbers of confirmed cases and deaths of each day of the recent past, which makes the forecast of the future extremely complicated. As we will discuss here, being able to precisely nowcast the current state of the pandemic in the largest country of South America would give policy makers around two weeks of information advantage when compared to the official reported numbers.

Globally and specially before vaccines, one of the most important indicators of the actual severity of the pandemic in each country, is the number of cases aggregated by the date of first symptoms felt by each person, as well as the daily casualties sorted by the day of the death. These numbers give a clear indication on how the disease is evolving in real-time. In most developed countries, these two variables mentioned above are very similar to the series of new cases and deaths sorted by the date of notification.

In Brazil, on the other hand, the true, final numbers are available with major delays (more or less 2 weeks). For some locations in Brazil, the delay can exceed a month. As we will argue in this section, part of the delay is due to the coronavirus cycle but most of it is related to the bureaucracy in the

register system.[1]

To begin illustrating the point that there is indeed a non superfluous delay in the notification of COVID-19 confirmed cases in Brazil, Figure 1.1 below shows the difference in days of the date of notification (the day in which the infected person enters the official register system) and the date of first symptoms felt by each individual.

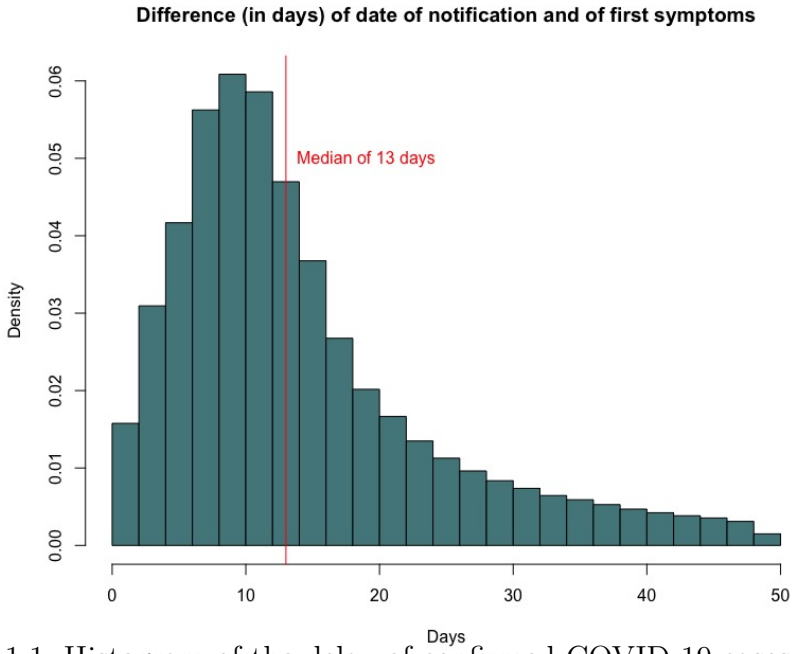

Figure 1.1: Histogram of the delay of confirmed COVID-19 cases notification in Brazil.

As one can see above, there is a significant portion of the infected population that are introduced into the official system with a major delay with respect to when that person started feeling COVID-19 symptoms. As a matter of fact, the median person gets reported almost two weeks after she started feeling symptoms, while in some less often cases, people might be computed into the official numbers only a month after being infected.

Of course, Brazilian official COVID-19 register system was not fully guilty for all this delay. There are several other reasons that can contribute for these numbers to be so high. For example, among the population of infected, some people might choose to wait a while before leaving home to get tested, which will impose a structural delay in their computation. Another thing that is important to highlight is that the date of first symptoms is a self reported variable, which means that it might not be extremely accurate in some cases.[2]

[1]For further reference with respect to the Brazilian case, please visit the following link.

[2]The infographic A.1 in Appendix A helps understanding better the reasons for the delay in Brazil's system. It will illustrate the usual timeline of a patient that ended up dying of COVID-19 in Brazil.

Moreover, when an individual dies with COVID-19 in Brazil, the same thing that happens with the date of notification versus the date of first symptoms, might happen with the date of notification and the date of the occurrence. It is harder to display the same histogram as in Figure 1.1 above for the case of a death, because for most people the "date of notification" variable in the official dataset (which we will discuss in the following section) represents the date of the first time that person entered the system. For instance, if an individual was first diagnosed with COVID-19 in day t_0, to only being computed into the system in day t_{0+h} and then ends up dying in day t_{0+d} (for $d > h$), her date of notification in the official dataset will be t_{0+h}.

However, there is a simpler and even more direct way to show the discrepancy in the curve of daily deaths by date of notification in respect to the (true) one aggregated by the date of the death. In fact, Figure 1.2 below compares both curves from the beginning of the pandemic in Brazil until mid-2021.

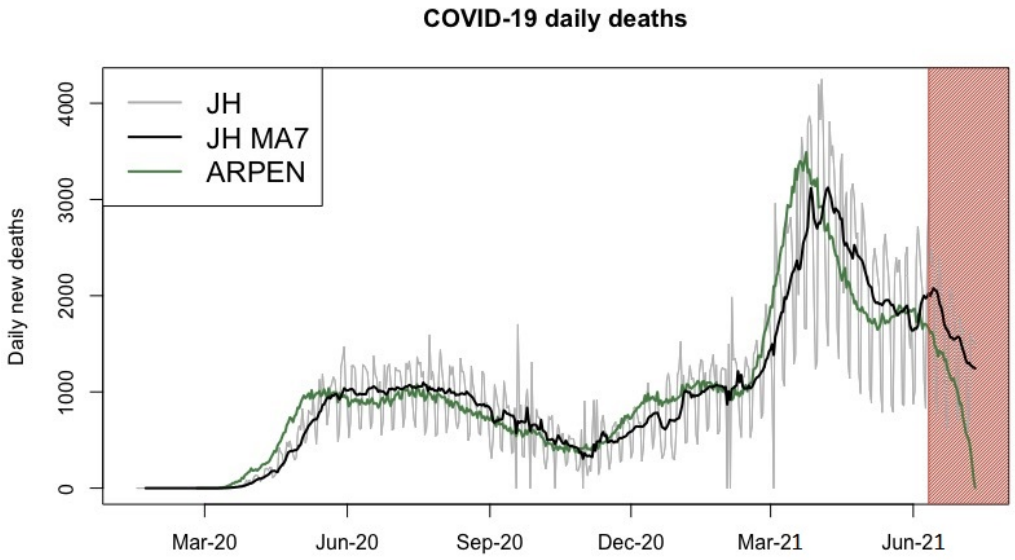

Figure 1.2: COVID-19 daily deaths in Brazil aggregated by date of death (green) and by date of notification (gray).

In Figure 1.2, the more volatile, gray curve is the daily deaths in Brazil from the John Hopkins data[3], which extracts data daily from the Brazilian Ministry of Health. Clearly, the gray line is reported with some noise. For instance, there is no virus-related reason why Monday's average daily deaths should be 57% smaller than Tuesday's figures. Indeed, the gray line displays

[3]For further reference, we refer to COVID-19 Data Repository by the Center for Systems Science and Engineering at https://github.com/CSSEGISandData/COVID-19

COVID-19 deaths curve by date of notification, which is limited by the infrastructure of Brazilian register system. As a matter of fact, this is is why the media usually reports the moving average of this curve, instead of the original series. In black, one can see its seven-day moving average.

Having described the official numbers from the Ministry of Health, we explain now the green line in Figure 1.2. It represents the daily new deaths caused by COVID-19 in Brazil, but now aggregated by the date of occurrence (or date of event). This dataset (further discussed in the next section) is made available for everyone by the ARPEN (Brazilian National Association of Registry Offices). Since the green curve displays the actual number of Brazilians that died from COVID-19 in each day since 2020 (and not the number of people that the system could compute in each day), it is a much more accurate proxy for the actual pandemic related deaths in Brazil than the curve representing deaths by date of notification.

Now, one might wonder that if ARPEN data is widely available daily for every researcher in Brazil and if it accurately represents the death curve, then the problem should be settled (or that ultimately there is not a problem to solve). However, as the red area in Figure 1.2 indicates (more on this soon in Figure 1.4 below), each new ARPEN's data point is subject to major (upwards) revisions for around 30 days after its first release until it stabilizes. This means that what one would see in mid-July/2021 above in the green curve inside the red area is not what she would see for the same timestamp (mid-June/2021 to mid-July/2021) in ARPEN's vintage one month later.

As one can see in Figure 1.2, even though the green and the black lines are similar and display the same patterns, the green one is always ahead of the black one. For example, in the first wave of April, 2020, the actual (green) curve of daily deaths started to rise (and then reached a plateau) much earlier than the curve by date of notification (black), i.e., the series of numbers of which Brazilians heard and read about everyday in the media.

In order to help make the case and clarify that the green curve has in fact being ahead of the black curve, Figure 1.3 below zooms in three specific important periods of the pandemic in Brazil: the rise of the first wave in April, 2020; the rise of the second big wave in March of 2021; and the decline of the same big wave in April, 2021.

By looking at the panel on the left, we conclude that the green (ARPEN) curve achieved its first peak (to then stay on a high plateau for around two weeks) three weeks before the black (John Hopkins/Ministry of Health) curve. Even taking into consideration that the black curve is a moving average of 7 days, the green curve reach its maximum value with more than two weeks in

advance of the other. Similar conclusions arrive if we analyze the other two panels, which display the rise and the fall of the true daily numbers of deaths by COVID-19 in the most recent big wave.[4]

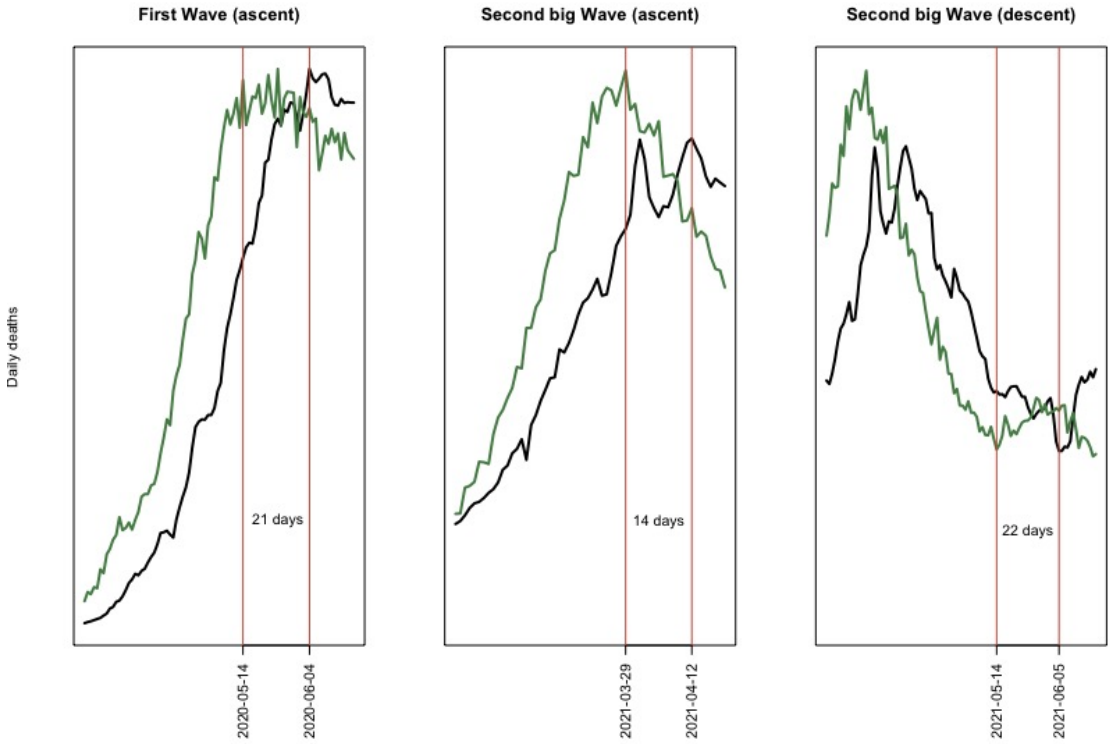

Figure 1.3: A zoom in three important moments of the pandemic's evolution in Brazil.

We believe that these images illustrate well the reasons why it may be so important to have Nowcasting models of the actual curve of deaths by COVID-19 by the date of the occurrence. Again, this curve seems to be the most accurate proxy for the actual pandemic status in Brazil, it seems to be always ahead of the reported curve (by date of notification), but it suffers from the problem of being constantly revised on its most recent tail.

In order to highlight how these figures evolve over time, Figure 1.4 below displays seven different vintages of the ARPEN data, each one spaced 1 month between each other. The most recent one used in this article, the one from mid-July/2021 (in black) is the green curve plotted in Figure 1.2. Every other colored curve below represents the same variable but from older vintages. Each vertical, dashed line indicates when the 30Th newest observation of that same color vintage starts.

[4]In fact, in the case of the right panel, when the black curve reached its bottom on June, 05, the actual series of COVID-19 deaths was already rising for almost 3 weeks.

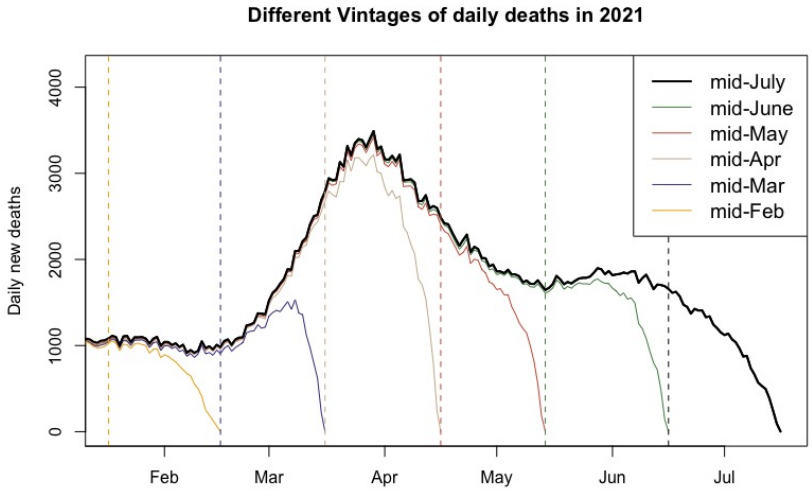

Figure 1.4: Six different ARPEN data vintages.

By looking at Figure 1.4, it becomes clear that the newest 30 observations of each vintage always rise, varying more the more recent it is.[5] We hope that this last image convinces the reader that an accurate Nowcasting model for the past 30 days of the curve of daily deaths by the date of occurrence seems to be a powerful tool for understanding the pandemic in Brazil and thus for the policy makers to adapt better for it.[6]

Now we will detail every dataset already mentioned in this section and the others we use to create the Nowcasting model.

1.3 Data

In the previous section we already mentioned two important variables that come from different datasets: the number of daily deaths by the date of notification (compiled from Brazil's Ministry of Health by the John Hopkins CSSE) and the number of daily deaths by the date of occurrence, made available by the ARPEN. While the former was used only to explain the reason Brazil needs a Nowcasting model, the latter is our target, the dependent variable of the model.

As previously mentioned, ARPEN stands for the Brazilian Association of Registry Offices. This means that its associates as a whole are responsible

[5]Of course, there is also some cases in which observations older than 30 days also change, but when they do, they change only marginally.

[6]While figure 1.4 shows vintages for the whole country of Brazil, figure A.2 illustrates the same idea, but for three different states. It highlights that there are some states that need Nowcasting models more than others.

for all death certificates in Brazil. During the pandemic, they started updating daily their data, including a section for deaths caused specifically by COVID-19. This made possible the daily download of different vintages of our series of interest, which we nowcasted for several months in order to understand the state of the pandemic in Brazil[7].

As we will explain in the methodology section, even though some of our proposed individual models are as simple as finding and applying past patterns of revisions to the dependent variable in order to nowcast it, there are other set of models that will nowcast the current state of the pandemic by relating the history of the ARPEN data with other independent variables (covariates) that come from different sources.

In these types of model, one of the most important variables used to nowcast COVID-19 deaths seem to be key topics of Google Trends data[8]. For each topic, Google Trends provides access to a sample of actual search requests made to Google, allowing one to look into the interest in that particular topic from around the globe or down to city-level geography almost in real time. The hypothesis behind having this variable in the set of potential predictors of the actual daily numbers of COVID-19 induced deaths is that when the situation start getting worse, people start searching more about COVID-19 in Google.

Another useful source of information is also widely made available by Google: Google Mobility data. It shows how visits and length of stay at different places (parks, residential areas, transit stations, groceries stores, etc) change compared to a baseline, before the outbreak of the pandemic. This variable allows us to have more understanding almost in real-time about when and where lockdown impositions were respected and if that had an impact on COVID-19 confirmed cases and deaths.

Besides the variables already described, holiday days are also part of the set of potential predictors. With the appropriate setup of temporal relations, holidays might help nowcast new deaths, as people tend to travel and meet each other more during these days. Also, disaggregated daily energy data[9] from the national interlinked system (SIN) (which allows the user to filter the data by federal unit, branch of activity, etc) is also taken into consideration. Moreover, several different moving averages of almost every covariate described above are

[7]The reader can find our model results in the following website: https://covid19analytics.com.br

[8]In the next article of this thesis we provide more information on how to gather and how properly to use Google Trends. In this paper, we gather the following keywords (in Portuguese): "COVID deaths", "COVID hospital", "COVID ICU", "COVID grief" and "COVID Brazil". We use package "gtrendsR" from software R for an easy download of each term.

[9]website of the Chamber of Electric Energy Trading.)

also part of the set of covariates.

Finally, lagged confirmed COVID-19 cases by the date of first symptom is also one important predictor. We create this variable using the SARS (syndrome respiratoire aigu sévère) Surveillance Portal, made available by the Ministry of Health.[10]

1.4 Methodology

In order to try to capture every source of variation on the variable of interest, our proposed methodology combines different kinds of models. The final Nowcasting numbers are an ensemble of several different models. In order to obtain the weights given to each model for each Nowcast window, we have structured a continuous backtest to be run every time a new Nowcast is done. The idea is to give highest weights to models that showed the smallest errors in the out-of-sample (the recent past). We start off this section by describing the High-Dimensional Machine Learning (ML) models.

1.4.1 High-Dimensional Models

We call High-Dimensional or ML models the usual models where the econometrician has a dependent variable that she wants to predict with a big set of available potential predictors. However, in our case, instead of forecasting unknown future realizations of the dependent variable, we nowcast its most recent realizations which we know will change over time (as in Figure 1.4). In other words, we train and validate the model in the past in which our dependent variable is already fully saturated (in-sample) and nowcast in the most recent past in which our dependent variable is subject to a lot of change (out-of-sample).

It is important to highlight that in the case of COVID-19 daily deaths, our dependent variable only gets revised upwards. In this sense, all our models, including other kind of models to be detailed next, have a lower bound, which is the number of deaths by COVID-19 that we already know using the most

[10]In order to construct this variable, we filter the original database to contain only people with the confirmation of COVID-19 diagnosis. Then, we count the number of individuals by the date of first symptoms felt. After that, we use a lagged moving average of that series to relate with the number of deaths by date of occurrence, as we will explain in the next section.

recent vintage. Therefore, if the nowcast for a certain day is smaller then the value we already know (when this happens, it is usually in the oldest half of the nowcast sample, which is way more stable than the first half), we change the original nowcast for this lower bound number.

As already briefly discussed in the previous section, in order to nowcast recent observations of our dependent variable, we need covariates that are also available in real-time (no delay) and that are stable (won't be systematically revised over time). For our purposes, we gathered Google Trends, Google Mobility, Lagged Cases (aggregated by date of first symptoms) and Energy data. All these variables are timely available and usually don't get revised. With respect to the actual models used in the Nowcasting, there is a variety of off-the-shelf Machine Learning models available with easy implementations[11]. For instance, Medeiros et al. (2021) and the latest paper in this thesis analyze the performance of several important benchmark models as the LASSO (Tibshirani (1996)), the Random Forest (Breiman (2001)), the Bagging (Breiman (1996)) and the Bayesian VAR (Banbura et al. (2010a)). Here, after a few out-of-sample performances comparisons, we ended up using only the LASSO and the Principal Component Analysis.

Figure 1.5 below helps illustrating the structure of ML models in a setup where variables are strongly subject to revision in the short term. As we highlight there, these models have the usual train sample (in-sample), where the ML model learns the important contemporary patterns between the dependent variable and its covariates, to then nowcast based on these past relations. We plot below two different vintages of our dependent variable[12] in order to show that it is the newest 30 observations (highlighted by the vertical, dashed, red line) which are more subject to revisions over time.

Note that for these types of model, except for the fact that the latest 30 data points of the orange curve (dependent variable) sets the lower bound for the final Nowcasting numbers, they are not useful for modeling purposes after the end of the in-sample. However, there are other types of models that will use information about to produce their nowcasts. We discuss them now.

[11]For further information about the implementation of these and other ML models, we refer to James and Tibshirani (2013).

[12]The orange line represents ARPEN's vintage of April 23, 2021, whereas the black one was extracted from the vintage of JUly 21, 2021

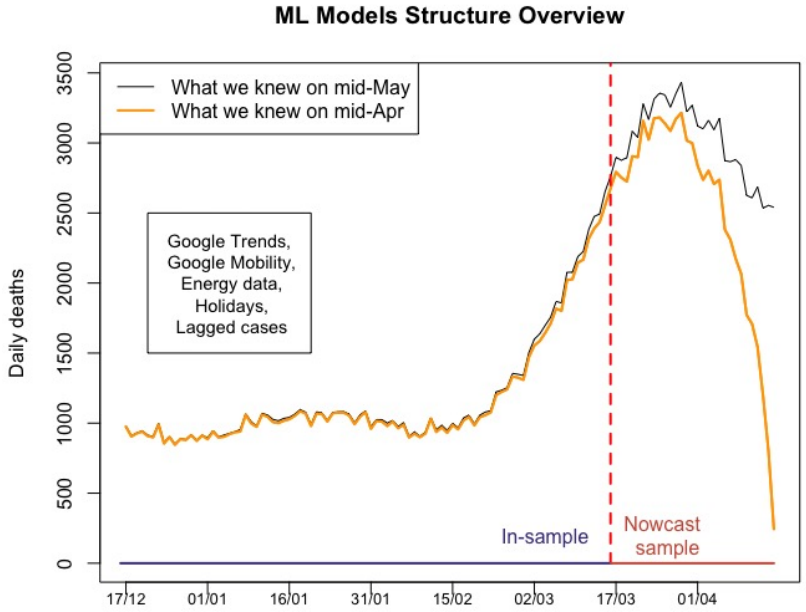

Figure 1.5: ML models where variables are strongly subject to revision in the short term.

1.4.2 Revision-based Models

In the High-Dimensional type of models presented before, we still didn't model anything related to the revisions of our dependent variable. We treated the most recent observations of that variable as if they still weren't available for the general public and tried to nowcast them using other variables that would indeed be available in the moment of the nowcast. However, there is a very important source of information which comes from past patterns of the revisions that would indeed improve the Nowcasting performance. We called revision-based models the first type of methodology that uses revision's patterns to nowcast.

In order to help explaining revision-based models, Table 1.1 below brings an example of a collection of vintages of a variable that can change over time. There, $y_h|_s$ is the observed value of variable y for the date of occurrence h known in day s (which can be different than the observed value of the same variable and date of occurrence in a day k, for $k \neq s$). Moreover, for all t the first release of observation y_t is always on the next day's vintage (day $t+1$), which is the pattern of our case of interest.

Table 1.1: Example of a collection of vintages of a variable that can change over time.

Date of occurrence	Vintage of ...			
	Day t+1	Day t+2	Day t+3	Day t+4
...	...	...	...	...
t-1	$y_{t-1\|t+1}$	$y_{t-1\|t+2}$	$y_{t-1\|t+3}$	$y_{t-1\|t+4}$
t	$y_{t\|t+1}$	$y_{t\|t+2}$	$y_{t\|t+3}$	$y_{t\|t+4}$
t+1	NA	$y_{t+1\|t+2}$	$y_{t+1\|t+3}$	$y_{t+1\|t+4}$
t+2	NA	NA	$y_{t+2\|t+3}$	$y_{t+2\|t+4}$
t+3	NA	NA	NA	$y_{t+3\|t+4}$
t+4	NA	NA	NA	NA

Another way of visualizing these revisions is through Figure 1.6. The black curve is the revision curve of the number of Brazilians who actually passed away from COVID-19 on March 16, 2021, which was one of the deadliest days in Brazil. The first released number (reported on March 17, 2021) was 250 deaths. Then, it was updated to 787 in the next day (March 18, 2021) and kept rising until it reached 2668 on April 16, 2021.[13] The red curve below illustrates the same idea, but for the amount of COVID-19 victims on April 27, 2021.

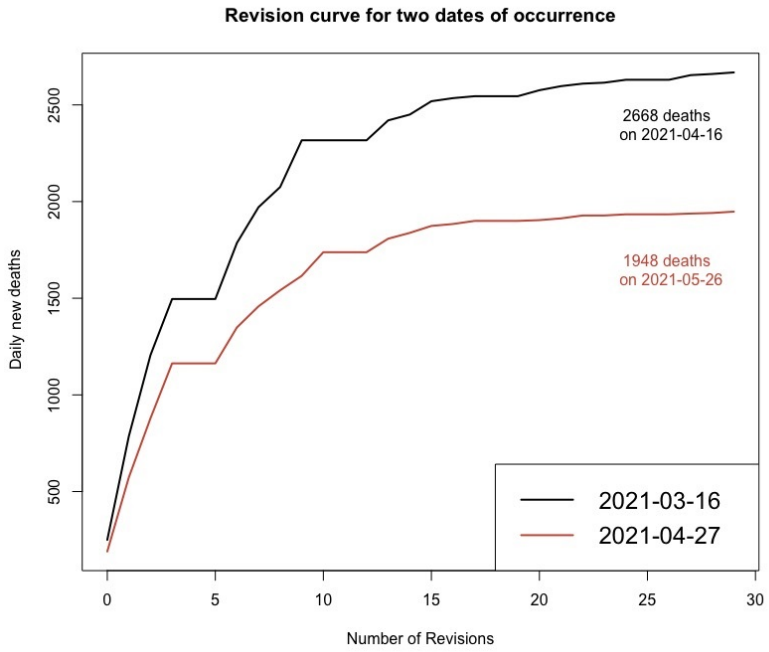

Figure 1.6: Number of individuals who died from COVID-19 on April 27, 2021 (red) and on March 16, 2021 (black).

