Now- and Backcasting Initial Claims with High-Dimensional Daily Internet Search-Volume Data

Daniel Borup
Aarhus University, CREATES, and DFI
dborup@econ.au.dk

David E. Rapach*
Washington University in St. Louis and Saint Louis University
david.rapach@wustl.edu

Erik Christian Montes Schütte
Aarhus University, CREATES, and DFI
christianms@econ.au.dk

September 11, 2020

*Corresponding author. Send correspondence to David Rapach, Olin School of Business, Washington University in St. Louis, St. Louis, MO 63130; e-mail: david.rapach@wustl.edu.

Now- and Backcasting Initial Claims with High-Dimensional Daily Internet Search-Volume Data

Abstract

We generate a sequence of now- and backcasts of weekly unemployment insurance initial claims (UI) based on a rich trove of daily Google Trends (GT) search-volume data for terms related to *unemployment*. To harness the information in a high-dimensional set of daily GT terms, we estimate predictive models using machine-learning techniques in a mixed-frequency framework. The sequence of now- and backcasts are made ten days to one day before the release of the UI figure on Thursday of each week. In a simulated out-of-sample exercise, now- and backcasts of weekly UI that incorporate the information in the daily GT terms substantially outperform those based on an autoregressive benchmark model, especially since the advent of the COVID-19 crisis. The improvements in predictive accuracy relative to the autoregressive benchmark generally increase as the now- and backcasts include additional daily GT data, with reductions in root mean squared error of up to approximately 50%. Variable-importance measures reveal that the GT terms become more relevant for predicting UI during the crisis, while partial-dependence plots indicate that linear specifications are largely adequate for capturing the predictive information in the GT terms. We are in the process of creating a website that will provide updated, real-time now- and backcasts of UI on a daily basis.

JEL classifications: C45, C53, C55, E24, E27, J65

Key words: Unemployment insurance, Internet search, Mixed-frequency data, LASSO, Elastic net, Neural network, Partial-dependence plot, Variable importance

1 Introduction

The COVID-19 crisis has created economic upheaval in the United States, including historically unprecedented levels of unemployment insurance initial claims (UI). After a national emergency was declared on March 13, 2020 and closures of non-essential retail establishments were ordered in many parts of the country, UI spiked in late March, reaching a (seasonally adjusted) record of 6,867,000 for the week ending March 28, 2020. By comparison, the peak in UI during the Great Recession was "only" 665,000 (for the week ending March 28, 2009). While UI has subsequently declined, it remains at elevated levels. Because it provides important information about the US labor market and is reported at the weekly frequency, UI has become perhaps the most closely watched economic variable during the COVID-19 crisis. Reflecting its relevance and timeliness, Lewis, Mertens, and Stock (2020) include UI in their recently developed weekly economic indicator for the United States.

In this paper, we use a rich trove of daily internet search-volume data from Google Trends (GT) to predict UI, with an eye toward improving prediction during the COVID-19 crisis. Because we cannot know ex ante the most relevant GT search terms for predicting UI, we employ a high-dimensional set of GT terms related to *unemployment*. We then rely on machine-learning techniques to harness the relevant information in the terms. Specifically, we use daily data for 103 *unemployment*-related GT terms for the most recent seven days to generate a sequence of now- and backcasts of a given week's (Sunday through Saturday) UI, in anticipation of the release of the figure by the Department of Labor on Thursday of the following week. The sequence of now- and backcasts incorporates the most recent daily GT data as they become available, which allows us to investigate the "term structure" of the flow of information with respect to predictive accuracy. The sequence of now- and backcasts of week-t UI are made from ten days to one day before the UI release on Thursday of week $t + 1$.

Each of our predictive models relates UI to its first (or second) lag—in recognition of the serial correlation in UI—as well as seven days of GT data.[1] Each model thus contains $7 \times 103 + 1 = 722$ predictors (or inputs), so that ours is a high-dimensional setting. We begin with a linear specification for the predictive models underpinning the now- and backcasts. To guard against overfitting in our high-dimensional setting, we estimate the linear predictive models via the *least absolute shrinkage and selection operator* (LASSO, Tibshirani 1996) and *elastic net* (ENet, Zou and Hastie 2005). The LASSO and ENet are popular machine-learning devices, which improve prediction in high-dimensional settings by including a penalty term in the objective function for estimating the model's parameters. Intuitively, the penalty term works to shrink the parameters toward zero, thereby helping to prevent overfitting. Because their penalty terms include an ℓ_1 component, the LASSO and ENet permit shrinkage to zero, so that they facilitate model interpretation by performing variable selection.

To allow for more complex, nonlinear predictive relationships, we also use artificial neural networks (ANNs) to generate now- and backcasts of UI based on the 722 predictors. ANNs contain one or more hidden layers, each of which contains multiple neurons that transmit predictive signals through the network. Under a reasonable set of restrictions, a single-layer ANN with a sufficient number of neurons can approximate any smooth function (e.g., Cybenko 1989; Funahashi 1989; Hornik, Stinchcombe, and White 1989; Hornik 1991; Barron 1994). Because ANNs with multiple hidden layers are often used in practice, we consider ANNs with one to three hidden layers (NN1, NN2, and NN3, respectively).[2] We fit the ANNs using the recently developed Adam stochastic gradient descent (SGD) algorithm (Kingma and Ba 2015).

Our use of mixed-frequency data in the predictive models is a version of the unrestricted mixed-data sampling (U-MIDAS) approach of Foroni, Marcellino, and Schumacher (2015). The restricted MIDAS approach (Ghysels, Santa-Clara, and Valkanov 2005) imposes a lag-

[1] The inclusion of the first or second lag of UI is determined by the timing of the UI release, as detailed in Section 3.

[2] ANNs with one or two (three or more) hidden layers are typically referred to as shallow (deep) neural networks.

polynomial structure on the higher-frequency data. Instead of somewhat arbitrarily imposing a lag-polynomial structure ex ante, we harness machine learning to fit the weights for the individual variables in the predictive models in our high-dimensional setting. The U-MIDAS framework also allows us to analyze how the flow of information affects the accuracy of the sequence of now- and backcasts, as each successive element in the sequence includes an additional day of GT data. In essence, we combine two branches of the economic forecasting literature, one that applies machine-learning methods (e.g., Diebold and Shin 2019; Kotchoni, Leroux, and Stevanovic 2019; Medeiros et al. forthcoming) and one that employs mixed-frequency data (e.g., Clements and Galvão 2008; Foroni and Marcellino 2014; Brave, Butters, and Justiniano 2019).

We find that the information in our high-dimensional set of daily GT terms is indeed useful for now- and backcasting weekly UI, even up to ten days before its release date. This finding holds for both non-seasonally and seasonally adjusted UI data. Specifically, the predictions for models that include GT terms generate substantial improvements in root mean squared error (RMSE) vis-à-vis an autoregressive (AR) benchmark model. For an out-of-sample period spanning the first week of 2019 through the last week of July of 2020, all of the now- and backcasts based on the daily GT trends deliver a lower RMSE than the AR benchmark, with reductions in RMSE of up to 53%. The improvement in predictive accuracy offered by the daily GT terms is dramatic during the advent of the COVID-19 crisis. In sum, our now- and backcasts of weekly UI based on daily GT terms dominate those based on an AR model that ignores the information in the GT terms, especially during the COVID-19 crisis.

The now- and backcasts based on linear models estimated via the LASSO and ENet generally perform better than those based on ANNs for seasonally adjusted UI. For non-seasonally adjusted UI, the ANNs have a slight performance edge in a number of cases. We also consider three *ensembles* for a given now- or backcast of UI.[3] The first takes the average

[3]Ensembles are often refereed to as *combinations* in the econometrics literature; see Timmermann (2006) for a survey of forecast combination.

of the LASSO and ENet predictions, the second is the average of the predictions generated by the three ANNs, and the last is the average of all five predictions (LASSO, ENet, NN1, NN2, and NN3). The ensemble approach works quite well, typically producing a reduction in RMSE vis-à-vis the AR benchmark that is nearly as large as the best individual model (which we cannot know ex ante).

The term structure of the information flow reveals a pronounced increase in predictive accuracy as more timely daily GT data are incorporated into the sequence of now- and backcasts. For the first nowcast, made ten days before the UI release, the GT terms improve the RMSE vis-à-vis the AR benchmark by approximately 20% (15%) on average for non-seasonally (seasonally) adjusted UI. For the backcast made three days before the UI release—which incorporates an additional seven days of GT data—the improvement in RMSE is around 50% on average for both non-seasonally and seasonally adjusted UI. The best overall performance obtains when there is full overlap between the seven days of GT terms and the UI week, corresponding to the backcast made three days before the UI release.

To look inside the "black box" of the fitted ANNs, we compute variable-importance measures (Greenwell, Boehmke, and McCarthy 2018) and partial-dependence plots (PDPs, Friedman 2001) for the individual predictors. The variable-importance measures allow us to see which predictors are the most relevant in the fitted ANNs, as well as the linear models fitted via the LASSO and ENet. Prior to the COVID-19 crisis, the lag of UI is the most important predictor in all of the fitted models. With the advent of the crisis, the situation changes markedly, as GT terms related to the application process for UI benefits become quite important. In fact, when we include data from the COVID-19 crisis, the GT terms dominate in the fitted models used to generate the backcast three days before the UI release, while the lag of UI becomes irrelevant. The PDPs allow us to analyze the strength of the nonlinearities in the fitted ANNs. The results indicate that the nonlinearities in the fitted ANNs are relatively weak, so that the fitted linear and nonlinear predictive models are fairly "close" to one another. This helps to explain why the now- and backcasts based on the ANNs do not dominate those based on a linear specification.

We contribute to an emerging literature that uses internet search-volume data to predict labor market variables. For example, D'Amuri and Marcucci (2017) show that GT search volume for terms including *jobs* improves predictions of the US unemployment rate, while Niesert et al. (2020) find that a broad array of GT terms are useful for predicting unemployment rates in a collection of developed countries. Borup and Schütte (forthcoming) use a large number of GT terms and machine-learning tools to improve predictions of US employment growth. In contrast to these studies, which predict variables available at the monthly frequency, our target is weekly UI—which has become perhaps the most closely watched variable since the advent of the COVID-19 crisis—and we use a mixed-frequency approach. A few recent studies use GT terms to predict UI during the crisis (Larson and Sinclair 2020; Aaronson et al. 2020). Unlike the present paper, these studies consider only a small number of GT terms, and they do not utilize machine learning and a mixed-frequency framework.[4]

The rest of the paper is organized as follows. Section 2 describes the data, while Section 3 explains the information flow for the sequence of now- and backcasts. Section 4 specifies the predictive models and outlines their estimation. Section 5 reports results for the out-of-sample exercise. Section 6 interprets the fitted predictive models via variable-importance measures and PDPs. Section 7 concludes.

2 Data

This section describes the data. The data span the first week of 2008 to the last week of July of 2020.

[4]Larson and Sinclair (2020) use a small number of GT terms in panel regressions to nowcast UI across US states; in contrast to our study, nowcasts based on the GT terms fail to outperform those based on an AR benchmark. Aaronson et al. (2020) use an event study design based on the sensitivity of UI to hurricanes to predict UI for the last two weeks of March 2020 using GT terms, while we analyze UI predictions over a period starting well before the COVID-19 crisis. Choi and Varian (2012) use a small number of GT terms to predict UI through mid 2011.

2.1 Unemployment Insurance Initial Claims

Our target variable is UI for the United States. UI is available at the weekly frequency, corresponding to initial claims for Sunday through Saturday of week t (Week_t). Each Thursday morning at 8:30 EST, the Department of Labor releases the UI figure for the previous week. We take this publication lag into account when computing our predictions.[5] As detailed in Section 3, we are careful in tracking the information flow, so that we only use information available at the time of prediction formation.

The choice between targeting non-seasonally adjusted (NSA) or seasonally adjusted (SA) data has been the subject of recent debate during the COVID-19 crisis (e.g., Rinz 2020). The issue is whether the conventional multiplicative seasonal adjustment process overstates the actual seasonality in the data during the COVID-19 crisis, when UI reached historically unprecedented levels. To address this issue, we generate predictions for both the NSA and SA cases.

2.2 Google Trends

Daily search-volume data are obtained from GT, which provides an index of the proportion of queries for a specific search term within a geographical area. The index is released with an approximately 36-hour delay. This delay is the result of Google filtering irregular search activity, such as automated searches or queries that may be associated with attempts to spam search; see "FAQ about Google Trends data."[6]

We construct a high-dimensional set of predictors based on daily GT terms. Starting with the source term *unemployment*, we use Google Keyword Planner to obtain the following top fifteen keywords associated with this term: (1) *unemployment*, (2) *unemployment benefits*, (3) *unemployment office*, (4) *unemployment insurance*, (5) *file for unemployment*, (6) *apply*

[5] Compared to other macroeconomic variables (e.g., gross domestic product and consumption), UI data are subject to relatively minor revisions, as UI is based on government administrative data (rather than surveys). UI is typically revised only once during the following week.

