UNIVERSITY OF COPENHAGEN
DEPARTMENT OF ECONOMICS

Master’s Thesis

Sofie Juel

Waldemar Schoustrup Schuppli

Beyond the Hype

A machine learning approach to macroeconomic nowcasting

Supervisor:	Andreas Bjerre-Nielsen
ECTS points:	30 ECTS per author
Submitted on:	30th December 2019
Keystrokes:	200,053 (~83 normal pages)

Acknowledgements

There are a number of people whom we would like to offer our gratitude for their helpful insights and comments throughout our entire thesis writing process.

First of all, we would like to thank *Danmarks Nationalbank* (The Central Bank of Denmark) and the department of *Data Analytics and Science* for providing facilities and guidance. In particular, Senior Data Scientist Alessandro Martinello has provided support, insights and challenging questions through weekly feedback sessions during the entirety of the thesis process, which has helped us tremendously in structuring the analysis and sharpening our results.

Furthermore, we would like to thank our supervisor Andreas Bjerre-Nielsen for his valuable feedback on how to motivate the analysis, and insights into how we should illustrate the results. The overall process has been very pleasant.

The thesis was written in close collaboration between the two authors, and we are jointly responsible for all the choices made throughout the thesis with regards to the written text, data processing, coding and the final results. Any errors made in the thesis rest solely on the shoulders of the authors.

Abstract

Many macroeconomic series such as unemployment rates are published with lags relative to the period that the statistic covers, which renders decision-makers that rely on these macroeconomic series blind for the duration of the publication lag period. Nowcasting is an approach designed to alleviate this issue by providing an early estimate of the desired statistic during the publication lag period.

In this thesis, we will develop a model framework to nowcast the regional unemployment rates of Denmark and also expand the analysis to the regions of Sweden. In order to this, we will incorporate two key aspects: Utilisation of alternative data with very short publication lags, and machine learning techniques.

We utilise job posts from the largest online job market in Denmark, Jobindex, along with search term intensity data from Google Trends. These data sources are available almost real-time, which enables the creation of nowcasting models of the regional unemployment rates during the publication lag period. By combining these data sources with machine learning techniques, we capture more variation and interdependencies in the data, which can allow for more accurate predictions.

Our findings suggest mixed results of applying novel real-time data combined with machine learning techniques versus a traditional econometric nowcasting model. The discovered improvements in nowcast precision of regional unemployment rates are contingent on geography and the choice of the baseline model.

Index of Individual Author Contributions

Sofie Juel:	2.1, 2.3, 3.1.1, 3.1.3, 3.2, 3.4, 4.2, 5.1, 5.3, 6.2, 6.3.1, 6.4, 6.4.2, 7.1, 7.3, 8.2, 8.2.2, 9.2, 9.4
Waldemar Schoustrup Schuppli:	2.2, 3.1, 3.1.2, 3.1.4, 3.3, 4.1, 4.3, 5.2, 6.1, 6.3, 6.3.2, 6.4.1, 6.5, 7.2, 8.1, 8.2.1, 9.1, 9.3
Collaborative:	Abstract, Introduction, Conclusion, Appendix

Contents

1 Introduction

Many actors in the economy, including central banks, government, financial institutions and private companies, are interested in both the current state and the future state of the economy. This is because many decisions facing such actors, even across very different sectors, are highly dependent on having a grasp on macroeconomic activity levels. One of the most important macroeconomic factors is the unemployment rate of the labour force. Central banks may adjust their monetary policies in the form of interest rates, governments may adjust their policies, financial institutions may adjust their lending policies and private companies may adjust their strategies with respect to production - all in accordance with some expectation to macroeconomic activity and in particular, the unemployment rate, which is one key indicator of the overall activity of economies.

Many indicators of macroeconomic activity, such as unemployment rates, GDP levels and investment levels, are published by governmental statistics agencies in the national accounts[1] - these constitute the official statistics that are used to asses the activity in a given economy. However, these statistics are often published with a significant lag (usually 1-6 months) relative to the period that they cover - meaning that e.g. the unemployment rate of January is not published until the end of February. The publication lag of the unemployment rate statistics gives rise to a so-called[2] *nowcasting* opportunity. If you have alternative data sources that are available ahead of publication of the unemployment rate, and this alternative data also has some predictive value on the unemployment rate, then it is possible to use these alternative data to provide an estimate of the unemployment rate ahead of the publication of the official statistic.

Nowcasts differ from traditional macroeconomic forecasts: We will define nowcasts as estimating the unemployment rate for a given period ahead of its publication using alternative data that covers the same periods as the unemployment rate statistic itself. A forecast would, on the other hand, estimate the future path of the unemployment rate multiple periods into the future.

In this thesis, we will develop a model framework to nowcast the regional unemployment rates for the period 2007-2019 by incorporating two aspects key aspects: Utilisation of alternative data with very short publication lags and machine learning techniques.

We will develop the initial nowcasting framework on monthly Danish data and subsequently expand to Sweden, where we will apply the model framework to nowcast the quarterly regional unemployment rates. We will utilise job posts from the largest online job market in Denmark,

[1] National accounts, of which GDP is one the main components, are compiled according to international standards to ensure comparability across countries (Statistics Denmark, 2014).

[2] Nowcasting is a borrowed term from meteorology, where the term is a contraction of *now* and *forecasting* (Giannone et al., 2008).

Jobindex[3], along with search term intensity data from Google Trends. These data sources are available almost real-time, which enables a nowcasting model of the regional unemployment rates during the publication lag period. By combining these data sources with machine learning techniques, we are able to capture more variation and interdependencies in the data, which can allow for more accurate predictions. By benchmarking our nowcasts relative to an autoregressive-based nowcast, we will asses if, when and why our model framework performs better or worse relative to the chosen baseline model.

Our findings suggest mixed results when applying machine learning models along with novel data sources to nowcasting of the regional unemployment rates as the results are contingent on geography and the choice of baseline model. There is a potential for using novel real-time data and machine learning to improve nowcasting of the unemployment rates - but there may be even more potential for other macroeconomic variables.

Nowcasting the monthly regional unemployment rates in Denmark using external data and machine learning reduces the mean root mean squared error, $RMSE$[4], across the 103 nowcasting periods by up to 81 percent (from 0.2144 percentage points to 0.1184 percentage points) compared to the simple autoregressive baseline model. Further, the results suggest that the potential of using machine learning are centred around periods with larger fluctuations in the unemployment rates.

However, when we extend the baseline model with additional autoregressive terms, the results are not robust and the extended baseline model outperforms all the nowcasting models for the Danish regions. When applying the model framework to the regions of Sweden, we find that the preferred machine learning model outperforms the extended baseline model by 10 percent (from 1.2296 percentage points to 1.1159 percentage points) when looking at the mean RMSE across the 34 nowcasting windows. For 14 of the 20[5] regions of Sweden, a gain in the mean RMSE of the machine learning model, XGBoost, over the extended baseline model is found. Though, while the XGBoost model has a lower RMSE for 80 percent of the nowcasting windows compared to the extended baseline model, it cannot be ruled out that this is a statistical anomaly when constructing a bootstrap confidence interval.

The overall implications of our results and the indications from the literature is that in many settings, an autoregressive based approach to nowcasting is sufficient to obtain precise nowcast predictions. We do find that there is a potential to apply machine learning along with novel, real-time data sources to improve nowcasting models, but that this potential is contingent on

[3] Named Jobbsafari in Sweden.

[4] See Section 6 for more details on our evaluation criteria.

[5] We have excluded the region Jämtland in the analysis due to a lack of data availability.

certain conditions. The potential is more likely to exist when the target variable of interest has sufficient cross-sectional variation such that there are enough targets for each nowcasting window. Further, the target itself should exhibit characteristics that renders pure autoregressive nowcasting difficult.

Focus and limitations

As part of the focus of the thesis is to incorporate machine learning techniques, we will delve into the terminology and the application of machine learning when accounting for a time dimension, which is vital for nowcasts. The terminology of machine learning differs somewhat from the terminology of econometrics, but contains many of the same aspects. Thus, we will introduce machine learning as if the reader is familiar with econometrics. We will not delve deeply into the technicalities of each of the utilised machine learning models - rather, we will focus on the intuitive aspects and the step-by-step approach that is necessary to ensure a machine learning framework that is applicable to a nowcasting setting. It is also important to note that we are primarily interested in the prediction of the unemployment rates in a nowcasting setting. Thus, causality of the included data is mostly ignored.

1.1 Modelling approaches in macroeconomic nowcasting

In this introductory subsection we will consider different modelling approaches in macroeconomic nowcasting to give an understanding of how our thesis fits into the broader framework.

The field of macroeconomic forecasting or nowcasting is vast and contains many different approaches, but in general, the goal is shared across the different applied methods: The prediction of the desired statistic must be as accurate as possible. There are many different methods that attempt to achieve this goal, but, in broad terms, most methods falls within or between the following two[6]: *Structural models* and *data-driven models.*

Pure structural models rely heavily on economic theory to establish causal links between macroeconomic variables through systems of equations (Pescatori and Zaman, 2011). Structural models can establish exact causal links between certain variables in the model, which can be used to forecast future development in the economy, given certain assumptions regarding changes (or lack thereof) in some of the variables. Structural models are derived from theoretical economics and is usually tested on empirical data - but it is not directly derived from observed data. Pure

[6] A third approach, which we will not cover, is consensus forecasting/nowcasting (Hall, 2018). In general, consensus forecasting/nowcasting does not directly rely on data-driven models or structural models. Common examples of consensus forecasting/nowcasting could be an expert's prediction based on their gut instinct, a CEO's prediction based on their own past experience or a random prediction from e.g. a monkey (Vermorken et al., 2013).

data-driven models rely on correlations and dependencies that can be found in the empirical data without explicitly incorporating economic theory and imposing structural equations with causal links (Pescatori and Zaman, 2011). A typical example of this is a pure time series model, which extrapolate into the future based on correlations between the past and the present - a so-called *autoregressive* model (Wooldridge, 2018). For example, one could model the unemployment rate as a function of the past values of the unemployment rate. This approach obviously has no direct causal link, but utilises the available historical data and the auto-correlation of the unemployment rate itself to make prediction about the future. Many of the large-scale macroeconomic forecast models in governments and central banks incorporate aspects from both approaches - where some aspects are modelled based on structural equation and other aspects are modelled using the historical data.

A relatively recent development in data-driven models is the application of machine learning methods into macroeconomic forecasting and/or nowcasting models (Hall, 2018). Machine learning techniques can, potentially, remove some of the necessary discretionary choices that are made under structural models and classical time series models. We will return to this in Sections 5 and 6. The rise of the internet has greatly increased the number of potential alternative data source that are available with much shorter lags (Bok et al., 2018). One example would be the number of job openings in an economy. Traditional macroeconomic indicators of this statistic rely on survey questionnaires of a random sample of companies across industries, which is then extrapolated to the entire population. Statistics based on survey questionnaires are usually associated with significant publication lag periods as with many of the national account statistics. With online job markets increasingly representing a larger and larger share of the labour market (Cedefop, 2019), it has become more popular to rely on aggregated statistics from the online job markets instead of survey questionnaires of companies. Aggregated statistics from the online job markets can be made available with a much shorter publication lag than traditional survey questionnaires, which allow online job market data to be applied in a nowcast setting. However, as always, there are drawbacks to relying on these newer alternative data sources, which are mainly procured online. We will return to this in Section 3.2.

Interpreting a prediction from a structural model is quite salient as one can trace the causal links to establish the exact reasons behind the prediction. On the other hand, it can be quite difficult to explain why a given prediction arises from a pure data-driven model besides stating that the model has captured the historical dependencies in the data and extrapolated this into the future. If the goal is purely to predict the future, then the interpretation, or the why, of a given prediction is not as important as the accuracy of the prediction itself. The most important aspect is that any predictive model's processes should be clearly defined and that it should be testable - preferably out-of-sample. We will return to this in Sections 2 and 5.

1.2 Literature review

This literature review serves three purposes. First, we provide an outlook of literature employing econometric models for nowcasting such as time series models. By exploring this, we can identify relevant baseline models. Secondly, we look into literature, which has used alternative real-time data in a nowcasting setting. Lastly, the state of the art of applying data-driven models, such as machine learning models, in a nowcasting setting are explored.

Nowcasting econometric modelling

The nowcasting literature of time dependent macroeconomic variables such as the unemployment rates primarily focus around applying time series analysis from econometric modelling for nowcasting purposes. The simple first order autoregressive model, *AR(1)* often constitute the baseline model in recent literature, which nowcast or forecast unemployment rates on a national level (see e.g. Son et al. (2010), Nagao et al. (2019), Tuhkuri (2015), Pavlicek and Kristoufek (2015)) and other macroeconomic variables (see e.g. Chakraborty and Joseph (2017) and Coulombe et al. (2019)). This simple baseline model has proven to have high predictive power as it by definition performs well in situations, where the underlying data-generating process is not too sensitive to shocks. Tuhkuri (2015) further extends the simple baseline when nowcasting unemployment rates in the US by introducing an autoregressive model, which include both the most recent lag and the one year lag of the depended variable.

For our nowcasting purposes, we are nowcasting the regional unemployment rates. This means that, unlike the papers mentioned above, we have a cross-sectional dimension to take into account. In order to mimic the simple autoregressive framework, we will utilise benchmark models which purely rely on the past values of the unemployment rates to predict the current unemployment rates, but also takes our cross-sectional dimension into account. We will return to this in more detail in Section 6.