[13]After 122 revisions, when we wrote this article, the number had stabilized at 2788, representing only a marginal increase.

Figure 1.6 highlights three important points. First, it becomes clear that, as they change a lot, the initial releases of each day are not very informative by themselves. Second, most part (around 90%) of the variation in these numbers happen in their first 15 revisions. And last, but not least, one could fit a linear model (using linear, logarithm or polynomial trends, for instance) using past, known revision curves in order predict future values of most recent released numbers.

Before heading to our proposed revision-based model, for illustration purposes, suppose we are in time t. As already explained, our Nowcasting window is formed by the 30 observations ranging from days $t-1$ to $t-30$ and is illustrated by the orange line in Figure 1.7 In t, the most recent day that has already completed 30 revisions (and therefore, the most recent one we could use to fit our model is $t-31$.

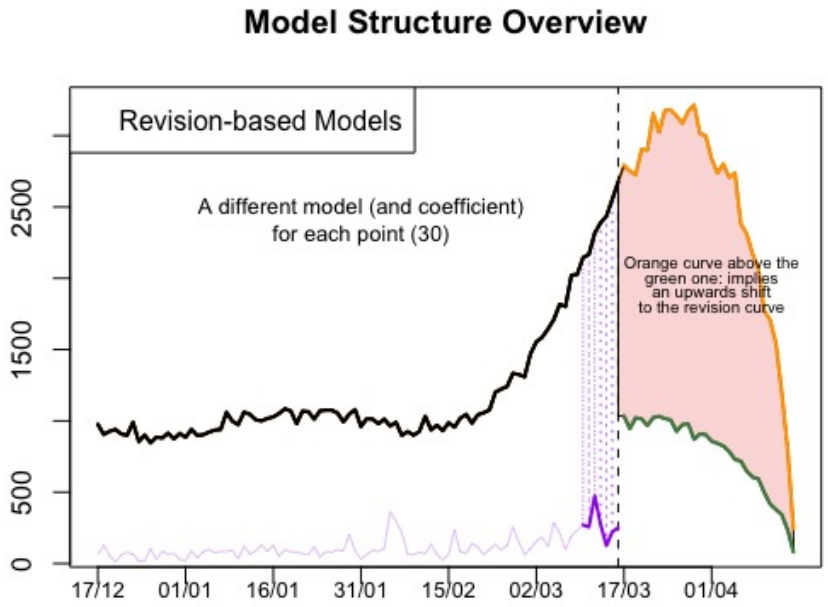

Figure 1.7: Overview of revision-based models structure.

If we gathered the first 30 revisions of y_{t-31} (i.e., $y_{t-31|t-30}$, $y_{t-31|t-29}$,..., $y_{t-31}|_{t-1}$, which is precisely the black curve in Figure 1.6), we could fit a trend linear model to it in order to project future, saturated values of the orange curve. However, by doing that directly, we would be missing an important source of information. In this sense, we propose a simple adjustment to be applied into these past, known revision curves prior to fitting a trend linear model into them.

The source of information that would have been left behind otherwise is

the comparison between what we know in t about the latest 30 observations (orange curve in Figure 1.7) and what we knew in $t-30$ about that moment's 30 most recent observations, which is represented by the green curve in Figure 1.7[14]. By computing this ratio and applying it uniformly to each of the 30 data points that we will use to fit our trend model, we shift its curve towards a more recent scenario.

In other words, if the sum of every data point from the orange curve is above the sum of the points from the green one, we would shift our revision curve (Figure 1.6) upwards and only then fit a trend model to it. Analogously, if the ratio was below one, we would shift the revision curve downwards. During our backtests[15], this transformation strongly raised the accuracy of our revision-based model.

1.4.3 Chain-ladder Models

The Chain Ladder Method (CLM) developed originally in Schmidt (1999) is a very popular algorithm among insurance companies originated in the actuarial field. It was originally used to determine the amount of reserves that must be established in order to cover forecasted future claims by projecting past claims experience into the future. The primary underlying assumption of the CLM is that historical loss patterns are indicative of future ones. Therefore, it only works when prior patterns of losses persists in the future.

For our COVID-19 exercise, we apply the same original idea of the CLM in two slightly different computations. For illustration purposes, suppose we are in time t. For our first CLM type model (henceforth, Naive model), we first calculate the proportion between each $y_{v-h}|_v$, for h $\in [1, 2, ...30]$ and $v \in [t-31, t-32, ..., t-37]$[16] and its saturated values $y_{v-h}|_t$.

Then, for each h, we compute the average growth across the seven observations (for each h) and apply it to our nowcasts. So, for example, if $y_{t-1}|_t = 100$ and we calculated that $\frac{1}{7}\sum_{v=t-37}^{t-31} \frac{y_{v-1}|_t}{y_{v-1}|_v} = 15$, our Naive method projects $y_{t-1}|_t$ to saturate in 1500. In the same way, if $y_{t-30}|_t = 2000$ and we calculated that $\frac{1}{7}\sum_{v=t-37}^{t-31} \frac{y_{v-30}|_t}{y_{v-30}|_v} = 1.05$, our Naive method projects $y_{t-30}|_t$ to saturate in 2100.

[14]The green curve is intentionally plotted below in the same x-axis range of the orange curve, even though they relate to different days chronologically

[15]We ended up using a polynomial trend model

[16]Note that we use the first 7 vintages in which the most recent observation has already saturated.

Figure 1.8 below illustrate[17] the regions where this computation extracts its information from. The green line between the dashed rectangle represents the last 30 observations of vintage $t-31$. By comparing these numbers to the ones they turned out to be (the black, solid line inside the same rectangle), we are able to apply the same patterns to our current nowcast window (the orange line in Figure 1.8).

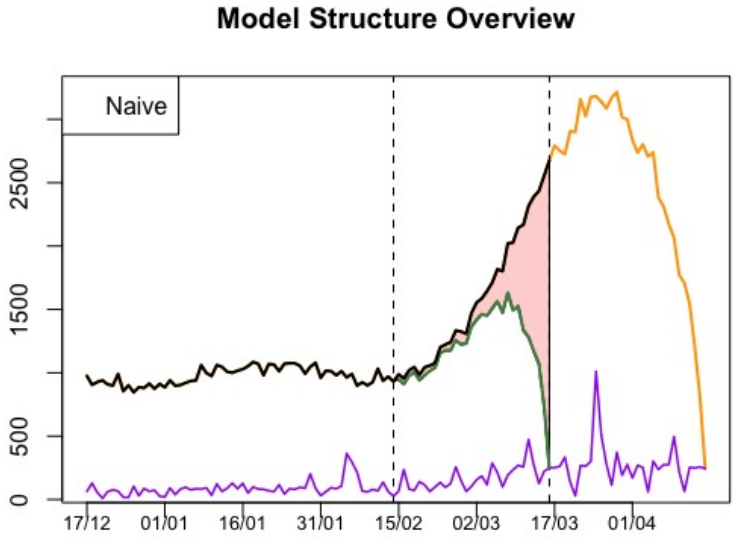

Figure 1.8: Illustration of the source of variation captured by both models proposed in this section.

Our second CLM-type model is very similar to the previous one and relates even more to the original CLM. It is worth noting that although it is more stable, it has the cost of responding less quicker to pattern changes. Here, instead of computing ratios between each data point and only then taking the average ratio across a week worth of vintages, this second approach computes the pattern in the following way: $\frac{\sum_{v=t-37}^{t-31} y_{v-h|t}}{\sum_{v=t-37}^{t-31} y_{v-h|v}}$ for each $h \in [1, 2, ..., 30]$. Then, we apply the same respective past growths to each data point in our nowcast window.

It is interesting to note that while CLM-type models extract information from the ration between the green curve and black curve (between dashed lines) in Figure 1.8, our revision-based model from previous subsection extracts valuable insights from the ratio between the green one and the orange one.

[17]The purple line indicates the curve of $y_t|_{t+1}$ $\forall t$, i.e., the curve of first realized values of each day (before they got any revision).

1.4.4
Simple combination schemes

Naturally, a kind of model that fits well our methodology are simple combination schemes models. One could think in several different schemes, as the simple average of all other models, a trimmed average or even the median. In our application for COVID-19 deaths, after initial tests, we ended up including only the simple average.

As of right now, we have detailed all type of models that could be used to extract valuable information to perform the Nowcasting. However, as previous explained, we perform a new backtesting (BT) every time we run a new Nowcast. We discuss now our BT procedure.

1.4.5
Continuous Backtest

Our continuous backtesting procedure consists on running the same models presented in previous subsections above, but in a past old enough, so that our dependent variable numbers is already saturated (using the latest available vintages). This gives us the opportunity to assess the accuracy of our models in the recent past, in a way we can give the highest ensemble weights to the best performing models and lowest weights to the worse ones. To illustrate the procedure, Figure 1.9 below gives an example of the windows (of nowcasting and BT) that would be used for a Nowcasting on July 1, 2021.

For this example, let's stick with our COVID-19 daily deaths by date of occurrence. If we were to Nowcast it on July 1st, this vintage would bring information until June 30rd, so that our Nowcast window would start on June 1st. For days older than June 1st, we consider that its information about daily deaths is reliable enough to consider it saturated.

That being said, we test our models in 30 BT windows of 30 days. The last one (the most recent one) starts on May 2 and ends on May 31. Every BT window takes a forward step until we reach 30 windows. In this example, the first one (the chronologically oldest one in Figure 1.9) begins in April 3 and ends on May 2.

It is important to highlight that in each of these 30 windows the nowcast procedure is performed exactly in the same way as in the nowcast window. The only difference is that we have available the target variable for the backtest, which allows us to compute each model's errors.

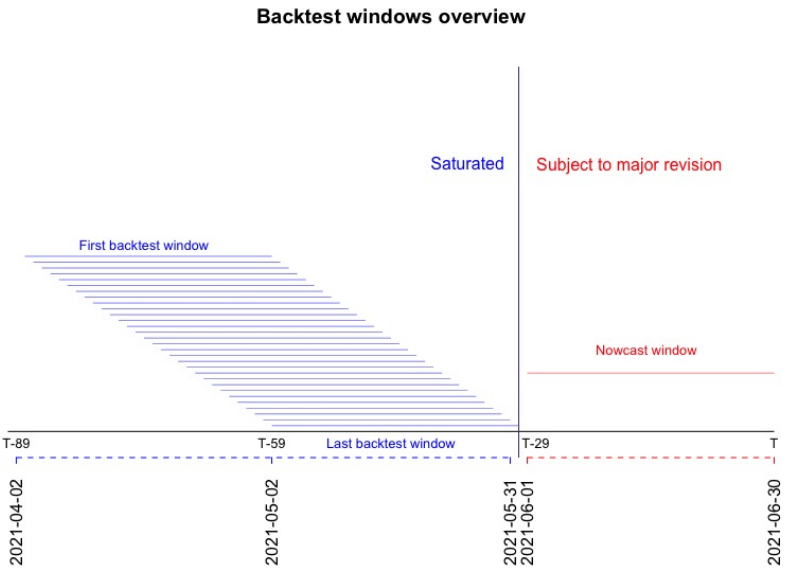

Figure 1.9: Nowcast and Backtest windows example of a vintage ending on June 30.

Another example that might help visualize the BT structure is presented in Figure 1.10 below. The yellow curve represents the number of COVID-19 victims using the ARPEN vintage of April 16, 2021. Naturally, the red box delimits the nowcast window (the 30 most recent days). The large black box enclosures all the days that are part of a BT window.

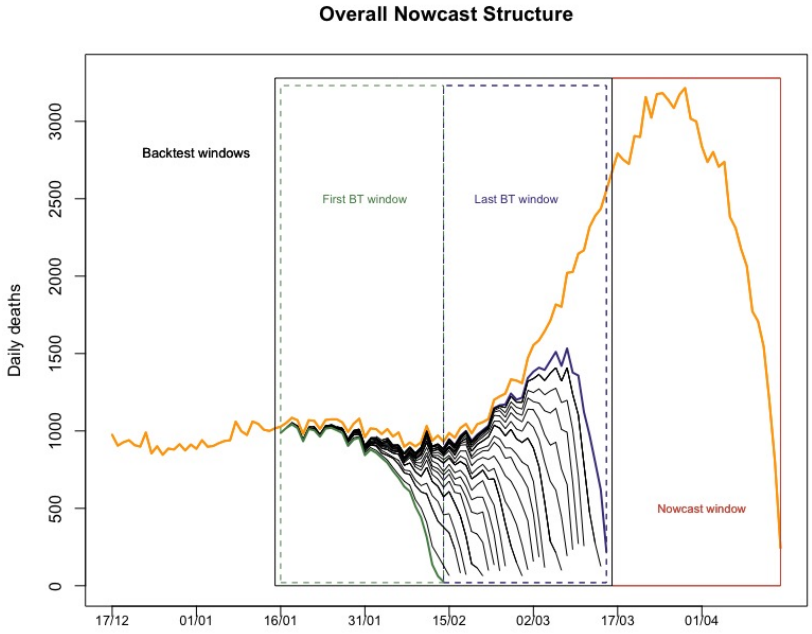

Figure 1.10: Our Nowcasting procedure overview.

Figure 1.10 highlights the first backtest window (in green) and the last (in

blue), as well as the information known at that time for each window. Again, even though the orange curve was already known and saturated in the BT period, in each of the 30 iterations we run every model using information only known in that vintage.[18] By doing that, we can measure errors for each model, which we use to construct the weights we give to each model in the actual nowcast window. We also use these errors to compute our nowcast prediction's interval, which we discuss next.

1.5 Results

We started thinking about applying our methodology to COVID-19 daily deaths in March, 2021 and after outstanding backtesting performances (illustrated in Figures A.4 and A.5), we were able to launch it online in mid-April of that year. In hindsight, it was an important moment for its release, because it was exactly the period when Brazil's second big wave was on its point of inflection, even though the official numbers of daily deaths (again, by date of notification) wasn't exactly showing that already.

Figure A.6 displays the nowcast results of its first release on April 16, 2021. After three months since this nowcast, we already know for a while the true numbers of daily COVID-19 victims for that period, which allows us to compare observed and nowcasted values. As illustrated in the aforementioned figure, our nowcasts were very accurate, obtaining a mean average percentage error (MAPE) of 3.1% in that period (3.8% if we consider only n_1 to n_{15}, the hardest windows to predict). It is important to highlight again that the nowcast window of that vintage is exactly the one when the daily deaths started to go down after a long, sharp ascendance in the previous month. Even so, our methodology performed very well, both in the backtesting and in reality.

Figures A.7 and A.8 illustrates the same idea as Figure A.6, but for nowcasts windows of April 23 (our model's second weekly release) and the ones of June 18. In both figures, our methodology seemed to be able to correctly nowcast the continuous downwards trend in Figure A.7 and the June plateau in Figure A.8. In fact, from its launch until the end of July, 2021, we had 12 weekly nowcasts, which numbers had already saturated, allowing us to compute the nowcast average errors. The black curve in Figure 1.11 below illustrates the Mean Absolute Percentage Error (MAPE) for each n_i ($i \in [1, 2, ..., 30]$)

[18]For example, the dependent variable for the last BT window is the curve in blue in Figure 1.10, going back to the beginning of the pandemic. This means that the model never get to see the orange line.

across the 12 weeks of our true out-of-sample, where, for each vintage,its n_1 is the nowcast of the most recent data point available (the one that still hadn't received any revisions) and its n_{30} is the oldest data point to be nowcasted.

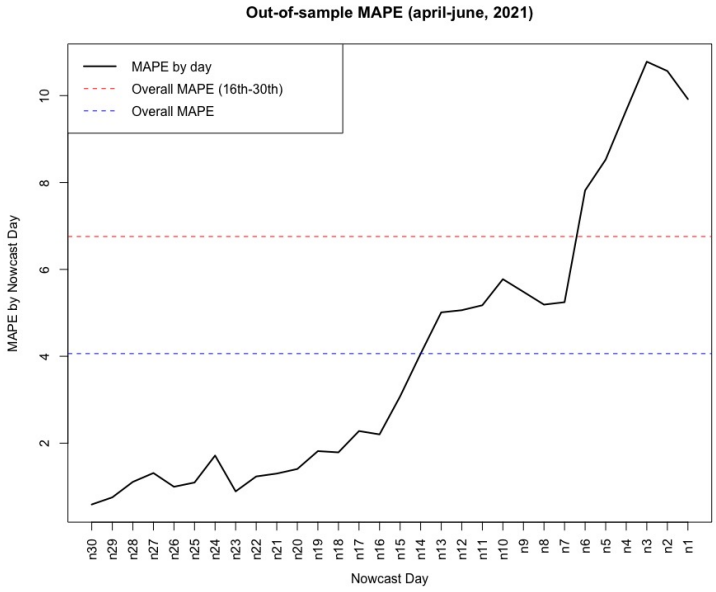

Figure 1.11: Error assessment across every nowcast done that has already saturated so far.

Figure 1.11 has the same idea as Figure A.5, with the difference that instead of plotting the errors for the backtesting procedure, it plots the actual out-of-sample errors across 12 different weeks of nowcasts. As expected, average errors are higher in more recent days (the ones that will vary a lot throughout the next weeks). Even so, our out-of-sample MAPE seems to be around 10% in the hardest moment to nowcast. If we look to our overall MAPE (simple average across our $30 * 12 = 360$ errors points) we get to a number around 4%. If we do the same, but filtering out nowcasts days n_i for $i \in [16, 17, ...30]$, i.e., the easier days to nowcast, we get to 6.5% of MAPE.

1.6 Conclusions

In this first article we propose a mixed Nowcasting methodology when the variable of interest is available almost in real-time, but is subject to huge revisions for a while. Our methodology combines Machine Learning (High-Dimensional) models, which will try to nowcast using past patterns of the relation of the dependent variable and a large set of important covariates available in real-time (variables which are not subject to revisions), with

simpler but also very important revision-based and chain-ladder models that will try to tackle the nowcasting problem from a different perspective.

Moreover, We explain how our methodology includes a continuous back-testing procedure, which allows us to keep updating the weights given for each model in the final nowcasting, based on their recent performance.

We apply our methodology to the case of nowcasting COVID-19 deaths by the date of occurrence, a variable that arises very small and is subject to daily revisions that might end up making it grow up to 25 times (but usually 10 times) its initial value around 30 days after its first release. We show good results for our model, highlighting that we were able to nowcast the peak of the second big wave of COVID-19 deaths in Brazil in April, 2021 two weeks before the official numbers were showing its peak.

Taking into consideration that we might actually never get completely rid of COVID-19 and that other pandemics might surge in the future, we believe it is of utmost importance to share what we have been learning with the Nowcasting of COVID-19 in Brazil. By having an accurate model that assists policy makers to better understand the current state of the pandemic and therefore prepare for the worst in advance, many lives can be saved.

2
The Proper Use of Google Trends in Nowcasting Models

2.1
Introduction

`Google Trends` is an incredible, free tool, that shows the most popular terms searched in Google the recent past. It provides access to a sample of actual search requests made to Google, allowing one to look into the interest in a particular topic from around the globe or down to city-level geography. Several studies from different areas used this resource mainly to improve forecast accuracy. See, for example, Askitas and Zimmermann (2009), Artola and Galan (2012), Baker and Fradkin (2017), Naccaratoa et al. (2018),Ferrara and Simoni (2019), Eraslan and Götz (2020), Woloszko (2020a).

The tool provides the frequency in which a particular term is searched for in several languages from various regions of the world. However, Google normalizes the search data to make comparisons between terms easier. This means that search results are normalized to the time and location of a query by the following process. First, each data point is divided by the total searches of the location and time range chosen by the user. Then, the resulting numbers are scaled from 0 to 100 based on the topic's proportion to all searches on all topics.

By doing so, `Google Trends`, (henceforth GT) data represent the relative popularity. This feature avoid the situation where places with the most search volume would always be ranked the highest. Naturally, this normalization also implies that, if different regions show the same search interest for a term, it does not always mean that they actually have the same total search volumes.

All these GT features described above are widely known by its users. However, what many do not seem to know is the fact that almost every time one looks for a `Google Trend`'s term and downloads it, she gets a different series of numbers. The reason for this is that Google handle billions of searches per day, which means that providing access to the entire data set would be too large to process quickly. To circumvent this obstacle, only a small sample of Google searches are actually used in GT. By sampling data, one can look at a dataset - generally - representative of all Google searches and that can be

processed within minutes of an event happening in the real world.

However, there are some cases when this feature can actually become a real problem if not properly handled. We show in this paper which are the situations where it is important to look at GT data with some care. Moreover, we show with both simulated and real data that it is possible to keep using GT as covariates (and indeed important ones) without having the problem of your dataset not representing the actual population of the term's search for some specific period of time and location.

The article is divided as follows. Section 2.2 explains the problem, highlighting how this sampling feature of `Google Trends` may be specially harmful for exercises involving nowcasts and the use of data vintages. Section 2.3 shows with a simulation exercise how can one strongly improve model's accuracy when using GT as covariates. Section 2.4, the last one before the conclusion, will illustrate with real data how our proposed strategy to deal with GT can improve forecast accuracy.

2.2 How representative of the actual data is each sample?

As previously explained, Google makes available only a small sample of its search database. What most researchers and practitioners seem not to know (or to ignore) is the fact that this small sample is not always the same. In fact, it is constantly changing. This means that someone who downloads `Google Trend` data today will not download the same data tomorrow, even if she filters the same topics, languages and location.

We show with two examples of search topics ("Refined Petroleum" and "GDP Growth") in two different regions (US and Brazil) that this difference in the data will be higher the less often the term is searched. Even though we do not have access to the actual number of searches for each term in each region, it is reasonable to assume that "Refined Petroleum" is less searched than "GDP Growth" and that less people use Google in Brazil than in the US (smaller population and percentage of people with access to the internet). This translates as the "Refined Petroleum" GT search data in Brazil being the more volatile and the "GDP Growth" data in the US the less.

It is important to note that this feature is not something that Google tries to hide. One can find information about the sampling of `Google Trends` as easy as in its FAQ. Nevertheless, most studies using `Google Trends` do not seem to take it into consideration. For example, Pelat et al. (2009), Vicente et al. (2015), Ferrara and Simoni (2019), and Doerr and Gambacorta (2020) do not even mention this fact that each `Google Trends` sampling may result in

different outcomes. Others like Heikkinen (2019), do recognize the importance of what this feature can imply, but do not seem to do anything about it in their estimations.

On the other hand, it is important to stress that there are extremely well executed papers that not only recognize the issue but also propose solutions to it. Good examples of such papers are Amuri and Marcucci (2017), Narita and Yin (2018), Woloszko (2020a), Borup and Schütte (2020), and Borup et al. (2021), among others.

Nevertheless, the important question is how bad can these different samples be to one's forecast model? To answer it, we gathered many different samples from Google searches in the US and in Brazil between January, 2009 to January, 2019 (this way we avoid the period from both the 2008 crisis and the COVID-19 pandemic). In Figure 2.1 we report the values from three different samples for the same term ("GDP Growth") in Brazil and same timestamp. It illustrates well the fact that each GT sample is different from the other, even when we set fixed the same topic, timestamp and region. As one can observe above, the black curve (random sample 1) does not even reach its maximum value in the same date as the other two. In case it is not very clear in the plot the lack of strong correlation between each series.

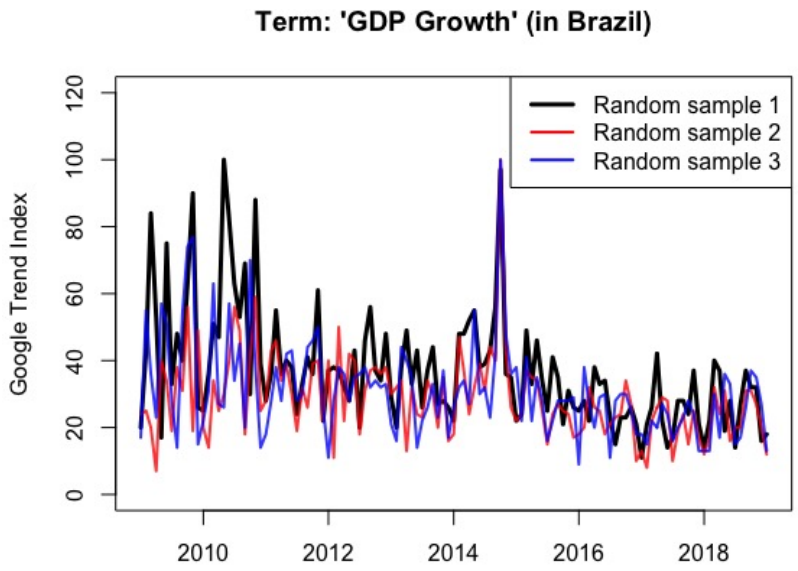

Figure 2.1: Three different samples of same topic and date in Brazil.