[6] Available at `https://support.google.com/trends/answer/4365533?hl=en`

for unemployment, (7) *unemployment claim*, (8) *how to file for unemployment*, (9) *ui online*, (10) *unemployment application*, (11) *unemployment weekly claim*, (12) *unemployment compensation*, (13) *unemployment number*, (14) *unemployment online*, (15) *employment insurance*. These "primitive terms" appear quite plausible, as they are associated with the actions of a person who becomes unemployed. Our out-of-sample period begins in 2019 in Section 5, so that, to avoid look-ahead bias, the set of primitive terms is based on GT data through the end of 2018.

We expand each of the primitive terms via a GT feature that provides a list of 25 related terms, again based on GT data through the end of 2018. We use the *top* category of related terms (instead of the *rising* category). This step adds terms that are specific to individual US states (e.g., *ny unemployment benefits*, *unemployment benefits california*); semantically related to the primitive terms (e.g., *how to apply for unemployment benefits*, *unemployment phone number*, *filing unemployment online*, *state unemployment office*); closely related to unemployment, such as health care coverage and tax policies (e.g., *unemployment health insurance*, *unemployment insurance taxes*); and narrowly defined (e.g., *edd online*, which refers to the Employment Development Department through which unemployment insurance benefits can be applied for in California). After excluding duplicates, this creates a total of 270 keywords.

After removing low-volume queries (defined as series with less than 95% non-zero values), we have 103 unique terms at the daily frequency. In the context of predicting employment growth, Borup and Schütte (forthcoming) find that minor variations in the wording of queries (like adding or removing an *s* for a plural or singular version of a word) can have a notable influence on their predictive power. We cannot know ex ante which specific terms or variations are the most relevant for predicting UI, so that we include a large number of related terms and rely on supervised machine learning to place greater weight on those that are deemed the most relevant. Terms that are specific to individual US states capture idiosyncrasies for each state, which is useful if, say, New York is suddenly the main driver of unemployment claims.

GT only allows for the downloading of daily data in blocks that do not cover the full sample period, so that we concatenate data from each download to construct complete time series. The downloaded data for each GT term are scaled to have a value of 100 for the day with the highest volume. We thus need to adjust the levels of each downloaded block of data to chain together series that are comparable over time. To accomplish this, we download seven-month blocks of data, with a month of overlap. For each GT term, we compute the average daily value for the current and preceding blocks for the overlapping month. We then use the ratio of the two averages to adjust the levels for all preceding blocks.[7]

Each block of downloaded GT data covering a particular period is based on a randomized sample (about 1%) of total search queries during the period. The values for the block corresponding to the period thus change according to the time and IP address of the request to download the data. To reduce sampling error, we make ten requests for a particular period and take the average of the values over the ten downloaded blocks.

Finally, we seasonally adjust each of the GT terms using the popular STL filtering procedure (Cleveland et al. 1990). To avoid look-ahead bias, we seasonally adjust the GT terms using data available at the time of prediction formation.

Figure 1 depicts seasonally adjusted UI, along with two selected GT terms (*file for unemployment* and *unemployment office*), starting from the first week of 2020 and extending through the last week of July of 2020.[8] The time stamp on the horizontal axis indicates the UI release. UI exhibits a dramatic increase for the March 26 release, corresponding to the week ending March 21, followed by another sharp increase in the next week, leading to an historical high of approximately 6.9 million for UI for the week ending March 28. UI then

[7]A simple example illustrates the basic idea. Suppose that the first set of downloaded data for an arbitrary series is 90 and 99 for periods 1 and 2, respectively; the next set of downloaded is 85 and 76 for periods 2 and 3, respectively. We take the ratio of the values for period 2, $85/99 = 0.86$; we then use the ratio to adjust the period 1 value to $90 \times 0.86 = 77.27$, which gives us a comparable series of 77.27, 85, and 76 for periods 1, 2, and 3, respectively.

[8]These are two of the most important terms for predicting UI in Section 6.

decreases gradually, although it remains quite elevated from an historical perspective though July of 2020.

The two GT terms in Figure 1 appear to track UI well. Specifically, the terms start to increase markedly in the weeks around the sharp increase in UI, and they follow the subsequent downward trajectory fairly closely. Figure 1 suggests that GT terms are relevant for predicting UI. This is economically intuitive, as individuals are likely to search for information about filing for unemployment benefits when they become (or anticipate becoming) unemployed. Such searches leave a footprint in the search volume of relevant queries, which we harness to predict UI.

3 Information Flow

Table 1 depicts the flow of information for generating our sequence of predictions. In terms of notation, we denote the days comprising Week_t by Sunday_t, Monday_t, ... , Saturday_t. The Department of Labor releases the UI figure for Week_{t-1} (UI_{t-1}) on Thursday_t. Since GT data are released with an approximately 36-hour delay, search-volume data for queries for, say, Saturday_t are available on Monday_{t+1}. When generating each prediction, we use the seven most recently available daily observations for each of the 103 GT terms.

We begin with a prediction of UI_t formed on Monday_t, which corresponds to a nowcast of UI_t. After accounting for the 36-hour reporting lag, the seven most recently available daily observations for the GT terms cover Sunday_{t-1} through Saturday_{t-1}. We compute the nowcast by first using historical data available at the time of prediction formation to estimate one of the predictive models described in Section 4, which relates UI for a given week to GT terms for the seven days in the previous week, as well as the second lag of UI. The UI lag accounts for the strong autocorrelation in UI. We use the second lag, because as indicated in the last column of Table 1, the most recent UI observation available for computing the nowcast of UI_t is for Week_{t-2} (due to the reporting lag for UI). We then plug the values for the GT terms for Sunday_{t-1} through Saturday_{t-1} and most recent UI observation (UI_{t-2}) into the fitted model to generate the nowcast of UI_t.

Next is a prediction of UI_t formed on Tuesday_t, which again corresponds to a nowcast. Because an additional day of GT data is available, this nowcast is based on terms for Monday_{t-1} to Sunday_t, so that there is now a one-day overlap between the GT terms and UI_t; see the fourth column of Table 1. To compute a nowcast, we first fit a predictive model relating UI in a given week to GT terms for Sunday of that week and Monday through Saturday of the previous week (as well as the second lag of UI). We then plug the values for the GT terms for Monday_{t-1} to Sunday_t (and UI_{t-2}) into the fitted model. We proceed analogously to compute the nowcast of UI_t formed on Wednesday_t, which is characterized by a two-day overlap between the GT terms and UI_t.

The next three nowcasts in Table 1 are formed on Thursday_t, Friday_t, and Saturday_t. In addition to incorporating GT data through Tuesday_t, Wednesday_t, and Thursday_t, respectively, the latest available UI release permits us to use UI for Week_{t-1}. We thus use the first (instead of the second) lag of UI in the predictive model. Observe that as we move from Thursday_t to Saturday_t when forming the nowcasts, we go from a three- to a five-day overlap between the available GT terms and UI_t. Otherwise, we compute the nowcasts in the same manner as the first three nowcasts in Table 1.

The remaining predictions of UI_t, formed on Sunday_{t+1} through Wednesday_{t+1}, constitute backcasts. The backcast formed on Monday_{t+1} employs the maximal overlap between the available GT terms and UI_t. The backcasts formed on Tuesday_{t+1} and Wednesday_{t+1} use GT data for one (two) day(s) from Week_{t+1} and six (five) days from Week_t.

The sequence of predictions in Table 1 allows us to investigate the term structure of the information flow with respect to predicting UI. As we proceed from the nowcast formed on Monday_t to the backcast formed on Monday_{t+1}, the degree of overlap between the days used to predict UI_t increases. For the final two backcasts in Table 1, we include GT terms from the first one or two days of Week_{t+1} when predicting UI_t. We are interested in seeing how the availability of more recent daily GT data affects the accuracy of the now- and backcasts.

4 Predictive Models

The general form of a predictive model is given by

$$\text{UI}_t = f^{(j)}\Big(\text{UI}_{t-1}, \boldsymbol{g}_t^{(j)}; \boldsymbol{\theta}^{(j)}\Big), \tag{4.1}$$

where $\boldsymbol{\theta}^{(j)}$ is a vector of model parameters specific to $f^{(j)}$,

$$\underbrace{\boldsymbol{g}_t^{(j)}}_{7K\times 1} = \begin{bmatrix} \boldsymbol{g}'_{t-j/7} & \boldsymbol{g}'_{t-(j+1)/7} & \cdots & \boldsymbol{g}'_{t-(j+6)/7} \end{bmatrix}', \tag{4.2}$$

$\boldsymbol{g}_{t-i/7}$ for $i = 0, \ldots, 6$ is a $K \times 1$ vector of GT terms for the $(7-i)$th day of Week_t, and $K = 103$.[9] The fifth column of Table 1 provides the value of j for each of the now- and backcasts; the data overlap in the fourth column is given by $7-|j|$. Based on data availability for UI, we use UI_{t-2} in lieu of UI_{t-1} in Equation (4.1) for the first three nowcasts in Table 1.[10]

We begin with a linear specification for the predictive model:

$$\text{UI}_t = \beta_0^{(j)} + \beta_{\text{AR}}^{(j)}\text{UI}_{t-1} + \boldsymbol{\beta}_g^{(j)\prime}\boldsymbol{g}_t^{(j)} + \varepsilon_t^{(j)}, \tag{4.3}$$

where $\boldsymbol{\beta}_g^{(j)}$ is a $7K \times 1$ vector of slope coefficients for the daily GT terms and $\varepsilon_t^{(j)}$ is zero-mean error term. In terms of Equation (4.1), the vector of model parameters is given by

$$\boldsymbol{\theta}^{(j)} = \begin{bmatrix} \beta_0^{(j)} & \beta_{\text{AR}}^{(j)} & \boldsymbol{\beta}_g^{(j)\prime} \end{bmatrix}'. \tag{4.4}$$

[9] The vector of GT terms for each day of Week_t is as follows: Sunday, $\boldsymbol{g}_{t-6/7}$; Monday, $\boldsymbol{g}_{t-5/7}$; Tuesday, $\boldsymbol{g}_{t-4/7}$; Wednesday, $\boldsymbol{g}_{t-3/7}$; Thursday, $\boldsymbol{g}_{5-2/7}$; Friday, $\boldsymbol{g}_{t-1/7}$; Saturday, $\boldsymbol{g}_t$.

[10] Including additional lags for UI in Equation (4.1) has little effect on the results, so that a single lag appears sufficient for capturing the autocorrelation in UI.

There are $7 \times 103 + 1 = 722$ regressors in Equation (4.3). In our high-dimensional setting, we use the LASSO and ENet from machine learning to estimate $\boldsymbol{\theta}^{(j)}$ in Equation (4.3) when generating the now- and backcasts.[11]

Equation (4.3) can be viewed as a U-MIDAS model (Foroni, Marcellino, and Schumacher 2015), as it allows each of the higher-frequency predictors in $\boldsymbol{g}_t^{(j)}$ to have its own coefficient. A restricted MIDAS specification imposes a lag-polynomial structure on the daily observations. We use a U-MIDAS approach for two reasons. First, the daily observations only naturally align with the calendar week when $j = 0$ (see Table 1). Second, we employ machine-learning methods that allow us to flexibly estimate the weights—rather than imposing a lag-polynomial structure ex ante—while guarding against overfitting.[12]

4.1 LASSO and Elastic Net

The LASSO (Tibshirani 1996) is a machine-learning device based on penalized regression. It alleviates overfitting by augmenting the objective function for estimating $\boldsymbol{\theta}^{(j)}$ in Equation (4.3) with an ℓ_1 penalty term:

$$\underset{\boldsymbol{\theta}^{(j)} \in \mathbb{R}^{7K+2}}{\arg\min} \; \frac{1}{2\mathcal{T}} \left\{ \sum_{t=1}^{\mathcal{T}} \left[\mathrm{UI}_t - \left(\beta_0^{(j)} + \beta_{\mathrm{AR}}^{(j)} \mathrm{UI}_{t-1} + \boldsymbol{\beta}_g^{(j)\prime} \boldsymbol{g}_t^{(j)} \right) \right] \right\}^2 + \lambda \|\boldsymbol{\beta}^{(j)}\|_1, \tag{4.5}$$

where

$$\boldsymbol{\beta}^{(j)} = \begin{bmatrix} \beta_{\mathrm{AR}}^{(j)} & \boldsymbol{\beta}_g^{(j)\prime} \end{bmatrix}', \tag{4.6}$$

$\mathcal{T}$ is the number of weekly UI observations available at the time of prediction formation, $\|\cdot\|_1$ is the ℓ_1 norm, and $\lambda \geq 0$ is a regularization parameter that controls the degree of shrinkage. Unlike the ℓ_2 penalty in ridge regression (Hoerl and Kennard 1970), the ℓ_1 penalty

[11] When we compute out-of-sample now- and backcasts starting in the first week of 2019, the number of weekly UI observations available for fitting the predictive model is always less than 722, so that the conventional ordinary least squares estimator fails.