Utilising novel real-time data

The digital age has expanded the possibility of looking for alternative data for prediction purposes. Recent literature justify the use of search queries for forecasting purposes. Ginsberg et al. (2009) first introduced the use of Google searches in academic research by including Google search queries to explore how influenza epidemics spread to improve early detection of disease activity.

Recent literature suggest that the inclusion of Google search terms has improved forecasting of unemployment rates in several countries such as Germany, the United States, Finland and Eastern European countries (see Askitas and Zimmermann (2009), D'Amuri and Marcucci (2017), Tuhkuri (2014) and Pavlicek and Kristoufek (2015)). Though to the best of our knowledge, no previous literature has explored whether Google search terms can improve nowcasting of unemployment rates in Denmark and Sweden.

Looking at the methodology to identify relevant search terms previous literature primarily identify words or collections of words which describe individual labour market status such as; *jobs*, specific job search engines, *unemployment benefits* and *employment offices* (see e.g. Askitas and Zimmermann (2009), D'Amuri and Marcucci (2017), Pavlicek and Kristoufek (2015) and Tuhkuri (2015)).

To identify whether Google searches can improve forecasting/nowcasting of unemployment rates, the most common approach is to augment a time series model such as an autoregressive model with additional search term variable(s). The forecasting/nowcasting models are usually estimated with a rolling window[7]. Some papers focus on the statistical significance of the search terms themselves, while others focus on the improvement in forecasting/nowcasting accuracy over a baseline model.

The main result is that Google search terms identifying individual labour market status are found to improve prediction of unemployment by at most a modest amount across explored countries. Some papers find coefficients of the search terms to be statistically significant while others do not. Similarly, not all improvements in prediction accuracy are found to be statistically significant[8] (See Tuhkuri (2015) and Nagao et al. (2019)). Further, Tuhkuri (2015) finds that the predictive power of applying Google search terms is most prevalent in periods with relatively large fluctuations such as in the financial crisis.

Another real-time data source which have the potential of improve the nowcasting of unemployment rates is to include an indicator of the demand for labour in the form of number of job posts on job search engines. To the best of our knowledge, no previous literature has explored the predictive power of this alternative data source in a nowcasting setting.

Machine learning for macroeconomic forecasting and nowcasting

Statistical learning, mostly known as machine learning, has garnered great interest for its applications across many fields including image recognition, translations, language detection, au-

[7] We will elaborate further on this workflow in Section 5.3

[8] based on a Diebold Marino test

tonomous vehicles etc. Recently, machine learning for macroeconomic forecasting has been applied by central banks (Chakraborty and Joseph, 2017). One of the case studies revolves around forecasting the consumer price index (CPI) inflation in the UK. This is one example of applying machine learning to relatively small data sets, but still obtain valuable results. One of the main conclusions is that machine learning models have better performance in unstable periods as the financial crisis. Further, they find that an a simple unweighted combination of the two best performing machine learning models improves performance after the crisis.

Fornaro and Luomaranta (2019) nowcasts multiple macroeconomic variables in Finland such as the quarterly GDP. They find that using machine learning techniques including both regression-based models and tree-based models in combination with real-time data such as traffic data provides competitive predictions of real economic activity. For performance evaluation the root mean squared error, RMSE and others are used. Coulombe et al. (2019) further finds that both penalised-based and tree-based machine learning algorithms outperforms the simple AR(1) model based on the out-of-sample RMSE when examining GDP-growth in Norway.

On the other hand, Makridakis et al. (2018) find that machine learning models are dominated by time series econometrics when it comes to forecasting various time series. They stress that machine models should always be evaluated against a proper baseline model in a clear and meaning full manner, such that the models are evaluated fair and objectively. In this thesis, we will evaluate the applied machine learning against classical econometric baseline models in order to ensure that the findings are compared to meaningful methods and that the conclusions are robust.

1.3 Thesis outline

The thesis is structured in the following manner: In Section 2, a brief theoretical framework of why unemployment exists in a competitive market is introduced. Further, the Danish unemployment insurance system is described to identify, which actions individuals take in order to insure themselves against the consequences of being unemployed. The section completes by identifying alternative data sources, which can be utilised in a nowcasting setting - in particular, we focus on data sources which are available in real time and can serve as early indicators of the unemployment rates. From this stand point the usage of Google searches and online job post data in a nowcasting setting are motivated. In Section 3, the data collection of all included characteristics are described before constructing the data set used in the analysis. In this process several data considerations are discussed before illustrating the overall model timing when nowcasting the Danish regional unemployment rate. In Section 4, we describe and visualise our primary sources data in more detail to provide insights into how these relatively novel data sources are defined, and how they should be included in the analysis. Section 5 outlines the the general methodological framework, terminology and workflow when applying machine learning in predictive modelling. The section ends by stating how the introduction of a time dimension affect the overall workflow and the consequences thereof. Section 6 outlines the prepossessing steps of the original data - this also includes feature engineering. Further, the models of the analysis are intuitively outlined including both the baseline model, and the five introduced machine learning models. Lastly, we are putting it all together by describing the full workflow of nowcasting the Danish regional unemployment rates. In Section 7, the overall results are presented by first presenting on the results of the baseline model. Next, we compare the results obtained by the machine learning models, which include the alternative data sources to this baseline model. Section 8 present the results of two robustness checks. First, the original baseline model is extended to include more information about the previous years unemployment rate. Next, the nowcasting framework is applied to the Swedish regions to test the robustness of the results for the Danish regions. In Section 9, our findings are compared to the literature and methodological considerations and limitations are discussed. The section ends by stating recommendations for future research. Section 10 concludes the thesis.

2 Unemployment

Unemployment is a macroeconomic problem of great interest for many economic institutions and an often discussed topic in the political debate. A person is characterised as unemployed if he is of working age, is able and available for work at the current wage rates, but does not have a job (Mankiw and Taylor, 2014). Being unemployed can have severe consequences on an individual level due to lower income levels, which leads to decreasing living standards. Furthermore, unemployment cause uncertainty about the future.

On a macroeconomic level, the unemployment rate is the term of interest. This refers to the number of unemployed individuals as a percentage of the labour force, where the labour force is the amount of people in the working age that are able and willing to work. A relatively high unemployment rate can result in lower overall private consumption, which is a key component in the national income accounts[9].

In this section a brief theoretical framework of unemployment is outlined focusing on theories explaining why this market imperfection exist. Next, the Danish unemployment insurance system is described. Lastly, we identify early indicators of the unemployment rate, which can be utilised in a nowcasting setting.

2.1 Theoretical framework

In the standard economic theory of markets, you expect a given market to be balanced in equilibrium. This refers to the case where market forces have secured the price of a good is at the level, where the quantity demanded matches the exact level of the quantity supplied. Further you assume that if a market is not in equilibrium, the price of a given good will adjust until equilibrium is obtained. The assumption of balanced markets does in reality not apply to the labour market. There are always some unemployed workers, who do not have jobs - even in periods of economic upswing due to labour market imperfections.

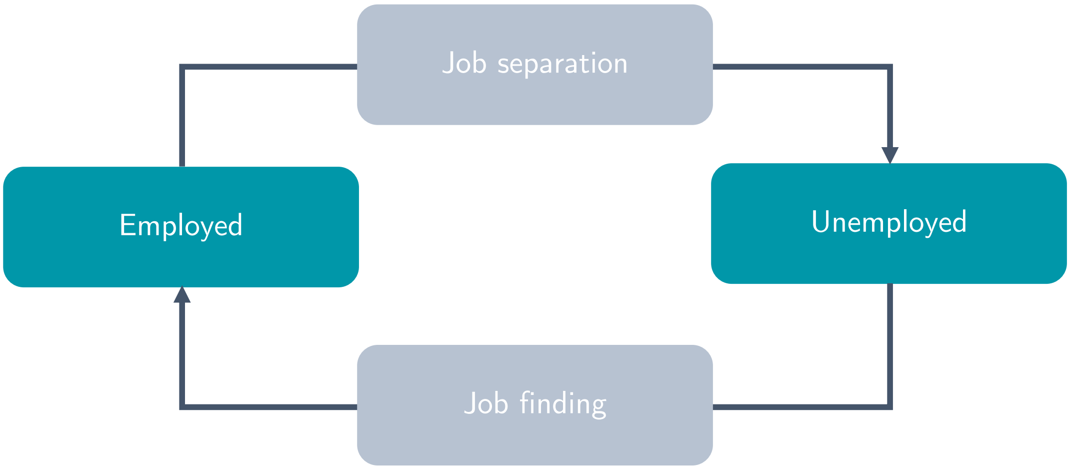
In standard economic theory, the unemployment rate fluctuates around some natural rate unemployment, which is the average rate of unemployment also referred to as *the long run level of unemployment.* In general, multiple theories of why the market for labour hold imperfections exist but they all agree upon two primary explanations for unemployment referred to as *frictional unemployment* and *structural unemployment.*

The first refers to the presence of unemployment due to imperfections in the matching process in the labour market. For an unemployed individual, it takes time to search for and find the

[9] The national accounts define a country's GDP as the sum of consumption, investment, government purchases and net exports. (Mankiw and Taylor, 2014)

most attractive jobs given his/her qualifications and preferences. On the other side of the table, it takes time for the employer to identify the best match for a given position and it often requires multiple job interviews with several candidates. The cycle of employment, which captures the transition between employment and unemployment is summarised in Figure 2.1. An employed individual transfer from being employed to being unemployed by job separation. To regain employment the individual must search for a new job, which can be time consuming.

Figure 2.1: Transition between employment and unemployment



Source: Inspired by Mankiw and Taylor (2014)

Due to constant matching occurring, unemployment in the labour market is inevitable. The faster information spread for job openings the faster the market can match a candidate to a given position and subsequently reduce the unemployment rate. Government policies can also affect the level of frictional unemployment by for instance introducing unemployment insurance. Such a government programme partially protects workers' income when they become unemployed. Unemployment insurance has the benefit of compensating for the decrease in income when being unemployed. Though, it is also argued to increase the level of unemployment as it lowers the cost of being unemployed and thereby adversely affect the incentive of finding a new job.

The other main reason for unemployment is named *structural unemployment* which refers to the scenario where the demand for labour does not meet the supply of labour. One reason for this is wage rigidity - the failure of wages to adjust such that the labour market obtain the equilibrium level of employment. Wages are not fully flexible and can therefore sometimes be stuck above the cleaning level. In this case, the supply of labour is higher than the demand for labour which leads to unemployment. Wage rigidity lowers the rate of job finding.

The main focus of this thesis is to predict the unemployment rate in a given month. No direct theoretical model will be constructed for the purpose of predicting unemployment rates with a structural model. Instead, a data-driven model will be constructed guided by the characteristics, setup and policies of the labour market in Denmark and later for Sweden.

2.2 The Danish unemployment insurance system

Every country has an unique unemployment insurance system. This subsection briefly outlines the main characteristics of the unemployment insurance system of Denmark.

In order to count as an unemployed person, one must be available to work and registered[10] as a recipient of any of the unemployment benefits. These include people who are registered with unemployment insurance funds (*A-kasser*), *cash benefits* (*kontanthjælp*) and any other benefits or programmes from the local or state government.

In order to receive unemployment benefits one must enroll in a voluntary insurance scheme, which requires that you are a member of one of the 25[11] Danish unemployment insurance funds and pay contributions. *Cash benefits*, on the other hand, covers all Danish citizens who are or have been a part of the Danish labour market. Second, you must be enrolled in a job centre from day one of the unemployment period. The primary role of the job centre is to minimise your duration of unemployment by supporting in the process of searching for a new job. This includes free access to computers and other information regarding job search. In 2018, 94 job centres existed which are administrated by the Danish municipalities. The specific criteria to be eligible to unemployment insurance benefits are summarised below. According to STAR (2019a), you must:

- be a member of an unemployment insurance fund
- be enrolled as an unemployed worker at a job centre from the first day of unemployment
- meet the employment requirement
- meet the income requirement
- not be self-inflicted in your status as unemployed
- check job proposals at least every 7 days
- have a complete and approved CV within 2 weeks after you are registered as unemployed
- have had legal residence in Denmark or another EU/EEA country for a total of at least 7 years within the past 12 years.

Cash benefits is a public benefit in Denmark with the primary goal of supporting citizens, who otherwise would be unable to provide for themselves or their families. *Cash benefits* is in principle a universal right for all Danish citizens as long as you meet the following criteria (Borger, 2019):

[10] This only applies for the unemployment statistics of the Danish regions. See Section 3.1.1 for more.

[11] STAR (2019b)

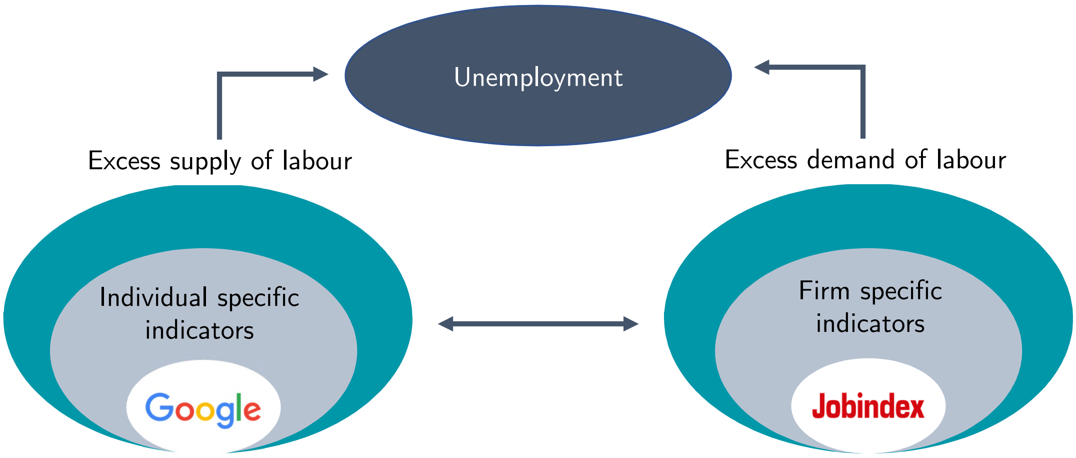
- you are a least 30 years old
- you have been through a critical situation, such as illness, unemployment or separation
- the critical situation has implied that you are unable to provide for yourself or your family and that you are not supported by others
- your need for support cannot be covered by other benefits such as unemployment insurance benefits or pensions.