The correlation among these series are reported in Panel (a) in Table 2.1. The table shows that the correlation between two different `Google Trends` sample can be as low as 0.496, even when we consider a relatively popular topic in Economics. Moreover, as the following plot and correlation matrix

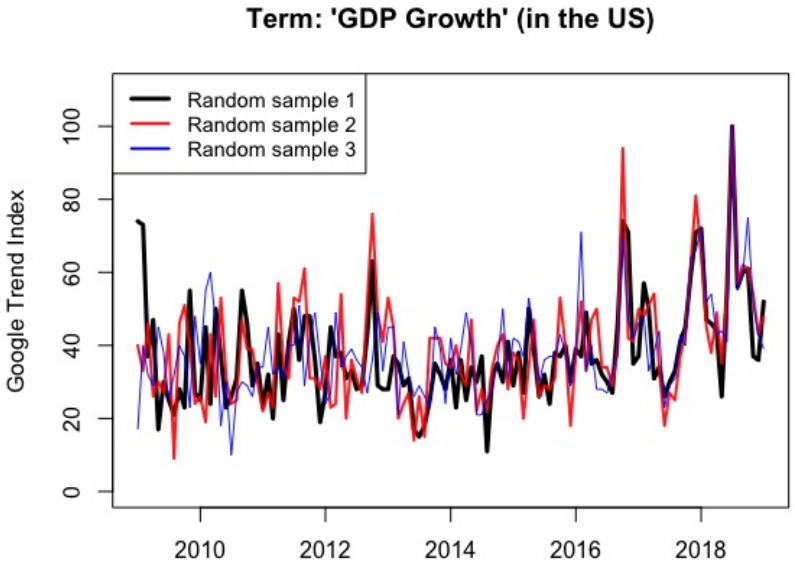

Figure 2.2: Three different samples of same topic and date in the US.

show, this finding is not an exclusivity of Brazil. See, Figure 2.2 and Panel (b) in Table 2.1, where we report the results for the case of US.

Table 2.1: Correlation between three different samples (S).

The table reports the correlation matrix among three different samples for "GDP Growth" searches. Panel (a) reports the cased of Brazil and Panel (b) shows the US case.

	Panel (a): Brazil		
	GDP Growth - S1	GDP Growth - S2	GDP Growth - S3
GDP Growth - S1	1	0.496	0.545
GDP Growth - S2	0.496	1	0.564
GDP Growth - S3	0.545	0.564	1
	Panel (b): US		
	GDP Growth - S1	GDP Growth - S2	GDP Growth - S3
GDP Growth - S1	1	0.655	0.516
GDP Growth - S2	0.655	1	0.575
GDP Growth - S3	0.516	0.575	1

But how can a researcher use such a volatile dataset in her forecast? We propose a very simple solution in the next subsection.

2.2.1
A simple way to circumvent the problem

Figures 2.1 and 2.2 previously displayed illustrate well the motivation of this article, by showing that each `Google Trend` sample may be very different from each other. However, we now argue that there is a simple way to treat this potential problem in order to use GT as a powerful research tool[1].

First of all, it is important to mention that very popular terms (e.g. COVID-19 in 2020) do not vary so much among different samples. However, in many situations our terms of interest are not these very popular ones. So how to overcome the possible problem shown in the plots above? The answer is very simple. By gathering many different samples and averaging across every term, one can get a more reliable time series of that term.

To illustrate how taking averages of many samples improve the series consistency, Figures 2.3 and 2.4 below plot the curve for the same term and same timestamp, but now comparing averages from different samples. Each curve represent the average of "GDP Growth" searches across seven different random samples. It is important to note that if some sample is in one average, it necessarily is not in the other one. As one can observe in the figures, taking averages strongly raise the correlation between the (averaged) samples. Besides that, as expected, the correlation between each series is a little higher in the US (0.95) than in Brazil (0.92).

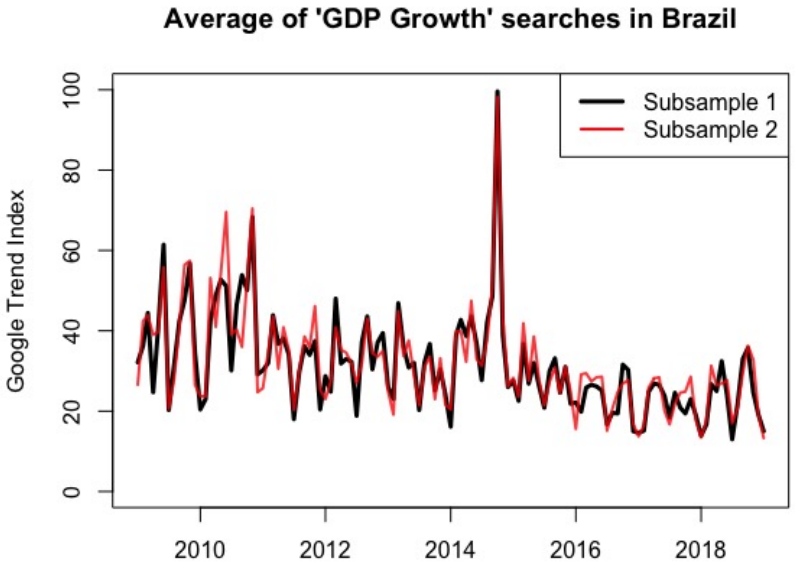

Figure 2.3: Averaged series using different samples in Brazil.

[1]In Appendix 2 we show another setup in which the sampling problem with `Google Trends` may arise and should be treated with care.

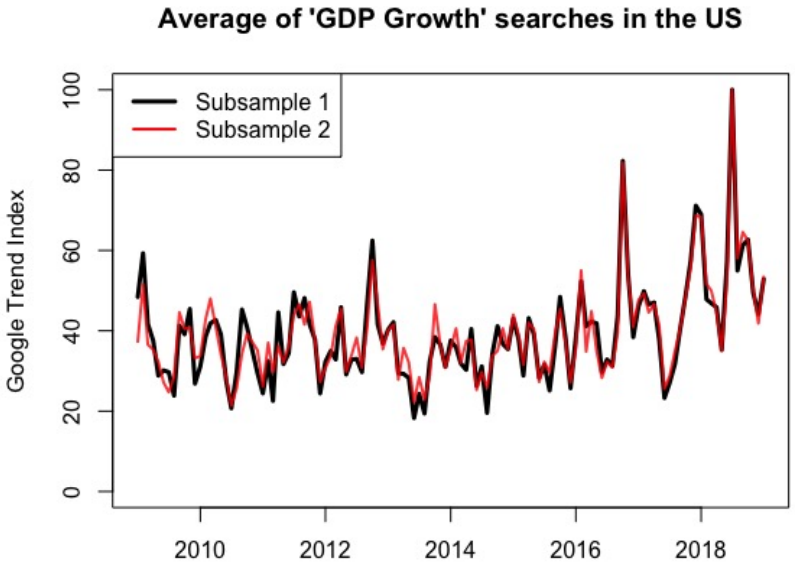

Figure 2.4: Averaged series using different samples in the US.

2.3 Simulation: Model selection improvement

The current section will exemplify through a simulation exercise how can one improve model selection performance simply by using as covariates averages of multiple `Google Trends` samples instead of a single sample. The idea here is to observe the capacity of LASSO (Least Absolute Shrinkage and Selection Operator, Tibshirani (1996)) to select the correct variables (i.e., variables that indeed make part of the Data Generation Process or DGP) in two different linear regression setups.

For this exercise, we gathered 28 different `Google Trends` samples (14 for Brazil and 14 for the US) collected in different days but using the same timestamps and terms (in its respective language). We consider 20 different search terms related to Economics. When the location is Brazil, we search for terms in Portuguese, while when the location is the US, we search for the same terms in English. The searches comprehend the period from January 2009 to December 2018 (120 months). The searches are conducted at the monthly frequency and we compare two different setups.

In the first setup, in each replication we construct different dependent variables using only a few GT terms extracted from a single random sample (from the 14 downloaded). Then, using each one of the 13 remaining samples, we run 13 different LASSO regressions in order to try to identify which variables (among the 20 terms in the full dataset) are indeed in the DGP of that replication. In Table 2.3 below we explain the methodology in an algorithm

format:

Table 2.2: Algorithm of GT simulation: setup 1

Algorithm 1

1: **for** *iteration* $= 1, 2, \ldots, 1000$ **do**
2: sample s = a random integer from 1 to 14
3: sample $\overrightarrow{c} = 5$ random variables from the 20 topics available
4: for k = 1, 2 and 3, construct each $Y^{US}_{k,t,s}$ and each $Y^{BR}_{k,t,s}$ using only variables $\in \overrightarrow{c}$
5: **for** $m \in \{1, 2 \ldots, 14\} - s$ **do**
6: Run a LASSO using sample m to predict each $Yk^{BR}_{t,s}$ and $Yk^{US}_{t,s}$
7: Save which variables were selected by the model to forecast each dependent variable
8: **end for**
9: **end for**
10: Compute average percentage of variables corrected selected by each LASSO.

More specifically, in the first setup, we simulate 1,000 replications with 120 observations each. In each iteration, a random number (s) ranging from 1 to 14 is drawn. This will indicate which one of the 14 `Google Trends` sample we will use to generate the data. Then, we randomly pick five of the 20 search terms to form the set of relevant variables of replication i that will actually be used to construct using sample s ($\boldsymbol{X}_{0,i,t,s}$). Finally, for each location (Brazil or US) we generate three different dependent variables given as

$$\begin{aligned} Y^{BR}_{i,k,t,s} &= \boldsymbol{\beta}^{BR}_{i,k} \boldsymbol{X}^{BR}_{0,i,t,s} + \epsilon^{BR}_{i,k,t,s} \\ Y^{US}_{i,k,t,s} &= \boldsymbol{\beta}^{US}_{i,k} \boldsymbol{X}^{US}_{0,i,t,s} + \epsilon^{US}_{i,k,t,s}, \end{aligned} \tag{2-1}$$

where $i = 1, 2, \ldots, 1000$, $k = 1, 2$ or 3 indexes the DGP and $t = 1, 2, \ldots, 120$. The linear coefficients are determined as follows. $\boldsymbol{\beta}_{i,1}$ is a vector of integers sampled from a discrete uniform distribution in the interval $[-10, 10]$; $\boldsymbol{\beta}_{i,2}$ has elements set to 1 or 2 with equal probabilities; and $\boldsymbol{\beta}_{i,3}$ is sampled from a continuous uniform distribution in the interval $[0, 1]$. Finally, $\epsilon^{BR}_{i,k,t,s}$ and $\epsilon^{US}_{i,k,t,s}$ are independent and normally distributed with zero mean and are set to have the same variance as $\boldsymbol{\beta}^{BR}_{i,k} \boldsymbol{X}^{BR}_{0,i,t,s}$ and $\boldsymbol{\beta}^{US}_{i,k} \boldsymbol{X}^{US}_{0,i,t,s}$.

Then, for each replication in setup 1, after all six dependent variables are constructed, we run 13 different LASSO regressions (each one using one of the 13 remaining samples that are not used in the GDP of that iteration) to then save the topics each one of them selected as predictors. It is important to remember that each of these 13 models predicts each $Y_{i,k,t,s}$ using the full database (which has 20 variables, each one being a time series related to a

different `Google Trends` term), whereas each $Y_{i,k,t,s}$ is constructed using only five random terms among the 20 possible.

In the second setup we do a similar replication exercise, but in this case in each iteration we construct the dependent variables using averaged terms of seven (randomly selected) `Google Trends` samples to then running only a single LASSO regression (using the remaining seven samples among the 14) in order to try to select the correct variables in each DGP. Table 2.3 below explains the procedure as an algorithm:

Table 2.3: Algorithm of GT simulation: setup 2

Algorithm 2

for $iteration = 1, 2, \ldots, 1000$ **do**
sample $\overrightarrow{s} = 7$ random integers from 1 to 14 without replacement
for every topic in each sample (20), compute its average across each sample $\in \overrightarrow{s}$
sample $\overrightarrow{c} = 5$ random variables from the 20 available
for k = 1, 2 and 3, construct each $Yk_{t,s}^{US}$ and each $Yk_{t,s}^{BR}$ using variables $\in \overrightarrow{c}$
Do the same as step 3 but for the other 7 samples $\ni \overrightarrow{s}$
Run a single LASSO using the averaged sample from step 6 to predict each $Y_{t,s}^{BR,US}$
Save which variables were selected by the model to forecast each dependent variable
end for
Compute average percentage of variables corrected selected by each LASSO.

As in setup 1, now in every replication the dependent variables will be constructed using only five random terms (among the 20 topics in each database). However, instead of randomly picking only one among the 14 `Google Trends` data, in each iteration we will randomly draw seven samples, take their average for each topic and then use the five previously selected topics to construct the dependent variables. Then, after all six dependent variables are constructed (as explained in equations 2-1), the algorithm will now run only one LASSO regression (instead of 13) which will use the average of the remaining 7 samples (the ones that are used to construct each Y) as covariates. Finally, the LASSO regression selects which variables (from the 20 topics available - each one being the average of that topic across the seven remaining samples) are in that iteration's DGP. The rest of the process is analogous to the first setup.

Finally, we compare the average (across replications) of correctly selected variables between setups 1 and 2. Our conclusions - which are summarized in

Table 2.4 - are that one can highly increase model selection performance (up to 27% in our simulations) with LASSO when using the average of multiple terms instead of a single sample from `Google Trend`'s website.

Table 2.4: Simulation results

Percentage of variables correctly selected by LASSO in each setup and dependent variable: average across 1000 replications.

Setup	US^1	US^2	US^3	BR^1	BR^2	BR^3
1	51.1	62.7	58.3	56.2	68.1	61.1
2	64.72	73.84	66.82	67.62	76.08	68.76

2.4 Empirical Application

By using a real example, this section will advocate both for the use of `Google Trends` as a powerful prediction tool as well for the use of the average of many samples of the same search pattern as your independent variable, instead of only one sample.

During the pandemic, many authors tried to predict and understand the pattern of COVID-19 cases and death counts with all sort of models, including both epidemiological and statistical ones; see, for example, Medeiros et al. (2022), Bertozzia et al. (2020), Anastassopoulou et al. (2020), Carneiro et al. (2020), and Matjaž et al. (2020b). One of the most important indicators of the actual severity of the pandemic, is the number of cases organized by the date of first symptoms felt by each person as well as the daily new deaths sorted by the day of the event (and not the date of registry). These numbers give a clear indication how the disease is evolving in real-time.

However, in some countries (and specifically in Brazil, as showcased in the first article of this thesis), the actual numbers are available with major delays. For some locations in Brazil, the average delay can exceed a month. Part of the delay is due to the coronavirus cycle but most of it is related to the bureaucracy in the register system.[2] As previously argued, this delay means that by using `Google Trends` (which is available almost in real-time) one could have a good insight of the direction in which the numbers would be moving towards two weeks in advance.

However, would it be possible to correctly nowcast the trend of SARS (syndrome respiratoire aigu sévère, strongly associated with COVID-19 in

[2]For further reference with respect to the Brazilian case, please visit the following link.

2020) cases in Brazil using merely a handful of `Google Trends` topics as predictors?[3]

To answer this question, we consider a simple LASSO regression to predict the trend of SARS new cases and new deaths in 2020 in Brazil. For each dependent variable (new cases and new deaths) we train eight LASSO regressions (each one using a different `Google Trends`' sample of the same topics, timestamps and location, but collected in different days) from February to the end of May 2020 using only `Google Trends` data with terms related to the pandemic, such as "COVID symptoms", "COVID ICU" and "coronavirus hospital" as predictors[4]. Figure 2.5 displays the results of the nowcasting model for SRAG's new cases. The black curve in the figure represents the actual number of new SARS cases aggregated by the reported day of the first symptoms felt by each person (known in the time we wrote this article, which was March, 2021). The colored curves represent the point forecasts of the eight different samples of the same `Google Trends` terms starting on June 1, 2020.

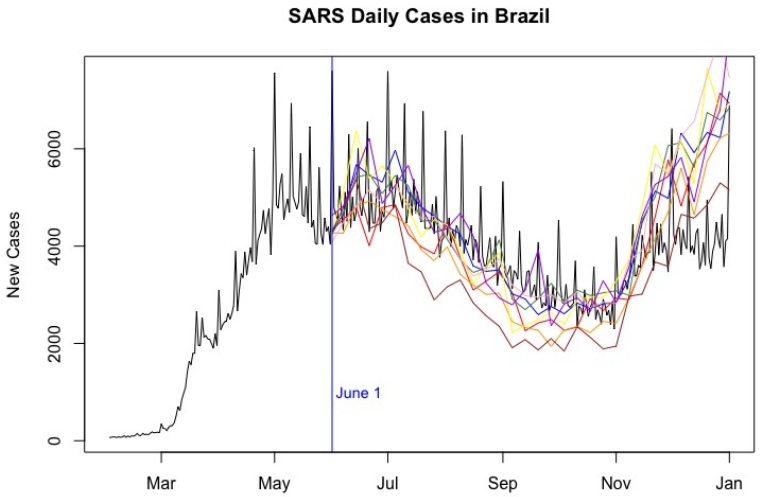

Figure 2.5: Nowcast of SARS daily cases in Brazil using only Google Trends.

As it is clear in Figure 2.5, it would have been possible to nowcast in real-time at least the trend of SARS new cases using either one of the samples (of course, some samples would have been better than others). In other words, by using only `Google Trends` data, one could predict that the number of new cases would continue on a high plateau until August, then it would start falling until Mid-November to then start rising again until the last day of the year. Moreover, by looking at Figure 2.6 below, we could conclude that it would

[3]Naturally, this would be a less sophisticated nowcasting model when compared to the one in our first article, but that is not the point.

[4]For obvious reasons, each of these terms translated to Portuguese.

also have been possible to predict the trend of SARS new deaths by using only `Google Trends` data in a linear model trained months before.

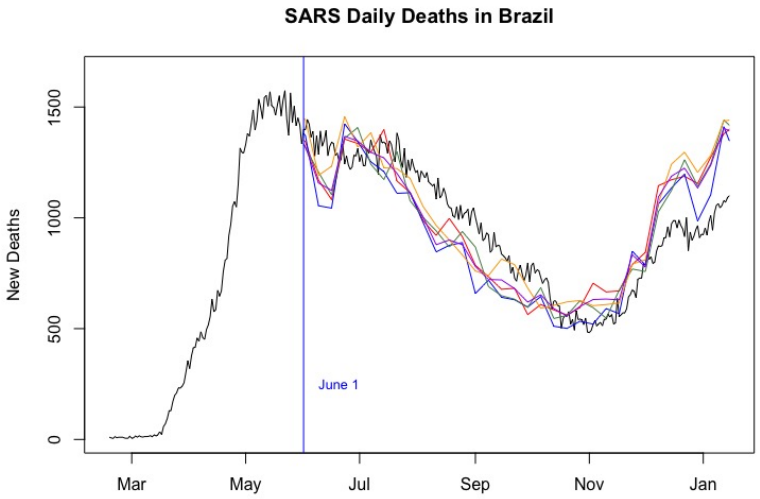

Figure 2.6: Nowcast of SARS daily deaths in Brazil using only Google Trends.

However, there is another thing to consider. Even though the eight predictions (using different samples) are similar, they are far from the same. This is specially meaningful if we take into consideration that all of the topics used in the models described above could be considered very popular terms in the year of 2020.

To understand how using the average of the eight samples across each term would improve the forecasts results for both dependent variables, we compared the Root Mean Squared Error (RMSE) of each model (New cases and New deaths) using averaged series (Proposed Model) with the other eight models that used `only one Google Trends` sample. The results are displayed in Table 2.5.

Table 2.5: Nowcasting results

The table reports the Root Mean Squared Error (RMSE) of the nowcasting LASSO regression using the average of the `Google Trend` samples as regressors as well as the RMSE for the sample-specific models with the worst, best and average performance.

	Proposed Model	Worst	Best	Average
New cases' RMSE	888.6	1340.97	900.6	1058.13
New deaths' RMSE	149.2	175.3	144.4	159.3

As one can infer above, for the eight new cases' models, not only none of them had a smaller RMSE than the average, but we found out that the RMSE was up to 51% higher in the models using only one sample. For the new death's models, the proposed method had a RMSE more than 6% smaller than the average RMSE of the other models and more than 17% smaller than the worse one.

These numbers above illustrate the fact that merely by chance an uninformed researcher could find herself in one of the following situations when using only a single sample of `Google Trends` in her studies. First, it could be possible that a given GT term was actually a good predictor of certain a variable of interest, but she may unfortunately find herself working with a specific random GT sample that does not represent very well the true population. In this case, she may end up concluding that there is not a significant relationship between the searches on the term and the dependent variable. The other situation is the one that she may find an apparent significant relation when in reality there is none. By averaging multiple samples, one can diminish this risk.

2.5 Conclusions

This article has reinforced the usefulness of internet search data as a powerful forecast tool. However, it has also highlighted some limitations of these data, such as the lack of information on the actual volume of searches, but mainly the fact that Google Trends' index is based on a subsample that is changing all the time. We explained that this feature might become a real problem for the forecaster when dealing with less frequent searched topics and/or when filtering the query to many years in the past.

We then illustrated with two examples using both simulated and real data that simply by taking averages of many different samples with the same specifications (i.e. same topic, timestamp and location) it is possible to improve both model selection and its forecast accuracy.

3
Real-time forecasting in a data-rich environment: the benefits of machine learning methods.

3.1
Introduction

Most traditional statistical techniques are usually intended for settings in which the number of observations is much greater than the number of covariates. According to James and Tibshirani (2013), this is due in part to the fact that, until recently, the bulk of scientific problems requiring the use of statistics have been low-dimensional. However, in the past two decades, new technologies have changed the way that data are collected. Collecting an almost unlimited number of feature measurements is now trivial, while the number of observations is often limited due to cost, sample availability, or other considerations. These data sets containing more features than observations are often referred to as high-dimensional or big data.

Simultaneously, throughout these past years many methods have been developed to perform particularly well in the high-dimensional setting. Three famous types of methods are the shrinkage operators, like the LASSO (Tibshirani (1996)), the dimension reduction methods, such as Principal Components Regression (PCA) and non-linear tree models, such as the Random Forecast Breiman (2001).

The main goal of this article is to access the forecasting accuracy of several economic variables in *real-time*,*horse-race* approach, using both linear ML models, as well as some of its nonlinear alternatives, such as the famous XGBoost (Chen and Guestrin (2016)). For this purpose, we compare *real-time* performances of ML models versus benchmark models not only in the full out-of-sample window, but also across recession (namely during the Great Financial Crisis and the recent COVID-19 pandemic) periods.

The following subsection will explain what we mean by *real-time forecasting*, while illustrating its modeling differences when compared to the usual *latest data available* approach.

3.1.1 Why real-time matters?

When we say our predictions are all made in real-time, we mean that a forecast for a certain variable in a given period must be computed by using only information available to the economist before that given moment.

With the purpose of illustrating how this setup may affect any type of prediction, the following Figure 3.1 displays the series of the percentage variation of the CPI and of the M1 Money Stock (M1SL) in the US, two of the dependent variables used in this paper. In both panels below, the black curve displays the real-time series of each variable[1] and the red one displays the series available in the last FRED-MD vintage used in this article.

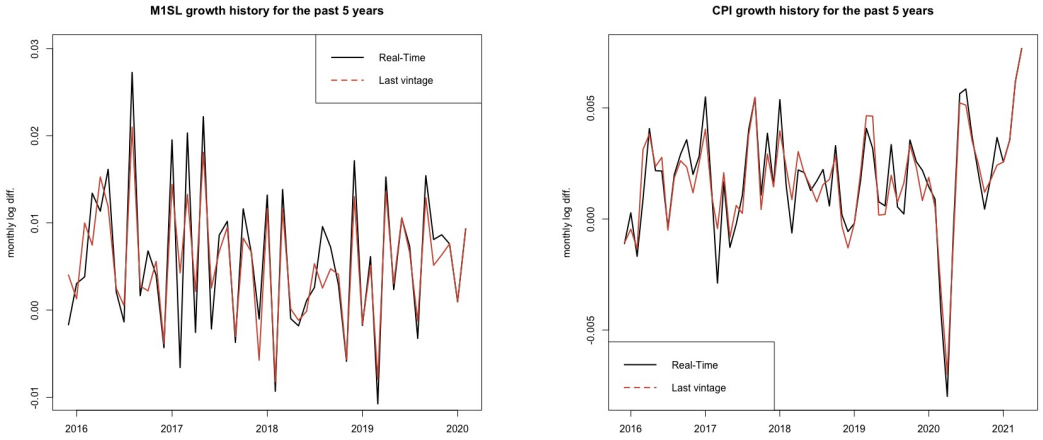

Figure 3.1: Real-time vs. last vintage approach for the M1 Money stock and the CPI.

As one can see above, the red and the black curves are considerably different in most of the months. This means that the two variables above are constantly subject to revision. In other words, most of the times, what we now know today about each of the series are not what the econometrician knew in each of the months plotted above.

Many relevant studies such as Medeiros et al. (2021), Coulombe et al. (a) and Coulombe et al. (b) (to only cite a few) already evaluated the performance of ML models vis-a-vis simpler benchmark models. However, some of the times the *real-time* factor was not taken into consideration and/or the evaluation was not done over several different target variables as we do here. That being said, the main message of this last article is that even in a real-time context, big-

[1]In order to construct this series, we took the last observation of each variable in each vintage since the end of 2015. More on the dataset in the next section.

data combined with the right ML tools can achieve substantial accuracy gains for many economic variables.

The remainder of this article is organized as follows. Section 2 gives an overview of the dataset and the dependent variables used in the estimations. Section 3 describes the forecasting methodology. Section 4 presents our results, dividing them in the CPI and the overall subsection . Finally, section 5 concludes.

3.2 Data

Our data consist of all variables from the FRED-MD database, a large monthly macroeconomic dataset designed for empirical analysis in data-rich environments, which is updated in real-time through the FRED database.[2]

Our sample goes from January 1960 to December 2020 (732 observations), in which the out-of-sample window starts in 2000 and goes until the end of 2020. Since our estimations are in real-time, we use a different vintage for each of these windows. The first vintage we use is the one of October, 2000 and the last one is the one of January, 2021. All variables are transformed as described in the end of this document.

One thing that is important to highlight is that since we are using a different vintage (dataset) for each window, the set of covariates may slightly change in each iteration. This is due to the fact that while some series are constantly being removed, others are being added and others revised.[3] As previously argued, we call this real-time because in this setup we replicate the models (with the same dataset, i.e., variables and observed values for them) the econometrician who would run them in those months would get.