[12] Foroni, Marcellino, and Schumacher (2015) find that a "small" difference in sampling frequency between the higher- and lower-frequency variables (as in our application) favors the U-MIDAS approach.

in Equation (4.5) permits shrinkage to zero (for sufficiently large λ), so that the LASSO performs variable selection.

Although the LASSO is effective at selecting relevant predictors in certain environments (e.g., Zhang and Huang 2008; Bickel, Ritov, and Tsybakov 2009; Meinshausen and Yu 2009), it tends to arbitrarily select one predictor from a group of highly correlated predictors. The ENet (Zou and Hastie 2005) is a refinement of the LASSO that mitigates this tendency by including both ℓ_1 (LASSO) and ℓ_2 (ridge) components in the penalty term for the objective function:

$$\underset{\boldsymbol{\theta}^{(j)} \in \mathbb{R}^{7K+2}}{\arg\min} \ \frac{1}{2\mathcal{T}} \left\{ \sum_{t=1}^{\mathcal{T}} \left[\mathrm{UI}_t - \left(\beta_0^{(j)} + \beta_{\mathrm{AR}}^{(j)} \mathrm{UI}_{t-1} + \boldsymbol{\beta}_g^{(j)\prime} \boldsymbol{g}_t^{(j)} \right) \right] \right\}^2 + \lambda P_\alpha\left(\boldsymbol{\beta}^{(j)}\right), \tag{4.7}$$

where

$$P_\alpha\left(\boldsymbol{\beta}^{(j)}\right) = 0.5(1-\alpha)\|\boldsymbol{\beta}^{(j)}\|_2^2 + \alpha\|\boldsymbol{\beta}^{(j)}\|_1, \tag{4.8}$$

$\|\cdot\|_2$ is the ℓ_2 norm, and $0 \le \alpha \le 1$ is a blending parameter for the ℓ_1 and ℓ_2 components of the penalty term. When $\alpha = 1$, $P_\alpha = \|\boldsymbol{\beta}^{(j)}\|_1$ in Equation (4.8), so that the ENet reduces to the LASSO. We follow the recommendation of Hastie and Qian (2016) and set $\alpha = 0.5$.[13]

After estimating $\boldsymbol{\theta}^{(j)}$ in Equation (4.3) via the LASSO or ENet using data available at the time of prediction formation, we plug the most recently available UI observation and seven most recently available daily observations for each of the 103 GT terms into the fitted model to generate a given now- or backcast in Table 1.

4.2 Artificial Neural Networks

We consider *feedforward* ANNs, the most well-known type of neural networks. ANNs can approximate complex nonlinear predictive relationships and have proven useful for prediction in numerous domains. An ANN architecture is comprised of multiple layers. The first, the

[13] To better guard against overfitting, we tune the regularization parameter, λ, for the LASSO and ENet in Equations (4.5) and (4.7), respectively, via the extended regularization information criterion (Hui, Warton, and Foster 2015), which is a refinement of the Bayesian information criterion (Schwarz 1978).

input layer, is the set of predictors, which we denote by $x_1, \ldots, x_{P_0}$. One or more *hidden* layers follow. Each hidden layer l contains P_l neurons, each of which takes signals from the neurons in the previous hidden layer to generate a subsequent signal:

$$h_m^{(l)} = g\left(w_{m,0}^{(l)} + \sum_{j=1}^{P_{l-1}} w_{m,j}^{(l)} h_j^{(l-1)}\right) \text{ for } m = 1, \ldots, P_l;\ l = 1, \ldots, L, \tag{4.9}$$

where $h_m^{(l)}$ is the signal corresponding to the mth neuron in the lth hidden layer;[14] $w_{m,0}^{(l)}, w_{m,1}^{(l)}, \ldots, w_{m,P_{l-1}}^{(l)}$ are weights; and $g(\cdot)$ is an activation function. The final layer is the *output* layer, which translates the signals from the last hidden layer into a prediction:

$$\hat{y} = w_0^{(L+1)} + \sum_{j=1}^{P_L} w_j^{(L+1)} h_j^{(L)}, \tag{4.10}$$

where $\hat{y}$ denotes the prediction of the target variable. For the activation function, we use the popular rectified linear unit (ReLU) function:

$$g(x) = \begin{cases} 0 & \text{if } x < 0, \\ x & \text{otherwise.} \end{cases} \tag{4.11}$$

In response to a sufficiently strong signal, Equation (4.11) activates a neuronal connection and relays the signal forward through the network.

Considerable judgment is involved in specifying an ANN architecture. A single hidden layer with sufficient nodes is theoretically sufficient for approximately any smooth function (under a reasonable set of assumptions). Nevertheless, ANNs with multiple hidden layers are often used in practice. We consider three ANNs with one, two, and three hidden layers (NN1, NN2, and NN3, respectively). Following the the popular "square-root" strategy, we include 26 neurons in the single hidden layer for NN1. Based on the "pyramid" strategy

[14]For the first hidden layer, $h_j^{(0)} = x_j$ for $j = 1, \ldots, P_0$.

(Masters 1993), NN2 (NN3) contains 78 and eight (139, 27, and five) neurons in its first and second (first, second, and third) hidden layers, respectively.

Fitting (or training) an ANN requires estimating the weights. We fit the ANNs by minimizing an objective function based on mean squared error for the training sample, which, to guard against overfitting, we augment with an ℓ_1 penalty term. Fitting an ANN is computationally demanding, and we use the recently developed Adam SGD algorithm (Kingma and Ba 2015).[15]

4.3 Ensembles

We also consider various ensemble predictions, which are popular in machine learning. In recognition of model uncertainty, instead of relying on a prediction from a single model, we take an average of the predictions generated by multiple models. For each of the now- and backcasts, we construct three ensemble forecasts. The first (Ensemble-Linear) is an average of the predictions for the linear models fitted via the LASSO and ENet. The second (Ensemble-ANN) is an average of the predictions for the fitted NN1, NN2, and NN3 models. The final ensemble (Ensemble-All) is an average of the predictions for the linear models fitted via the LASSO and ENet and the fitted NN1, NN2, and NN3 models.

5 Predictive Accuracy

We generate simulated now- and backcasts for UI for the first of week of 2019 through the last week of July of 2020. Simulating the situation of a forecaster in real time, we proceed as follows using a rolling-window approach. We first use UI and GT data available for the

[15] We fit the ANNs via the Adam algorithm in `Python` using the `keras` package. To implement the algorithm, we need to specify a handful of hyperparameters. We set the number of epochs to 700 and use the `keras` default of 32 for the batch size. Considering a grid of potential values, we select the regularization parameter via three-fold cross validation for randomly selected (and non-overlapping) validation samples with a length of 15% of the estimation sample. To reduce the influence of the starting values for the random-number generator in the SGD algorithm, we compute an ensemble prediction by fitting a given model ten different times with a different seed each time and taking the average of the ten predictions.

first week of 2008 through the last week of 2018 to fit the predictive models. We then plug the most recent UI value and relevant GT values into the fitted models to generate the now- and backcasts for UI for the first week of 2019. Next, we refit the predictive models using UI and GT data available for the second week of 2008 through the first week of 2019 and plug the most recent UI and relevant GT values into the fitted models to compute the UI now- and backcasts for the second week of 2019. We continue in this manner through the end of the out-of-sample period. We reiterate that the simulated now- and backcasts only use information available at the time of prediction formation, as described in Section 3. An AR model based on the first or second lag of UI is used to generate the benchmark now- and backcasts, where we also fit the AR models using a rolling-window approach.[16] An AR model is a standard benchmark in the economic forecasting literature.

Panel A (B) of Table 2 reports results for non-seasonally (seasonally) adjusted UI.[17] The second column reports the RMSE for the AR benchmark for each of the now- and backcasts. For the nowcasts formed on Monday_t through Wednesday_t, the RMSE for the AR benchmark is approximately 1.01 million (1.13 million) for non-seasonally (seasonally) adjusted UI. Beginning with the nowcast on Thursday_t, the AR model is based on the first (instead of the second) lag of UI, as the UI_{t-1} figure becomes available on Thursday_t. This leads to a substantial reduction in RMSE to 611,730 (761,679) for the non-seasonally (seasonally) adjusted case.[18]

The third through tenth columns of Table 2 report the RMSE ratio for the machine-learning method in the column heading vis-à-vis the AR benchmark. The ratios are all below one, so that the now- and backcasts based on machine learning—which incorporate the information in the daily GT terms—always outperform the AR benchmark in terms

[16]Based on the information flow (see Section 3), the AR benchmark model is given by $\text{UI}_t = \alpha_0 + \alpha_2 \text{UI}_{t-2} + \varepsilon_t$ for the nowcasts formed on Monday_t through Wendesday_t; it is given by $\text{UI}_t = \alpha_0 + \alpha_1 \text{UI}_{t-1} + \varepsilon_t$ for the now- and backcasts formed on Thursday_t through Wendesday_{t+1}. Because they are univariate regressions, we estimate the AR models via ordinary least squares.

[17]We assess the accuracy of the now- and backcasts using revised UI data.

[18]Due to the significant autocorrelation in UI, the AR is a relevant benchmark. Indeed, the AR benchmark performs substantially better than a naïve model that ignores the autocorrelation in UI and simply uses the rolling mean to predict UI; the reduction in RMSE is up to 50%.

of RMSE. The improvements in predictive accuracy are sizable. For the initial nowcast formed on Monday$_t$, which is formed ten days before the UI release, the reductions in RMSE relative to the AR benchmark range from approximately 15% to 20%. The RMSE ratios decrease nearly monotonically as we move from the nowcast formed on Monday$_t$ to the backcast formed on Monday$_{t+1}$, which uses the largest data overlap (seven days) between the daily GT terms and weekly UI. For seasonally adjusted UI, there are often additional improvements in RMSE for the backcasts formed on Tuesday$_{t+1}$ and Wednesday$_{t+1}$. Overall, the predictions generally become more accurate in Table 2 as we include additional days of GT data. The reduction in MSFE vis-à-vis the AR benchmark reaches as high as 53.3% (for the LASSO backcast formed on Monday$_{t+1}$ for non-seasonally adjusted UI).

The different machine-learning methods in Table 2 perform similarly. The LASSO often provides the most accurate predictions, especially for seasonally adjusted UI. This suggests that a linear model is largely sufficient for capturing the information in the GT terms; we address this issue further in Section 6. The ensemble approaches also perform well overall. In particular, the Ensemble-All approach, which is an average of the LASSO, ENet, NN1, NN2, and NN3 predictions, often produces close to the lowest RMSE for the individual now- and backcasts. Because we cannot know ex ante the best method, the Ensemble-All approach provides a promising practical strategy for now- and backcasting UI.

Figures 2 and 3 provide perspective on the performance of the machine-learning methods vis-à-vis the AR benchmark over time. The figures show the cumulative difference in squared errors (CDSE) for a machine-learning method relative to the AR benchmark (Goyal and Welch 2003, 2008). Each curve allows us to conveniently analyze relative performance for any subsample by comparing the height of the curve at the beginning and end of the interval corresponding to the subsample. If curve is higher (lower) at the end of the interval, then the machine-learning method is more (less) accurate than the AR benchmark for the subsample.

Figure 2 plots CDSE curves for selected now- and backcasts and machine-learning methods for the full out-of-sample period. The figure includes results for nowcasts formed on

Tuesday$_t$ and Thursday$_t$ (which have one and three days, respectively, of data overap), as well as the backcast formed on Monday$_{t+1}$ (which has the maximum seven days of data overlap). The machine-learning methods are the LASSO and NN1. The plots in Figure 2 are dominated by sharp increases in the early months of the COVID-19 crisis. This is the time when we expect the GT terms to provide relevant information for anticipating the increases in UI associated with the crisis. With the onset of the crisis, an historically large number of people were laid off, many of whom likely became unemployed for the first time or were rarely unemployed previously. They are less familiar with the application process for unemployment benefits, so that their Google search histories leave a footprint as they gather information on applying for unemployment benefits. Our machine-learning, mixed-frequency approach harnesses the information in daily GT terms to substantially improve prediction near the start of the COVID-19 crisis.