Knowledge about the local policies and conditions of the labour market of Denmark will be used to construct hypothesised relevant Google search terms - see Section 3.1.2 for more.

2.3 Early indicators of the unemployment rate

The level of unemployment is argued in economic theory to be determined by several factors as outlined in the beginning of this section. As the aim of this thesis is to predict the regional unemployment rate in a nowcast setting, it is crucial to include characteristics, which are indicators of the changes in the unemployment level. It is a necessary condition for the purpose of nowcasting that the included indicators are available in close to real time such that it can be utilised during the publication lag. To get a better understanding of possible data source to include when nowcasting the unemployment rates, Figure 2.2 displays the overall link between supply and demand for labour and how sub-components of each category can be used as indicators to explain changes in the overall unemployment level.

Figure 2.2: Indicators of excess demand and supply of labour



Note: The circles for both excess supply and excess demand of labour represent the set of actions affecting each component. Here, individual specific indicators within the set of excess supply of labour is a subset of all possible actions. Individuals' actions on Google is again a subset of the set of individual specific indicators. The same logic applies for the set representing excess demand of labour.

The possible indicators of unemployment can be found both from data describing the excess supply, and from data describing excess demand of labour. One such example is Google searches.

An individual often turns to Google to seek out information when experiencing changes in their daily life. Individuals will also search for unemployment specific terms in the case of actual unemployment or merely uncertainty regarding the labour market status. Thus, it can function as an early indicator of unemployment from the supply side.

Online job markets serve as an indicator of the demand side of the labour market. To get real-time data one source is to identify online platforms, where such actions are public. An example is the number of online job posts from Jobindex.

Jobindex and Google will serve as two novel, real-time data sources utilised in the analysis.

3 Data processing

This section describes the overall collection process of each of the separate data sources. Secondly, the overall data considerations are discussed. Thirdly, the process of constructing the master data set is outlined with the focus of securing no data leakage in a nowcast setting. Fourthly, the resulting model timing is outlined to motivate nowcasting.

3.1 Data collection

This subsection describes the process of collecting data used in the analysis to predict regional unemployment in the five regions of Denmark. The primary data sources consist of the regional unemployment rates, Google searches indicating the population's labour force status as well as an indicator of the demand for labour represented by vacant jobs on an online job posting site.

3.1.1 Unemployment rates

Danish regional monthly unemployment rates are published by Statistics Denmark[12] with a publication lag of around one month - meaning that the unemployment rate in a given region in e.g. June is published at the end on July. This particular measure of the unemployment rates represent the actual number of unemployed persons (measured in full-time persons aged 16-64) relative to the labour force aged 16-64 in a given period for each of the five Danish regions. The statistic is of October 2019 available for the period of January 2007 to September 2019 and covers close to the entire population of unemployed individuals. The exceptions would be individuals opting out of receiving any form of benefits while being unwillingly unemployed and actively seeking employment. These instances are assumed to be rare - thus, the unemployment rates based on the registries[13] are associated with high a degree of certainty. The methodological approach from Statistics Denmark is the same throughout the entirety of the covered period, meaning that are no concerns with respect to comparability between periods and across regions. We will be using the unemployment data that is unadjusted with respect to seasonality as we are interested in short-term nowcast predictions as in Tuhkuri (2015). There is no consensus as to whether or not to seasonally adjust the data when doing short-term nowcast predictions. Different authors use either adjusted (see e.g. D'Amuri and Marcucci (2017), Nagao et al. (2019)) or unadjusted data (see e.g. Tuhkuri (2016), Askitas and Zimmermann (2009)) with respect to seasonality when predicting unemployment rates. Many of the authors that use seasonally

[12] The relevant statistics is found in Table AUS08 (Statistics Denmark, 2019a).

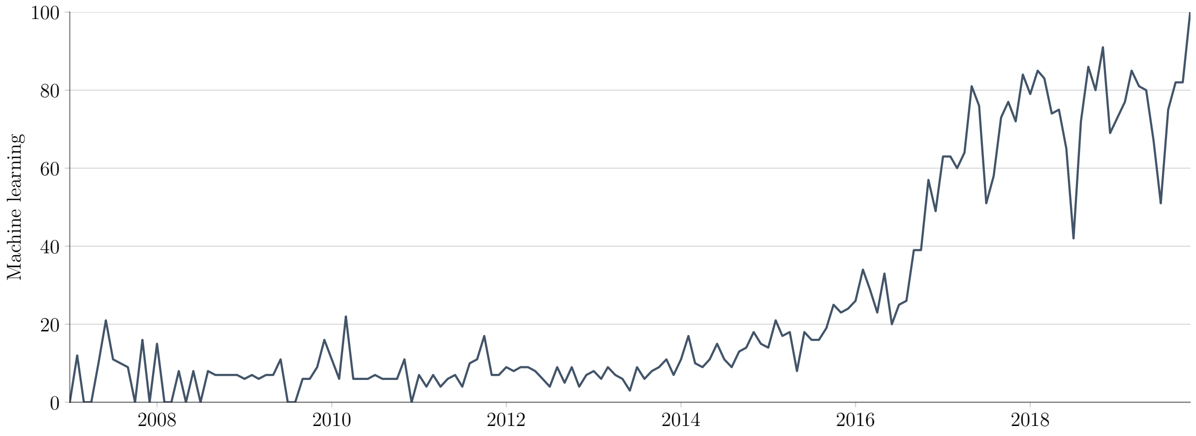
[13] As the unemployment rate used in this paper relies on a unique Danish registry system, the unemployment rates themselves are not suitable for comparisons with other nations. For comparisons, the Labour Force Survey (LFS) would be utilised instead as these are standardised across nations.

adjusted data for nowcasting implicitly introduce data leakage as the seasonal adjustment process itself often relies on the entire data series in order to estimate the seasonal effects (Monsell, 2017).

3.1.2 Google search terms

Google Trends (GT henceforth) is a tool created by Google that allows one to investigate the number of Google searches on specific words or terms over time. The tool allows one to examine how search terms have changed in popularity over time for specified geographical area[14]. Search term intensity is measured as an index from 0-100, where 100 will represent the highest search term intensity for the specified time period and geographical area. An example of a Google Trends time series is shown in Figure 3.1, where the search term is *machine learning* and the geography is Denmark. The search intensity is highest in the last month shown, November 2019, which means that searches for *machine learning* peaked Denmark at this time.

Figure 3.1: *Machine Learning* GT index, Denmark



Source: GT

The search intensity index for *machine learning* is 0 for multiple months during the period 2007-2009. An index of 0 can mean that there is not enough data to create the index - this is important to keep in mind when examining a geographical area with a low population as an index of 0 does not *necessarily* mean that there is no or close to zero search activity - just that the data is insufficient to construct the index taking privacy concerns into account. From Figure 3.1 with the entire country of Denmark as the geography (i.e. high enough population), the occurrences of index 0 in 2007-2009 most likely points to a very low number of searches for *machine learning* in these periods - where the index then is censored to 0 for privacy concerns.

A search term is not restricted to a single word - GT allows certain boolean conditioning

[14] Geographies can be countries or sub-regions for a given country. The granularity depends on the data quality in the country. In the U.S., state level and city levels are available, whereas regional level is available for Denmark

of searches such that search terms can be narrowed down (e.g. removing *steve* when searching for *jobs*) or expanded with an *or* condition (e.g. include an additional spelling of *centre* when examining *center*). Terms can also be restricted to the entire combination of words in the specified order (e.g. *job vacancy*) such that the search term is restricted to the exact phrase without including searches for *vacancy.* These flexibilities allows one to narrow down searches to only include results which are relevant for the specific setting. A note for the terminology in this thesis: A search term can encapsulate a single word (e.g. *jobs*), but also multiple variations of a combination of words that cover the same area (e.g. *open position* or *job openings*).

Data availability and limitations

GT allows users to download the data in .csv-format containing single time series - meaning an index for a search term over time for a specified geographical area. Thus, it is cumbersome to create a rich data set containing many search term time series across multiple geographies. Unfortunately, no official API exists for GT, but it possible to access the data using web scraping methods - i.e. to create a psuedo-API. In this thesis, we have used an already existing psuedo-API for GT called *pytrends* (Elkins and Sonnek, 2019), which web scrapes[15] the contents of the GT web page.

There are certain data limitations from both GT's definitions and backend, and from the method with which we obtain the data that are important to keep in mind.

It is not possible to create a rich longitudinal data sets that preserves the indexation of search intensity along both the cross-sectional dimension (across geographies) and the time dimension simultaneously. This arises from GT's definitions when web scraping[16], where one can either look up the cross-sectional dimension at a given time or look to the time dimension for a given geography. As the search term intensity themselves are measured as a index 0-100, it is not possible to stitch multiple look-ups together in order to get a true longitudinal data set[17].

As our analysis is focused on short-term nowcast predictions, the time dimension is the most vital - thus, we have collected the data such that we have a time series for each search term for each of the five Danish regions. This has some unfortunate, but unavoidable, consequences that are important to keep in mind: Across geographies, we cannot directly compare the index values of the search intensity - regardless of whether or not it is the same search term and/or at

[15] Web scraping or scraping is a coding process of extracting data that is displayed on a website to create data sets that the scraper can then utilise (Heydt and Zeng, 2018).

[16] When web scraping, in general, it is not always possible to fetch the same content as you see when you browse the web page itself.

[17] In principle, it is possible to look up a set of five search terms at a time and then chain them together in order to rank the indices. But this is unfortunately very difficult in practice as the relative search term intensity needs to be almost equal in absolute search term volume - otherwise, a single search term with high volume will push all other indices towards 0.

which time frame the comparison is made.

For example: A search term intensity index of 75 for *machine learning* in Denmark in August 2019 and a search term intensity index of 68 for Sweden in August 2019. From this, we cannot infer that Denmark has a higher search intensity than Sweden because the respective indexes are only indexed relative to each respective geography-specific time series. However, the time dimension within a geography is preserved, which is the most important aspect in terms of being utilised for nowcasting, where we hypothesise that individuals' job market status may change their behaviour with respect to Google searches (Tuhkuri, 2015). Thus, in Figure 3.1, we can infer that the search intensity for *machine learning* is almost twice as high as in November 2019 (index 100) than in July 2017 (index 51).

The data on GT is based on a random sample drawn daily for the given time period and geography - this means that data is not replicable over time as the samples will differ. However, the cross-correlation between values drawn over time is between 0.97-0.99 (D'Amuri and Marcucci (2017), Nagao et al. (2019)), so it is not a major concern. The series themselves have also been de-trended by Google (Tuhkuri, 2015) in order to account for the very rapid increase in overall search volumes from 2004 to the present (40-60 percent annual growth rate from 2002-2010, 10-15 percent annual growth rate since 2010 (99firms, 2019)). The random sample drawn each day for construction of search indices is corrected to attempt to be more representative of actual human internet users. This is done by removing repeated searches of the same search term from the same IP-address within a short time frame - this approach removes most automated searches made by programs and machines.

Selection of search terms

The number of Google searches made per day is astoundingly great and the search terms cover every conceivable topic. This fact, combined with our data collection process, necessitates a hypothesis-driven approach to selecting which search terms to include in the analysis. As we are attempting to include many search terms in a machine learning setting, we will have more search terms than other papers such as D'Amuri and Marcucci (2017) or Nagao et al. (2019), whom only include the general search terms *jobs* and/or *job offer*.

Similar to Tuhkuri (2015), we will use a theoretical, hypothesis-driven approach to constructing a gross-list of relevant search terms, where we hypothesise salient searches that may occur around changes in employment status as well a terms related to the Danish unemployment system. This includes search terms such as *cash benefits*, *unemployment insurance*, *job openings* and more Danish specific words such as *KRIFA* (a Danish unemployment insurance fond) and

Jobnet (a job market run by the Danish Agency for Labour Market and Recruitment). The full list can be found in Table A.2 in the Appendix.

We then iteratively check the data on each search term in the gross-list and how they appear to correlate[18] with the historical data on the regional unemployment rates. We also check each search term time series for whether or not there appears to be enough data behind it. If the index values are fluctuating wildly between high values and 0 - then this indicates that there is a very low search volume (relative to number of total searches in the geography) and this invalidates the search term for use in our analysis. We also require that there is sufficient data for all five Danish Regions, which means that some search terms are excluded in the analysis because there was not sufficient data for e.g. North Denmark (the least populated Danish region).

3.1.3 Job posts

The number of job posts can give valuable insight into a country's labour market even before it is available in the statistics as is illustrates the unmet demand for labour (Tainer, 2006). When a vacant job is posted on an online job posting site, it often takes several months before the position is occupied. Given this, changes in the number of job posts can be useful to identify economic fluctuations and bottlenecks in a economy prior to the occurrence and thereby valuable in nowcasting the unemployment.