In addition to running everything in real-time, this study constructs several different models in order to forecast a number of different economic variables and horizons. More specifically, we have results for 12 dependent variables, from several different groups of FRED database. From all these variables, five are related to the Interest and Exchange rates group, two to Output and Income, another two to Prices group, one to the Labor Market and another to the Money and Credit group. We will describe in the next subsection which are these variables and how they were selected.

[2]It is available from Michael McCraken's webpage. For further details, we refer to McCracken and Ng (2015)

[3]For further details, we refer to https://files.stlouisfed.org/files/htdocs/fred-md/fredmdchanges.pdf

3.2.1
Choosing a dependent variable

In order to select our group of dependent variables, two main filters were considered. As previously briefly explained, some variables are often removed. In this sense, our first filter was that all of our dependent variable must be present in every vintage of FRED-MD database. Second, it should not contain any missing value in any of the vintages. Many variables contained missing values either in the most earlier years (during the 1960's, for example) or in the last month of each vintage.

After applying these two filters, we ended up using the 12 dependent variables displayed in Table 3.1 below:

Table 3.1: Set of dependent variables used in this article.

Variable Description	**Variable Code**	**Group**
CPI : All Items	CPIAUCSL	Prices
Crude Oil, spliced WTI and Cushing	OILPRICEx	Prices
S&P's Common Stock Price Index: Composite	S&P 500	Stock Market
Effective Federal Funds Rate	FEDFUNDS	Interest and exchange rates
1-Year Treasury Rate	GS1	Interest and exchange rates
10-Year Treasury Rate	GS10	Interest and exchange rates
Moody's Seasoned AAA	AAA	Interest and exchange rates
U.S. / U.K. Foreign Exchange Rate	EXUSUKx	Interest and exchange rates
M1 Money Stock	M1SL	Money and credit
Civilian Unemployment Rate	UNRATE	Labor market
IP Index	INDPRO	Output and income
Capacity Utilization: Manufacturing	CUMFNS	Output and income

3.3
Methodology

The methodology follows an usual setup of forecasts based on a rolling window framework of fixed length, except for the fact that we use a different dataset (vintage) for each window. Consider the following model:

$$Y_{t+h} = M_h(\mathbf{x}_t) + u_{t+h}, \tag{3-1}$$

where Y_{t+h} is the dependent variable in month $t+h$; h $\in [1, ..., 12]$ is the forecasting horizon; $\mathbf{x}_t = (\mathbf{x}_{1t}, ..., \mathbf{x}_{kt})'$ is a k-vector of covariates, containing lags of Y_t and/or common factors [4], as well as a large set of potential predictors;

[4]following the best practices of macroeconomic data transformations as described in Coulombe et al. (b).

$M_h(\cdot)$ is the mapping between covariates and the future dependent variable; finally, u_{t+h} is a zero-mean random error.

The target function $M_h(\mathbf{x}_t)$ indicates that there is a different mapping for each forecasting horizon. Since we do not try to predict any of the predictors, we only consider direct forecasts.[5]

Even though our out-of-sample period is fixed between September, 2000 and December, 2020, the actual in-sample number of observations depends on the forecasting horizon. As our forecasts are based on a rolling-window framework of fixed length[6], the number of observations in each window is given by W_h = 488-h-p-1, where p is the number of lags in the model and 488 is the number of months between 1960 and September, 2000.

In addition to two benchmark specifications (random walk (RW), auto-regressive models (AR)), we consider several different types of ML methods[7] including factor-augmented AR models (a standard one, two varieties of target factor similar to Bai and Ng (2008) and a boosted factor model as in Bai and Ng (2009)), models related to the Lasso-family (LASSO, adaLASSO, ElNet, adaElNet and Ridge), a Bayesian VAR, the Complete Subset Regression (CSR from Elliot et al. (2013)), the RF, the XGBoost and Neural Nets.

3.4 Results

As previously argued, one of the main objectives of this article is to expand on previous findings of ML models accuracy in forecasting, specially on those which use a similar set of models and dataset, but do not do it in real-time. That being said, the first part of the results will be exclusively directed to the CPI as the dependent variables. Only then, in the following subsection we will address the overall results.

3.4.1 CPI Results

This section will discuss the results for the CPI as the dependent variable. We start analyzing Table 3.2, which displays the results for the full out-of-sample period. In short, several of the main conclusions of Medeiros et al. (2021) still holds in a real-time setup and expanding the estimations until the end of 2020, a year that we started facing a huge global pandemic.

[5]The only exception is the BVAR model, where joint forecasts for all variables are computed as in Banbura et al. (2010a)

[6]As usual in this literature (Medeiros et al. (2021)), we choose this framework mainly to attenuate the effects of potential structural breaks.

[7]More information about each method is available in Appendix 3

For instance, apart from a few horizons, the RF alternative delivers the smallest ratios in most of the cases, followed closely by shrinkage models. Moreover, the fact that Factor models have very poor results still holds for most horizons and therefore are included in the MCS[8] less often[9].

We also find that when factors are combined with boosting (Boost model in the table), there is a small gain, even though it still shows an inferior accuracy when compared to the aforementioned models. We will see in the next section that the factors combined with boosting do perform well overall. Another thing that stands out and is comparable to the literature is that all the competing models outperform the RW for the vast majority of the horizons.

Table 3.2: Forecasting Errors for CPIAUCSL since 2000.
RMSE are displayed in odd lines (with 1 as a superscript besides the model name) while MAE are displayed in even lines (with 2 as a superscript besides the model name). Cells in blue (gray) show the models included in the 50% model confidence set (MCS) using the squared error (absolute error) as loss function.

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	MCS Count
RW¹	1	1	1	1	1	1	1	1	1	1	1	1	0
RW²	1	1	1	1	1	1	1	1	1	1	1	1	0
AR¹	0.89	0.8	0.76	0.77	0.75	0.77	0.76	0.75	0.75	0.79	0.81	0.76	9
AR²	0.86	0.76	0.72	0.73	0.73	(0.74)	(0.71)	0.71	0.73	(0.76)	(0.8)	(0.73)	5
RF¹	0.82	0.73	0.69	0.71	0.71	0.71	0.71	0.7	0.69	0.72	0.74	0.69	12
RF²	(0.73)	(0.66)	(0.64)	(0.68)	(0.68)	(0.69)	(0.66)	(0.65)	(0.65)	(0.67)	(0.7)	(0.64)	12
LASSO¹	0.77	0.73	0.7	0.72	0.72	0.72	0.73	0.71	0.71	0.76	0.79	0.71	12
LASSO²	(0.71)	0.67	(0.64)	(0.68)	(0.68)	(0.69)	(0.67)	(0.66)	(0.67)	(0.71)	(0.75)	(0.69)	11
RIDGE¹	0.78	0.84	1.18	0.92	1.14	1.27	0.79	0.81	0.76	0.8	0.84	0.76	8
RIDGE²	(0.73)	0.75	0.83	0.92	0.98	0.88	(0.75)	0.76	0.74	(0.77)	(0.85)	0.77	4
ELNET¹	0.77	0.72	0.7	0.71	0.71	0.72	0.73	0.71	0.71	0.76	0.79	0.7	12
ELNET²	(0.71)	(0.66)	(0.64)	(0.67)	(0.68)	(0.69)	(0.67)	(0.65)	(0.67)	(0.71)	(0.75)	(0.67)	12
ADA¹	0.77	0.73	0.7	0.72	0.72	0.71	0.72	0.71	0.71	0.76	0.78	0.71	12
ADA²	(0.7)	0.67	(0.64)	(0.68)	(0.69)	(0.68)	(0.67)	(0.66)	(0.67)	(0.7)	(0.75)	(0.69)	11
ADAELNET¹	0.77	0.72	0.7	0.71	0.71	0.72	0.73	0.71	0.71	0.76	0.79	0.7	12
ADAELNET²	(0.71)	(0.66)	(0.64)	(0.67)	(0.68)	(0.69)	(0.67)	(0.65)	0.67	(0.71)	(0.75)	(0.68)	11
RF.OLS¹	0.77	0.72	0.7	0.71	0.72	0.72	0.74	0.73	0.73	0.78	0.8	0.76	12
RF.OLS²	(0.72)	(0.66)	(0.65)	(0.68)	0.71	(0.7)	(0.69)	(0.69)	0.7	(0.74)	(0.79)	(0.77)	10
LASSO.RF¹	0.8	0.77	0.71	0.76	0.76	0.75	0.73	0.71	0.74	0.76	0.78	0.71	11
LASSO.RF²	(0.71)	0.7	(0.66)	0.72	0.72	(0.72)	(0.69)	(0.67)	0.72	(0.74)	(0.77)	(0.69)	8
BOOST¹	0.96	0.75	0.72	0.73	0.73	0.76	0.76	0.73	0.72	0.76	0.79	0.71	10
BOOST²	0.85	0.69	0.67	0.71	(0.7)	(0.72)	(0.7)	(0.68)	(0.68)	(0.72)	(0.77)	(0.7)	8
BVAR¹	0.77	0.78	0.79	0.79	0.78	0.79	0.8	0.81	0.8	0.82	0.82	0.8	8
BVAR²	(0.72)	0.75	0.76	0.76	0.77	(0.76)	0.75	0.78	0.78	0.79	(0.79)	0.78	3
FACTORS¹	0.82	0.79	0.86	0.82	0.85	0.88	0.89	0.84	0.84	0.87	0.89	0.81	9
FACTORS²	0.78	0.71	0.74	0.78	0.8	0.81	(0.79)	0.76	0.78	(0.81)	(0.86)	(0.78)	4
TFct¹	0.81	0.77	0.76	0.75	0.81	0.74	0.79	0.78	0.78	0.81	0.8	0.74	10
TFct²	0.8	0.71	0.7	0.74	0.79	(0.72)	(0.75)	0.75	0.75	0.78	(0.8)	0.73	3
ATFct¹	0.79	0.86	0.77	0.76	0.89	0.8	0.79	0.82	0.78	0.84	0.79	0.75	10
ATFct²	0.76	0.74	0.71	0.77	0.81	0.81	(0.75)	0.78	0.76	(0.82)	(0.81)	0.77	3
CSR¹	0.83	0.72	0.69	0.71	0.71	0.72	0.73	0.71	0.73	0.76	0.78	0.69	11
CSR²	0.78	(0.67)	(0.64)	(0.67)	(0.68)	(0.69)	(0.68)	(0.67)	0.69	(0.72)	(0.74)	(0.65)	10
XGB¹	0.84	0.77	0.72	0.71	0.75	0.74	0.73	0.74	0.73	0.78	0.78	0.71	11
XGB²	0.76	0.7	(0.66)	(0.69)	0.72	(0.71)	(0.69)	(0.69)	0.71	(0.74)	(0.77)	(0.68)	8
NN¹	1	0.96	0.85	0.94	0.93	0.86	0.81	0.77	0.81	0.93	0.81	0.77	7
NN²	0.88	0.82	0.8	0.88	0.86	0.84	(0.77)	0.74	0.78	(0.8)	(0.83)	0.78	3

[8]for further reference, we refer to Hansen et al. (2011)
[9]We will see that this finding also holds for the overall results.

The aforementioned findings are well corroborated and supplemented by Figure 3.2 below. There, one can see rolling-windows of 24 observations of both the RMSE and the MAE of the CPI forecasts made by the RW, the AR and by the "median" ML model. By "median" ML model we mean the model which presented the median error (RMSE or MAE, dependind on the panel) across all ML models for that window.

Considering that we showed that some models didn't show much improvements over the AR (like the three factor models, a result that was also found in other studies), having the "median" ML error almost always below the AR error is a good finding. More specifically, if we were to plot the RW and the AR errors against the best models such as the LASSO or the RF, the curves would have been even more far apart.

3.4.1.1 Model accuracy during recession times for the CPI

Finally, one thing that might be interest to study is the accuracy of the models during recession times. In Figure 3.3, one can see a 6 months rolling window of forecast errors for the CPI variable. The black line indicates the AR's RMSE relative to the RW's, while the green one displays the median relative RMSE of all Machine Learning models for the 1-step ahead forecast.

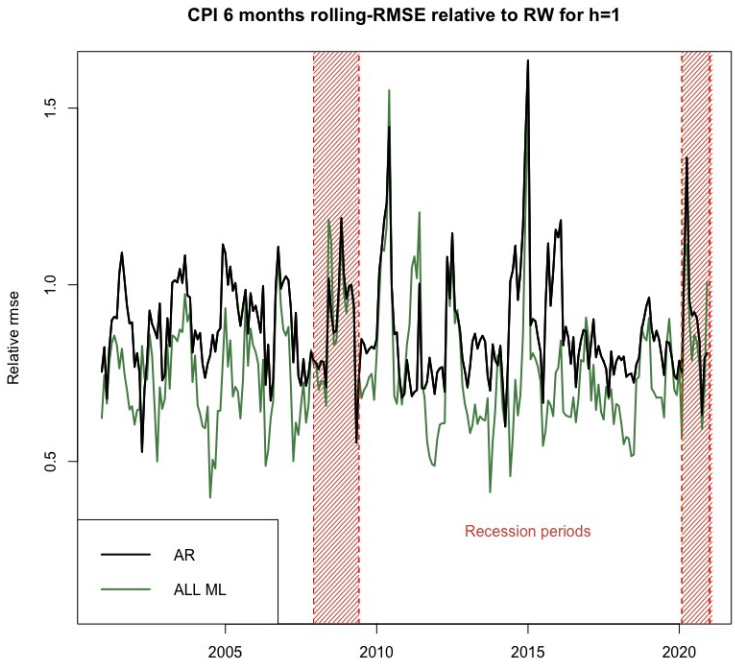

Figure 3.3: The figure displays the 1-step ahead (h=1) relative RMSE, computed over rolling windows of 6 months. Shaded red lines indicate recession periods.

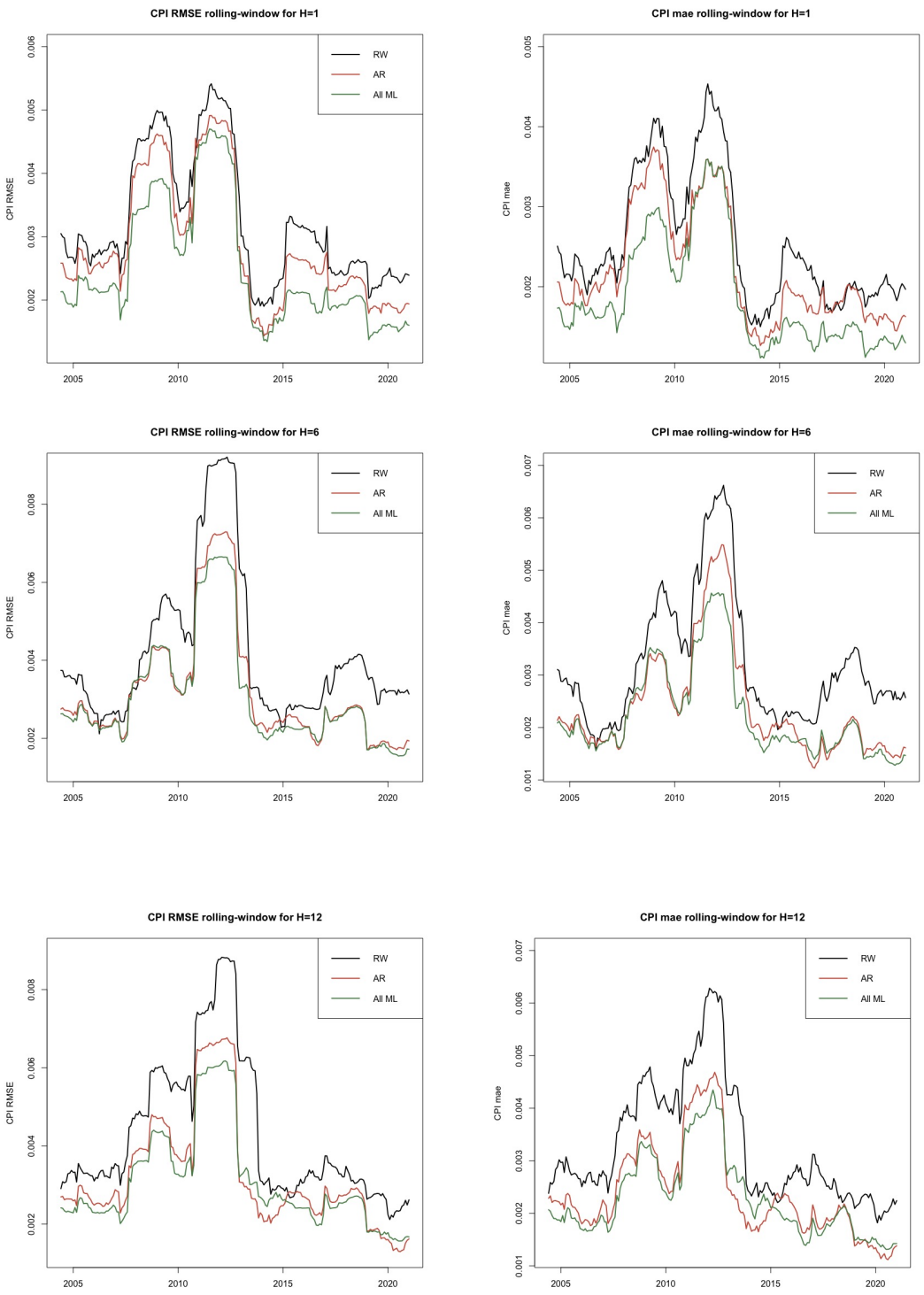

Figure 3.2: The figure displays the root mean squared errors (RMSE) and mean absolute error (MAE) computed over rolling windows of 24 observations. Panel (a) displays the results for one-month-ahead forecasts (h = 1), panel (b) displays the results for six-months-ahead forecasts (h = 6), while panel (c) displays the results for twelve-months-ahead forecasts (h = 12).

The idea of the above plot is to illustrate the following results: while the AR errors are smaller than the median ML's ones only 11% of the time when the American economy was not facing an recession, it is 25% of the time smaller during recessions[10]. The same idea seems to be true when we do the same analysis, but instead of just looking to the CPI, we do it for all other dependent variable: the AR seems to be more accurate than the median of all ML models (naturally, this includes models that don't seem to perform very well) 51% of the time during recessions. This ratio halves to 25% during non-recession periods.

In other words, it seems that ML models seem to forecast the CPI one period ahead relatively worse in periods of recession. However, this finding don't seem to be valid for all other forecast horizons. For example, for 6-steps ahead forecast (h=6) we find that the number of the months in which the AR predicts the CPI better than the median ML model is actually 50% higher in periods of no recession. At the same time, when we verify the same results but for h=12, ML models again seem to better predict CPI during times of no recession. In these months, AR's RMSE is only 25.7% of the time smaller than the median ML model's RMSE. This frequency grows to 40.7% during recessions.

3.4.2 Overall Results

In the previous subsections we presented the results specifically to the CPI variable. Now we are going to discuss our findings related to the other dependent variables.

In general, the fact from the previous subsection that ML models tend to perform better than the benchmark models still holds. Table 3.3 summarizes this idea.

It is important to highlight that most of the ML models tested in this paper (NN being the worse one) didn't show the highest RMSE or MAE in none of the 142 forecasting possibilities.[11] More specifically, none of the models belonging to the LASSO family (LASSO, El-net, Adalasso, Ada-El-net and Lasso-RF), the RF, the Boosting or the the CSR had the highest error not even once. At the same time, these models accounted for most of the percentage of minimum errors.

[10]The recession periods considered here are the 19 months between December, 2007 and June, 2009 and the 11 months between February, 2020 and December, 2020.

[11]One interest fact is that the Ridge model was constantly a model that showed both the highest and the lowest errors in the estimations, depending on the dependent variable and/or horizon.

Table 3.3: Overall individual performances.

Frequency of the times each model showed the smallest or highest (measured by RMSE or MAE) across all 142 forecasting possibilities (12 variables and 12 horizons).

Error Measure	RMSE		MAE	
Model	Min	Max	Min	Max
RW	0%	**57%**	0%	**73%**
AR	9.7%	0%	6.25%	0%
RF	**16%**	0%	8.3%	0%
LASSO	7.5%	0%	9.7%	0%
RIDGE	**18%**	**18%**	**18.7%**	**11.8%**
ELNET	**10.4%**	0%	**16.7%**	0%
ADA	**10.4%**	0%	9%	0%
ADAELNET	6.9%	0%	9.7%	0%
RF-OLS	3.5%	3.5%	1.4%	2.1%
LASSO-RF	2.1%	0%	2.1%	0%
Boosting	**10.4%**	0%	**10.4%**	0%
BVAR	0%	2.8%	0%	2.1%
Factors	1.4%	**11.8%**	1.4%	**3.4%**
Target Factors	0%	4.9%	0%	**3.4%**
Alt. Target Factors	0%	2.1%	0%	2.1%
CSR	3.5%	0%	5.5%	0%
XGB	0%	0%	0%	0%

On the other hand, the RW[12] was by far the model showing the worse accuracy most of the times and it didn't show the best result not even once. Besides that, even though the AR, the other benchmark model, didn't show any worse forecast results, generally speaking it also didn't perform extremely well. It showed the ninth smallest MAE (from 17 models excluding RW) and the seventh smallest RMSE, while, for instance, the RF (Elastic-Net) had the seventh (second) and the second (third) position, respectively.

Figure 3.4 below illustrates a similar idea presented in previous sections, i.e., the one that ML models tend to outperform both benchmark models studied here. It displays a 24 month rolling-window of RMSEs and MAEs.

It is important to highlight that now we are comparing errors across different dependent variables. So, in order to arrive to each series plotted in panel (a) and (b), we computed their "relative errors". For instance, to arrive at each point of the black curve of panel (a), we computed 24 rolling-window of

[12]Table 3.3 leaves the NN model out, as it seems to be outperformed many times by a simple Random Walk model.

AR's RMSE in relation to RW's RMSE over that same period for all dependent variables, to only then take the median value of these numbers. We do that for the median ML model (the red curve, the same way we did with the CPI plots) and for the Elastic-net (the green one).

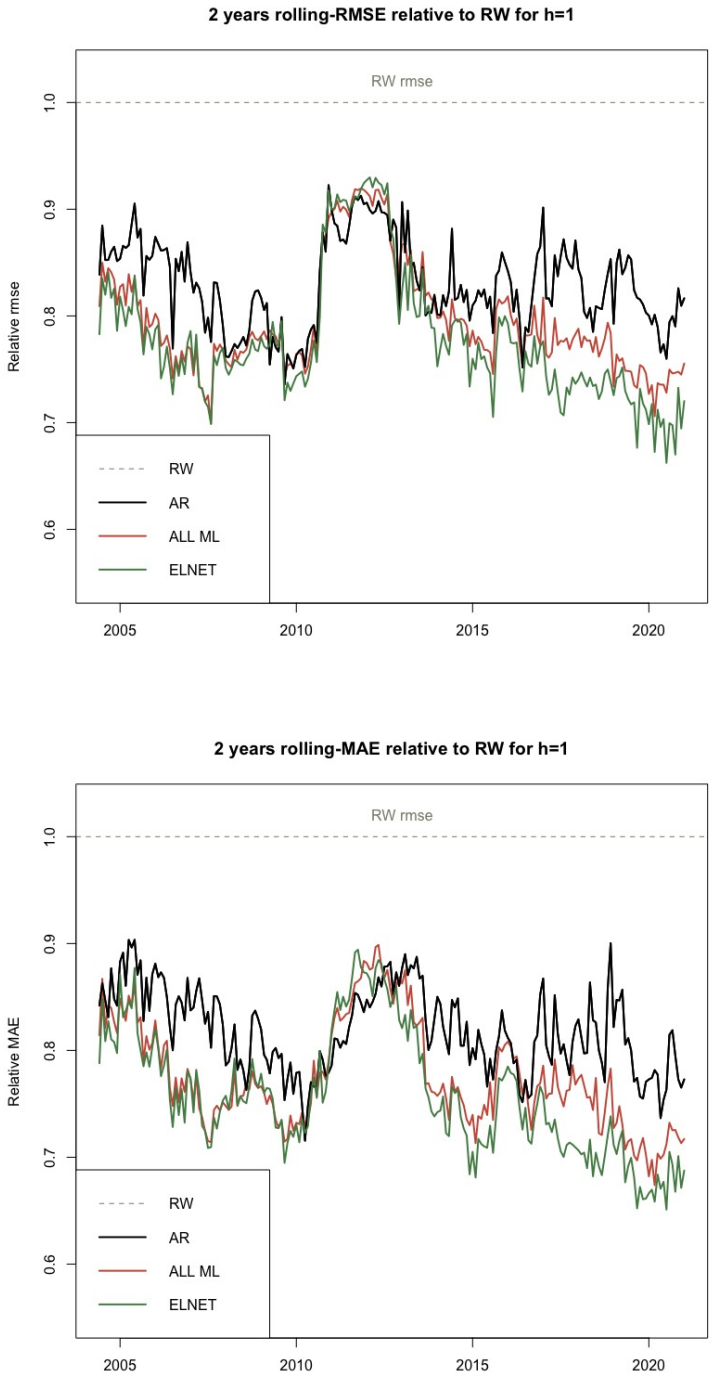

Figure 3.4: Displays the 1-step ahead (h=1) median relative errors across every used dependent variable, computed over rolling windows of 24 observations. Panel (a) shows the relative (to the Random Walk model errors) RMSE, while panel (b) displays the relative MAE.

We can see in Figure 3.4 above that ML models tend to present smaller errors than the benchmark models overall. This is even more highlighted when we plot only the elastic-net model (one of the models that had a best overall performance).

Finally, and just as a matter of illustration, Table 3.4 displays the results for the S.P.500. The reader can find the results table for every other dependent variable in Appendix 3.

Table 3.4: Forecasting Errors for SP500 since 2000.
RMSE are displayed in odd lines, while MAE are displayed in even lines (with 2 as a superscript besides the model name). Cells in blue (gray) show the models included in the 50% model confidence set (MCS) using the squared error (absolute error) as loss function.