The CDSE curves in Figure 2 are dominated by the advent of the COVID-19 crisis, making it difficult to assess the out-of-sample gains in the "normal" period before the crisis. To examine the pre-crisis period more closely, Figure 3 depicts CDSE curves for the first week of 2019 through the second week of March of 2020. The curves are predominantly positively sloped in most cases, so that the LASSO and NN1 outperform the AR benchmark on a reasonably consistent basis over time outside of the COVID-19 crisis. Taken together, Figures 2 and 3 indicate that the information in GT terms provide moderate gains during relatively quiescent times and striking gains during turbulent times.

6 Interpreting the Fitted Models

While the LASSO and ENet facilitate the interpretation of fitted linear models by performing variable selection, fitted ANNs are black boxes that are difficult to interpret. In this section, we use PDPs (Friedman 2001) and variable-importance measures (Greenwell, Boehmke, and

McCarthy 2018) to peer into the black box of the fitted ANNs and compare them to the fitted linear models.

6.1 Partial-Dependence Plots and Variable Importance

For ease of exposition, we gather the predictors in the 722×1 vector,

$$\boldsymbol{x}_t^{(j)} = \left[\begin{array}{cc} \mathrm{UI}_{t-1} & \boldsymbol{g}_t^{(j)} \end{array} \right]'. \tag{6.1}$$

Furthermore, we denote the $722 \times \mathcal{T}$ data matrix by

$$\boldsymbol{X}_{\mathcal{T}}^{(j)} = \left[\begin{array}{ccc} \boldsymbol{x}_1^{(j)} & \cdots & \boldsymbol{x}_{\mathcal{T}}^{(j)} \end{array} \right]'. \tag{6.2}$$

Suppose that we are interested in analyzing the marginal effect of a given predictor, $x_s^{(j)}$, on the expected value of UI_t for a fitted model. Letting $\boldsymbol{x}_{C(s)}^{(j)} = \boldsymbol{x}^{(j)} \setminus x_s^{(j)}$, the partial dependence for $x_s^{(j)}$ is defined as

$$\begin{aligned} \mathrm{PD}\big(x_s^{(j)}\big) &= \mathbb{E}_{\boldsymbol{x}_{C(s)}^{(j)}}\left[\hat{f}^{(j)}\left(x_s^{(j)}, \boldsymbol{x}_{C(s)}^{(j)}\right)\right] \\ &= \int_{\boldsymbol{x}_{C(s)}^{(j)}} \hat{f}^{(j)}\left(x_s^{(j)}, \boldsymbol{x}_{C(s)}^{(j)}\right) p_{C(s)^{(j)}}\left(\boldsymbol{x}_{C(s)}^{(j)}\right) d\boldsymbol{x}_{C(s)^{(j)}}, \end{aligned} \tag{6.3}$$

where

$$p_{C(s)^{(j)}}\left(\boldsymbol{x}_{C(s)}^{(j)}\right) = \int_{x_s^{(j)}} p\big(\boldsymbol{x}^{(j)}\big)\, dx_s^{(j)}, \tag{6.4}$$

$p\big(\boldsymbol{x}^{(j)}\big)$ is the joint probability density for $\boldsymbol{x}^{(j)}$, and $\hat{f}\big(\boldsymbol{x}^{(j)}\big)$ is the prediction function for the fitted model. Equation (6.3) gives the marginal relationship between the expected value of the target and $x_s^{(j)}$. It is typically estimated via Monte Carlo integration using the training sample, $\boldsymbol{X}_{\mathcal{T}}^{(j)}$:

$$\widehat{\mathrm{PD}}\big(x_s^{(j)}\big) = \frac{1}{\mathcal{T}}\sum_{t=1}^{\mathcal{T}} \hat{f}^{(j)}\Big(x_s^{(j)}, \boldsymbol{x}_{C(s),t}^{(j)}\Big), \tag{6.5}$$

where Equation (6.5) is evaluated at the training sample values of $x_s^{(j)}$ (i.e., $x_{s,t}^{(j)}$ for $t = 1, \ldots, \mathcal{T}$) or a set of quantiles.

Of course, the PDP for a fitted linear model will have a constant slope, while it will be a horizontal line for a predictor that is not selected by the LASSO or ENet. By comparing the PDPs for the fitted ANNs to those for the fitted linear models, we can gauge the relative importance of nonlinearities in the former.

Greenwell, Boehmke, and McCarthy (2018) develop a variable-importance metric based on Equation (6.5):

$$\hat{\mathcal{I}}\big(x_s^{(j)}\big) = \left\{\frac{1}{\mathcal{T}-1}\sum_{t=1}^{\mathcal{T}}\left[\widehat{\mathrm{PD}}\Big(x_{s,t}^{(j)}\Big) - \frac{1}{\mathcal{T}}\sum_{t=1}^{\mathcal{T}}\widehat{\mathrm{PD}}\Big(x_{s,t}^{(j)}\Big)\right]^2\right\}^{0.5}. \tag{6.6}$$

Equation (6.6) measures the importance of a predictor via the variation (i.e., standard deviation) in the PDP around its average value. For a predictor with a horizontal PDP, the expected value of the target does not vary with the predictor, so that its variable importance is zero. As the conditional expectation fluctuates more about its average value, the variable importance increases. To facilitate comparison across predictors, we scale Equation (6.6) using the sum of the individual measures:

$$\tilde{\mathcal{I}}\big(x_s^{(j)}\big) = \frac{\hat{\mathcal{I}}\Big(x_s^{(j)}\Big)}{\sum_{p=1}^{P}\hat{\mathcal{I}}\Big(x_p^{(j)}\Big)}, \tag{6.7}$$

where P is the total number of predictors, so that $\tilde{\mathcal{I}}\Big(x_s^{(j)}\Big)$ ranges from zero to one.

6.2 Importance of Google Trends Terms

Figures 4 and 5 depict variable-importance measures based on Equation (6.7) for the top 25 predictors for linear models fitted via the LASSO and fitted NN1 models, respectively (for seasonally adjusted UI). The left-hand panels of the figures correspond to models fitted with data through the end of 2019, before the start of the COVID-19 crisis; the right-hand panels are for models estimated using data through the end of July of 2020, so that the training sample includes the crisis. Note that a given GT term can appear up to seven times in the same plot, due to our mixed-frequency framework. To conserve space, we focus on fitted models for nowcasts formed on Monday_t, Tuesday_t, and Thursday_t and the backcast formed on Monday_{t+1} (with data overlaps of zero, one, three, and seven days, respectively).

For the pre-crisis sample, the lag of UI (*lag*) is the most important predictor for both the LASSO and NN1 in all cases in Figures 4 and 5. For the linear models fitted by the LASSO in Figure 4, the $\tilde{\mathcal{I}}\left(x_s^{(j)}\right)$ scores for *lag* are all above 0.75, while the GT terms play a more limited role. The GT terms are more important in the fitted NN1 in Figure 5, but *lag* still predominates, with $\tilde{\mathcal{I}}\left(x_s^{(j)}\right)$ scores ranging from around 0.1 to 0.4.

The GT terms become substantively more important in the right-hand panels of Figures 4 and 5 when we include data from the COVID-19 crisis. Although *lag* remains the most or next-to-most important predictor for the nowcasts formed on Monday_t, Tuesday_t, and Thursday_t across both models, it drops out of top 25 for the Monday_{t+1} backcast. In other words, when we use the maximum data overlap in computing the predictions, the fitted models assign essentially no importance to the autocorrelation in UI, so that the predictive signals in the GT terms dominate those in the AR component.

Table 3 provides additional information on the growing importance of GT terms in the sequence of now- and backcasts, especially for the sample that includes the COVID-19 crisis. The table reports the joint importance of the GT terms for each prediction-formation day (for seasonally adjusted UI). For linear models fitted via the LASSO for the pre-crisis sample in the third column, the GT terms grow in importance from 0.12 for the initial nowcasts to around 0.25 for the later backcasts. When the training sample includes the COVID-19 crisis (see the fourth column), the GT terms again grow in importance, but the level is markedly higher for each prediction-formation day. For the Monday$_t$ nowcast, the joint importance score for the GT terms is 0.725. It reaches 1.000 for the Saturday$_t$ nowcast through the Wednesday$_{t+1}$ backcast, so that the AR component becomes completely unimportant. A similar pattern holds for the fitted NN1 models in the last two columns, with the joint importance measures nearly always larger than the corresponding values in the third and fourth columns.

The increasing importance of the GT terms since the start of the COVID-19 crisis is also evident in Figure 6, which shows the number of predictors selected by the LASSO and ENet for rolling-window estimation of the linear predictive models underlying the Monday$_t$, Tuesday$_t$, and Thursday$_t$ nowcasts and Monday$_{t+1}$ backcast. In general, the number of selected predictors increases in the fitted linear models with the advent of the crisis. For example, for the SA case, the LASSO typically selects fewer than 20 predictors before the crisis; the number jumps to around 40 to 60 once data from the crisis are included in the estimation window. As expected, the ENet usually selects more predictors than the LASSO. Before the COVID-19 crisis, the ENet selects about 40 to 60 (50 to 70) predictors for the NSA (SA) case, while around 70 to 110 (80 to 130) are selected when data from the crisis are included.

Returning to Figures 4 and 5, for the prediction-formation days where the GT terms matter the most in (i.e., the Thursday$_t$ nowcast and, especially, the Monday$_{t+1}$ backcast), there is a tendency for both the fitted linear and NN1 models to place more weight on recently available search queries. For example, the Monday$_{t+1}$ backcast attaches the greatest importance to GT terms for Saturday and Friday (recall the two-day lag in the availability of the GT data). Similarly, the Thursday$_t$ nowcast emphasizes GT terms for Monday and Tuesday. To explore this issue further, Figure 7 shows heatmaps for the joint importance of the GT terms organized according to the day of the week. For the pre-crisis sample in the left-hand panels, there is no discernible pattern in the importance of the GT terms across the days of the week. For the sample that includes the COVID-19 crisis in the right-hand panels, a strong pattern is evident: with the exception of the Wednesday$_{t+1}$ backcast, the now- and backcasts attach relatively high importance to GT terms for the most recent day of available data. This is most evident for the Monday$_{t+1}$ backcast, which uses the maximal data overlap of seven days, where the collective importance of GT terms for Saturday is 0.69 (0.54) for the linear model fitted via the LASSO (fitted NN1 model). Figure 7 highlights the relevance of high-frequency GT data for anticipating UI during the crisis.

Looking back to Figures 4 and 5, an interesting pattern emerges in the types of GT terms that appear important across the two samples. For the fitted linear and NN1 models, the pre-COVID-19 sample is characterized by a wide variety of search queries with no clear commonalities (e.g., some geographical terms and some generic terms like *unemployment rate*, *employment*, and *workers compensation*). In contrast, for the sample that includes the COVID-19 crisis, there is an emphasis on search queries related to the application process for unemployment insurance benefits, as terms such as *how to file for unemployment*, *apply for unemployment benefits*, *unemployment application*, *unemployment office*, and variations thereof consistently appear as the most relevant predictors. These results further help to explain the usefulness of high-frequency GT data for predicting UI during turbulent times.

6.3 Nonlinearities

Finally, in order to get a sense of the strength of the nonlinearities in the fitted ANNs, we investigate PDPs for some of the most relevant predictors. We again report results for linear models fitted via the LASSO and fitted NN1 models for training samples that exclude and include the COVID-19 crisis. Figure 8 presents PDPs for the AR component (*lag*), while Figure 9 provides PDPs for *how to file for unemployment* and *unemployment office illinois*. The figures report results for the Monday$_t$, Tuesday$_t$, and Thursday$_t$ nowcasts and

Monday$_{t+1}$ backcast (for seasonally adjusted UI). The two GT terms in Figure 9 are for the most recently available day of GT data. The first GT term, *how to file for unemployment*, is included because it is the most important predictor in the fitted linear and NN1 models for the Monday$_{t+1}$ backcast during the sample that includes the crisis. We include *unemployment office illinois* because it provides an example of an important geographical search query that enters the top 25 for the Monday$_{t+1}$ backcast.[19] Note that Figures 8 and 9 use different scales for the two training samples, as average UI is much higher for the sample that includes the COVID-19 crisis.