Data on the number of job posts is scraped from the Danish job posting site *Jobindex*, which has existed since 1996 and today is the largest job posting site in Denmark with 290.293 job posts in 2018 corresponding to around 85 percent of all job post in the given year[19]. The site has no publication lag, which makes it possible to easily obtain high frequency data about the labour market - and the historic data is available as well, which enables the creation of rich data sets[20]. The retrieved data can be used as an indicator of the total amount of job posts in Denmark and includes monthly job posts in a given region for the period January 2007 to June 2019.

Figure 3.2 illustrate how the data is collected from Jobindex for a given region and month. The historical data is retrieved from the archive. The first step, represented by the number 1 in the Figure, is to specify and retrieve the month of interest. Second, you must specify the region of interest to get data on a regional level. The third and fourth step both constitute of obtaining the results. Here both the total number of jobs and the total number of jobs within

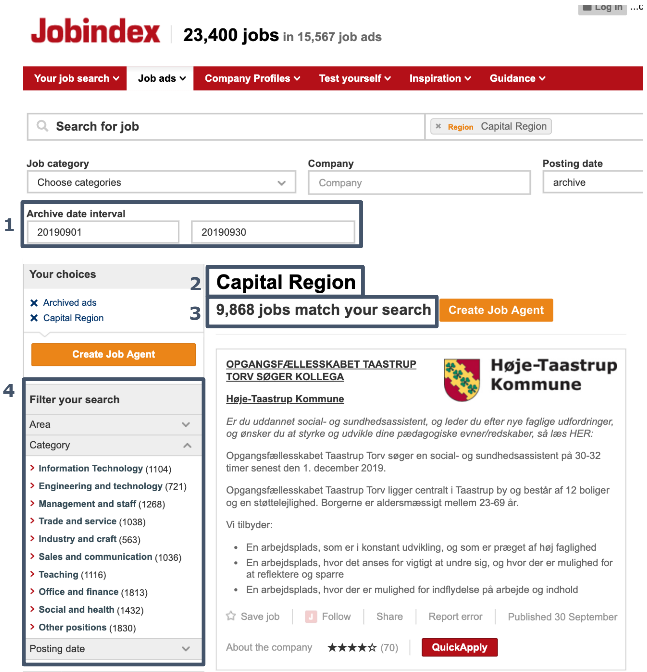
[18] See Section 4.2 for more in depth details about the process.

[19] According to an analysis from *Dansk HR* has the number of job posts in the first 11 months of 2018 been 309.385. This number has been scaled to reflect 12 months and is used as the total amount of job posts in 2018

[20] In an applied nowcasting setting, one would need to scrape both the historical archive as well as the active job posts month-to-month in order to ensure that all job posts for a given month are included.

10 pre-specified sectors are retrieved from the site. This allows one to segment job posts into relevant industries without labelling them first. Job posts can, however, have multiple sector designations, which means that the sum of all job posts across each sector will be larger than the total number of unique job posts. We will return to this in Section 4.3.

Figure 3.2: Scraping Jobindex



Jobindex 23,400 jobs in 15,567 job ads

Search for job

× Region Capital Region

Job category: Choose categories

Company: Company

Posting date: archive

Archive date interval: 20190901 – 20190930

Your choices

× Archived ads
× Capital Region

Filter your search

Area

Category

- Information Technology (1104)
- Engineering and technology (721)
- Management and staff (1268)
- Trade and service (1038)
- Industry and craft (563)
- Sales and communication (1036)
- Teaching (1116)
- Office and finance (1813)
- Social and health (1432)
- Other positions (1830)

Posting date

Capital Region

9,868 jobs match your search

OPGANGSFÆLLESSKABET TAASTRUP TORV SØGER KOLLEGA

Høje-Taastrup Kommune

Høje-Taastrup Kommune

Er du uddannet social- og sundhedsassistent, og leder du efter nye faglige udfordringer, og ønsker du at styrke og udvikle dine pædagogiske evner/redskaber, så læs HER:

Opgangsfællesskabet Taastrup Torv søger en social- og sundhedsassistent på 30-32 timer senest den 1. december 2019.

Opgangsfællesskabet Taastrup Torv ligger centralt i Taastrup by og består af 12 boliger og en støttelejlighed. Borgerne er aldersmæssigt mellem 23-69 år.

Vi tilbyder:

- En arbejdsplads, som er i konstant udvikling, og som er præget af høj faglighed
- En arbejdsplads, hvor det anses for vigtigt at undre sig, og hvor der er mulighed for at reflektere og sparre
- En arbejdsplads, hvor der er mulighed for indflydelse på arbejde og indhold

Published 30 September

About the company ★★★★☆ (70)

Note: The squares captures the collected elements from Jobindex

Source: Jobindex

The scraped data from Jobindex consists of: The number of regional job posts and the number of regional job posts in specific sectors. Here the division of sectors are restricted to the ones displayed on the web page; *Information Technology, Engineering and technology, Management and staff, Trade and service, Industry and craft, Sales and communication, Teaching, Office and finance, Social and health and Other positions.*

The overall data considerations regarding sampling issues when employing Google search terms and job posts as indicators to nowcast the Danish unemployment rate will be elaborated in Section 3.2.

3.1.4 Regional characteristics

The Danish regions are characterised by differences in demographics, urbanisation rates, labour force attachment etc. In order to control for these differences, we include the following information: *population, share of population in the labour force, share of population with a higher education* and *degree of urbanisation* for each region over time.

The characteristics are retrieved from Statistics Denmark. Below, definition and data availability of the individual regional characteristics are described.

Regional population

One important regional characteristic is the population size which can be retrieved from Statistics Denmark Table FOLK1A. The data is available on a quarterly level in the period 2008Q1-2019Q3 and is published medio in the current quarter. It is extracted at the first day of each quarter and is defined as the residential population in the respective region measured by individuals' permanent addresses.

Regional labour force

The regional labour force is included as an control variable as it gives information of the regional composition of the population. The statistic is retrieved from Statistics Denmark's Table AUS08 and is available on monthly basis for the period January 2007 to September 2019 and is published with a one month lag. The labour force is measured as the number of full-time[21] people aged 16-64 that are either employed or registered unemployed in a given region at a given time.

Regional population with a higher education

To get and indicator for regional wealth and living conditions, we include the share of the population with a higher education included as an control variable. The metric is retrieved from Table HFUDD10 in Statistics Denmark, which is published on a yearly level for the period 2008-2019. The statistic refers to the population the first day of January in a given year and is published in the middle of June of the year. Higher education here includes the following education levels: *Vocational bachelors educations, Bachelors programmes, Master programmes* and *PhD programmes.*

Regional degree of urbanisation

The degree of urbanisation can tell about how densely populated a given region is. The metric refers to the share of the population in a densely populated area or urban settlement - A hub of buildings is registered as an urban settlement if it is inhabited by at least 200 persons (60 - 70 dwellings).

The statistic is retrieved from the Ministry of Social Affairs and Interiors platform for key municipality figures and is constructed by using the Tables: BEF1A, BEF1A07, FOLK1, BY1,

[21] For our purposes, we use the available statistics *unemployment rate as a percentage of the labour force* and the *number of full-time people that are unemployed* to calculate the number of full-time people in the labour force.

BEF4A and BEF44 from Statistics Denmark. The data is published once a year and refer to the level by the first day of January in a given year and is published ultimo April.

3.2 Data considerations

As previously stated, the rise of the internet along with far more data being readily available has changed empirical analysis in social sciences (Foster et al., 2016). However, there are certain drawbacks and elements to be cautious of when working with these new data sources - especially if they are web scraped as the Google and Jobindex data of this thesis are.

When doing web scraping, it is important to respect the original owner of the data and not take all the data on the website. Instead, you should only take smaller parts or aggregations of the data as you might otherwise be violating rights of the original owners. We are only scraping relatively few search terms from Google Trends and only scrape aggregated measures from Jobindex, so we are comfortable that we are not infringing on the rights of the respective owners of the data.

One should also always be cautious regarding the privacy and sensitivity of data. In this thesis, we have deliberately only chosen to use data that is publicly available with no privacy issues. The data scraped from Jobindex only includes aggregated descriptive statistics across regions, so we do not use any data, which could be considered sensitive. The Google Trends data on search term frequencies is already filtered, aggregated and indexed to ensure privacy as described in section 3.1.2.

The lack of privacy issues combined with negligible data right considerations for the utilised data in this thesis, greatly increases the value of our nowcasting model of regional unemployment rates as any institution or company will be able to deploy and implement versions of the models themselves.

Many forecasting methods and especially nowcasting methods rely on macroeconomic indicators gathered from surveys. For instance, Statistics Denmark estimate the number of job openings in the private sector based on survey questionnaires to around 7000 companies across industries[22]. This metric, as with many other Statistics Denmark metrics, is associated with a significant publication lag due to sampling corrections and other data processing. In this thesis, we choose to instead use the Jobindex data on the number of online job posts, which is available with almost no publication lag. The Jobindex data, unlike the data from Statistics Denmark, is not a representative sample from the population. This trade-off is difficult to avoid, but we choose to use the Jobindex data as online job banks represent as increasingly larger and larger

[22] See Statistics Denmark (2019c).

share of the overall job market. Combining this with nearly no publication lag, Jobindex data becomes very well suited to nowcasting - despite not being a random sample from the entire job market. The online job market must be considered to have some sampling bias when comparing to the entire labour market. Jobs that are posted online are typically made by large firms, institutions and public sector organisations in a somewhat standardised way. It is most likely that jobs posts from e.g. small, single-person firms are not posted online in a job bank, but rather posted on e.g. Linked in or some other (online or offline) network approach.

Gathering data on individuals, such as employment sentiment, through survey questionnaires present some difficult challenges especially with regards to the veracity of the data. Google searches have some potential advantages in this area. Google searches has been shown to have a high degree of veracity, where it can used to reveal otherwise unacceptable attitudes such as racism that cannot be inferred from survey data (Stephens-Davidowitz, 2014). For nowcasting purposes, surveys also suffer from the lagged publication, which decreases their utility for nowcasting - here, Google searches also provide useful alternatives as the data is available with almost no lag. As with the Jobindex data, GT data is not representative as it only pertains to the part of the population that uses the Google search engine - and we cannot verify GT's sampling methods (Dilmaghani, 2019).

3.3 Constructing the master data set

Section 3.1 outlined the process of retrieving the unemployment rates, Google variables, job variables and regional characteristics. To construct one data set including all data sources, we need to keep the nowcasting framework in mind as we can not include any data points that would not have been available at each point in time - i.e., we need to avoid data leakage. Therefore, we cannot naively merge the data according the dates - we need to ensure that we take any publication lags into account, such that any nowcasts only use data that would have been available at the time as if to mimic an actual application of a monthly nowcast.

Both the Google variables and the job posts can be merged without further actions as they are published without a publication lag. However, the regional control characteristics are all published with different lags in the range of one to six months lag, which means that these variables must be lagged relative to the unemployment rate for each month in order to ensure a nowcast without data leakage.

For instance, to get the population metric in the Capital Region of Denmark in January 2019 lagged statistics are used. As, the population statistic is published in medio of the quarter of interest (mid-February if looking at 2019Q1) the statistic is not available in January 2019.

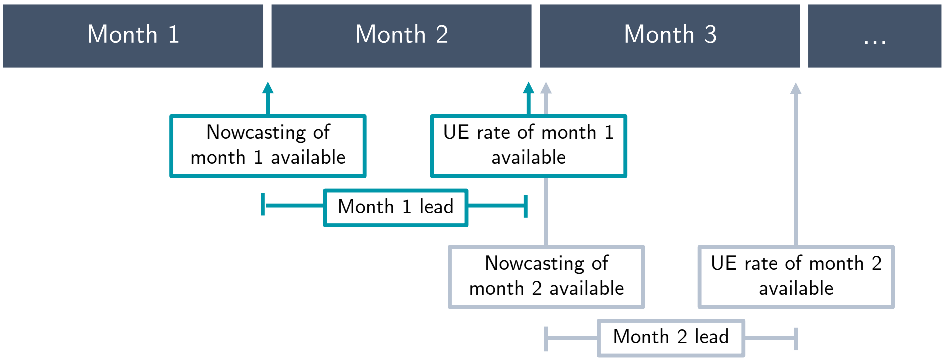
Given this, the population for 2018Q4 is used as metric for the population for January 2019. This exercise is performed with all features published with a lag before merging on to the master data in order to ensure no data leakage.

3.4 Model timing

Many agents in the economy are interested in the unemployment rate because it is one of many indicators of economic activity and stability. As the unemployment rate is published in the official national statistics for a given month by the end of the following month, nowcasting is useful in order to obtain a prediction of the unemployment rate ahead of the publication. In the period before the publication agents often estimates the unemployment rate. To obtain valuable information before the following publication alternative data sources, which are available in real-time are included. Nowcasting is a made possible by introducing Google searches and job posts in a given month as these are both published with a one day delay. Thus, the prediction of a given month can be obtained the day after the beginning of the following month.

To get a precise overview of the model timing in the analysis Figure 3.3 summarises the release of the actual unemployment rates from Statistics Denmark as well as the nowcasting model timing. The model timing is the same for every period in the testing period March 2011 to September 2019 as we will expand upon later in Section 7.