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	MCS Count
RW¹	1	1	1	1	1	1	1	1	1	1	1	1	0
RW²	1	1	1	1	1	1	1	1	1	1	1	1	0
AR¹	0.8	0.70	0.72	0.75	0.75	0.69	0.72	0.72	0.7	0.71	0.72	0.72	10
AR²	(0.77)	0.70	0.69	(0.71)	(0.75)	(0.71)	(0.72)	(0.73)	(0.68)	(0.73)	(0.71)	(0.69)	10
RF¹	0.83	0.78	0.8	0.8	0.81	0.72	0.77	0.75	0.73	0.74	0.74	0.76	6
RF²	(0.8)	0.77	0.77	0.78	0.82	0.76	0.78	0.77	(0.71)	0.76	0.73	0.73	2
LASSO¹	0.79	0.70	0.71	0.73	0.75	0.68	0.71	0.72	0.7	0.71	0.73	0.72	11
LASSO²	(0.76)	0.69	0.69	(0.7)	(0.75)	(0.7)	(0.71)	(0.73)	(0.69)	(0.72)	(0.71)	(0.68)	10
RIDGE¹	0.89	0.97	0.98	0.73	0.75	0.75	0.79	0.76	0.75	0.72	0.72	0.89	7
RIDGE²	0.86	0.94	0.95	0.7	(0.75)	(0.76)	0.79	(0.78)	(0.72)	(0.73)	(0.7)	0.82	6
ELNET¹	0.78	0.70	0.71	0.73	0.75	0.68	0.71	0.72	0.7	0.71	0.72	0.72	11
ELNET²	(0.76)	0.69	0.69	0.7	(0.75)	(0.7)	(0.71)	(0.72)	(0.68)	(0.72)	(0.71)	(0.68)	10
ADA¹	0.79	0.70	0.71	0.73	0.75	0.68	0.71	0.72	0.7	0.71	0.73	0.72	10
ADA²	(0.76)	0.69	0.69	0.7	(0.75)	(0.7)	(0.71)	(0.73)	(0.69)	(0.72)	(0.71)	(0.68)	10
ADAELNET¹	0.78	0.70	0.71	0.73	0.75	0.68	0.71	0.72	0.7	0.71	0.72	0.72	10
ADAELNET²	(0.75)	0.69	0.69	0.7	(0.75)	(0.7)	(0.71)	(0.72)	(0.68)	(0.72)	(0.71)	(0.68)	9
RF.OLS¹	0.82	0.74	0.76	0.75	0.75	0.69	0.69	0.81	0.74	0.73	0.74	0.73	10
RF.OLS²	(0.78)	0.73	0.74	(0.72)	(0.76)	(0.72)	(0.72)	(0.77)	0.71	0.74	0.73	(0.7)	7
LASSO.RF¹	0.83	0.74	0.74	0.74	0.77	0.71	0.73	0.72	0.72	0.74	0.75	0.74	6
LASSO.RF²	(0.8)	0.73	0.72	(0.73)	(0.77)	(0.74)	0.73	(0.73)	(0.7)	0.75	0.73	0.71	6
BOOST¹	0.8	0.71	0.72	0.74	0.76	0.68	0.71	0.73	0.71	0.71	0.72	0.72	9
BOOST²	(0.78)	0.70	0.69	(0.71)	(0.75)	(0.71)	(0.72)	(0.73)	(0.68)	(0.72)	(0.7)	(0.68)	10
BVAR¹	0.82	0.76	0.76	0.76	0.81	0.76	0.76	0.77	0.75	0.77	0.77	0.77	4
BVAR²	(0.79)	0.76	0.76	0.75	0.8	0.78	0.77	0.79	0.73	0.79	0.75	0.74	1
FACTORS¹	0.98	0.87	0.78	0.77	0.79	0.77	0.77	0.76	0.77	0.75	0.75	0.74	5
FACTORS²	(0.83)	0.77	0.74	(0.74)	(0.78)	(0.78)	0.77	0.77	0.73	0.75	0.74	0.7	4
TFct¹	0.88	0.79	0.75	0.88	0.79	0.87	0.75	0.79	0.74	0.86	0.75	0.76	5
TFct²	(0.82)	0.75	0.73	0.81	0.78	0.83	0.76	0.79	0.73	0.82	0.75	0.73	1
ATFct¹	0.95	0.81	0.8	0.83	0.82	0.79	0.83	0.79	0.76	0.79	0.8	0.78	3
ATFct²	0.83	0.79	0.77	0.8	0.84	0.83	0.83	0.81	0.75	0.8	0.8	0.76	0
CSR¹	0.82	0.74	0.74	0.74	0.76	0.68	0.72	0.73	0.71	0.73	0.74	0.74	7
CSR²	(0.77)	0.72	0.71	(0.71)	(0.75)	(0.7)	(0.71)	(0.73)	(0.69)	(0.73)	0.72	0.7	8
XGB¹	0.85	0.78	0.77	0.77	0.82	0.74	0.77	0.78	0.76	0.78	0.75	0.77	4
XGB²	0.82	0.77	0.74	0.76	0.84	0.79	0.8	0.8	0.77	0.83	0.76	0.76	0
NN¹	1.04	0.87	1.1	0.86	1.04	0.87	0.92	0.98	1.07	1.22	0.87	0.93	1
NN²	0.98	0.89	0.98	0.89	0.98	0.9	0.89	0.93	0.95	0.97	0.9	0.9	0

As one can see above, when we consider the SP's Common Stock Price Index: Composite, many ML models seem to perform very well when comparing to benchmark's results. Specifically in this case and in terms of both RMSE and MAE, the Elastic-Net seem to have beaten the AR in every horizon.

3.5 Conclusion

Our results corroborate previous findings showing that with the recent advances in ML methods and the availability of big data, it is possible to improve model forecasts in a real-time environment. More specifically, we confirm that the US inflation (measured by the CPI) is one variable which ML models can be used to greatly increase its forecast accuracy for most of the horizons.

When we look at other types of dependent variables (some related to industrial production, while others to the labor market, etc), we verify that, on average, ML models can also outperform benchmark models. We highlighted that some models such as the ones belonging to the LASSO-family and the RF are often models with the smallest forecast errors across all kinds of dependent variables. On the other hand, we tested some ML models that were constantly outperformed by a simple AR and sometimes even by a Random Walk model, such as the factor models and NN.

Finally, we also tried to shed some light on the relative performance of ML models compared to the AR during periods of recessions. When we look at the 1-step ahead forecast and compared to the median ML model, we find that the AR errors are smaller twice as often during recessions than in regular periods. We find similar results for the 12-step ahead forecast. However, when we look at the h=6 models, the results seem to be the other way around.

Bibliography

Adam, T., Michálek, O., Michl, A., and Slezáková, E. (2021). The rushin index: a weekly indicator of czech economic activity. *CNB working paper series*.

Alexander, M., Polimis, K., and Zagheni, E. (2020). Combining social media and survey data to nowcast migrant stocks in the united states. *Population Research and Policy Review*.

Amuri, F. and Marcucci, J. (2017). The predictive power of google searches in forecasting us unemployment. *International Journal of Forecasting*, 33:801–816.

Anastassopoulou, C., Russo, L., Tsakris, A., and Siettos, C. (2020). Data-based analysis, modelling and forecasting of the covid-19 outbreak. *PLoS ONE*, 15(3).

Artola, C. and Galan, E. (2012). Tracking the future on the web: Construction of leading indicators using internet searches. *Documentos Ocasionales (Bank of Spain)*, 1203:255—271.

Askitas, N. and Zimmermann, K. (2009). Google econometrics and unemployment nowcasting. *DIW Berlin*, Discussion Paper 899.

Bai, J. and Ng, S. (2008). Forecasting economic time series using targeted predictors. *Journal of Econometrics*, 146:304–317.

Bai, J. and Ng, S. (2009). Boosting diffusion indexes. *Journal of Applied Econometrics*, 24:607–629.

Baker, S. and Fradkin, A. (2017). The impact of unemployment insurance on job search: Evidence from google search data. *The Review of Economics and Statistics*, 99 (5):756–768.

Banbura, M., Giannone, D., and Reichlin, L. (2010a). Large bayesian vector autoregressions. *Journal of Applied Econometrics*, 25:71–92.

Banbura, M., Giannone, D., and Reichlin, L. (2010b). Nowcasting. *Oxford Handbook on Economic Forecasting.*

Bertozzia, A., Franco, E., Mohlerd, G., Shorte, M., and Sledgef, D. (2020). The challenges of modeling and forecasting the spread of covid-19. *Proceedings of the National Academy of Sciences*, 117 (29):16732–16738.

Borup, D., Rapach, D., and Schütte, E. (2021). Now- and backcasting initial claims with high-dimensional daily internet search-volume data. Working Paper 3690832, SSRN.

Borup, D. and Schütte, E. (2020). In search of a job: Forecasting employment growth using google trends. *Journal of Business Economic Statistics*. forthcoming.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24:123–140.

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

Burns, A. and Mitchell, W. (1956). *Measuring Business Cycles*. Springer Press.

Bühlmann, P. and Yu, B. (2002). Analysing bagging. *The Annals of Statistics*, 30:927–961.

Carneiro, C. B., Ferreira, I. H., Medeiros, M. C., Pires, H. F., and Zilberman, E. (2020). Lockdown effects in US states: an artificial counterfactual approach. *arXiv e-prints*, page arXiv:2009.13484.

Carvalho, M. and Rua, A. (2017). Real-time nowcasting the us output gap: Singular spectrum analysis at work. *International Journal of Forecasting*, 33 (1):185–198.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 785–794.

Coulombe, P., Leroux, M., Stevanovic, D., and Surprenant, S. How is machine learning useful for macroeconomic forecasting? *Accepted, Journal of Applied Econometrics*.

Coulombe, P., Leroux, M., Stevanovic, D., and Surprenant, S. Macroeconomic data transformations matter. *International Journal of Forecasting, Special Issue*.

Doerr, S. and Gambacorta, L. (2020). Identifying regions at risk with google trends: the impact of covid-19 on us labour markets. *BIS Bulletin*, 8:1–6.

Elliot, G., Garganob, A., and Timmermann, A. (2013). Complete subset regressions. *Journal of Econometrics*, 177(2):357–373.

Eraslan, S. and Götz, T. (2020). An unconventional weekly economic activity index for germany. *Bundesbank Technical Report*.

Evans, M. D. (2005). Where are we now? real-time estimates of the macroeconomy. *International Journal of Central Banking.*, 1(2).

Ferrara, L. and Simoni, A. (2019). When are google data useful to nowcast gdp? an approach via pre-selection and shrinkage. Working paper, Center for Research in Economics and Statistics.

Hansen, P., Lunde, A., and Nason, J. (2011). The model confidence set. *Econometrica*, 79:453–497.

Heikkinen, J. (2019). Nowcasting gdp growth using google trends. Master's thesis, Jyväskylä University School of Business and Economics.

James, G., W. D. H. T. and Tibshirani, R. (2013). *An Introduction to Statistical Learning with applications in R*. Springer Press.

Lewis, D., Mertens, K., Stock, J., and Trivedi, M. (2020). Measuring real activity using a weekly economic index. *Journal of Applied Econometrics*.

Matjaž, P., Nina, G., Mitja, S., and Andraž, S. (2020a). Forecasting covid-19. *Frontiers in Physics.*, 8.

Matjaž, P., Nina, G., Mitja, S., and Andraž, S. (2020b). Forecasting covid-19. *Frontiers in Physics .*, 8.

McCracken, M. and Ng, S. (2015). Fred-md: A monthly database for macroeconomic research. Working paper, St. Louis FED.

Medeiros, M., Street, A., Valladao, D., Vasconcelos, G., and Zilberman, E. (2022). Short-term covid-19 forecast for latecomers. *International Journal of Forecasting*, 38 (2):467–488.

Medeiros, M., Vasconcelos, G., Veiga, A., and Zilberman, E. (2021). Forecasting inflation in a data-rich environment: The benefits of machine learning methods. *Journal of Business and Economic Statistics*, 39:98–119.

Naccaratoa, A., Falorsib, S., Lorigab, S., and Pierinia, A. (2018). Combining official and google trends data to forecast the italian youth unemployment rate. *Technological Forecasting & Social Change*, 130:114–122.

Narita, F. and Yin, R. (2018). In search of information: Use of google trends' data to narrow information gaps for low-income developing countries. Working Paper 18/286, IMF.

Pelat, C., Turbelin, C., Bar-Hen, A., Flahault, A., and Valleron, A. (2009). More diseases tracked by using google trends. *Emerging Infectious Diseases*, 15(8):1327–1328.

Richardson, A., Mulder, T., and Vehbi, T. (2018). Nowcasting new zealand gdp using machine learning algorithms. Working Paper 47, Centre for Applied Macroeconomic Analysis.

Schmidt, D. (1999). Chain ladder prediction and asset liability management. *Blätter DGVFM*, 24:1–9.

Stock, J. H. and Watson, M. W. (2002). Forecasting using principal components from a large number of predictors. *Journal of the American Statistical Association*, 97(460):1167–79.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. R. Statistical Society*, 58:267–288.

Vicente, M., López-Menéndez, A., and Pérez, R. (2015). Forecasting unemployment with internet search data: Does it help to improve predictions when job destruction is skyrocketing? *Technological Forecasting & Social Change*, 92:132–139.

Woloszko, N. (2020a). Tracking activity in real time with google trends. *OECD Economics Department Working Papers*, 1064.

Woloszko, N. (2020b). Tracking activity in real time with google trends. Working Paper 1634, OECD Economics Department.

algorithm

A
Appendix

A.1
Appendix A

A.1.1
Additional figures

Figure A.1 below illustrates the usual path of a victim of COVID-19 in Brazil, which might help understand the reasons for the register delays of COVID-19 confirmed cases and deaths.

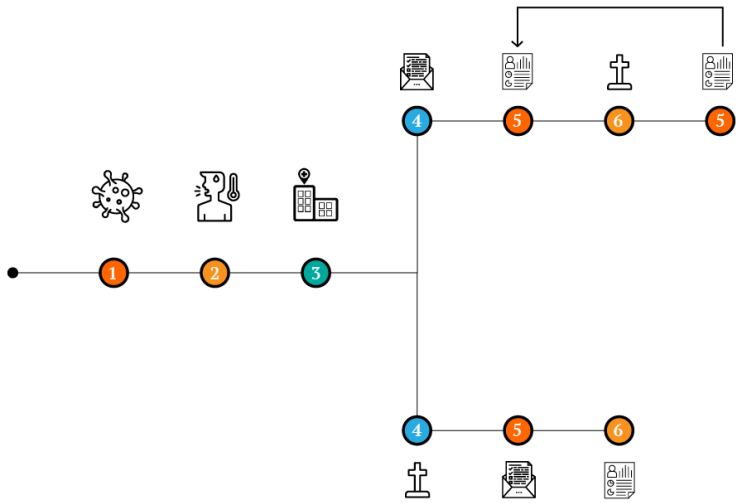

Figure A.1: Illustration of the reasons for the delay in notifications.

It all starts with the day of contagion in circle 1. Usually, it takes around 4 to 5 days for that person to start feeling some symptoms (circle 2), which already imposes a natural, virus-related delay in the notification. On top of that, some people might choose to wait for a while (maybe to see if the symptoms will worsen or not) before going to the hospital (to be treated and/or get tested).

After circle 3, there may be at least two different trajectories. In the top one, after a few days (in most cases) the infected person will receive the confirmation of the diagnosis (4). After a while (and here enters the official Registry system's delay), she will be computed into the system, even if she got

infected weeks ago (5). After that, she ends up dying of COVID-19 (6) and, potentially after some days, her file will be updated with the date of death (7).

The bottom path illustrates the other alternative in which the person ends up dying before her diagnosis (4). Only after the confirmation of the death by COVID-19 (5), she will be computed into the system with the date of her death (6).[1]

Now, Figure A.2 below show the same patterns of revisions as in Figure 1.4 for seven different vintages in the recent past. The idea of this figure is to show that the patterns of revision change from state to state.

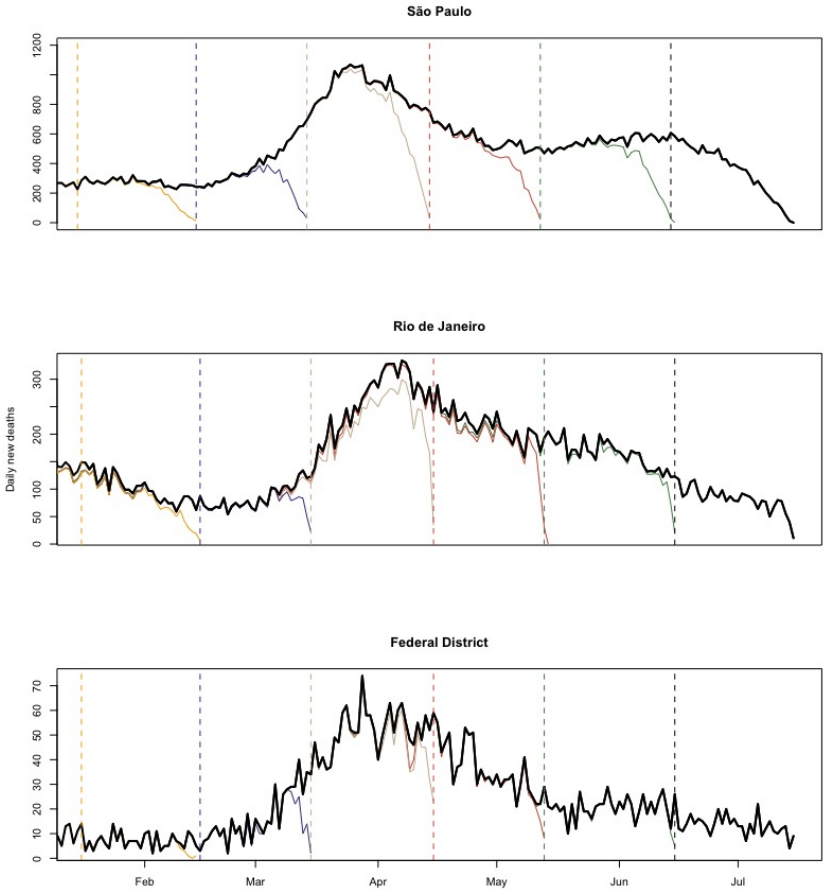

Figure A.2: Six different ARPEN data vintages for some of the largest Brazilian states. Some states need nowcast models more than others.

As we see above, in São Paulo, the last 20 observations are almost always subject to strong revisions, while usually in Rio de Janeiro only the last 10 days are. In Brazil's Federal District, death's registers are even quicker: usually only the most recent week of days in a certain vintage is subject to revisions. This

[1]Note that in this case, she will only enter the system after her death, which means the "date of notification" variable in the Open Data SUS database will reflect the date of notification of her death and not the notification of the confirmation of infection.

indicates that a Nowcasting model for the Federal District is less important than one for the state of Rio, which in turns is less important than one for São Paulo.

Figure A.4 below displays six different BT windows (1st, 7th, 13th, 19th, 25th and 30th) from the COVID-19 daily deaths' Nowcasting in April 16, 2021. All these windows, among the other 24, were used to compute the weights used in that nowcast window (March 16 to April 16), which we can see in Figure A.6. They were also the inputs necessary to generate the MAPEs by day in Figure A.5, and therefore the nowcasting interval.

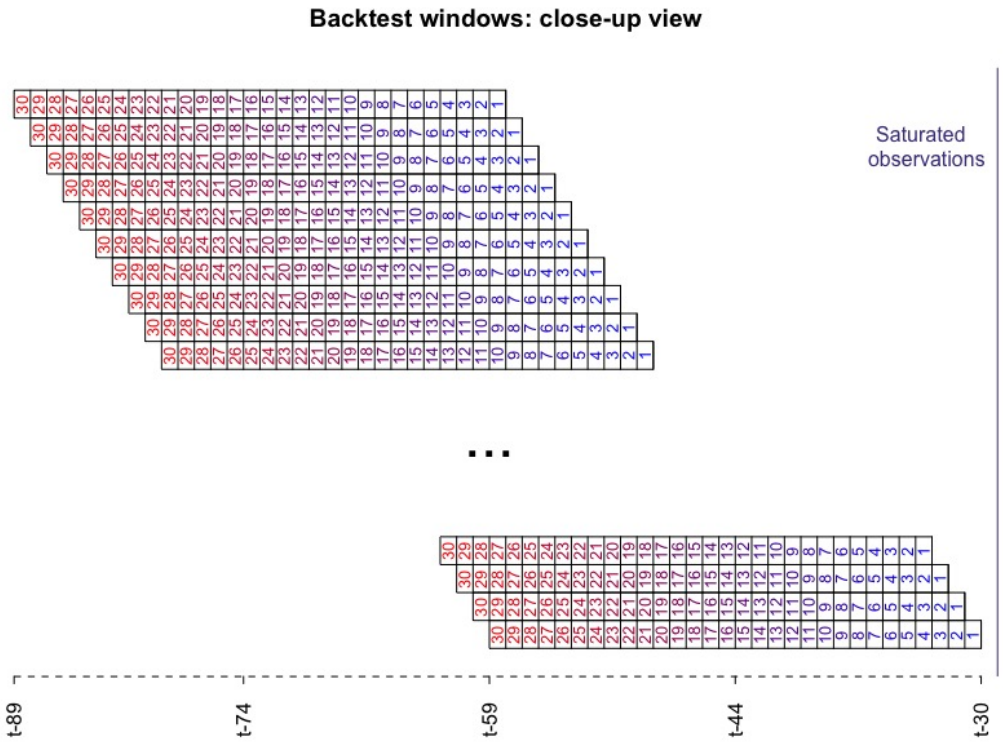

Figure A.3: This figure shows a zoomed in version of Figure 1.9, with the purpose to highlight that we have 30 different nowcasts' points for each window. Therefore, we have 30 observations for each revision type (i.e., points with 29 to 0 revisions)[a], which allows us to compute backtest MAPEs and nowcast intervals for each nowcast point.

[a]Again, nowcast a data point with 29 revisions is much easier than nowcast one with 0 revisions.

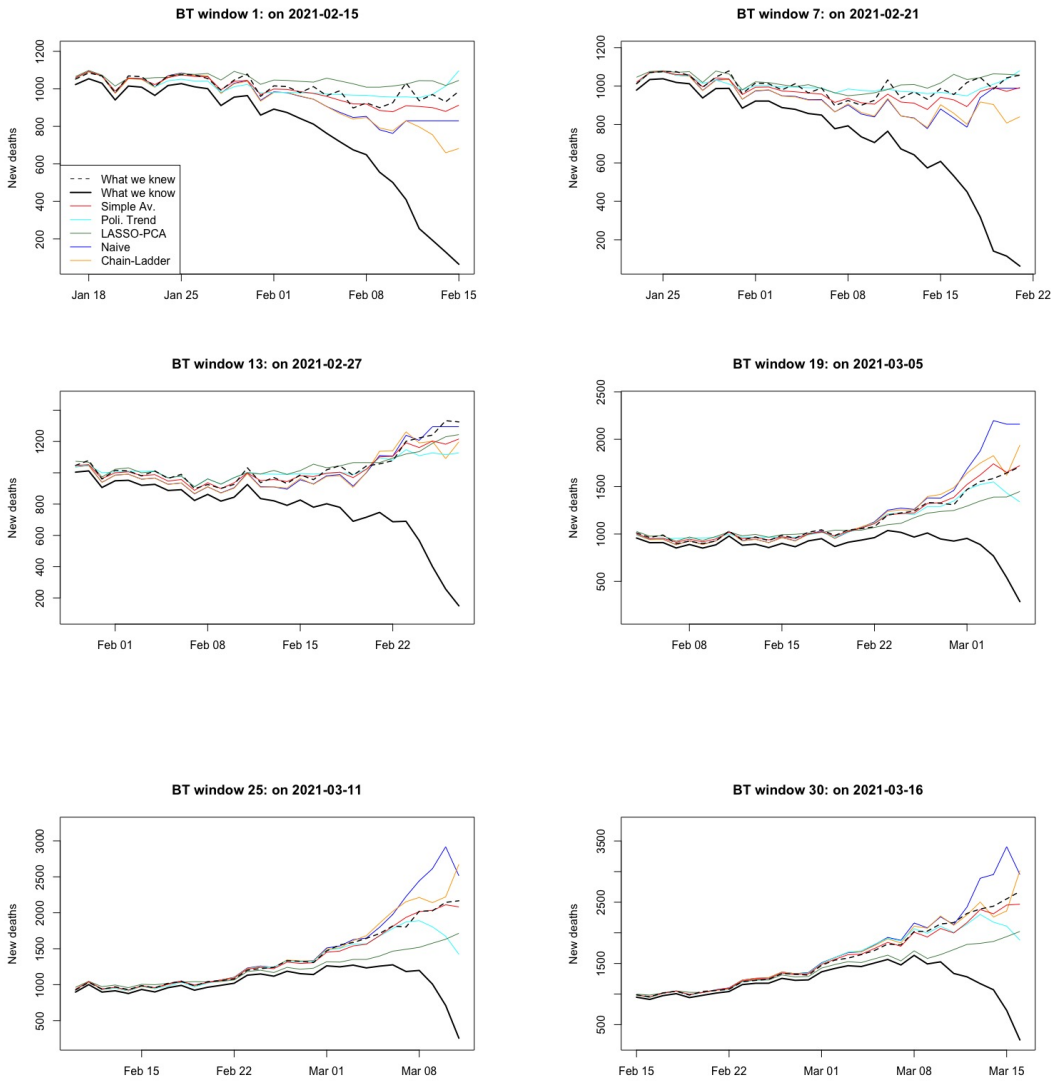

Figure A.4: Six BT windows obtained in the Nowcasting of April 16, 2021. The "Simple Average" model performs exceptionally well and thus receives the highest weight in that actual nowcast window.

Figure A.5 below shows MAPEs obtained by each model during the BT procedure. As highlighted in its legend, each color represent a different model's MAPE across each Nowcast window (n_1 to n_{30}). As expected, almost every curve increases monotonically, i.e., the most recent each observation is (or the less revisions it was subject to), the hardest it is to predict it.