By construction, the PDPs are linear for the linear models fitted via the LASSO. Figure 8 indicates that the predictive relationship for the AR component, *lag*, is quite close to linear for the fitted NN1 models. In other words, although the NN1 allows for nonlinear relationships, when we train the model using available data, the predictive relationship involving the AR component is essentially linear. In the bottom-right panel of Figure 8, the PDP for *lag* is horizontal for the sample that includes the COVID-19 crisis, as the LASSO does not select *lag*. The PDP for *lag* is also nearly horizontal for the fitted NN1, so that no meaningful nonlinearity is evident.

With respect to *how to file for unemployment* in the left-hand panels of Figure 9, there are only slight curvatures for some of the PDPs for the fitted NN1 models; the predictive relationships between the GT term and UI are thus essentially linear. The fitted linear and NN1 models generally indicate a positive relationship, and the effects are larger (smaller) for the latter for the Tuesday$_t$ nowcast (Thursday$_t$ nowcast and Monday$_{t+1}$ backcast). The positive relationship makes economic sense, as an increase in GT searches involving *how to file for unemployment* portends an increase in UI. With respect to *unemployment office illinois* in the right-hand panels of Figure 9, we again fail to see significant nonlinearities in the predictive relationships for the fitted NN1 models. The fitted NN1 models imply a

[19] At the end of August of 2020, Illinois had the sixth (seventh) highest number of confirmed cases (deaths) across US states (`https://www.nytimes.com/interactive/2020/us/coronavirus-us-cases.html`).

positive relationship between UI and *unemployment office illinois* for the Tuesday$_t$ nowcast and Monday$_{t+1}$ backcast, but the relationships are essentially linear.

Overall, the similar performances of the fitted linear and ANN models in terms of RMSE in Table 2, together with the lack of substantive nonlinearites for the fitted NN1 models in Figures 8 and 9, indicate that a linear structure is generally sufficient for predicting UI based on the information in a large number of daily GT terms.

7 Conclusion

We show that the information in high-dimensional daily internet search-volume data can be used to substantially improve predictions of weekly UI in anticipation of its Thursday release by the Department of Labor, especially since the advent of the COVID-19 crisis. We construct a sequence of now- and backcasts that are formed ten days to one day ahead of the UI release on Thursday of each week. To effectively utilize the information in a large number of daily GT terms related to *unemployment*, we estimate the predictive models underpinning the now- and backcasts using machine-learning techniques in a mixed-frequency framework. The mixed-frequency framework allows us to incorporate daily GT data as they become available, thereby providing us with more timely information for predicting weekly UI, while machine learning is appropriate for our high-dimensional setting. In a simulated out-of-sample exercise, now- and backcasts based on daily GT terms substantially outperform an AR benchmark in terms of RMSE. As the sequence of now- and backcasts incorporates more recent daily GT data, predictive accuracy generally improves, leading to reductions in RMSE of up to approximately 50% vis-à-vis the AR benchmark.

Variable-importance measures for the fitted predictive models reveal that the GT terms become more relevant with the advent of the COVID-19 crisis. GT terms for the most recently available day are typically the most germane, highlighting the value of our mixed-frequency approach. GT terms relating to the application process for unemployment insurance benefits also become more important during the COVID-19 crisis.

We are in the process of creating a website that will provide updated, real-time now- and backcasts of UI on a daily basis using the methods developed in this paper.[20] The website will also include historical data for the now- and backcasts.

[20] Available at `https://www.uinowcast.org/`

References

Aaronson, D., S. A. Brave, R. A. Butters, D. W. Sacks, and B. Seo (2020). Using the Eye of the Storm to Predict the Wave of Covid-19 UI Claims. Federal Reserve Bank of Chicago Working Paper No. 2020-10.

Barron, A. R. (1994). Approximation and Estimation Bounds for Artificial Neural Networks. *Machine Learning* 14:1, 115–133.

Bickel, P. J., Y. Ritov, and A. B. Tsybakov (2009). Simultaneous Analysis of Lasso and Dantzig Selector. *Annals of Statistics* 37:4, 1705–1732.

Borup, D. and E. C. M. Schütte (forthcoming). In Search of a Job: Forecasting Employment Growth Using Google Trends. *Journal of Business and Economic Statistics*.

Brave, S. A., R. A. Butters, and A. Justiniano (2019). Forecasting Economic Activity with Mixed Frequency BVARs. *International Journal of Forecasting* 35:4, 1692–1707.

Choi, H. and H. Varian (2012). Predicting the Present with Google Trends. *Economic Record* 88:s1, 2–9.

Clements, M. P. and A. B. Galvão (2008). Macroeconomic Forecasting with Mixed-Frequency Data: Forecasting Output Growth in the United States. *Journal of Business and Economic Statistics* 26:4, 546–554.

Cleveland, R. B., W. S. Cleveland, J. E. McRae, and I. Terpenning (1990). STL: A Seasonal-Trend Decomposition Procedure Based on Loess. *Journal of Official Statistics* 6:1, 3–73.

Cybenko, G. (1989). Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control, Signals, and Systems* 2:4, 303–314.

D'Amuri, F. and J. Marcucci (2017). The Predictive Power of Google Searches in Forecasting US Unemployment. *International Journal of Forecasting* 33:4, 801–816.

Diebold, F. X. and M. Shin (2019). Machine Learning for Regularized Survey Forecast Combination: Partially-Egalitarian Lasso and its Derivatives. *International Journal of Forecasting* 35:4, 1679–1691.

Foroni, C., M. Marcellino, and C. Schumacher (2015). Unrestricted Mixed Data Sampling (MIDAS): MIDAS Regressions with Unrestricted Lag Polynomials. *Journal of the Royal Statistical Society. Series A (Statistics in Society)* 178:1, 57–82.

Foroni, C. and M. Marcellino (2014). A Comparison of Mixed Frequency Approaches for Nowcasting Euro Area Macroeconomic Aggregates. *International Journal of Forecasting* 30:3, 554–568.

Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics* 29:5, 1189–1232.

Funahashi, K.-I. (1989). On the Approximate Realization of Continuous Mappings by Neural Networks. *Neural Networks* 2:3, 183–192.

Ghysels, E., P. Santa-Clara, and R. Valkanov (2005). There is a Risk-Return Trade-Off After All. *Journal of Financial Economics* 76:3, 509–548.

Goyal, A. and I. Welch (2003). Predicting the Equity Premium with Dividend Ratios. *Management Science* 49:5, 639–654.

Goyal, A. and I. Welch (2008). A Comprehensive Look at the Empirical Performance of Equity Premium Prediction. *Review of Financial Studies* 21:4, 1455–1508.

Greenwell, B. M., B. C. Boehmke, and A. J. McCarthy (2018). A Simple and Effective Model-Based Variable Importance Measure. Working Paper (arXiv:1805.04755v1).

Hastie, T. and J. Qian (2016). Glmnet Vignette. Working Paper.

Hoerl, A. E. and R. W. Kennard (1970). Ridge Regression: Applications to Nonorthogonal Problems. *Technometrics* 12:1, 69–82.

Hornik, K. (1991). Approximation Capabilities of Multilayer Feedforward Networks. *Neural Networks* 4:2, 251–257.

Hornik, K., M. Stinchcombe, and H. White (1989). Multilayer Feedforward Networks Are Universal Approximators. *Neural Networks* 2:5, 359–366.

Hui, F. K. C., D. I. Warton, and S. D. Foster (2015). Tuning Parameter Selection for the Adaptive Lasso Using ERIC. *Journal of the American Statistical Society* 110:509, 262–269.

Kingma, D. P. and J. Ba (2015). *Adam: A Method for Stochastic Optimization.* Third Annual International Conference on Learning Representations. San Diego.

Kotchoni, R., M. Leroux, and D. Stevanovic (2019). Macroeconomic Forecast Accuracy in a Data-Rich Environment. *Journal of Applied Econometrics* 34:7, 1050–1072.

Larson, W. D. and T. M. Sinclair (2020). Nowcasting Unemployment Insurance Claims in the Time of COVID-19. CAMA Working Paper No. 63/2020.

Lewis, D., K. Mertens, and J. H. Stock (2020). US Economic Activity During the Early Weeks of the SARS-Cov-2 Outbreak. NBER Working Paper No. 26954.

Masters, T. (1993). *Practical Neural Network Recipes in C++.* Boston, MA: Academic Press.

Medeiros, M. C., G. F. R. Vasconcelos, A. Veiga, and E. Zilberman (forthcoming). Forecasting Inflation in a Data-Rich Environment: The Benefits of Machine Learning Methods. *Journal of Business and Economic Statistics.*

Meinshausen, N. and B. Yu (2009). Lasso-Type Recovery of Sparse Representations for High-Dimensional Data. *Annals of Statistics* 37:1, 246–270.

Niesert, R. F., J. A. Oorschot, C. P. Veldhuisen, K. Brons, and R.-J. Lange (2020). Can Google Search Data Help Predict Macroeconomic Series? *International Journal of Forecasting* 36:3, 1163–1172.

Rinz, K. (2020). Understanding Unemployment Insurance Claims and Other Labor Market Data During the COVID-19 Pandemic. Working Paper.

Schwarz, G. (1978). Estimating the Dimension of a Model. *Annals of Statistics* 6:2, 461–464.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the LASSO. *Journal of the Royal Statistical Society. Series B (Methodological)* 58:1, 267–288.

Timmermann, A. (2006). Forecast Combinations. In *Handbook of Economic Forecasting*. Ed. by G. Elliott, C. W. J. Granger, and A. Timmermann. Vol. 1. Amsterdam: Elsevier, pp. 135–196.

Zhang, C.-H. and J. Huang (2008). The Sparsity and Bias of the Lasso Selection in High-Dimensional Linear Regression. *Annals of Statistics* 36:4, 1567–1594.

Zou, H. and T. Hastie (2005). Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 67:2, 301–320.

Table 1: Information flow for prediction

(1) Prediction formation	(2) Backcast/ nowcast	(3) Google Trends used for prediction	(4) Data overlap (days)	(5) j	(6) Latest available UI release
Monday_t	Nowcast	Sunday_{t-1} to Saturday_{t-1}	0	7	Week_{t-2}
Tuesday_t	Nowcast	Monday_{t-1} to Sunday_t	1	6	Week_{t-2}
Wednesday_t	Nowcast	Tuesday_{t-1} to Monday_t	2	5	Week_{t-2}
Thursday_t	Nowcast	Wednesday_{t-1} to Tuesday_t	3	4	Week_{t-1}
Friday_t	Nowcast	Thursday_{t-1} to Wednesday_t	4	3	Week_{t-1}
Saturday_t	Nowcast	Friday_{t-1} to Thursday_t	5	2	Week_{t-1}
Sunday_{t+1}	Backcast	Saturday_{t-1} to Friday_t	6	1	Week_{t-1}
Monday_{t+1}	Backcast	Sunday_t to Saturday_t	7	0	Week_{t-1}
Tuesday_{t+1}	Backcast	Monday_t to Sunday_{t+1}	6	-1	Week_{t-1}
Wednesday_{t+1}	Backcast	Tuesday_t to Monday_{t+1}	5	-2	Week_{t-1}

The table reports the information flow for daily Google Trends data and data releases of unemployment insurance initial claims (UI) used for now- and backcasts of UI for week t (Week_t). The first column provides the prediction-formation day, where the subscript denotes the week when the prediction is made. The second column provides the classification as a nowcast or backcast of Week_t UI. The third column provides the range of daily Google Trends terms used for prediction. The fourth column provides the number of days of overlap between the Google Trends terms in the third column and Week_t UI. The fifth column provides the value for j in Equation (4.1) The sixth column provides the latest release of UI available at the time of prediction formation.