Figure 3.3: Model timing


Month 1
Month 2
Month 3
...
Nowcasting of month 1 available
UE rate of month 1 available
Month 1 lead
Nowcasting of month 2 available
UE rate of month 2 available
Month 2 lead

Note: UE is an abreviation of unemployment.

For instance, following the end of month 1, both the Google searches and the job posts are available for month 1 the first day of month 2 and can be retrieved and the nowcasting model for month 1 can be predicted and used as an precise estimate until the official statistic of month 1 is published on Statistics Denmark by the end of month 2.

4 Data & Descriptive

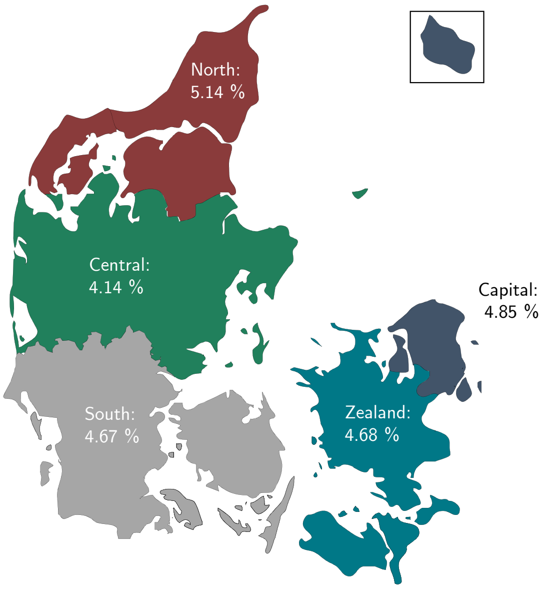
In this section, we will describe and visualise our primary sources data in more detail to provide insights into how these relatively novel data sources are defined. The three primary data sources are Statistics Denmark, Google Trends and Jobindex. We will also show how the different data sources are connected and motivate the predictive value of both the GT and Jobindex data on the regional unemployment rates. For the GT data in particular, we will also motivate the choices of the search terms and show off the iterative process with which we have chosen and included our search terms.

Throughout the section, the data will only be described for the demarcated period from January 2007 to September 2019 as this is the intersection of data availability between the three primary sources as described in section 3.3. The merged data set consists of 760 observations, corresponding to 152 months of data for each of the five Danish regions.

4.1 Unemployment rates & regions

Denmark has five regions in total as shown in Figure 4.1: The Capital Region, Zealand, Southern Denmark, Central Denmark and North Denmark. Each of the regions' average unemployment rate from 2007-2019 is also shown in Figure 4.1, where North Denmark has the highest average unemployment at 5.14 percent, where Central Denmark has the lowest average unemployment rate at 4.14 percent. This indicates that the regions may have different structural characteristics that result in different structural unemployment rates.

Figure 4.1: Average monthly regional unemployment rate, 2007-2019

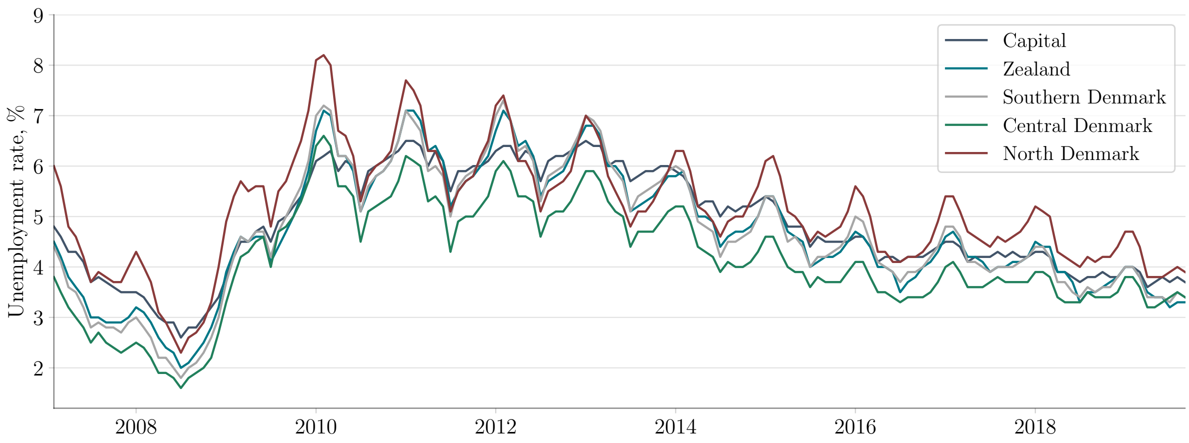


Source: Statistics Denmark

As the period of analysis from 2007 to 2019 covers both the recession stemming from the financial crisis from mid-2008 and the subsequent recovery period from around 2012 onward, naturally, the unemployment rates for each of the regions has fluctuated substantially as shown in Figure 4.2. Note that series shown are not seasonally adjusted as we are focusing on a nowcast prediction, which is only relevant short-term - thus, we are focusing on describing the data in the same manner as it utilised in the models[23]. This means that Figure 4.2 also shows the seasonal fluctuations in the unemployment, but it is still quite prevalent to spot the trend: The boom period up until mid-2008 with a sharp decline in the unemployment rate to levels of around 1.8-2.5 percent for all of the regions followed by a rapid increase in the unemployment for all regions up until around the beginning of 2010, where the unemployment rates were 6.1-8.1 percent. From there, a steady decline in the unemployment rates is observed until September 2019, where the unemployment rates are between 3.3-3.9 percent.

We again see the pattern shown in Figure 4.1, where North Denmark tends to have the highest unemployment rate and Central Denmark tends to have the lowest unemployment rate of the five regions and the remaining three regions tend to fluctuate in the middle. We also notice that the regional unemployment rates tend to co-move over time - but also that this is not always the case, which again points to regional specific tendencies.

Figure 4.2: Monthly regional unemployment rate, %



Source: Statistics Denmark

We try to capture some of the regional variation by including certain background characteristics of the regions and their populace as shown in Table 4.1 with averages of monthly regional characteristics from 2007 to 2019. The Capital region, which contains the Danish capital and also the most densely populated city, Copenhagen, is the largest region with about 1.7 million inhabitants, where North Denmark is about a third in size with a population of 580,000. 21 per-

[23] See Section 3 for more details.

cent of the population of the Capital region have a higher degree of education, where the share is around 12-15 percent for the other regions. This indicates the known tendency of academics gathering in the largest cities. The labour force attachment is highest for the Capital region, where around 51 percent of the population is in labour force, whereas it is around 47-49 for the other regions. The Capital region also differentiates itself from the other regions with respect to the degree of urbanisation, where 97 percent of the populace living in cities for the Capital region, where it is 80-84 percent for the other regions. This also shows how Denmark, overall, is relatively urbanised country.

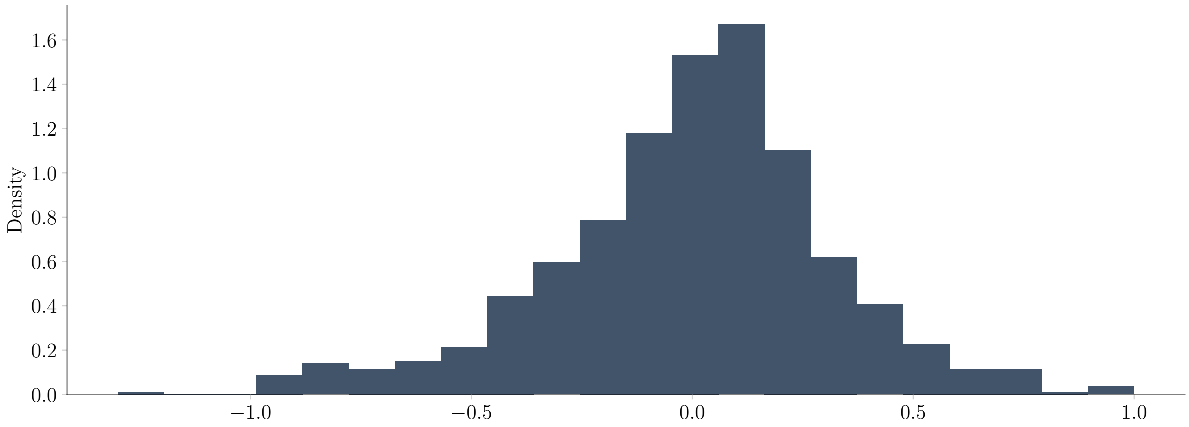
Table 4.1: Average regional characteristics, 2007-2019

Region	Population	Higher education, %	Labour force, %	Urbanisation, %
Capital	1,737,053	21.3	50.7	97.4
Central Denmark	1,274,867	15.1	48.8	84.5
North Denmark	582,403	13.0	47.3	80.4
Southern Denmark	1,205,761	13.2	46.8	83.3
Zealand	823,312	12.4	46.7	80.6

Note: The listed characteristics are the averages of each monthly data series for each region. For the exact definitions and sources of each metric, see Appendix Table A.1.
Source: Statistics Denmark

As indicated in Figure 4.2, the monthly changes in the unemployment rate for a given region are relatively small. Figure 4.3 shows the distribution of the monthly changes in the unemployment rate across the five Danish regions. The unemployment rates almost never jump or fall with multiple percentage points month-to-month - in fact, the largest change in the unemployment rate occurred in April 2010 , where the unemployment rate jumped with 1.3 percentage points from 6.7 percent to 8 percent in North Denmark. In the period of analysis, the mode of the change in the unemployment rate is 0.1 percentage points, which occurs in 17 percent of the months, where the second and third most frequent changes in unemployment rates are 0 and -0.1 percentage points, which occurs in 16 percent and 12 percent of covered months, respectively. This indicates that it might be easier to model the changes in the unemployment rates rather than the level themselves - which is a very common approach in the literature and also one we will utilising in this thesis.

Figure 4.3: Histogram of changes in the regional unemployment rate (%-points), 2007-2019



Source: Statistics Denmark

4.2 Google search terms

Google search term intensity is measured as index (0-100). In this subsection, we will describe and visualise the GT data and expand upon some of the discussions and limitations behind the GT data as described in Section 3.1.2. We will visualise the data behind search terms that are applied in our predictive model, as well as some search terms from our initial gross-list that ended up being dropped.

As previously noted, it is important to keep in mind that the method of gathering data on search term intensity requires a hypothesis-driven approach, where we have to fetch data for a limited amount[24] of search terms that we come up with ourselves. The data gathering method is somewhat restrictive, making it infeasible to e.g. get data for 100,000 search terms and then data-mine these to find those that has some significant correlation with the unemployment rate. Thus, we have applied a hypothesis-driven approach, where we only include search terms that we hypothesise having some relation to the labour market - i.e. where it is feasible that changes in the labour market would change individuals search behaviour on Google. These search terms are primarily related to the supply side of the labour - i.e. potential employees.

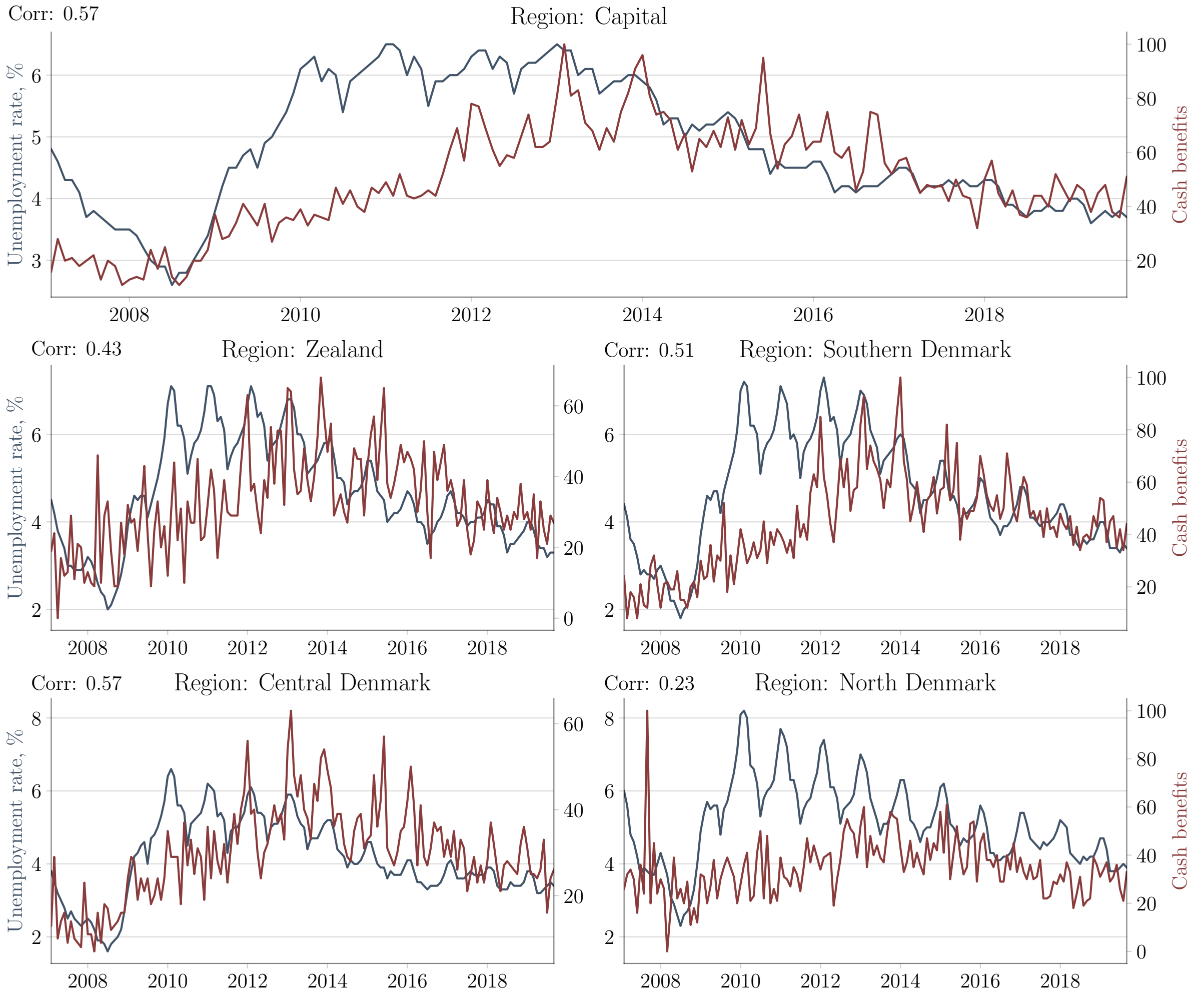
The search terms include aspects of the Danish Social Insurance system - such as searches for *cash benefits*, *unemployment insurance*, *unemployment insurance rates*, *unemployment insurance fund* and specific names of the largest unemployment insurance funds. One would assume that these search terms are positively correlated with the current unemployment rate. As more and more individuals in the labour force feel at-risk of losing their jobs or out-right lose their jobs, they be more actively searching Google for answers about their social benefits associated with

[24] Data fetching is limited for both time concerns and ethical issues - see Section 9 for more.

unemployment - and it is this specific behaviour that may be utilised to nowcast the current unemployment rate. Other search terms such as *job openings* may be negatively correlated with the unemployment rate as more individuals may be searching for jobs when there excess supply of labour, which may indicate a fall in the unemployment rate - but the correlation could also be positive as there would be searches on job openings, when individuals lose their jobs. Most of our gross-list search terms are, however, assumed to be positively correlated with the unemployment rates. The full gross-list is shown in Table A.2 in the Appendix.