Another information brought by Figure A.5 below are the dashed, colored horizontal lines. They represent each model's average MAPE across n_{16} to n_1, the nowcasting window hardest to predict. These lines give an idea of the overall performance of each model. As one can see below, the Simple Average model achieves an overall MAPE lower than 5% across every BT window. Even though the other models have a higher MAPE, every other overall MAPE was below 10% in the vintage of April 16, 2021.

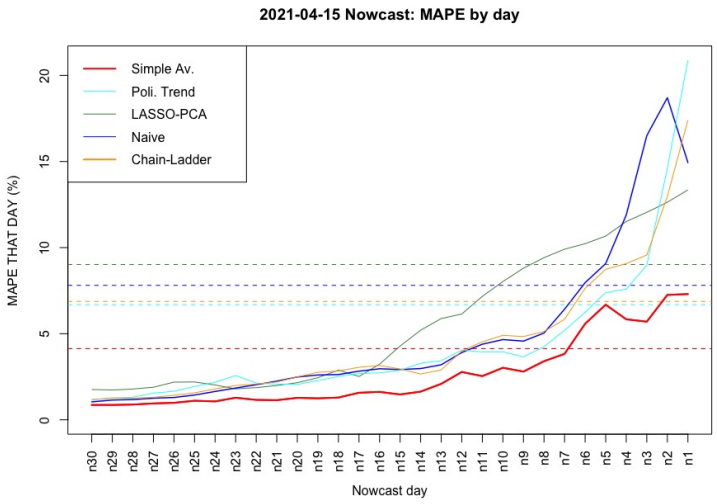

Figure A.5: Overall MAPE and MAPEs by nowcast day (n_1 to n_{30} for the 30 BT windows of April 16, 2021's nowcast.

Figure A.6 below displays the nowcast results of its first release on April 16, 2021.

In order give more examples of our methodology performance, Figures A.7 and A.8 below shows the same idea as A.6, but for nowcasts of April 23 (its second release) and of June 18 (the most recent already saturated nowcasts we have so far).

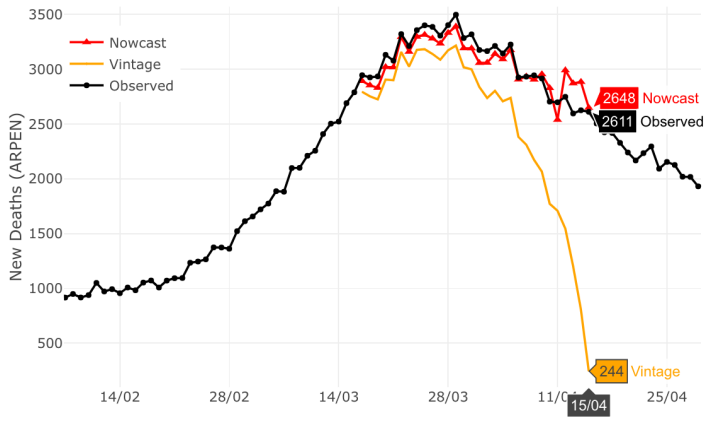

Figure A.6: Nowcast results of its first release on April 16, 2021.

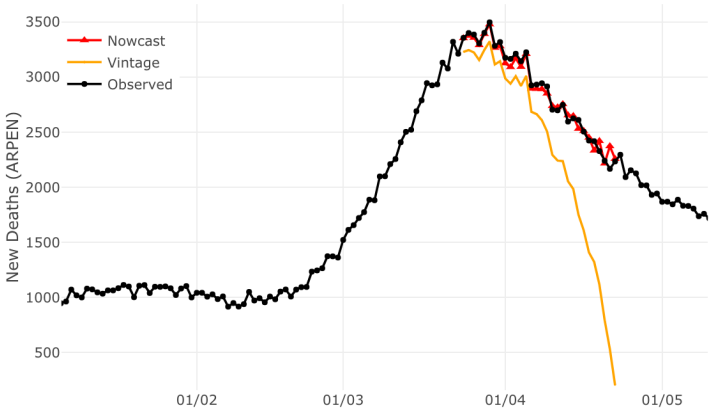

Figure A.7: Nowcast results of its second release on April 23, 2021.

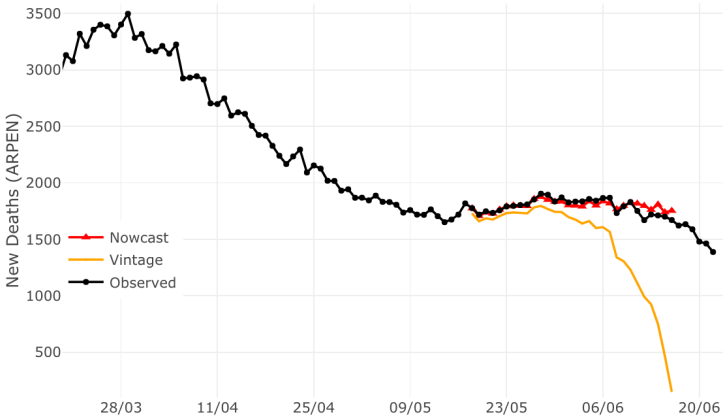

Figure A.8: Nowcast results of a more recent release on June 18, 2021.

A.2
Appendix 2

A.2.1
Forecast models in Real Time

In 2015, `Google Trends` data was made available in real time. It is no doubt that it is helping people around the world explore the global reaction to major events. But not only that. The way `Google Trends` is constructed allows one to verify how was the search patterns in the past for a specific term and location. This feature makes possible the construction of vintages of the same search patterns, reproducing for any given day in that range, information that was available to a real-time forecaster.

That being said, it becomes clear that due to the fact that the use of real-time forecast and nowcast has been increasingly rising (some examples are Carvalho and Rua (2017), Richardson et al. (2018), Woloszko (2020b), and Alexander et al. (2020)), the possibility of constructing vintages of `Google Trends` is a very powerful feature of this tool. However, the same possible issue shown in previous sections may arise in a setup of real-time forecast and/or nowcast.

To illustrate the issue, we display below a series of figures. Each one contains three search pattern for the same term one month ahead. For example, the first figure below brings the time series of "Refined Petroleum" searches in the US from: 1) Jan, 2004 to Jan, 2014; 2) Feb, 2004 - Feb, 2014; 3) Mar, 2004 - Mar, 2014. This would be what a real-time forecaster could gather when constructing the vintages.

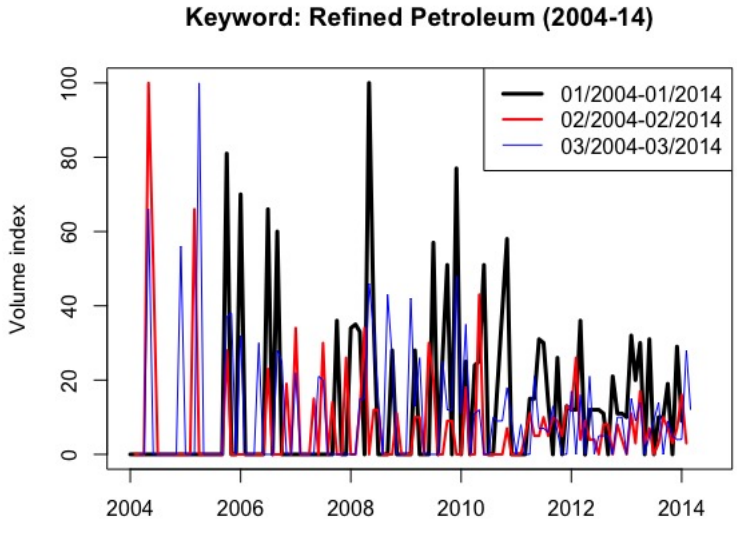

Figure A.9: Vintages of "Refined Petroleum" topic in the US.

It is very clear from Figure A.9 that each series differ a lot from each other. As a matter of fact, the highest correlation between these series is 0.27 (between black and blue lines) and the lowest is 0.01! It is important to remember that the values of the volume index displayed by `Google Trends` is normalized and therefore could indeed differ from one vintage to another if the period of search was different. However, it is clear from the plot above that no new maximum search volume index is achieved neither in February, 2014 nor in March, 2014, i.e., the maximum value (100) of the three plots should be the same, when it is not.

Another important thing to mention is that we are showing a very extreme example for two main reasons: 1) we went back very further in time, when fewer people used Google, which means their database should be less consistent; 2) we searched for a very-low volume term, contributing to the low consistency of the data. However, if one intends to perform actual real-time analysis with `Google Trends`, it is very likely that she will go back that further in time. Besides that, depending on what she is trying to forecast (e.g., volume or value of exports), she may try to use those low-volume terms in `Google Trends` as covariates in her forecast.

In Figure A.10 we display the same plot as above, but now with timestamps going from Jan, 2009 - Jan, 2019 to Mar, 2009 - Mar, 2019. We can see that the correlations between each sample becomes a little bit higher: the highest one is one 0.43 and the lowest 0.29.

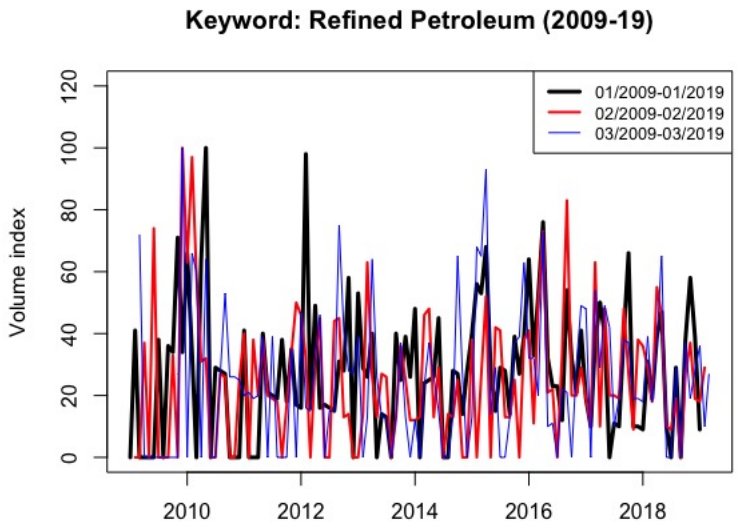

Figure A.10: More recent vintages of 'Refined Petroleum' topic in the US.

Figure A.11 displays search patterns of "US inflation", a far more popular term. We can see that even for the 2004-2014 period, the correlations are

already higher (maximum of 0.64 and minimum of 0.44) than the ones from the "Refined Petroleum" topic.

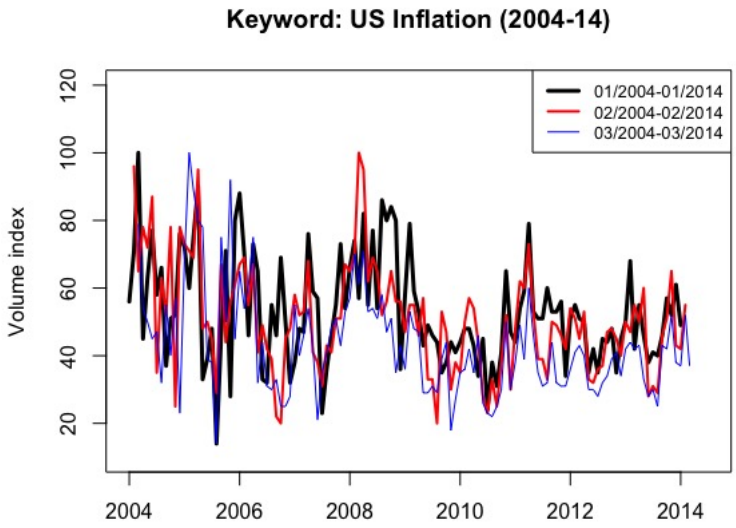

Figure A.11: Vintages of 'US Inflation' topic in the US.

Figure A.12 makes it clear that in recent years correlations between random samples can be very high. However, this does not mean that taking averages of many samples of these series could not improve their consistency with respect to the actual population search pattern of the term in each period of time.

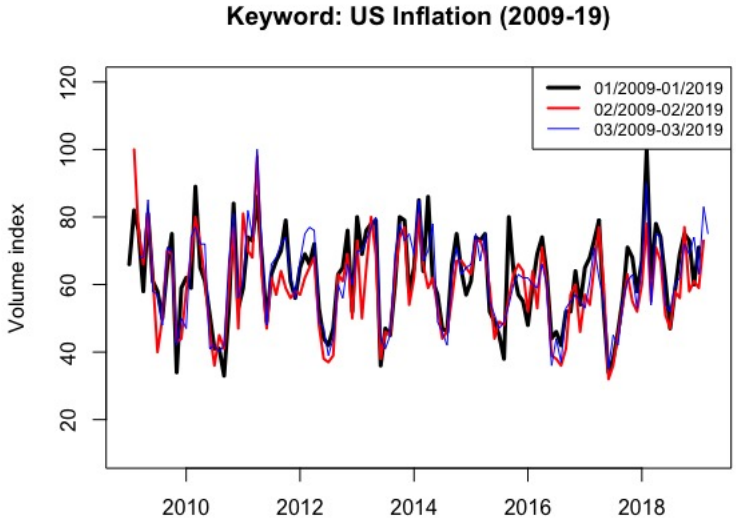

Figure A.12: More recent vintages of "US inflation" topic in the US.

A.3
Appendix 3

A.3.0.1
Additional tables

Table A.1: Forecasting Errors for FEDFUNDS since 2000.

RMSE are displayed in odd lines (with 1 as a superscript besides the model name) while MAE are displayed in even lines (with 2 as a superscript besides the model name). Cells in blue (gray) show the models included in the 50% model confidence set (MCS) using the squared error (absolute error) as loss function.

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	MCS Count
RW¹	1	1	1	1	1	1	1	1	1	1	1	1	1
RW²	1	1	1	1	1	1	1	1	1	1	1	1	0
AR¹	0.9	0.69	0.71	0.69	0.67	0.68	0.68	0.7	0.73	0.74	0.73	0.7	12
AR²	0.89	(0.67)	0.68	(0.66)	(0.65)	0.66	0.67	(0.69)	(0.69)	(0.75)	(0.73)	(0.72)	8
RF¹	0.79	0.7	0.75	0.74	0.73	0.73	0.72	0.76	0.77	0.74	0.73	0.72	8
RF²	(0.78)	(0.68)	0.72	0.69	(0.69)	0.71	0.7	0.74	0.74	(0.73)	(0.72)	(0.73)	6
LASSO¹	0.79	0.69	0.71	0.69	0.66	0.66	0.67	0.7	0.73	0.73	0.72	0.7	12
LASSO²	(0.78)	(0.67)	(0.68)	(0.65)	(0.63)	(0.65)	(0.65)	(0.68)	(0.69)	(0.72)	(0.71)	(0.71)	12
RIDGE¹	0.96	0.99	1.04	0.69	0.66	0.66	0.66	0.7	0.73	0.73	0.72	1.3	8
RIDGE²	0.94	0.92	0.96	(0.65)	(0.63)	(0.65)	(0.65)	(0.68)	(0.69)	0.72	(0.71)	1.11	7
ELNET¹	0.82	0.69	0.71	0.69	0.66	0.66	0.67	0.7	0.73	0.73	0.72	0.7	10
ELNET²	(0.79)	(0.67)	0.68	(0.65)	(0.63)	(0.65)	(0.65)	(0.68)	(0.69)	0.72	(0.71)	0.71	9
ADA¹	0.8	0.7	0.71	0.69	0.66	0.67	0.67	0.7	0.73	0.73	0.72	0.7	10
ADA²	(0.79)	(0.68)	(0.67)	(0.66)	(0.64)	(0.65)	(0.65)	(0.68)	(0.69)	0.72	(0.71)	0.71	10
ADAELNET¹	0.8	0.69	0.71	0.69	0.66	0.67	0.67	0.7	0.73	0.73	0.72	0.7	10
ADAELNET²	(0.79)	(0.68)	(0.68)	(0.65)	(0.64)	(0.65)	(0.65)	(0.68)	(0.69)	0.72	(0.71)	0.71	10
RF.OLS¹	0.85	0.73	0.76	0.75	0.71	0.73	0.72	0.76	0.75	0.76	0.76	0.84	7
RF.OLS²	(0.81)	(0.7)	0.72	0.71	0.68	0.72	0.7	0.73	0.71	(0.75)	(0.75)	0.78	4
LASSO.RF¹	0.83	0.71	0.74	0.74	0.68	0.7	0.69	0.76	0.78	0.76	0.75	0.74	7
LASSO.RF²	0.83	(0.7)	0.72	0.72	(0.66)	0.69	0.68	0.75	0.76	(0.75)	0.76	0.77	3
BOOST¹	0.83	0.69	0.71	0.69	0.68	0.67	0.67	0.71	0.73	0.73	0.72	0.69	12
BOOST²	(0.81)	(0.67)	(0.67)	(0.66)	(0.65)	(0.65)	(0.65)	(0.68)	(0.69)	(0.72)	(0.71)	(0.71)	12
BVAR¹	0.89	0.82	0.81	0.8	0.81	0.83	0.82	0.87	0.9	0.85	0.83	0.83	1
BVAR²	0.85	0.78	0.79	0.79	0.78	0.8	0.79	0.85	0.86	0.84	0.82	0.84	0
FACTORS¹	1.26	1.14	0.96	0.95	0.89	0.94	0.91	0.86	0.82	0.76	0.75	0.76	3
FACTORS²	0.96	0.84	0.87	0.85	0.8	0.84	0.82	0.79	0.74	(0.74)	0.75	0.78	1
TFct¹	1.24	0.99	0.83	0.84	0.79	0.96	0.78	0.74	0.85	0.75	0.79	0.77	5
TFct²	0.95	0.77	0.78	0.75	0.74	0.82	0.74	0.73	0.76	(0.75)	0.8	0.8	1
ATFct¹	0.88	0.8	0.85	0.85	0.82	0.8	0.8	0.8	0.8	0.8	0.81	0.8	1
ATFct²	0.89	0.8	0.82	0.81	0.78	0.8	0.78	0.77	0.77	0.8	0.83	0.82	0
CSR¹	0.85	0.7	0.74	0.72	0.7	0.71	0.71	0.73	0.73	0.73	0.73	0.7	7
CSR²	(0.81)	(0.69)	0.7	0.68	(0.66)	0.69	0.69	0.71	(0.69)	(0.73)	(0.73)	(0.73)	7
XGB¹	0.87	0.75	0.81	0.77	0.77	0.78	0.77	0.84	0.82	0.78	0.79	0.77	2
XGB²	0.84	0.75	0.8	0.74	0.74	0.77	0.76	0.82	0.78	0.77	0.8	0.8	0
NN¹	1.18	1.2	1.18	1.13	1.51	1.42	1.05	1.05	1.32	1.33	1.29	1.74	0
NN²	1.13	1.04	1.06	1	1.09	1.09	0.99	1.03	1.08	1.12	1.17	1.24	0

Table A.2: Forecasting Errors for GS10 since 2000.

RMSE are displayed in odd lines (with 1 as a superscript besides the model name) while MAE are displayed in even lines (with 2 as a superscript besides the model name). Cells in blue (gray) show the models included in the 50% model confidence set (MCS) using the squared error (absolute error) as loss function.

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	MCS Count
RW¹	1	1	1	1	1	1	1	1	1	1	1	1	1
RW²	1	1	1	1	1	1	1	1	1	1	1	1	0
AR¹	0.9	0.69	0.71	0.69	0.67	0.68	0.68	0.7	0.73	0.74	0.73	0.7	12
AR²	0.89	(0.67)	0.68	(0.66)	(0.65)	0.66	0.67	(0.69)	(0.69)	(0.75)	(0.73)	(0.72)	8
RF¹	0.79	0.7	0.75	0.74	0.73	0.73	0.72	0.76	0.77	0.74	0.73	0.72	8
RF²	(0.78)	(0.68)	0.72	0.69	(0.69)	0.71	0.7	0.74	0.74	(0.73)	(0.72)	(0.73)	6
LASSO¹	0.79	0.69	0.71	0.69	0.66	0.66	0.67	0.7	0.73	0.73	0.72	0.7	12
LASSO²	(0.78)	(0.67)	(0.68)	(0.65)	(0.63)	(0.65)	(0.65)	(0.68)	(0.69)	(0.72)	(0.71)	(0.71)	12
RIDGE¹	0.96	0.99	1.04	0.69	0.66	0.66	0.66	0.7	0.73	0.73	0.72	1.3	8
RIDGE²	0.94	0.92	0.96	(0.65)	(0.63)	(0.65)	(0.65)	(0.68)	(0.69)	0.72	(0.71)	1.11	7
ELNET¹	0.82	0.69	0.71	0.69	0.66	0.66	0.67	0.7	0.73	0.73	0.72	0.7	10
ELNET²	(0.79)	(0.67)	0.68	(0.65)	(0.63)	(0.65)	(0.65)	(0.68)	(0.69)	0.72	(0.71)	0.71	9
ADA¹	0.8	0.7	0.71	0.69	0.66	0.67	0.67	0.7	0.73	0.73	0.72	0.7	10
ADA²	(0.79)	(0.68)	(0.67)	(0.66)	(0.64)	(0.65)	(0.65)	(0.68)	(0.69)	0.72	(0.71)	0.71	10
ADAELNET¹	0.8	0.69	0.71	0.69	0.66	0.67	0.67	0.7	0.73	0.73	0.72	0.7	10
ADAELNET²	(0.79)	(0.68)	(0.68)	(0.65)	(0.64)	(0.65)	(0.65)	(0.68)	(0.69)	0.72	(0.71)	0.71	10
RF.OLS¹	0.85	0.73	0.76	0.75	0.71	0.73	0.72	0.76	0.75	0.76	0.76	0.84	7
RF.OLS²	(0.81)	(0.7)	0.72	0.71	0.68	0.72	0.7	0.73	0.71	(0.75)	(0.75)	0.78	4
LASSO.RF¹	0.83	0.71	0.74	0.74	0.68	0.7	0.69	0.76	0.78	0.76	0.75	0.74	7
LASSO.RF²	0.83	(0.7)	0.72	0.72	(0.66)	0.69	0.68	0.75	0.76	(0.75)	0.76	0.77	3
BOOST¹	0.83	0.69	0.71	0.69	0.68	0.67	0.67	0.71	0.73	0.73	0.72	0.69	12
BOOST²	(0.81)	(0.67)	(0.67)	(0.66)	(0.65)	(0.65)	(0.65)	(0.68)	(0.69)	(0.72)	(0.71)	(0.71)	12
BVAR¹	0.89	0.82	0.81	0.8	0.81	0.83	0.82	0.87	0.9	0.85	0.83	0.83	1
BVAR²	0.85	0.78	0.79	0.79	0.78	0.8	0.79	0.85	0.86	0.84	0.82	0.84	0
FACTORS¹	1.26	1.14	0.96	0.95	0.89	0.94	0.91	0.86	0.82	0.76	0.75	0.76	3
FACTORS²	0.96	0.84	0.87	0.85	0.8	0.84	0.82	0.79	0.74	(0.74)	0.75	0.78	1
TFct¹	1.24	0.99	0.83	0.84	0.79	0.96	0.78	0.74	0.85	0.75	0.79	0.77	5
TFct²	0.95	0.77	0.78	0.75	0.74	0.82	0.74	0.73	0.76	(0.75)	0.8	0.8	1
ATFct¹	0.88	0.8	0.85	0.85	0.82	0.8	0.8	0.8	0.8	0.8	0.81	0.8	1
ATFct²	0.89	0.8	0.82	0.81	0.78	0.8	0.78	0.77	0.77	0.8	0.83	0.82	0
CSR¹	0.85	0.7	0.74	0.72	0.7	0.71	0.71	0.73	0.73	0.73	0.73	0.7	7
CSR²	(0.81)	(0.69)	0.7	0.68	(0.66)	0.69	0.69	0.71	(0.69)	(0.73)	(0.73)	(0.73)	7
XGB¹	0.87	0.75	0.81	0.77	0.77	0.78	0.77	0.84	0.82	0.78	0.79	0.77	2
XGB²	0.84	0.75	0.8	0.74	0.74	0.77	0.76	0.82	0.78	0.77	0.8	0.8	0
NN¹	1.18	1.2	1.18	1.13	1.51	1.42	1.05	1.05	1.32	1.33	1.29	1.74	0
NN²	1.13	1.04	1.06	1	1.09	1.09	0.99	1.03	1.08	1.12	1.17	1.24	0

Table A.3: Forecasting Errors for OILPRICEx since 2000.

RMSE are displayed in odd lines (with 1 as a superscript besides the model name) while MAE are displayed in even lines (with 2 as a superscript besides the model name). Cells in blue (gray) show the models included in the 50% model confidence set (MCS) using the squared error (absolute error) as loss function.