Table 2: RMSE ratios

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
		Linear		ANN			Ensemble		
Prediction formation	AR RMSE	LASSO	ENet	NN1	NN2	NN3	Linear	ANN	All
Panel A: Non-seasonally adjusted UI									
Monday_t	1,005,527	0.784	0.790	0.842	0.812	0.879	0.840	0.787	0.805
Tuesday_t	1,005,527	0.714	0.726	0.764	0.722	0.763	0.743	0.719	0.727
Wednesday_t	1,005,527	0.647	0.648	0.668	0.656	0.642	0.652	0.646	0.642
Thursday_t	611,730	0.702	0.690	0.596	0.630	0.705	0.631	0.692	0.658
Friday_t	611,730	0.639	0.609	0.521	0.651	0.746	0.616	0.623	0.614
Saturday_t	611,730	0.541	0.548	0.522	0.690	0.559	0.562	0.542	0.545
Sunday_{t+1}	611,730	0.598	0.517	0.570	0.668	0.647	0.618	0.555	0.577
Monday_{t+1}	611,730	0.467	0.536	0.470	0.551	0.504	0.503	0.498	0.494
Tuesday_{t+1}	611,730	0.530	0.509	0.634	0.622	0.550	0.591	0.518	0.552
Wednesday_{t+1}	611,730	0.638	0.587	0.478	0.686	0.617	0.567	0.609	0.586
Panel B: Seasonally adjusted UI									
Monday_t	1,131,559	0.867	0.855	0.971	0.897	0.947	0.874	0.924	0.882
Tuesday_t	1,131,559	0.753	0.771	0.863	0.891	0.878	0.755	0.877	0.807
Wednesday_t	1,131,559	0.712	0.715	0.769	0.753	0.749	0.711	0.769	0.729
Thursday_t	761,679	0.790	0.718	0.802	0.911	0.723	0.757	0.771	0.757
Friday_t	761,679	0.676	0.653	0.682	0.981	0.684	0.666	0.806	0.725
Saturday_t	761,679	0.556	0.581	0.719	0.687	0.611	0.565	0.649	0.596
Sunday_{t+1}	761,679	0.528	0.569	0.627	0.646	0.627	0.541	0.614	0.564
Monday_{t+1}	761,679	0.488	0.535	0.540	0.619	0.640	0.499	0.584	0.536
Tuesday_{t+1}	761,679	0.483	0.516	0.621	0.577	0.687	0.492	0.556	0.519
Wednesday_{t+1}	761,679	0.500	0.534	0.552	0.553	0.615	0.505	0.552	0.526

The table reports out-of-sample results for now- and backcasts of weekly unemployment insurance initial claims (UI) formed on the day indicated in the first column, where the subscript denotes the week when the prediction is made. The second column reports the root mean squared error (RMSE) for an autoregressive (AR) benchmark model. The third through tenth columns report the RMSE ratio for the machine-learning method in the column heading vis-à-vis the AR benchmark. The third and fourth columns are for linear models fitted via the LASSO and elastic net (ENet), respectively. The fifth through seventh columns are for fitted artificial neural networks (ANNs) with one (NN1), two (NN2), and three (NN3) hidden layers, respectively. The ensemble in the eighth (ninth) column is an average of the predictions in the third and fourth (fifth through seventh) columns. The ensemble in the tenth column is an average of the predictions in the third through seventh columns. The out-of-sample period begins in the first week of 2019 and ends in the last week of July of 2020.

Table 3: Importance of GT terms

(1)	(2)	(3)	(4)	(5)	(6)
Prediction formation	Data overlap (days)	LASSO excluding COVID-19 crisis	LASSO including COVID-19 crisis	NN1 excluding COVID-19 crisis	NN1 including COVID-19 crisis
Monday_t	0	0.122	0.725	0.570	0.948
Tuesday_t	1	0.124	0.731	0.858	0.935
Wednesday_t	2	0.214	0.730	0.864	0.947
Thursday_t	3	0.121	0.758	0.859	0.968
Friday_t	4	0.123	0.859	0.825	0.943
Saturday_t	5	0.112	1.000	0.844	0.997
Sunday_{t+1}	6	0.136	1.000	0.821	1.000
Monday_{t+1}	7	0.251	1.000	0.917	1.000
Tuesday_{t+1}	6	0.243	1.000	0.922	0.999
Wednesday_{t+1}	5	0.236	1.000	0.925	1.000

The table reports joint variable-importance measures for all of the daily Google Trends terms in fitted linear models estimated via the LASSO and fitted artificial neural networks with a single hidden layer (NN1). The second column provides the number of days of overlap between the daily Google Trends terms and week-t unemployment insurance initial claims. The third and fifth (second and sixth) columns report results for estimation samples excluding (including) the COVID-19 crisis. Results pertain to seasonally adjusted unemployment insurance initial claims.

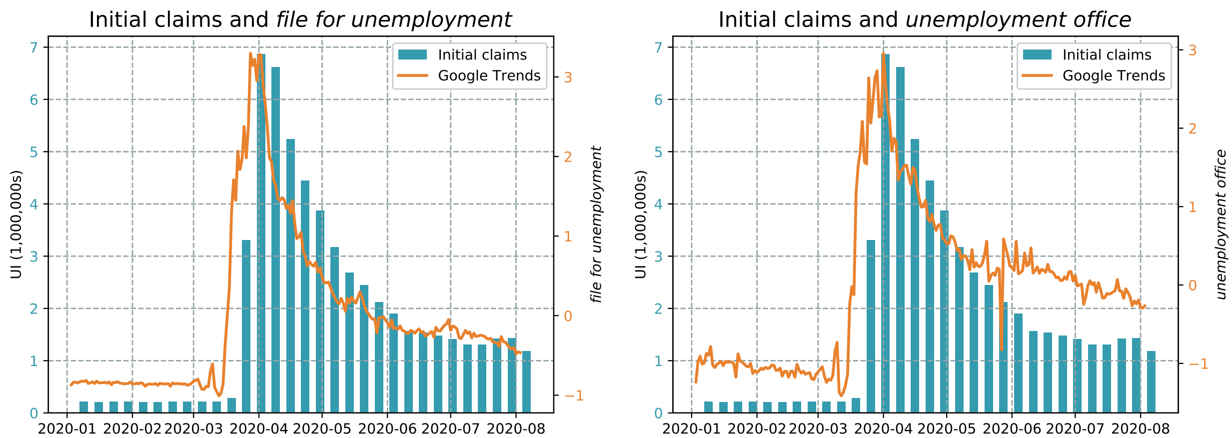

Figure 1: UI and GT terms

The figure depicts seasonally adjusted weekly unemployment insurance initial claims (UI, left-hand axis) at their release date and (standardized) daily search volume for two Google Trends terms (right-hand axis): *file for unemployment* (left panel) and *unemployment office* (right panel). The panels span the first week of January 2020 through the last week of July 2020.

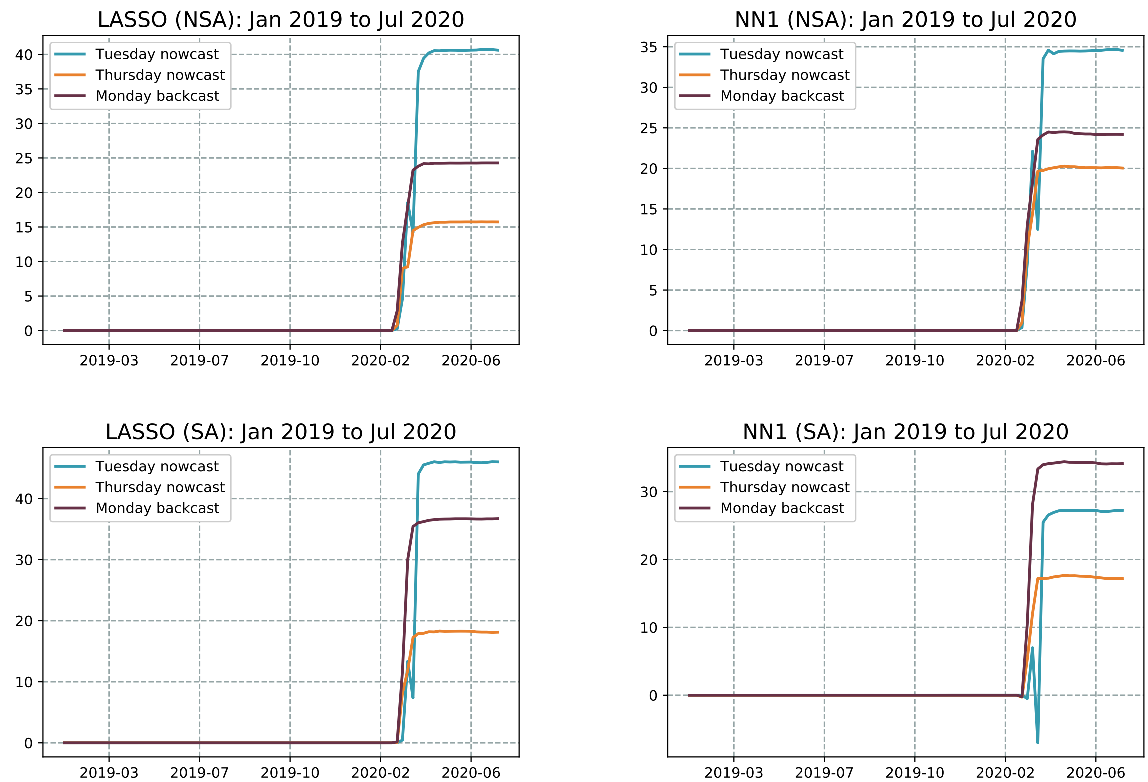

Figure 2: CDSE of selected models vis-à-vis AR benchmark

The figure depicts the cumulative difference in squared errors (CDSE) for now- and backcasts based on linear models fitted via the LASSO and fitted single-layer artificial neural networks (NN1) vis-à-vis those based on an autoregressive (AR) benchmark model. The upper and lower panels report results for non-seasonally adjusted (NSA) and seasonally adjusted (SA) weekly unemployment insurance initial claims, respectively. Each panel plots the CDSE for the Tuesday$_t$ and Thursday$_t$ nowcasts and Monday$_{t+1}$ backcast. The out-of-sample period spans the first week of 2019 through the last week of July 2020.

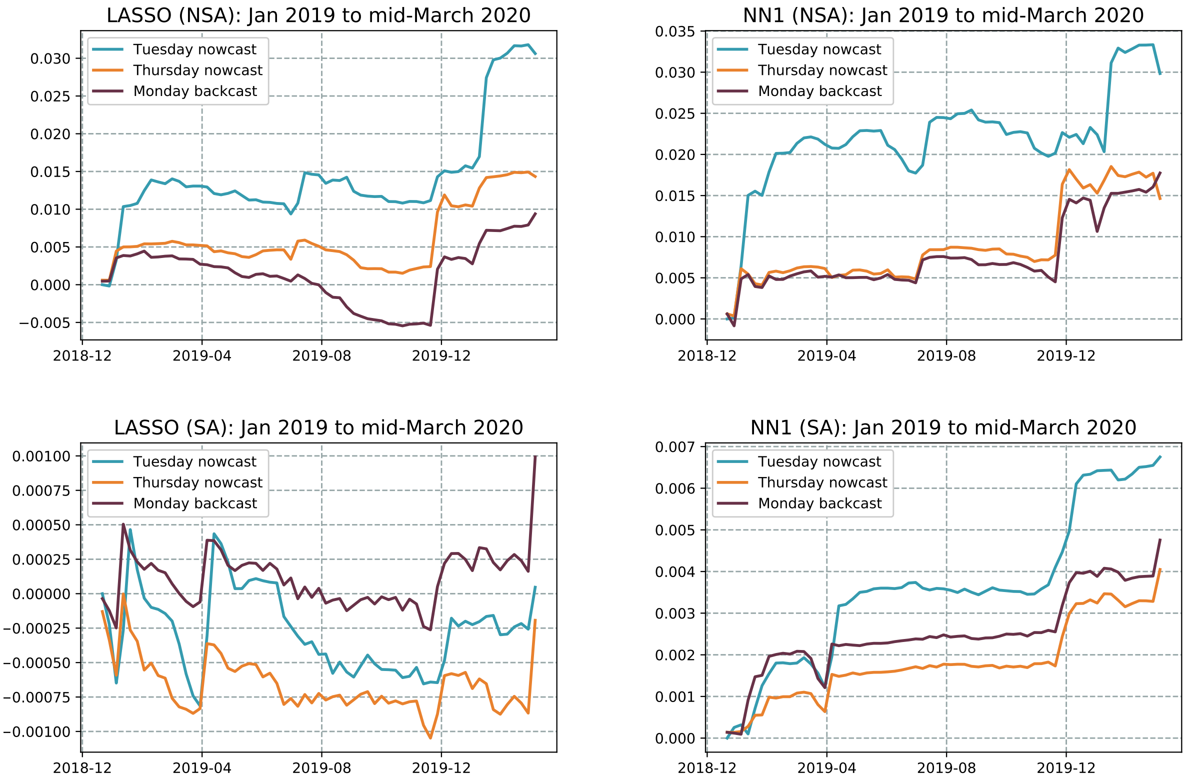

Figure 3: CDSE of selected models vis-à-vis AR benchmark, pre-crisis period

The figure depicts the cumulative difference in squared errors (CDSE) for now- and backcasts based on linear models fitted via the LASSO and fitted single-layer artificial neural networks (NN1) vis-à-vis those based on an autoregressive (AR) benchmark model. The upper and lower panels report results for non-seasonally adjusted (NSA) and seasonally adjusted (SA) weekly unemployment insurance initial claims, respectively. Each panel plots the CDSE for the Tuesday$_t$ and Thursday$_t$ nowcasts and Monday$_{t+1}$ backcast. The out-of-sample period spans the first week of 2019 through the week ending March 14, 2020.