Figure 4.4 shows the co-movement of search term intensity of *cash benefits* and the unemployment for each region from 2007-2019 along with the sample cross-correlation.

Figure 4.4: Regional sample correlation, Unemployment rates vs. *Cash benefits* GT index

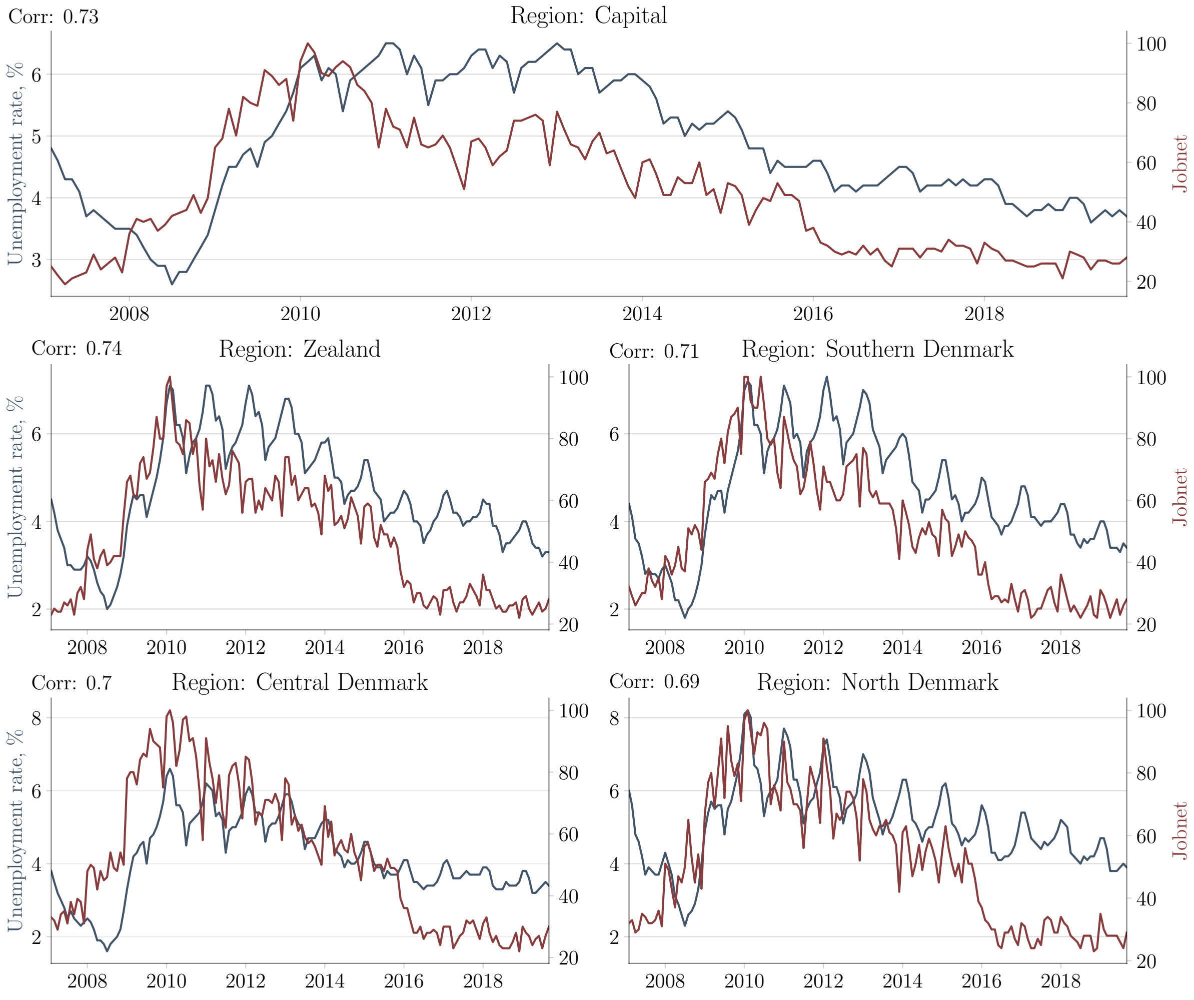


Source: Statistics Denmark, GT

Cash benefits is a unemployment assistance for those that are no longer eligible (or are uninsured) for unemployment benefits from unemployment insurance - and you must be unemployed and not studying to qualify for cash benefits. The search intensity of *cash benefits* correlates positively with respective unemployment rates across all regions, but there is also some observed heterogeneity. The correlation is 0.57 for both the Capital Region as well as Central Denmark, whereas the correlation is quite low for North Denmark with only 0.23. We also notice that the variation in the search term intensity month-to-month is greater than the variation of the unemployment rate itself. Figure 4.4 shows how the search term *cash benefits* may have some predictive value on the unemployment rate - but also significant heterogeneity in the potential predictive value across regions. The correlation also appears to be stronger across all regions from about 2012 and on wards, which might be problematic as the usefulness of the search term as a predictor may be significantly lessened during the early nowcast windows of our analysis.

Figure 4.5 shows the co-movement between the search intensity of *Jobnet* and the regional unemployment rates.

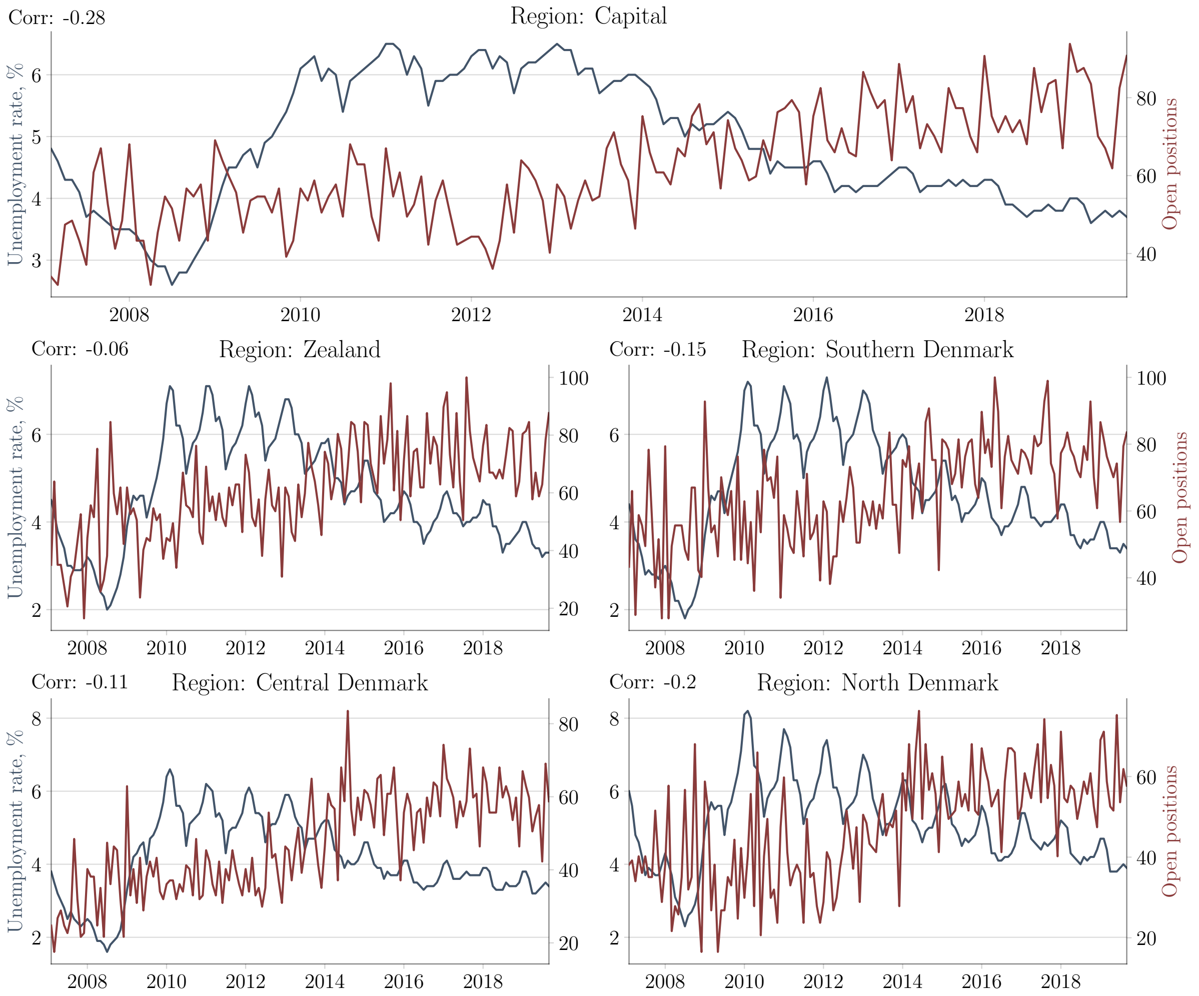
Figure 4.5: Regional sample correlation, Unemployment rates vs. *Jobnet* GT index



Source: Statistics Denmark, GT

Jobnet is a website run by the Danish Agency for Labour Market and Recruitment. The website serves multiple functions[25] in the policies regarding the labour market. It serves as a job market with companies posting their job openings and also as a resume-bank for potential applicants that also allows companies to directly contact potential employees. It also serves as a general information site about the Danish labour market, its job centres and how to navigate it as e.g. an unemployed person or as a foreigner. Thus, *Jobnet* as a website is a very important part of the local municipalities work with unemployment policies. The search term *Jobnet* has a quite strong positive correlation with the respective unemployment rates across the five Danish regions - with correlation coefficient of 0.69-0.74. The co-movement also appears quite similar across regions, where there is much less heterogeneity across regions compared to the case of *cash benefits* from Figure 4.4. It appears that *Jobnet* may be a more valuable predictor of the unemployment rates than *cash benefits* - although *cash benefits* may still hold some predictive value.

Figure 4.6: Regional sample correlation, Unemployment rates vs. *open positions* GT index



Source: Statistics Denmark, GT

[25] See Jobnet (2019)

Figure 4.6 shows another example with the search term *open positions*, which is similar[26] to one of the search terms, *job offer* used in Nagao et al. (2019), D'Amuri and Marcucci (2017) and Askitas and Zimmermann (2009).

The co-movement between *open positions* and the unemployment rates is quite low with an observed in-sample cross-correlation between -0.28 and -0.06. This indicates that the potential predictive value of the search term intensity of *open positions* is quite low - and we have excluded this search for our nowcasting models.

Inclusion of search terms

The previous example of co-movement between the regional unemployment rates and the respective search terms describes part of our method of choosing which search terms to include. The initial inclusion criteria for the search terms as to whether or not they are included in the subsequent can be summarised as:

- The search term intensity must be available and sufficient for all five Danish regions
- The search term intensity must not have too frequent occurrences of index value 0
- The search term intensity must have correlations of a certain magnitude with the regional unemployment rates

First, we check that there is sufficient data to construct a time series for each region for each search term. If the absolute level[27] of search volume for a search is very low for most months, then the time series will not be constructed at all by Google for privacy considerations. This happens for North Denmark with the search term *resume assistance* - thus, we have excluded the search term entirely.

Secondly, given that the time series is feasible to construct by Google, we check there is not too frequent occurrences of index values of 0 as this would indicate very low search volume. Some of these search term series have index values either in the range of 50 and then 0's for the rest - this indicates that there is just enough search volume to construct the time series based on Google's criteria. But these time series exhibit too discrete characteristics, rendering them less useful for our analysis - thus, we also exclude these types of series.