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	MCS Count
RW¹	1	1	1	1	1	1	1	1	1	1	1	1	1
RW²	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	2
AR¹	0.88	0.68	0.67	0.68	0.71	0.7	0.7	0.69	0.7	0.77	0.79	0.76	12
AR²	(0.88)	(0.69)	(0.65)	(0.65)	(0.71)	(0.66)	(0.68)	(0.67)	(0.67)	(0.72)	(0.75)	(0.75)	12
RF¹	0.86	0.73	0.72	0.69	0.72	0.71	0.73	0.71	0.72	0.78	0.82	0.8	6
RF²	(0.84)	(0.74)	(0.7)	(0.68)	(0.73)	(0.69)	(0.71)	(0.69)	(0.7)	(0.74)	(0.78)	(0.78)	7
LASSO¹	0.82	0.69	0.67	0.67	0.71	0.7	0.7	0.69	0.7	0.77	0.79	0.76	11
LASSO²	(0.81)	(0.7)	(0.65)	(0.65)	(0.71)	(0.66)	(0.67)	(0.66)	(0.67)	(0.71)	(0.74)	(0.73)	11
RIDGE¹	0.94	0.81	0.8	0.67	0.71	0.7	0.7	0.69	0.7	0.77	0.79	1.05	9
RIDGE²	(0.97)	(0.86)	(0.82)	(0.65)	(0.7)	(0.66)	(0.67)	(0.66)	(0.67)	(0.71)	(0.75)	(0.92)	9
ELNET¹	0.84	0.69	0.67	0.67	0.71	0.7	0.7	0.69	0.7	0.77	0.79	0.76	11
ELNET²	(0.82)	(0.7)	(0.65)	(0.65)	(0.7)	(0.66)	(0.67)	(0.66)	(0.67)	(0.71)	(0.74)	(0.73)	11
ADA¹	0.83	0.69	0.67	0.67	0.71	0.7	0.7	0.69	0.7	0.77	0.79	0.76	11
ADA²	(0.82)	(0.7)	(0.65)	(0.65)	(0.71)	(0.66)	(0.67)	(0.66)	(0.67)	(0.71)	(0.75)	(0.73)	11
ADAELNET¹	0.83	0.69	0.67	0.67	0.71	0.7	0.7	0.69	0.7	0.77	0.79	0.76	11
ADAELNET²	(0.81)	(0.7)	(0.65)	(0.65)	(0.71)	(0.66)	(0.67)	(0.66)	(0.67)	(0.71)	(0.75)	(0.73)	11
RF.OLS¹	0.88	0.72	0.69	0.68	0.71	0.7	0.71	0.7	0.72	0.89	1.99	1.96	7
RF.OLS²	(0.83)	(0.73)	(0.68)	(0.67)	(0.71)	(0.68)	(0.69)	(0.68)	(0.69)	(0.77)	(0.97)	(1)	8
LASSO.RF¹	0.85	0.73	0.7	0.7	0.72	0.71	0.7	0.69	0.7	0.78	0.8	0.77	8
LASSO.RF²	(0.83)	(0.72)	(0.69)	(0.69)	(0.71)	(0.68)	(0.69)	(0.67)	(0.68)	(0.74)	(0.76)	(0.76)	9
BOOST¹	0.85	0.69	0.68	0.67	0.71	0.7	0.7	0.69	0.71	0.77	0.79	0.76	9
BOOST²	(0.83)	(0.7)	(0.66)	(0.65)	(0.71)	(0.67)	(0.67)	(0.67)	(0.67)	(0.71)	(0.75)	(0.73)	10
BVAR¹	0.86	0.76	0.73	0.73	0.75	0.74	0.75	0.75	0.76	0.8	0.82	0.78	4
BVAR²	(0.85)	(0.75)	(0.71)	(0.73)	(0.76)	(0.72)	(0.73)	(0.72)	(0.73)	(0.76)	(0.79)	(0.77)	5
FACTORS¹	1.02	0.76	0.71	0.71	0.73	0.74	0.73	0.7	0.75	0.79	0.8	0.78	6
FACTORS²	(0.87)	(0.76)	(0.72)	(0.7)	(0.74)	(0.71)	(0.71)	(0.68)	(0.71)	(0.74)	(0.76)	(0.77)	7
TFct¹	0.95	0.81	0.74	0.7	0.76	0.76	0.9	0.75	0.73	0.79	0.81	0.81	3
TFct²	(0.9)	(0.79)	(0.71)	(0.7)	(0.79)	(0.74)	(0.77)	(0.74)	(0.69)	(0.73)	(0.76)	(0.78)	4
ATFct¹	0.95	0.82	0.74	0.73	0.77	0.77	0.75	0.82	0.77	0.83	0.89	0.85	2
ATFct²	(0.9)	(0.82)	(0.74)	(0.73)	(0.79)	(0.74)	(0.74)	(0.74)	(0.75)	(0.78)	(0.84)	(0.81)	4
CSR¹	0.89	0.71	0.69	0.69	0.73	0.71	0.71	0.69	0.71	0.78	0.9	0.87	6
CSR²	(0.84)	(0.72)	(0.67)	(0.67)	(0.73)	(0.68)	(0.69)	(0.68)	(0.69)	(0.73)	(0.81)	(0.79)	7
XGB¹	0.90	0.76	0.71	0.70	0.74	0.72	0.76	0.74	0.76	0.83	0.86	0.82	4
XGB²	(0.88)	(0.75)	(0.69)	(0.68)	(0.77)	(0.70)	(0.75)	(0.73)	(0.73)	(0.77)	(0.81)	(0.79)	7
NN¹	0.85	0.91	0.92	0.82	0.90	0.81	0.90	0.88	0.93	0.89	1.05	1.19	2
NN²	(0.90)	(0.93)	(0.89)	(0.84)	(0.88)	(0.81)	(0.86)	(0.82)	(0.88)	(0.84)	(0.97)	(1.00)	1

Table A.4: Forecasting Errors for M1SL since 2000.

RMSE are displayed in odd lines (with 1 as a superscript besides the model name) while MAE are displayed in even lines (with 2 as a superscript besides the model name). Cells in blue (gray) show the models included in the 50% model confidence set (MCS) using the squared error (absolute error) as loss function.

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	MCS Count
RW¹	1	1	1	1	1	1	1	1	1	1	1	1	0
RW²	1	1	1	1	1	1	1	1	1	1	1	1	0
AR¹	0.87	0.8	0.82	0.74	0.79	0.83	0.85	0.86	0.9	0.83	0.84	0.87	12
AR²	(0.69)	(0.67)	(0.71)	(0.64)	(0.68)	(0.71)	(0.68)	(0.71)	(0.77)	(0.68)	(0.68)	(0.73)	12
RF¹	0.87	0.8	0.79	0.72	0.75	0.79	0.83	0.88	0.94	0.83	0.84	0.88	12
RF²	(0.68)	(0.67)	(0.7)	(0.64)	(0.68)	(0.71)	(0.67)	(0.71)	(0.78)	(0.67)	(0.66)	(0.73)	12
LASSO¹	0.89	0.81	0.81	0.75	0.81	0.84	0.86	0.88	0.92	0.81	0.84	0.86	12
LASSO²	(0.67)	(0.67)	(0.7)	(0.64)	(0.68)	(0.7)	(0.66)	(0.69)	(0.77)	(0.65)	(0.67)	(0.72)	12
RIDGE¹	1.04	0.85	0.85	0.93	1.04	1.16	0.92	0.91	0.95	0.84	0.86	0.88	9
RIDGE²	0.79	(0.72)	0.75	0.84	(0.85)	(0.89)	(0.74)	(0.73)	(0.81)	(0.7)	(0.71)	(0.76)	9
ELNET¹	0.9	0.82	0.81	0.76	0.81	0.84	0.86	0.88	0.92	0.81	0.85	0.86	12
ELNET²	(0.68)	(0.67)	(0.7)	(0.64)	(0.69)	(0.7)	(0.66)	(0.69)	(0.76)	(0.65)	(0.67)	(0.72)	12
ADA¹	0.89	0.81	0.81	0.76	0.81	0.84	0.86	0.88	0.92	0.81	0.84	0.86	12
ADA²	(0.67)	(0.67)	(0.7)	(0.64)	(0.68)	(0.7)	(0.66)	(0.69)	(0.77)	(0.65)	(0.67)	(0.72)	12
ADAELNET¹	0.9	0.82	0.81	0.76	0.81	0.84	0.86	0.89	0.92	0.81	0.84	0.86	12
ADAELNET²	(0.68)	(0.67)	(0.7)	(0.64)	(0.68)	(0.71)	(0.66)	(0.69)	(0.77)	(0.65)	(0.67)	(0.72)	12
RF.OLS¹	0.89	0.82	0.81	0.76	0.8	0.84	0.87	0.88	0.92	0.82	0.84	0.87	12
RF.OLS²	(0.69)	(0.67)	(0.71)	(0.63)	(0.67)	(0.7)	(0.66)	(0.69)	(0.76)	(0.65)	(0.66)	(0.72)	12
LASSO.RF¹	0.9	0.82	0.8	0.76	0.77	0.83	0.86	0.9	0.91	0.83	0.84	0.87	11
LASSO.RF²	(0.73)	(0.7)	(0.69)	0.66	(0.68)	(0.71)	(0.69)	(0.71)	(0.77)	(0.68)	(0.69)	(0.73)	11
BOOST¹	0.92	0.84	0.85	0.77	0.8	0.84	0.86	0.89	0.92	0.82	0.84	0.87	12
BOOST²	(0.69)	(0.69)	(0.72)	(0.63)	(0.67)	(0.71)	(0.66)	(0.69)	(0.76)	(0.66)	(0.66)	(0.73)	12
BVAR¹	0.9	0.84	0.88	0.83	0.86	0.9	0.86	0.85	0.96	0.83	0.85	0.9	10
BVAR²	(0.69)	(0.71)	0.75	0.7	(0.71)	(0.73)	(0.67)	(0.71)	(0.83)	(0.68)	(0.69)	(0.78)	10
FACTORS¹	0.94	0.8	0.81	0.76	0.9	0.91	0.9	0.9	0.9	0.81	0.82	0.86	12
FACTORS²	(0.71)	(0.7)	(0.72)	(0.64)	(0.74)	(0.74)	(0.69)	(0.72)	(0.77)	(0.66)	(0.65)	(0.71)	12
TFct¹	0.93	0.81	0.8	0.74	0.93	0.87	1	0.95	0.91	0.83	0.83	0.87	10
TFct²	(0.71)	(0.7)	0.72	0.66	(0.75)	(0.75)	(0.76)	(0.78)	(0.81)	(0.68)	(0.66)	(0.72)	10
ATFct¹	0.93	0.81	0.81	0.74	0.8	0.83	0.88	0.87	0.92	0.83	0.84	0.87	11
ATFct²	(0.72)	(0.7)	0.74	(0.66)	(0.72)	(0.73)	(0.71)	(0.74)	(0.8)	(0.69)	(0.68)	(0.74)	11
CSR¹	0.9	0.82	0.82	0.76	0.8	0.84	0.87	0.88	0.92	0.82	0.83	0.86	12
CSR²	(0.69)	(0.68)	(0.71)	(0.63)	(0.67)	(0.7)	(0.66)	(0.68)	(0.76)	(0.65)	(0.66)	(0.72)	12
XGB¹	1.01	1.03	1	0.99	1.01	1	1	0.99	0.99	0.99	1	1	10
XGB²	(0.74)	(0.78)	0.89	0.85	(0.89)	(0.88)	(0.79)	(0.79)	(0.84)	(0.78)	(0.78)	(0.84)	10
NN¹	1.05	1.03	1	0.98	1	1	1	1	1	1	1.01	1.01	3
NN²	(0.81)	0.87	0.91	0.78	(0.85)	(0.82)	0.81	0.83	0.87	0.83	0.84	0.91	3

Table A.5: Forecasting Errors for GS1 since 2000.

RMSE are displayed in odd lines (with 1 as a superscript besides the model name) while MAE are displayed in even lines (with 2 as a superscript besides the model name). Cells in blue (gray) show the models included in the 50% model confidence set (MCS) using the squared error (absolute error) as loss function.

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	MCS Count
RW¹	1	1	1	1	1	1	1	1	1	1	1	1	4
RW²	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	4
AR¹	1.1	0.86	0.88	0.83	0.78	0.8	0.78	0.78	0.87	0.84	0.8	0.77	11
AR²	(1.11)	(0.87)	(0.94)	(0.85)	(0.78)	(0.8)	(0.78)	(0.72)	(0.81)	(0.78)	(0.74)	(0.73)	11
RF¹	0.92	0.8	0.82	0.78	0.8	0.82	0.83	0.92	0.92	0.89	0.85	0.81	9
RF²	(0.96)	(0.84)	(0.9)	(0.82)	(0.84)	(0.86)	(0.86)	(0.91)	(0.9)	(0.89)	(0.82)	(0.78)	9
LASSO¹	1.1	1	0.85	0.79	0.84	0.87	0.79	0.81	0.86	0.84	0.8	0.75	12
LASSO²	(1.11)	(0.94)	(0.89)	(0.81)	(0.82)	(0.85)	(0.8)	(0.76)	(0.8)	(0.77)	(0.74)	(0.7)	12
RIDGE¹	1.64	1.56	1.6	0.77	0.76	0.77	0.78	0.81	0.86	0.84	0.8	1.77	8
RIDGE²	(1.74)	(1.64)	(1.81)	(0.76)	(0.74)	(0.76)	(0.76)	(0.76)	(0.8)	(0.77)	(0.74)	(1.68)	8
ELNET¹	1.2	0.88	0.82	0.77	0.83	0.9	0.79	0.81	0.86	0.84	0.8	0.75	12
ELNET²	(1.12)	(0.88)	(0.85)	(0.78)	(0.81)	(0.84)	(0.79)	(0.76)	(0.8)	(0.77)	(0.74)	(0.7)	12
ADA¹	1.22	1.12	0.89	0.83	0.93	1.11	0.83	0.81	0.86	0.84	0.8	0.76	9
ADA²	(1.28)	(1.05)	(0.94)	(0.85)	(0.9)	(0.94)	(0.86)	(0.76)	(0.81)	(0.78)	(0.74)	(0.71)	9
ADAELNET¹	1.2	1.09	0.88	0.82	0.92	1.06	0.82	0.81	0.86	0.84	0.8	0.76	11
ADAELNET²	(1.24)	(1.02)	(0.94)	(0.84)	(0.89)	(0.93)	(0.85)	(0.76)	(0.81)	(0.78)	(0.74)	(0.71)	11
RF.OLS¹	1.09	1.03	0.93	0.88	0.85	0.89	0.87	0.84	0.86	0.82	0.8	0.77	10
RF.OLS²	(1.1)	(1)	(1.07)	(0.99)	(0.89)	(0.96)	(0.94)	(0.84)	(0.84)	(0.79)	(0.78)	(0.75)	10
LASSO.RF¹	1.01	0.94	0.9	0.89	0.83	0.9	0.88	0.87	1	0.93	0.86	0.87	5
LASSO.RF²	(1.12)	(1.03)	(1.05)	(0.98)	(0.93)	(0.98)	(0.94)	(0.93)	(1)	(0.98)	(0.88)	(0.93)	5
BOOST¹	1.27	0.93	0.82	0.79	0.81	0.79	0.81	0.83	0.84	0.84	0.8	0.77	11
BOOST²	(1.38)	(1.02)	(0.91)	(0.81)	(0.85)	(0.87)	(0.91)	(0.81)	(0.86)	(0.83)	(0.75)	(0.76)	11
BVAR¹	1.32	1.15	1.18	1.13	1.11	1.14	1.12	1.21	1.23	1.16	1.12	1.08	1
BVAR²	(1.52)	(1.33)	(1.4)	(1.26)	(1.25)	(1.3)	(1.29)	(1.34)	(1.36)	(1.26)	(1.21)	(1.19)	1
FACTORS¹	2.34	2.48	1.51	1.43	1.23	1.31	1.22	1	1	0.97	1	1.01	0
FACTORS²	(1.85)	(1.7)	(1.52)	(1.42)	(1.19)	(1.35)	(1.26)	(1.09)	(1.06)	(1.02)	(1.01)	(0.99)	0
TFct¹	1.68	1.34	1.31	1.31	1.08	1.68	1.21	1.03	1.03	0.97	1.07	0.9	1
TFct²	(1.73)	(1.44)	(1.35)	(1.52)	(1.28)	(1.34)	(1.2)	(1.08)	(1.11)	(1.05)	(1.13)	(0.94)	1
ATFct¹	1.87	1.82	1.46	1.47	1.15	1.9	1.17	1.03	1.06	1.02	1.08	0.99	0
ATFct²	(1.89)	(1.57)	(1.59)	(1.54)	(1.31)	(1.56)	(1.41)	(1.1)	(1.15)	(1.11)	(1.21)	(1.15)	0
CSR¹	1.31	1.03	0.98	0.87	0.89	0.84	0.82	0.82	0.87	0.85	0.85	0.8	10
CSR²	(1.16)	(1.02)	(1.14)	(0.98)	(0.95)	(0.89)	(0.85)	(0.82)	(0.86)	(0.87)	(0.88)	(0.83)	10
XGB¹	0.89	0.86	0.89	0.88	0.91	0.85	0.87	0.87	0.88	0.84	0.86	0.85	10
XGB²	(0.97)	(0.93)	(1.06)	(0.98)	(0.99)	(0.94)	(0.92)	(0.94)	(0.93)	(0.90)	(0.90)	(0.90)	10
NN¹	2.31	2.76	1.88	1.61	2.43	1.77	1.79	1.52	1.81	1.77	2.01	2.33	0
NN²	(1.98)	(1.71)	(1.80)	(1.56)	(1.71)	(1.75)	(1.67)	(1.57)	(1.60)	(1.58)	(1.75)	(1.71)	0

Table A.6: Forecasting Errors for UNRATE since 2000.

RMSE are displayed in odd lines (with 1 as a superscript besides the model name) while MAE are displayed in even lines (with 2 as a superscript besides the model name). Cells in blue (gray) show the models included in the 50% model confidence set (MCS) using the squared error (absolute error) as loss function.

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	MCS Count
RW¹	1	1	1	1	1	1	1	1	1	1	1	1	0
RW²	1	1	1	1	1	1	1	1	1	1	1	1	0
AR¹	0.8	0.74	0.73	0.71	0.73	0.73	0.73	0.74	0.73	0.99	2.41	0.96	12
AR²	(0.79)	(0.82)	(0.72)	(0.73)	(0.75)	(0.72)	(0.72)	(0.72)	(0.72)	(0.83)	(0.66)	(0.79)	12
RF¹	0.82	0.71	0.79	0.76	0.77	0.75	0.76	0.77	0.74	0.98	1.23	0.96	11
RF²	(0.73)	(0.77)	(0.81)	(0.81)	(0.81)	(0.74)	(0.75)	(0.74)	(0.73)	(0.81)	1.23	(0.78)	11
LASSO¹	0.78	0.68	0.71	0.67	0.7	0.7	0.71	0.73	0.72	0.98	1.36	0.96	11
LASSO²	(0.71)	(0.72)	(0.69)	(0.68)	(0.7)	(0.66)	(0.69)	(0.69)	(0.7)	(0.82)	1.36	(0.78)	11
RIDGE¹	0.72	0.67	0.7	0.67	0.7	0.7	0.71	0.73	0.72	0.98	1.3	0.97	10
RIDGE²	(0.71)	(0.75)	(0.71)	(0.68)	(0.69)	(0.65)	(0.67)	(0.67)	(0.68)	(0.81)	1.3	0.88	10
ELNET¹	0.81	0.67	0.7	0.67	0.7	0.7	0.71	0.73	0.72	0.99	1.31	0.96	11
ELNET²	(0.73)	(0.71)	(0.68)	(0.68)	(0.69)	(0.65)	(0.67)	(0.68)	(0.69)	(0.82)	1.31	(0.78)	11
ADA¹	0.78	0.68	0.71	0.67	0.7	0.7	0.71	0.74	0.72	0.99	1.36	0.96	11
ADA²	(0.71)	(0.73)	(0.7)	(0.68)	(0.7)	(0.66)	(0.7)	(0.71)	(0.71)	(0.83)	1.36	(0.79)	11
ADAELNET¹	0.78	0.68	0.71	0.67	0.7	0.7	0.71	0.73	0.72	0.99	1.37	0.96	11
ADAELNET²	(0.71)	(0.72)	(0.7)	(0.68)	(0.7)	(0.66)	(0.7)	(0.71)	(0.7)	(0.83)	1.37	(0.79)	11
RF.OLS¹	1.13	0.73	0.73	0.7	0.71	0.71	0.72	0.74	0.73	0.99	1.4	0.96	9
RF.OLS²	(0.88)	(0.79)	(0.72)	(0.71)	(0.73)	(0.69)	(0.72)	(0.73)	(0.73)	0.85	1.4	0.82	9
LASSO.RF¹	0.84	0.68	0.7	0.68	0.73	0.7	0.71	0.73	0.72	0.98	1.07	0.96	9
LASSO.RF²	(0.75)	(0.73)	(0.69)	(0.71)	(0.75)	(0.66)	(0.69)	(0.69)	(0.73)	0.84	1.07	0.81	9
BOOST¹	0.83	0.71	0.71	0.68	0.7	0.7	0.71	0.73	0.72	0.99	2.39	0.96	12
BOOST²	(0.78)	(0.77)	(0.69)	(0.69)	(0.69)	(0.66)	(0.68)	(0.69)	(0.7)	(0.83)	(0.66)	(0.8)	12
BVAR¹	0.89	0.87	0.86	0.86	0.86	0.86	0.86	0.87	0.87	0.98	2.4	0.96	12
BVAR²	(0.75)	(0.79)	(0.74)	(0.78)	(0.79)	(0.78)	(0.79)	(0.79)	(0.82)	(0.85)	(0.68)	(0.82)	12
FACTORS¹	1.19	0.82	0.82	0.74	0.81	0.76	0.73	0.75	0.74	0.99	2.4	0.96	11
FACTORS²	(0.94)	(0.89)	(0.82)	(0.77)	(0.84)	(0.75)	(0.71)	(0.71)	(0.73)	(0.85)	(0.68)	0.81	11
TFct¹	1.29	0.81	0.75	0.7	0.81	0.73	0.72	0.77	0.76	0.99	2.4	0.96	8
TFct²	(0.98)	(0.89)	(0.74)	(0.73)	0.83	0.74	(0.71)	(0.75)	(0.75)	0.86	(0.69)	0.83	8
ATFct¹	1.2	0.76	0.8	0.74	0.8	0.79	0.79	0.76	0.77	1	2.41	0.97	7
ATFct²	(0.96)	(0.84)	(0.82)	0.81	(0.86)	(0.79)	(0.8)	0.77	0.77	0.88	(0.69)	0.84	7
CSR¹	0.92	0.73	0.73	0.69	0.72	0.7	0.71	0.73	0.72	0.98	2.4	0.96	12
CSR²	(0.76)	(0.79)	(0.74)	(0.72)	(0.74)	(0.68)	(0.69)	(0.7)	(0.71)	(0.83)	(0.66)	(0.79)	12
XGB¹	1	0.7	1.09	1.01	1.04	0.91	0.94	0.84	0.84	0.99	0.98	0.96	12
XGB²	(0.8)	(0.77)	(0.99)	(0.99)	(1)	(0.87)	(0.83)	(0.8)	(0.81)	(0.83)	(0.88)	(0.8)	12
NN¹	0.94	0.67	0.76	0.7	0.71	0.75	0.73	0.74	0.73	0.99	1	0.99	3
NN²	0.91	(0.79)	0.84	0.79	(0.76)	0.84	0.82	(0.76)	0.78	0.92	0.98	0.93	3

Table A.7: Forecasting Errors for INDPRO since 2000.

RMSE are displayed in odd lines (with 1 as a superscript besides the model name) while MAE are displayed in even lines (with 2 as a superscript besides the model name). Cells in blue (gray) show the models included in the 50% model confidence set (MCS) using the squared error (absolute error) as loss function.

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	MCS Count
RW¹	1	1	1	1	1	1	1	1	1	1	1	1	0
RW²	1	1	1	1	1	1	1	1	1	1	1	1	0
AR¹	0.97	0.72	0.71	0.76	0.78	0.78	0.78	0.74	0.87	0.9	0.9	0.91	12
AR²	(0.82)	(0.75)	(0.77)	(0.76)	(0.75)	(0.76)	(0.7)	(0.73)	(0.79)	(0.75)	(0.75)	(0.76)	12
RF¹	0.96	0.74	0.78	0.83	0.81	0.81	0.79	0.76	0.87	0.9	0.9	0.92	10
RF²	(0.81)	(0.77)	(0.86)	(0.84)	0.81	0.82	(0.74)	(0.76)	(0.8)	(0.75)	(0.76)	(0.77)	10
LASSO¹	0.94	0.74	0.71	0.76	0.77	0.79	0.77	0.75	0.87	0.9	0.9	0.91	12
LASSO²	(0.81)	(0.76)	(0.79)	(0.74)	(0.74)	(0.76)	(0.7)	(0.72)	(0.78)	(0.73)	(0.74)	(0.76)	12
RIDGE¹	1.49	0.92	1.16	1.16	1.2	1.08	0.81	0.77	0.91	0.92	0.93	0.94	3
RIDGE²	1.09	0.87	0.99	1.15	1.08	1	0.79	(0.78)	0.86	(0.78)	(0.81)	0.84	3
ELNET¹	0.96	0.74	0.7	0.75	0.77	0.78	0.77	0.75	0.87	0.9	0.9	0.91	12
ELNET²	(0.82)	(0.75)	(0.78)	(0.74)	(0.74)	(0.75)	(0.7)	(0.72)	(0.77)	(0.73)	(0.74)	(0.75)	12
ADA¹	0.92	0.75	0.72	0.75	0.77	0.79	0.77	0.75	0.87	0.9	0.9	0.91	12
ADA²	(0.81)	(0.76)	(0.79)	(0.74)	(0.74)	(0.76)	(0.7)	(0.72)	(0.77)	(0.73)	(0.74)	(0.75)	12
ADAELNET¹	0.96	0.74	0.72	0.75	0.77	0.78	0.77	0.75	0.87	0.9	0.9	0.91	12
ADAELNET²	(0.82)	(0.76)	(0.78)	(0.73)	(0.74)	(0.76)	(0.7)	(0.72)	(0.78)	(0.73)	(0.74)	(0.75)	12
RF.OLS¹	1.19	0.82	0.79	0.79	0.79	0.8	0.79	0.77	0.9	0.93	0.94	0.91	10
RF.OLS²	(0.88)	(0.79)	(0.84)	(0.79)	0.79	0.8	(0.73)	(0.75)	(0.8)	(0.76)	(0.78)	(0.76)	10
LASSO.RF¹	0.95	0.71	0.73	0.77	0.8	0.78	0.78	0.75	0.87	0.9	0.91	0.92	10
LASSO.RF²	(0.85)	(0.75)	(0.77)	(0.77)	0.79	(0.77)	(0.71)	(0.73)	(0.79)	(0.78)	(0.78)	0.78	10
BOOST¹	1.09	0.81	0.75	0.76	0.76	0.79	0.77	0.75	0.87	0.9	0.9	0.91	12
BOOST²	(0.87)	(0.8)	(0.81)	(0.75)	(0.72)	(0.76)	(0.7)	(0.73)	(0.77)	(0.74)	(0.75)	(0.77)	12
BVAR¹	1.07	0.9	0.89	0.9	0.88	0.93	0.92	0.9	0.88	0.9	0.91	0.94	11
BVAR²	(0.86)	(0.76)	(0.82)	(0.8)	(0.79)	(0.84)	(0.8)	(0.81)	(0.83)	(0.77)	(0.8)	0.82	11
FACTORS¹	1.78	1.14	0.89	0.87	0.93	0.93	0.8	0.78	0.88	0.91	0.91	0.93	10
FACTORS²	(1.08)	(0.91)	(0.88)	(0.82)	(0.84)	0.86	(0.74)	(0.75)	(0.79)	(0.77)	(0.78)	0.81	10
TFct¹	1.7	0.93	0.82	0.83	0.82	0.83	0.78	0.8	0.88	0.91	0.92	0.93	7
TFct²	(1.04)	(0.83)	(0.86)	0.83	0.82	0.82	(0.71)	(0.77)	(0.8)	(0.78)	0.8	0.82	7
ATFct¹	1.44	0.91	0.86	0.83	0.85	0.84	0.8	0.76	0.89	0.93	0.92	0.93	3
ATFct²	(0.92)	(0.83)	0.91	0.85	0.87	0.85	0.76	(0.77)	0.84	0.8	0.81	0.8	3
CSR¹	1.2	0.82	0.82	0.79	0.78	0.8	0.78	0.75	0.87	0.91	0.9	0.91	12
CSR²	(0.89)	(0.8)	(0.84)	(0.77)	(0.75)	(0.77)	(0.71)	(0.74)	(0.78)	(0.75)	(0.77)	(0.77)	12
XGB¹	1.01	0.76	0.93	0.97	0.95	0.87	0.85	0.85	0.83	0.87	0.92	0.92	8
XGB²	(0.84)	(0.77)	(0.96)	0.93	0.88	0.84	(0.76)	(0.82)	0.82	(0.75)	(0.79)	(0.78)	8
NN¹	1.11	0.73	0.78	0.86	0.83	0.81	0.77	0.81	0.84	0.99	1.04	1.01	1
NN²	0.96	(0.83)	0.93	0.93	0.88	0.85	0.76	0.83	0.91	0.85	0.95	0.89	1

Table A.8: Forecasting Errors for EXUSUKx since 2000.