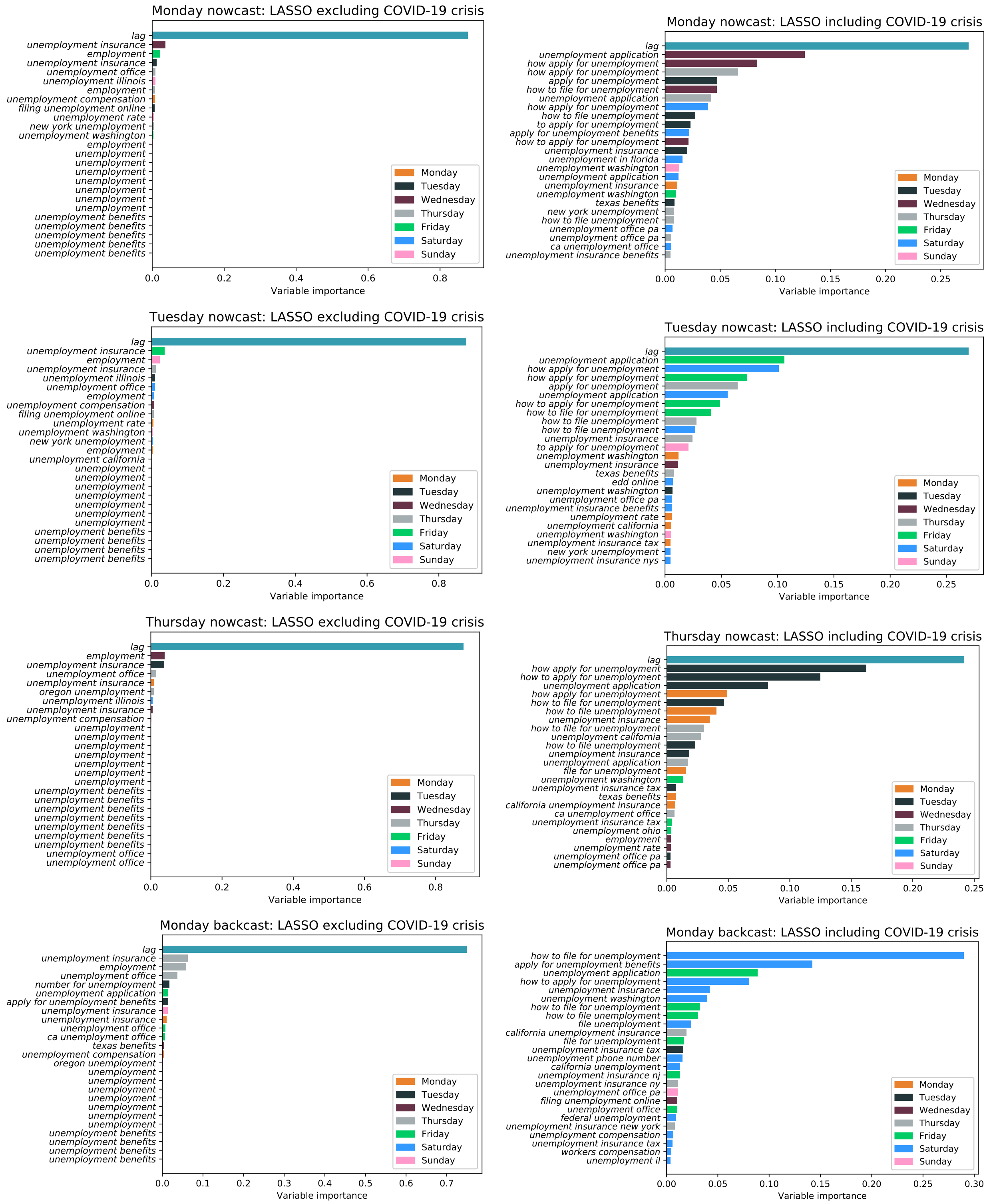

Figure 4: Variable importance, linear models

The figure depicts variable-importance measures for the top 25 Google Trends terms for linear models fitted via the LASSO. Results are reported for Monday_t, Tuesday_t, and Thursday_t nowcasts and the Monday_{t+1} backcast. The estimation sample for the left-hand (right-hand) panels ends in the last week of 2019 (last week of July of 2020), thereby excluding (including) the COVID-19 crisis. Results pertain to seasonally adjusted unemployment insurance initial claims.

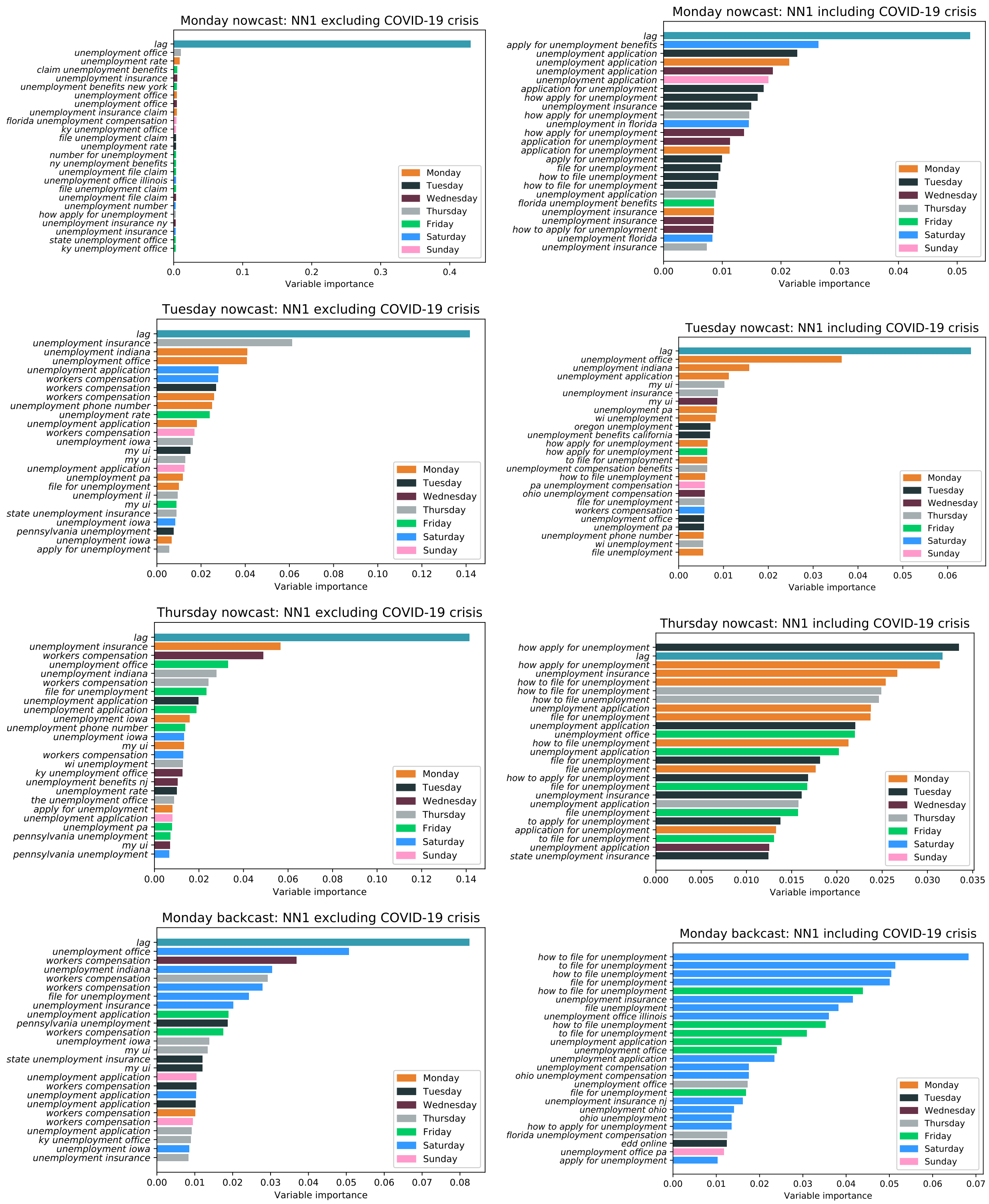

Figure 5: Variable importance, artificial neural networks

The figure depicts variable-importance measures for the top 25 Google Trends terms for fitted artificial neural networks with a single hidden layer (NN1). Results are reported for Monday$_t$, Tuesday$_t$, and Thursday$_t$ nowcasts and the Monday$_{t+1}$ backcast. The estimation sample for the left-hand (right-hand) panels ends in the last week of 2019 (last week of July of 2020), thereby excluding (including) the COVID-19 crisis. Results pertain to seasonally adjusted unemployment insurance initial claims.

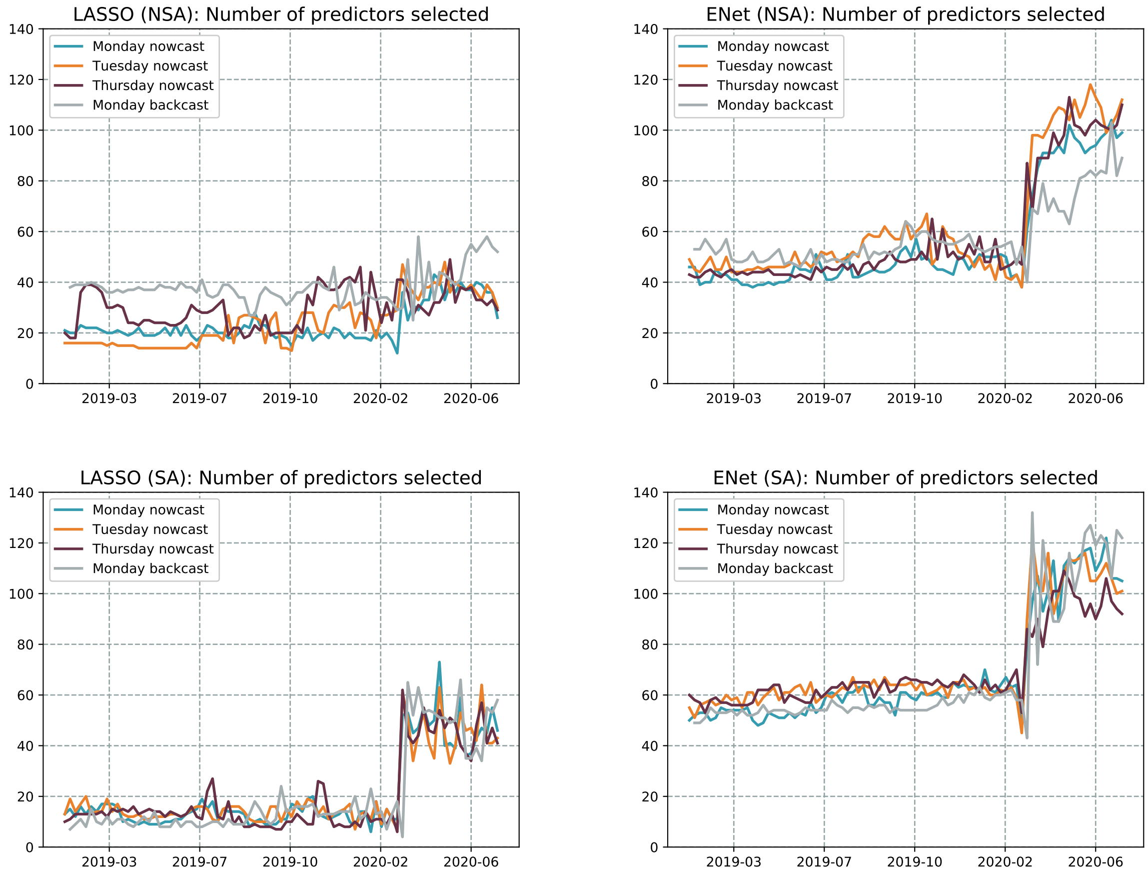

Figure 6: Number of predictors selected by the LASSO and elastic net

The figure depicts the number of predictors selected by the LASSO and elastic net (ENet) for rolling-window estimation of linear predictive models, where each linear predictive model can include up to 722 predictors. Results are reported for linear predictive models used to generate the Monday$_t$, Tuesday$_t$, and Thursday$_t$ nowcasts and Monday$_t$ backcast. The upper and lower panels report results for non-seasonally adjusted (NSA) and seasonally adjusted (SA) weekly unemployment insurance initial claims, respectively. The out-of-sample period spans the first week of 2019 through the last week of July 2020.