Lastly, we check the co-movement between the each search term and the unemployment rate to ensure that there appears to be correlation with some magnitude that can indicate the predictive value of the respective search terms. This correlation magnitude criteria has been set

[26] The two terms are quite similar as the meaningful translation of *job offer* into Danish would be equivalent to *open positions*

[27] The exact search volume criteria is unknown (D'Amuri and Marcucci, 2017).

as |0.35|. Looking at correlation coefficient is in no way definitive as a correlation of any given magnitude may have very low predictive value, but it can point one in the right direction when doing an empirical analysis.

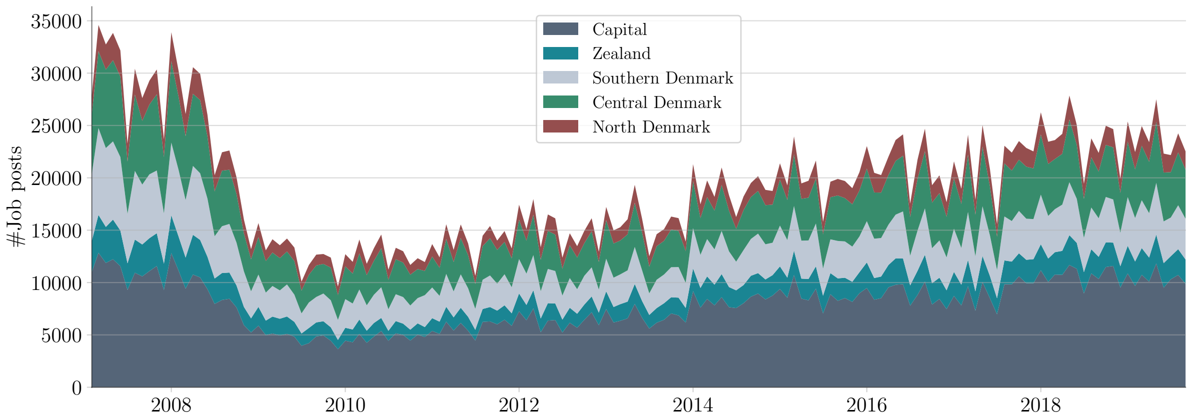
Some search terms that we hypothesised having a predictive value on the unemployment end up being discarded because we do not actually see the relationship in the data. Example of these are *open positions* as shown in Figure 4.6 and also search terms such as *resume* and *unemployment insurance* end up being discarded. Table A.2 in the Appendix shows an overview of the search terms, which ended up passing all of our inclusion criteria.

4.3 Job posts

As described in Figure 3.2 in Section 3.1.3, we have collected summary data regarding the number of job posts across the five Danish regions. Throughout the period 2007-2019, there has been about 19,500 jobs posted per month on average in Denmark. The Capital Region has the largest share of these 19,500 job posts on average, where the job posts of the Capital Region make up approximately 41 percent of the total number of job posts. Central Denmark and Southern Denmark has job post shares of 22 and 19 percent, respectively, and Zealand and North Denmark has job post shares of 10 and 8 percent, respectively. If we compare the job post shares to the population shares[28] of each the regions, then we can see that the Capital Region is slightly over-represented and North, Southern and Zealand are slightly under-represented. Figure 4.7 shows the monthly evolution of the total number of job posts across the five regions. It is notable that the number of job posts peaked in 2007-2008 and has never recovered (in absolute levels at least) since the financial crisis. This most likely is cause of the exceptional boom in the year 2005-2008 before the outset of the financial crisis. It could also be that Jobindex itself has lost market share relative to other job banks during the period.

[28] The monthly average share of Denmark's population is 31, 23, 21, 15 and 10 percent for the Capital Region, Central Denmark, Southern Denmark, Zealand and North Denmark, respectively. See Table 4.1.

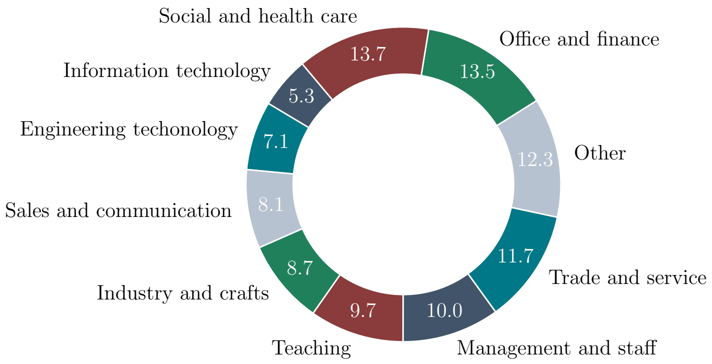
Figure 4.7: Total number of job posts across Danish regions



Source: Jobindex

The job posts at Jobindex can be assigned to one or more sectors as also shown in Figure 3.2 in Section 3.1.3. These sectors include *industry and crafts*, *office and finance* etc. as shown in Figure 4.8.

Figure 4.8: Job posts sector decomposition (%), 2007-2019



Source: Jobindex

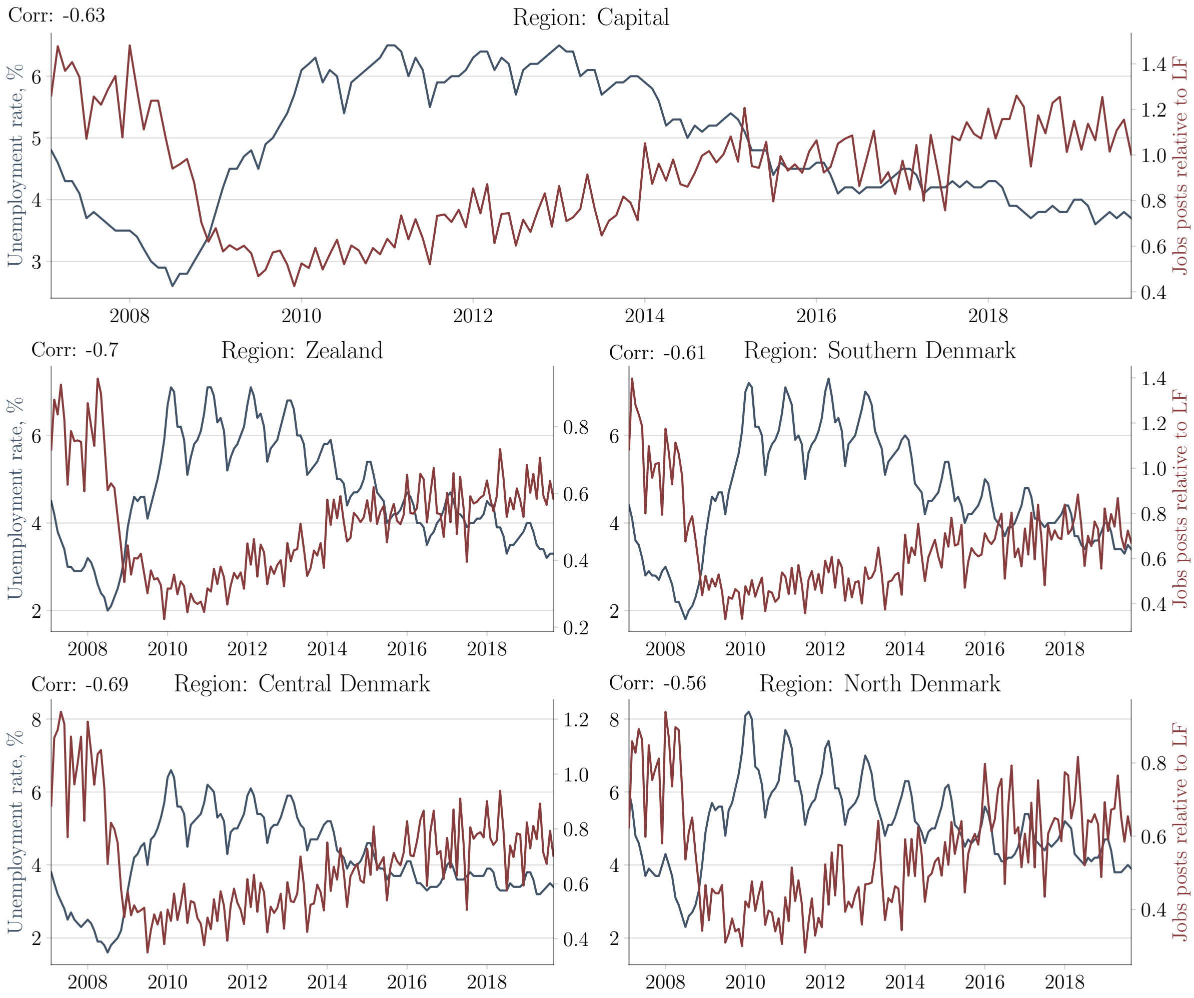
With our web scraping method, we only fetch the summary statistics - thus, the sum of all sector job posts will be greater[29] than the total number of job posts in a given region in a given month. However, we can still utilise these sector segmentation, as some sectors may be more or less cyclical with respect to the business cycle of the economy - thus, even though they may be slightly mismeasured due to double-classification of sectors, the relative number of job posts in various sector may still have predictive value on the unemployment rate. Indeed, some sectors, such as *industries and crafts*, would be assumed to be more cyclical than e.g. *social and*

[29] This could be addressed, but would require fetching background information about every single job posts, which most likely would violate fair usage of Jobindex' data.

health care, which is more determined by the policies of the government. Figure 4.8 also shows that the distribution across sectors is quite even with only a few percentage points separating the shares of each sector.

In order to account for the differences in the sizes of the regions, we calculate the number of job posts relative to the labour force in each region such that the metrics are more comparable across regions. The number of job posts relative to the labour force in a given region may have some predictive value on the current unemployment rate. This is shown in Figure 4.9.

Figure 4.9: Regional sample correlation, Unemployment rates vs. Job posts relative to labour force



Source: Statistics Denmark, Jobindex

We observe a robust, negative correlation between -0.56 and -0.70 across the five regions between the job posts rate and the unemployment rate, which is in line with what one would hypothesise. This is because more job posts indicates a higher demand for labour, which in turn indicates that the unemployment rate should fall. However, as outlined previously in Section 2.3, it is likely that jobs are not filled within a month of the post because friction in searching and matching between potential employees and employers. Therefore, we lag the job rates with three months to reflect this friction in the labour, as shown in Figure A.1 in Appendix. Around three months has been found to be the average number of days a job post is online before it the position is filled according to Cedefop (2019). By lagging the job posts the correlation becomes much more robust with values between -0.77 and -0.86 across the five regions, which indicates that the friction between job posts and actual employment is present. The potential predictive value may also be higher when we lag the relative number of job posts - which is the approach of the models as will be described later.

A similar pattern appears when looking at the number of jobs across the sectors, all relative to the respective labour forces. We follow a similar approach as in Section 4.2, where we check that the correlation between the given sector job posts relative to the labour force has some significant correlation with the unemployment rate. All sectors listed in Figure 4.8 except *sales and communication* and *other* display large negative correlation above -0.35 with the unemployment rates across all regions. The actual correlations are shown in Appendix Table A.3. This indicates that the sector shares can be utilised in a predictive model.

5 Machine learning in social science

The digital age has, as noted in Salganik (2019), reformed the approach to social science research. Researchers can now run experiments in ways that were simply not possible in the recent past before the existence of cheap high-speed computers. The science of economics has also evolved rapidly with the modern technology and big data - from being driven mainly by theoretical assumptions and structural models to a more data-driven approach. Machine learning models are a collection of algorithms that has the ability to learn from data to e.g. make predictions about future events. In this section, the goals, terminology and general workflow in machine learning will introduced as if the reader is familiar with the basics of traditional econometric modelling.

Before describing the, at times, complex workflow of doing machine learning, the differences between traditional econometrics and machine learning are discussed to motivate the use of machine learning for nowcasting. Next, the general machine learning terminology is introduced and explained as well as initial considerations with regards to model evaluation and choosing the optimal model.

Economtrics and machine learning: Goals in predictive modelling

In a classical econometric analysis for prediction one would, as described in Wooldridge (2018), define an outcome variable and explanatory variables, which by some underlying functional form are assumed to capture the variation in the outcome variable - thus allowing for predictions of the outcome. Given a random sample of the population, we can find the optimal parameter values, which represent the best fit of the actual outcome variable by some objective function such as the sum of squares. To understand the implications of this method, let us look at an example, which can be estimates using ordinary least squares (OLS):

$$y = X\beta + \epsilon \tag{1}$$

In traditional econometrics, the main focus is on the quality of the estimators and the interpretability of the parameter estimates, $\hat{\beta}$, as well as determining the uncertainty of the estimates as stated in Athey and Imbens (2019). If the underlying model is correctly specified and other assumption are met[30], then the ordinary least squares estimator is the best linear unbiased estimator. In contrast, machine learning methods are more concerned with the prediction of the outcome, $\hat{y}$ - especially out-of-sample. We will return to the exact definition of this later.

[30] These assumptions are known as the Gauss-Markow assumptions - see Wooldridge (2018) for more.

In machine learning three overall categories of models exist which is employed depending on the overall problem; *supervised learning, unsupervised learning and reinforcement learning*[31]. Supervised machine learning is the preferred method when considering a prediction problem (Raschka, 2015). The goal supervised learning models is to predict a known, measureable[32] outcome, which is the exact problem specification when predicting the unemployment rate.

Supervised machine learning consists of two subcategories: Classification analysis for binary or discrete outcomes and regression analysis for continuous outcomes. The latter is the focus in the analysis as the unemployment rate is a continuous variable. The task is to find the relationship between explanatory variables and the actual outcome variable to predict a given unemployment rate. In contrast to the traditional econometric analysis, you are not as restricted by some underlying functional form when employing different machine learning models as they rely on numerical optimisation approaches.