RMSE are displayed in odd lines (with 1 as a superscript besides the model name) while MAE are displayed in even lines (with 2 as a superscript besides the model name). Cells in blue (gray) show the models included in the 50% model confidence set (MCS) using the squared error (absolute error) as loss function.

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	MCS Count
RW¹	1	1	1	1	1	1	1	1	1	1	1	1	10
RW²	1	1	(1)	(1)	1	1	1	1	1	1	1	(1)	3
AR¹	1.26	1.03	1.01	0.97	0.9	0.85	0.82	0.88	0.84	0.78	0.78	0.75	12
AR²	1.37	1.1	(1.13)	(1.02)	0.9	(0.85)	0.8	(0.84)	0.79	0.73	(0.74)	(0.74)	7
RF¹	0.86	0.8	0.85	0.94	0.89	1.02	0.95	1	0.87	0.83	0.83	0.82	12
RF²	0.89	0.85	(1.02)	(1.06)	1.02	1.11	0.99	0.97	0.86	0.83	0.85	(0.86)	5
LASSO¹	1.32	0.95	1	1.17	1.02	0.92	0.91	0.82	0.79	0.77	0.77	0.8	12
LASSO²	1.11	1.08	(1.06)	(1.11)	1.03	0.95	0.93	(0.81)	(0.73)	0.69	(0.71)	(0.87)	7
RIDGE¹	2.75	2.58	2.37	0.87	0.88	0.87	2.05	2.01	2.02	1.96	1.93	1.71	3
RIDGE²	2.81	2.7	2.74	(0.84)	0.85	(0.83)	2.26	2.12	2.18	2.13	2.13	2.02	3
ELNET¹	1.14	0.91	0.95	0.92	0.94	0.89	0.89	0.86	0.81	0.77	0.77	0.79	12
ELNET²	1.02	0.96	(1)	(0.96)	0.96	0.9	0.84	(0.79)	(0.73)	0.69	(0.7)	(0.82)	6
ADA¹	1.44	1.08	1.07	1.42	1.13	0.98	0.94	0.82	0.77	0.78	0.78	0.91	11
ADA²	1.23	1.32	(1.14)	(1.29)	1.21	1.05	1.01	(0.84)	0.76	0.7	(0.72)	(1.04)	5
ADAELNET¹	1.42	1.04	1.06	1.39	1.11	0.97	0.95	0.82	0.77	0.78	0.78	0.9	11
ADAELNET²	1.2	1.27	(1.13)	(1.28)	1.18	1.04	1	(0.84)	0.76	0.7	(0.72)	(1.03)	5
RF.OLS¹	0.97	1.12	1.09	1.08	1.1	0.98	1.06	0.97	0.86	0.81	0.81	0.87	11
RF.OLS²	1.11	1.31	1.38	(1.23)	1.23	1.15	1.07	0.92	0.93	0.91	0.93	1.06	1
LASSO.RF¹	1.18	0.97	0.96	0.94	0.94	1.09	0.98	0.99	0.96	0.95	0.89	0.87	10
LASSO.RF²	1.23	1.11	(1.14)	(1.14)	1.14	1.2	1.11	1.1	0.99	1.01	0.96	(0.96)	3
BOOST¹	2.12	1.18	1.01	0.97	1.03	1.06	0.96	0.93	0.9	0.84	0.8	0.81	10
BOOST²	2.03	1.43	1.26	(1.11)	1.22	1.21	1.18	0.99	1.06	0.97	0.9	(1.01)	2
BVAR¹	1.54	1.32	1.31	1.34	1.33	1.3	1.28	1.29	1.23	1.17	1.16	1.16	4
BVAR²	2.04	1.63	1.69	1.65	1.65	1.67	1.52	1.5	1.43	1.35	1.36	1.44	0
FACTORS¹	1.95	3.07	1.99	2.98	2.62	1.92	1.87	1.08	0.97	0.92	0.92	0.95	6
FACTORS²	2.22	2.34	2.25	2.37	1.98	1.69	1.54	1.23	1.12	1.08	1.06	1.09	0
TFct¹	1.37	1.67	1.87	2.73	1.34	1.37	1.4	1.49	1.13	0.91	1.08	1.07	3
TFct²	1.97	2.03	2.06	2.35	1.73	1.66	1.44	1.67	1.22	1.07	1.22	1.36	0
ATFct¹	2.14	2.99	2.19	2.45	1.53	1.48	1.36	1.33	1.37	1.1	1.1	1.07	3
ATFct²	2.69	2.63	2.36	2.22	1.89	1.67	1.49	1.44	1.52	1.35	1.32	1.42	0
CSR¹	1.29	1	0.97	1.21	1.25	0.98	0.9	0.83	0.79	0.77	0.78	0.85	12
CSR²	1.3	1.1	(1.14)	(1.25)	1.35	1.1	0.97	0.92	0.83	0.8	(0.79)	(0.96)	4
XGB¹	0.93	0.89	0.88	1.35	0.96	0.98	0.91	0.95	1.35	1.13	1.09	1.23	11
XGB²	1.14	1.07	(1.08)	(1.19)	1.09	1.08	1.02	1.04	1.05	0.92	0.96	(1.04)	3
NN¹	2.49	2.28	4.1	3.24	2.05	2.82	2.21	2.18	1.66	1.57	2.19	2.32	0
NN²	2.26	2.1	2.52	2.14	2.04	1.99	2.01	1.83	1.64	1.39	1.71	1.86	0

Table A.9: Forecasting Errors for CUMFNS since 2000.

RMSE are displayed in odd lines (with 1 as a superscript besides the model name) while MAE are displayed in even lines (with 2 as a superscript besides the model name). Cells in blue (gray) show the models included in the 50% model confidence set (MCS) using the squared error (absolute error) as loss function.

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	MCS Count
RW¹	1	1	1	1	1	1	1	1	1	1	1	1	0
RW²	1	1	1	1	1	1	1	1	1	1	1	1	0
AR¹	0.99	0.81	0.76	0.76	0.78	0.79	0.75	0.76	0.77	0.86	0.91	0.92	11
AR²	(0.85)	(0.86)	(0.79)	(0.78)	(0.76)	(0.77)	(0.7)	(0.78)	(0.74)	(0.76)	(0.78)	0.78	11
RF¹	0.94	0.75	0.78	0.8	0.8	0.81	0.78	0.78	0.77	0.85	0.91	0.92	12
RF²	(0.8)	(0.83)	(0.83)	(0.83)	(0.79)	(0.81)	(0.72)	(0.79)	(0.72)	(0.74)	(0.76)	(0.75)	12
LASSO¹	0.86	0.74	0.71	0.74	0.76	0.77	0.75	0.75	0.75	0.84	0.91	0.91	12
LASSO²	(0.78)	(0.81)	(0.76)	(0.75)	(0.75)	(0.76)	(0.68)	(0.74)	(0.69)	(0.72)	(0.76)	(0.74)	12
RIDGE¹	0.95	0.73	0.73	0.73	0.75	0.77	0.74	0.76	0.76	0.85	0.91	0.96	11
RIDGE²	(0.89)	(0.87)	(0.85)	(0.76)	(0.74)	(0.77)	(0.69)	(0.76)	(0.72)	(0.75)	(0.77)	0.85	11
ELNET¹	0.94	0.69	0.7	0.73	0.75	0.77	0.74	0.76	0.76	0.84	0.9	0.91	12
ELNET²	(0.79)	(0.78)	(0.76)	(0.76)	(0.73)	(0.75)	(0.68)	(0.75)	(0.71)	(0.73)	(0.76)	(0.75)	12
ADA¹	0.89	0.77	0.73	0.75	0.79	0.78	0.76	0.76	0.76	0.84	0.92	0.91	12
ADA²	(0.79)	(0.83)	(0.78)	(0.77)	(0.77)	(0.77)	(0.7)	(0.75)	(0.71)	(0.72)	(0.77)	(0.74)	12
ADAELNET¹	0.89	0.77	0.72	0.75	0.78	0.78	0.76	0.76	0.76	0.84	0.92	0.91	12
ADAELNET²	(0.79)	(0.83)	(0.77)	(0.77)	(0.77)	(0.76)	(0.69)	(0.75)	(0.71)	(0.72)	(0.77)	(0.74)	12
RF.OLS¹	1.19	0.83	0.76	0.75	0.77	0.78	0.76	0.79	0.77	0.86	0.91	0.91	12
RF.OLS²	(0.87)	(0.85)	(0.8)	(0.79)	(0.77)	(0.78)	(0.71)	(0.79)	(0.73)	(0.75)	(0.76)	(0.74)	12
LASSO.RF¹	0.96	0.71	0.73	0.75	0.78	0.77	0.75	0.76	0.77	0.85	0.92	0.93	9
LASSO.RF²	(0.86)	(0.82)	(0.78)	(0.8)	0.8	(0.79)	(0.7)	(0.8)	(0.74)	(0.77)	0.82	0.8	9
BOOST¹	1.08	0.81	0.75	0.75	0.76	0.78	0.74	0.76	0.76	0.84	0.9	0.91	12
BOOST²	(0.86)	(0.86)	(0.8)	(0.77)	(0.74)	(0.75)	(0.67)	(0.74)	(0.7)	(0.73)	(0.75)	(0.74)	12
BVAR¹	1.05	0.9	0.87	0.87	0.86	0.92	0.88	0.89	0.89	0.85	0.93	0.95	11
BVAR²	(0.83)	(0.83)	(0.81)	(0.8)	(0.79)	(0.83)	(0.77)	(0.84)	(0.79)	(0.77)	(0.81)	0.82	11
FACTORS¹	1.83	1.17	1.03	0.98	0.81	0.82	0.76	0.78	0.78	0.85	0.91	0.92	12
FACTORS²	(1.06)	(0.99)	(0.89)	(0.86)	(0.78)	(0.8)	(0.7)	(0.78)	(0.74)	(0.76)	(0.78)	(0.77)	12
TFct¹	1.72	1	0.91	0.76	0.8	0.81	0.78	0.76	0.77	0.86	0.92	0.93	8
TFct²	(1.05)	(0.95)	(0.87)	(0.81)	(0.78)	(0.8)	(0.72)	(0.78)	0.76	0.79	0.81	0.81	8
ATFct¹	1.36	0.88	0.89	0.75	0.82	0.86	0.89	0.83	0.86	0.87	0.92	0.94	4
ATFct²	(0.87)	(0.91)	(0.86)	(0.79)	0.83	0.84	0.79	0.84	0.81	0.8	0.81	0.82	4
CSR¹	1.09	0.77	0.75	0.73	0.76	0.77	0.74	0.76	0.75	0.84	0.9	0.91	12
CSR²	(0.85)	(0.83)	(0.78)	(0.75)	(0.73)	(0.76)	(0.68)	(0.75)	(0.71)	(0.74)	(0.76)	(0.75)	12
XGB¹	0.91	0.81	0.94	1	0.97	0.89	0.82	0.82	0.84	0.86	0.92	0.93	8
XGB²	(0.8)	(0.85)	(0.93)	(0.91)	(0.89)	0.85	0.76	0.82	0.78	(0.76)	(0.77)	(0.74)	8
NN¹	1.28	0.81	0.79	0.81	0.88	0.79	0.82	0.87	0.84	0.9	1.08	0.98	2
NN²	1.02	(0.94)	(0.88)	0.91	0.93	0.86	0.82	0.96	0.88	0.86	1.01	0.88	2

Table A.10: Forecasting Errors for AAA since 2000.

RMSE are displayed in odd lines (with 1 as a superscript besides the model name) while MAE are displayed in even lines (with 2 as a superscript besides the model name). Cells in blue (gray) show the models included in the 50% model confidence set (MCS) using the squared error (absolute error) as loss function.

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	MCS Count
RW¹	1	1	1	1	1	1	1	1	1	1	1	1	0
RW²	1	1	1	1	1	1	1	1	1	1	1	1	0
AR¹	0.84	0.68	0.7	0.71	0.69	0.71	0.68	0.74	0.73	0.71	0.69	0.71	11
AR²	0.83	(0.7)	(0.68)	(0.69)	0.67	(0.7)	(0.65)	(0.7)	0.72	0.73	(0.69)	(0.69)	8
RF¹	0.76	0.68	0.73	0.74	0.74	0.74	0.71	0.78	0.76	0.71	0.71	0.72	9
RF²	(0.76)	(0.71)	0.7	0.72	0.71	(0.73)	0.69	0.75	0.75	(0.71)	(0.69)	(0.7)	6
LASSO¹	0.8	0.68	0.69	0.7	0.68	0.69	0.68	0.73	0.72	0.71	0.7	0.7	12
LASSO²	0.76	(0.71)	(0.67)	(0.68)	(0.65)	(0.68)	(0.64)	(0.7)	(0.7)	(0.71)	(0.69)	(0.69)	11
RIDGE¹	0.93	0.98	1.13	0.7	0.68	0.69	0.68	0.73	0.72	0.71	0.69	1.12	9
RIDGE²	0.92	1	0.99	(0.68)	(0.65)	(0.68)	(0.65)	(0.7)	(0.7)	(0.71)	(0.68)	1.01	8
ELNET¹	0.78	0.67	0.69	0.7	0.68	0.69	0.68	0.73	0.72	0.71	0.7	0.7	11
ELNET²	(0.75)	(0.7)	0.67	(0.68)	(0.65)	(0.68)	(0.64)	(0.7)	(0.7)	(0.71)	(0.69)	(0.69)	11
ADA¹	0.83	0.68	0.69	0.7	0.68	0.69	0.68	0.73	0.72	0.71	0.71	0.7	11
ADA²	0.79	(0.71)	0.67	(0.68)	(0.65)	(0.68)	(0.65)	(0.69)	(0.7)	(0.72)	(0.69)	(0.69)	10
ADAELNET¹	0.81	0.68	0.69	0.7	0.68	0.69	0.68	0.73	0.72	0.71	0.7	0.7	11
ADAELNET²	0.78	(0.71)	0.67	(0.68)	(0.65)	(0.68)	(0.65)	(0.69)	(0.7)	(0.71)	(0.69)	(0.69)	10
RF.OLS¹	0.82	0.72	0.76	0.75	0.82	0.76	0.74	0.79	0.74	0.74	0.7	0.73	8
RF.OLS²	0.79	0.76	0.73	0.73	0.75	0.75	0.7	0.74	0.72	(0.73)	(0.7)	0.72	2
LASSO.RF¹	0.8	0.68	0.71	0.74	0.71	0.72	0.72	0.76	0.77	0.76	0.7	0.73	7
LASSO.RF²	0.82	(0.71)	0.7	0.71	0.69	(0.72)	0.7	0.73	0.77	0.76	0.72	0.72	2
BOOST¹	0.77	0.67	0.7	0.71	0.69	0.7	0.68	0.74	0.72	0.71	0.69	0.7	11
BOOST²	(0.75)	(0.69)	(0.67)	(0.69)	(0.65)	(0.67)	(0.65)	(0.7)	(0.7)	(0.71)	(0.68)	(0.69)	12
BVAR¹	0.88	0.82	0.78	0.8	0.82	0.86	0.82	0.86	0.86	0.81	0.79	0.8	2
BVAR²	0.84	0.82	0.78	0.78	0.79	0.83	0.79	0.81	0.84	0.82	0.79	0.79	0
FACTORS¹	1.18	0.93	0.87	0.88	0.84	0.93	0.96	0.91	0.84	0.77	0.74	0.78	1
FACTORS²	0.87	0.81	0.8	0.81	0.78	0.87	0.86	0.84	0.78	0.78	0.74	0.77	0
TFct¹	0.97	1.3	0.95	1.05	0.77	1.28	0.78	0.79	0.84	0.75	0.76	0.8	4
TFct²	0.89	0.88	0.78	0.81	0.74	0.96	0.72	0.76	0.79	0.77	0.78	0.81	0
ATFct¹	0.83	0.8	0.82	0.86	0.82	0.95	0.78	0.8	0.78	0.77	0.76	0.83	2
ATFct²	0.81	0.82	0.79	0.83	0.79	0.83	0.76	0.77	0.8	0.77	0.75	0.8	0
CSR¹	0.77	0.67	0.71	0.73	0.71	0.76	0.7	0.75	0.73	0.72	0.71	0.71	9
CSR²	(0.74)	(0.7)	(0.68)	(0.7)	0.68	(0.71)	(0.66)	(0.7)	(0.71)	(0.73)	(0.7)	(0.71)	11
XGB¹	0.81	0.76	0.8	0.79	0.76	0.77	0.75	0.85	0.81	0.77	0.77	0.79	4
XGB²	0.83	0.78	0.8	0.79	0.74	0.77	0.74	0.84	0.82	0.78	0.78	0.79	0
NN¹	1.23	1.17	1.46	1.12	1.26	1.09	1.44	1.06	0.98	1.19	1.26	1.06	0
NN²	1.05	1.06	1.11	1	1.05	1	1.1	0.96	0.98	1.03	1	0.99	0

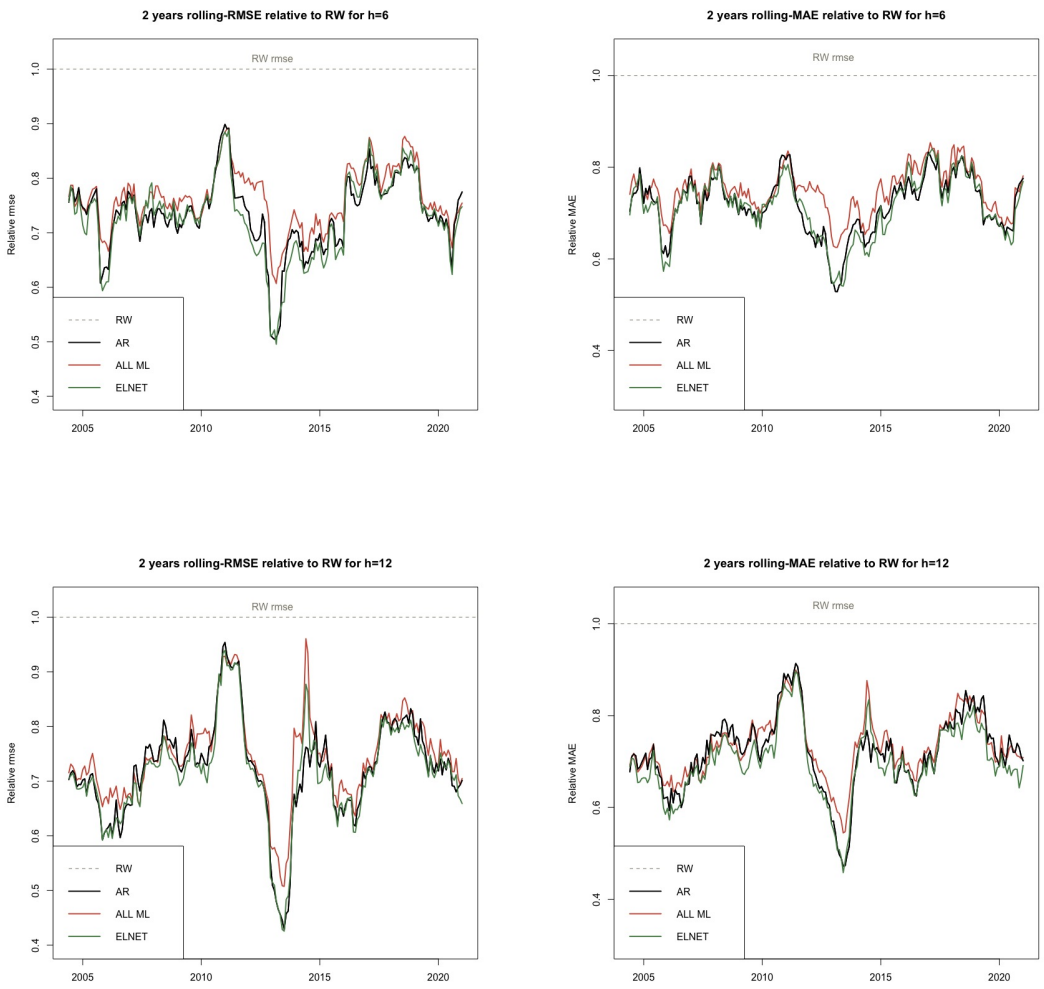

Figure A.13: The figure displays the root mean squared errors (RMSE) and mean absolute error (MAE) computed over rolling windows of 24 observations. Panels (a) and (b) display the results for six-months-ahead forecasts (h = 6), while panels (c) and (d) display the results for twelve-months-ahead forecasts (h = 12).

A.3.1
Benchmark Models

We had two benchmark models in this paper. The first one is the Random Walk model, where for h=1,...,12, the forecasts are computed as follows:

$$\hat{Y}_{t+h|t} = Y_t \tag{A-1}$$

The second benchmark is the AR of order p, where p is chosen by the Bayesian information criterion (BIC). The forecast equation for horizon h is the following OLS:

$$\hat{Y}_{t+h|t} = \hat{\alpha}_h + \hat{\beta}_{1,h}Y_t + ... + \hat{\beta}_{p,h}Y_{t-p+1} \tag{A-2}$$

A.3.2
Machine Learning Models

Almost every ML model used in this study can be find in detail in the Appendix B of Medeiros et al. (2021), which we refer to for morei information. It is important to note that our specifications were the same as those detailed in that study. For instance, for the CSR, we used 25 candidate variables, four selected variables and the same pretesting procedure as they did.

However, there is a single model used here that is new and deserves an explanation, even though its performance was poor compared to other ML models. It is the Alternative Target Factors.

Factor models using principal components is a very popular approach for deriving a low-dimensional set of features (usually referred to as factors) from a large set of variables. The idea is to extract common components from all variables, thus reducing the model dimension.

To improve the forecasting performance of factor models, Bai and Ng (2008) proposed targeting the predictors. The authors write that at that time, standard procedure was so that the principal components were always extracted from the same large data set, regardless of the series to be forecasted. In this way, they developed an algorithm (among others) that essentially uses only variables whose marginal predictive power for the dependent variable is significant at some prescribed level in the factor analysis. The assumption behind this idea is that the principal components estimated from a large group of variables can be dominated by principal components estimated from a smaller set of predictors.

It is important to explain that Medeiros et al. (2021) (and this study) already include a slightly different version of Bai and Ng (2008) target factors in the set of ML models. Our alternative target factors is another different

approach. In short, the idea of our approach is to perform a Bagging in the pretest procedure Breiman (1996) instead of a simple t-test, as in the usual target factors model.

Note that the idea behind our procedure is one of the main conclusions of Bühlmann and Yu (2002), i.e., the fact that the subset model selection through t-tests generate unstable predictors, in a way that we expected that bagging should reduce the estimator's variance. The idea is to "turn" the hard threshold procedure of the t-test into a soft one. We achieve that by averaging the same t-statistics over many bootstrap samples.

Let K be the number of possible predictors, N the number of observations in our data, B the number of bootstraps and A a null-matrix with BxK dimension. We can describe the procedure as follows:

Table A.11: Algorithm of our alternative target factor model

Algorithm 3

1: Set j=1
2: Sample N observations from the data with replacement (standard bootstrap)
3: **for** $i \in \{1, 2 \ldots, k\}$ **do**
4: regress y_t on y_{t-1} and $X_{i,t-1}$. Let t_i denote the t-statistic associated with $X_{i,t-h}$
5: Save the K-vector of the t-statistics (in their absolute value) in the j-th line of A.
6: Set j=j+1
7: **end for**
8: Repeat procedures above B times
9: Take the mean of each column of A and choose a threshold significance level α
10: Estimate the factors from the selected variables by the standard method of principal components.