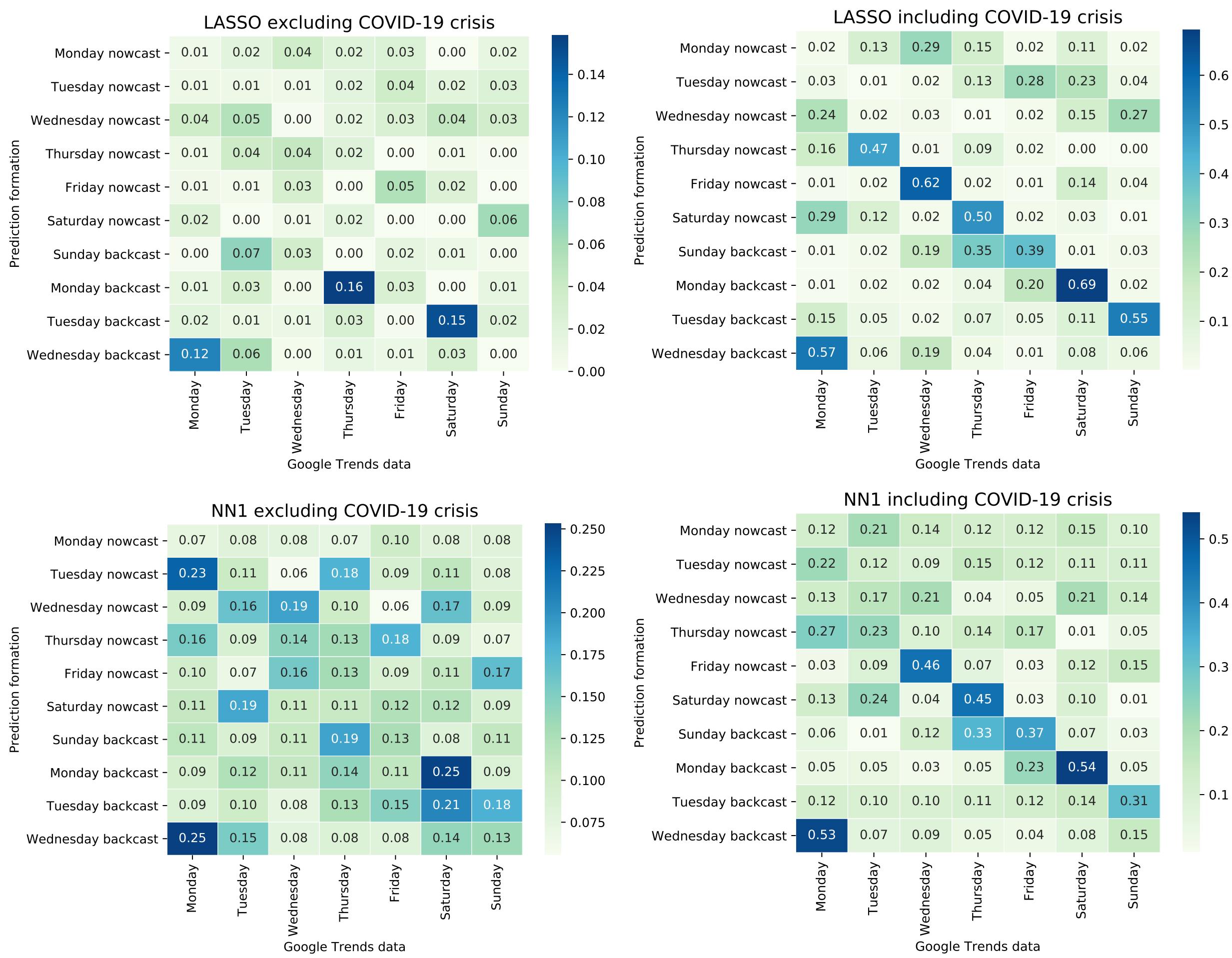

Figure 7: Day-of-the-week effects

The figure depicts heatmaps for joint variable-importance measures of Google Trends terms grouped by the day of the week. The results are for linear models fitted via the LASSO and fitted artificial neural networks with a single hidden layer (NN1). The fitted models underpin the now- and backcasts indicated on the vertical axes. The estimation sample for the left-hand (right-hand) panels ends in the last week of 2019 (last week of July of 2020), thereby excluding (including) the COVID-19 crisis. Results pertain to seasonally adjusted unemployment insurance initial claims.

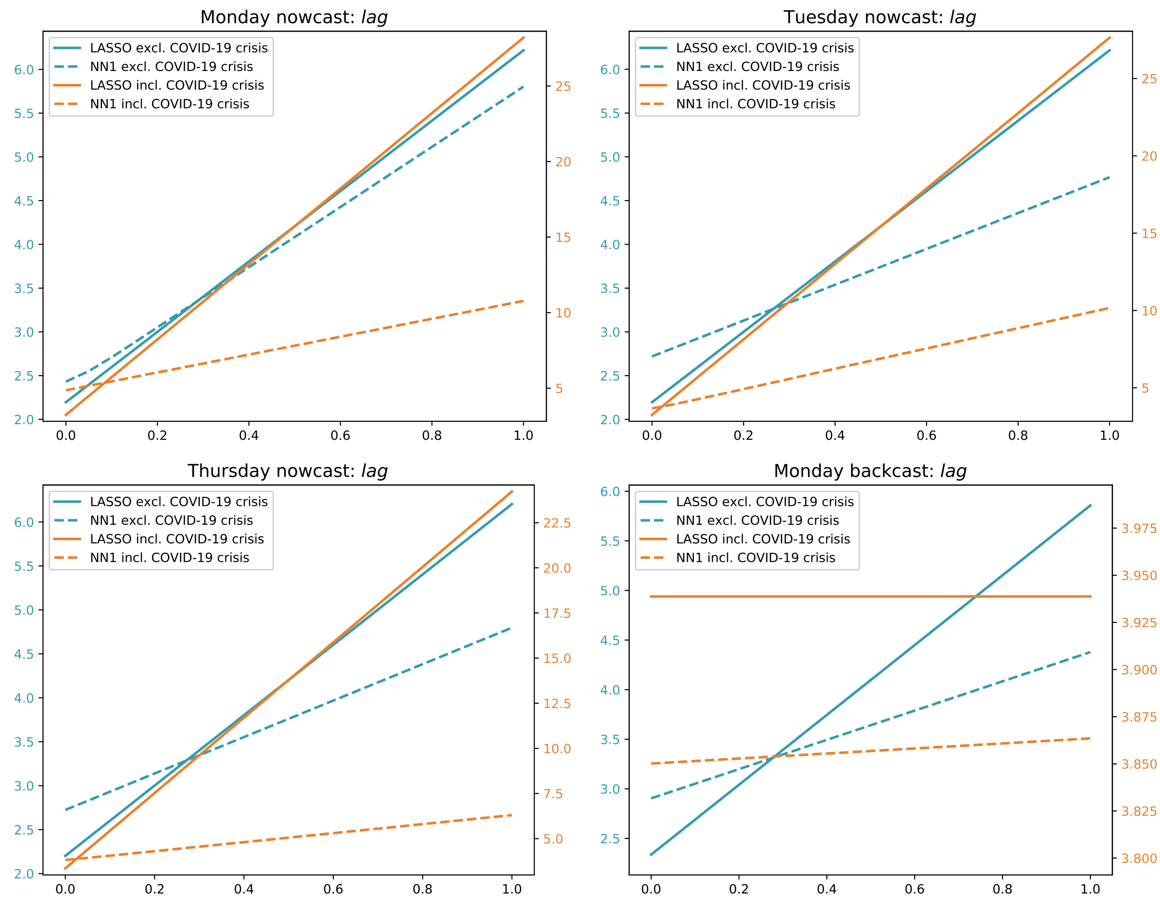

Figure 8: Partial-dependence plots for the AR component

The figure depicts partial-dependence plots for the autoregressive component (*lag*) in linear models fitted via the LASSO and fitted artificial neural networks with a single hidden layer (NN1). The values on the horizontal axis are normalized to lie between zero and one. Results are reported for Monday_t, Tuesday_t, and Thursday_t nowcasts and the Monday_{t+1} backcast. Results are reported for an estimation sample ending in the last week of 2019 (last week of July of 2020), thereby excluding (including) the COVID-19 crisis. Results pertain to seasonally adjusted unemployment insurance initial claims.

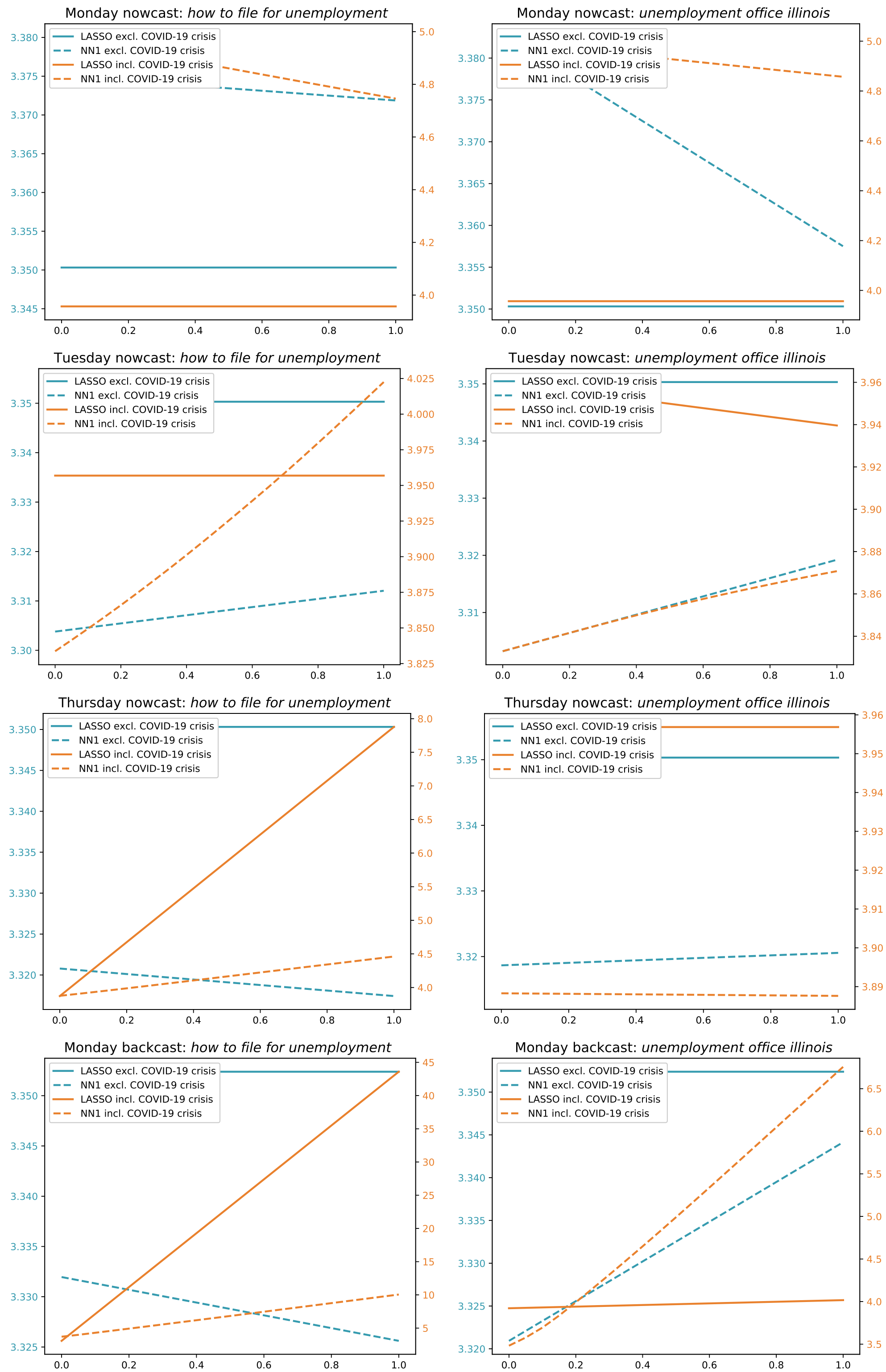

Figure 9: Partial-dependence plots for two GT terms

The figure depicts partial-dependence plots for two Google Trends terms in linear models fitted via the LASSO and fitted artificial neural networks with a single hidden layer (NN1): *how to file for unemployment* (left-hand panels) and *unemployment office illinois* (right-hand panels). The values on the horizontal axis are normalized to lie between zero and one. Results are reported for Monday$_t$, Tuesday$_t$, and Thursday$_t$ nowcasts and the Monday$_{t+1}$ backcast. Results are reported for an estimation sample ending in the last week of 2019 (last week of July of 2020), thereby excluding (including) the COVID-19 crisis. Results pertain to seasonally adjusted unemployment insurance initial claims.