5.1 Machine learning terminology

The basic terminology and framework in machine learning differ slightly from that of econometric modelling. First, with respect to the phrasing and standard notation, the actual outcome variable, y, is called the *target variable* or simply *target* and the explanatory variables, X, is referred to as *features*. Figure 5.7 illustrates the standard workflow in machine learning, which will be described in more detail in Section 5.2. Before diving into the technical workflow, a number of methods used in machine learning, which all have the purpose of increasing model prediction performance will be introduced.

The learning aspect

The *learning* of machine learning stems from how the parameters, usually called weights in machine learning lingo, of a model are estimated. Let us return to example of a linear equation:

$$y = X\beta + \epsilon = \beta_0 + \beta_1 x_1 + ... + \beta_m x_m + \epsilon$$

In econometrics, the parameters of the equation above can estimated with OLS, where the analytical solution, i.e. the method of actually calculating the parameters, can be found by e.g. minimising the sum of squared errors, method of moments or by maximum likelihood

[31] We will only focus on supervised learning in this thesis.

[32] Unsupervised learning, on the other hand, concerns models, where the there is no labelled outcome variable with which to model. A typical example could be customer-segmentation, where you attempt to cluster customers into segments. But these cluster-segmentations are not known beforehand - thus, you cannot model customer-segmentation after labelled data as it does not exist.

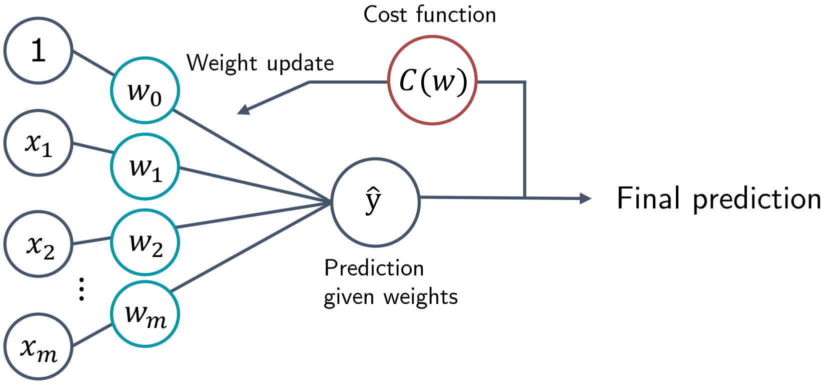
estimation (Wooldridge, 2018). Under certain assumptions, these approaches will result in the same estimators and thus, the same estimated parameters for the above equation.

In machine learning, the parameters are estimated using numerical approximation with a *learning* aspect. A bit of notation rewrite to match the machine learning lingo (Raschka, 2015):

$$y = w_0 + w_1 x_1 + ... + w_m x_m m$$

Where the individual w_i are sometimes called weights. The numerical approximation, at its core, is shown in Figure 5.1.

Figure 5.1: The learning process in machine learning



Source: Inspired by Raschka (2015)

At first, the weights are randomly initialised from e.g. a standard normal distribution. Then the predicted target, $\hat{y}$, is evaluated against the actual target, y, and the errors are calculated, i.e. $y - \hat{y}$. These errors can then e.g. be squared and summed, which means we will get a familiar concept from econometrics, the sum of squared errors:

$$C(w_0, w_1, ..., w_m) = \sum_{i=1}^{m} (y - \hat{y})^2$$

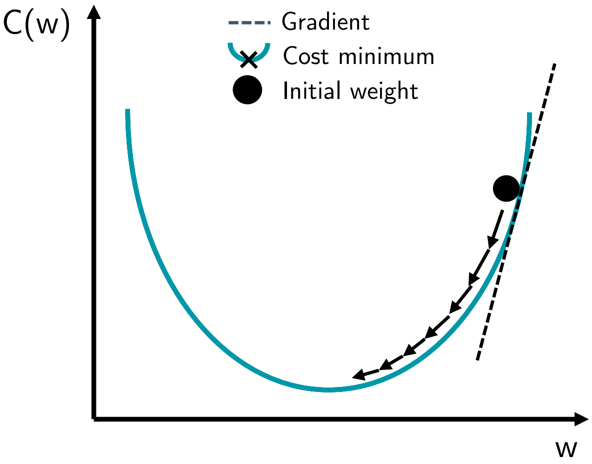
In machine learning, this is known as a cost-function[33], which captures how wrong a model's predictions are relative to the true target values for a given set of weights. The cost-function is a function of the weights in the model, i.e. $C(W)$. For most machine learning models, this cost-function is differentiable and sometimes even convex (Raschka, 2015) - which means that we can change the weights in order to minimise the cost-function - i.e. set the weights such that our model's predictions are as close to the true targets as possible. The weights can be updated by approximating the first-order derivative of the cost-function, which is known gradient descent

[33] A cost function can be specified in many different ways - but it must always capture the relationship between a model's prediction and the true target.

or stochastic gradient descent (depending on the exact implementation of the approximation) as shown in Figure 5.2 (Raschka, 2015). By updating the weights with an approximation of the first-order derivative at the current level of the weights, we will converge to some local minimum step-by-step for each weight update. How large each step is (length of arrow in Figure 5.2) is called the *learning rate*. This process is repeated many times until the calculated weight updates are sufficiently small, which indicates a convergence to a local minimum.

It should be noted that the described process is the most simplified version - most machine learning models include more steps, but the logic still holds: You allow the model to learn which weights gives the smallest error between the actual outcome, y, and the predicted outcome, $\hat{y}$, within the sample of the data measured by some cost-function.

Figure 5.2: Simple gradient descent



Note: The cost minimum can here represent both a local minimum and global minimum depending on the chosen learning rate.

Source: Inspired by Raschka (2015)

The process of minimising a cost-function by gradient descent is not necessarily that different from some maximum likelihood estimators from classical econometrics that are estimated with numerical optimisation. What does distinguish supervised machine learning methods is the focus on out-of-sample predictions and the presence of additional parameters/model specifications, called hyperparameters, which are set before the weighted are actually fitted - and the focus on systematically tuning these hyperparameters. These two aspects are described below.

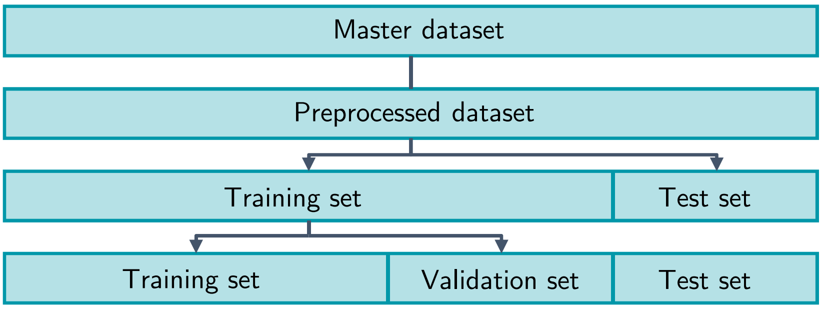
Hyperparameters

Hyperparameters, sometimes called choice parameters, are parameters or specifications of the machine learning models that must be chosen prior to actually training the model by fitting the best weights. In Figure 5.2, the size of the steps with which to update the weights is an example of hyperparameter that is chosen prior to running the model. This hyperparameter is called the *learning rate* for most machine learning models. What value the *learning rate* should take is not obvious and it cannot be trained in similar manner as the weights in Figure 5.2 - but different choices of values of the *learning rate* will affect the optimal weights and thus also the predictive performance of the model. A key aspect of machine learning models is choosing these hyperparameters for the models - sometimes called hyperparameter tuning (Raschka, 2015), which will be described in more detail below. There are numerous hyperparameters and most of them are model-specific - thus, we will return to the specific hyperparameters of our models and how we tune them in Section 6 and Appendix B.

Out-of-sample: The hold-out method and validation

One characteristic of supervised machine learning is the introduction of out-of-sample testing often referred to as the hold-out method (Raschka, 2015). The idea is to test a given model's predictive performance on unseen data. To enforce this, the data is split randomly into training set and test set as seen in the first step of Figure 5.3. The former is used for model training (i.e. fitting the weights) and the latter is used to test the overall performance of the final model by testing its performance on unseen data by using the weights and hyperparameters from the training set to make predictions on the test set.

Figure 5.3: Train, validation and test set splits



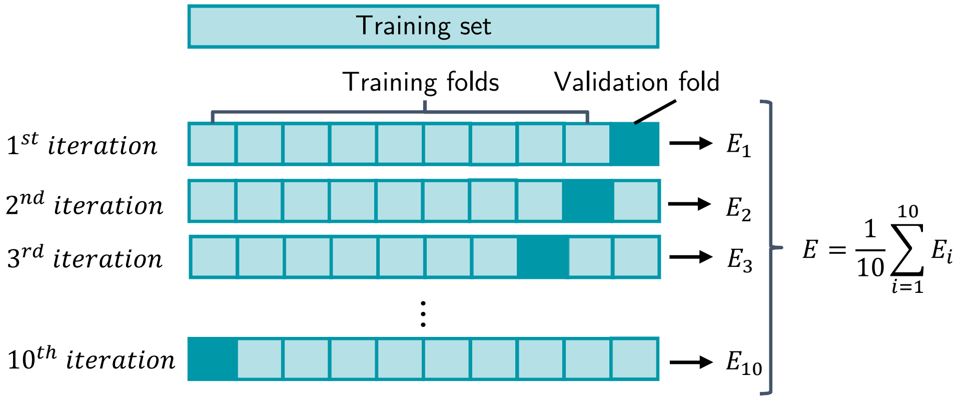
Source: Inspired by Raschka (2015)

As some of the hyperparameters of a given machine learning model can have great influence on the model's out-of-sample performance, it is also vital to tune the hyperparameters. However, as the hyperparameters are chosen as set values ahead of fitting the weights in the training set, we cannot directly tune the hyperparameter within the training set. Tuning refers to picking a set of values of the hyperparameters (a single value for each hyperparameter), train the model with these values and fit the weights, and then check the predictive performance on the test set. However, if one were to tune the hyperparameters on the test set, then you would invoke data leakage, where the test set then no longer is truly *unseen* by the model, which should only see the training set.

Therefore, it is common practice to further split the training set further into a training set and validation set as shown in Figure 5.3. With this split, you can find the optimal hyperparameters by tuning on the validation set - i.e. pick the hyperparameter values that gives the best out-of-sample prediction on the validation set. Afterwards, these hyperparameter values can then be used on entire training set to fit the final weights (including the validation set from step 3 in Figure 5.3), and the model's predictive performance can be evaluated on the test set. This process ensures a more optimal choice of hyperparameters in model and is vital to keep in mind for all predictive machine learning models.

This process of splitting the training set into both a training set and a validation set can be done multiple times by imposing what is known as *cross-validation* (Raschka, 2015). An example of this is K-fold cross-validation, which is a method to optimise hyperparameter tuning across multiple validation splits. In short, the hold-out method is repeated multiple times on a number of subsets of the training set. The actual method is illustrated in Figure 5.4. The training set is randomly split in k folds without replacement, where $k = 10$ in the example figure. $k - 1$ folds are used for model training and one is used for validation. This splitting technique is repeated k times where the fold used for validation varies for each iteration.

Figure 5.4: K-fold cross-validation



Source: Inspired by Raschka (2015)

By introducing this procedure you obtain k models with the same hyperparameters and k predictive accuracy measures, E_i, from which it is possible to construct an overall performance estimate by taking the average. The hyperparameters are then tuned to perform best on average by maximising the average predictive accuracy measures across the k validation folds. K-fold cross-validation will most of the time result in a better out-of-sample model on the test set compared to just a single validation split as the K-fold is tested out-of-sample multiple times.

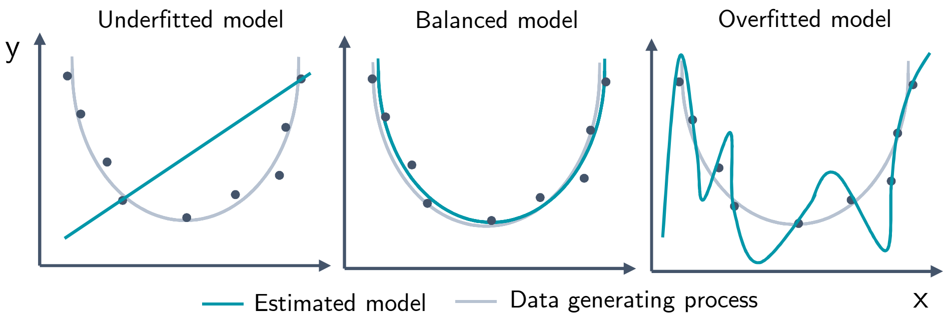
Predictive model evaluation

In general, there exists no one superior machine learning model for prediction purposes. This gives rise to the importance of methods to evaluate individual model performance. As will be explained in Section 6, one should always strive for simplicity and transparency and only include complexity if it has proven to increase model performance. Next, methods and trade-offs to consider when evaluating model performance will be explained.

The curse of overfitting

One aspect to consider in the case of model evaluations is the implication of adding more complexity to a model. As stressed above, a vital part of machine learning model for prediction is that the results are evaluated on unseen data, i.e. out-of-sample. This gives rise to a trade-off between the fit of the model in-sample and out-of-sample, usually denoted as overfitting versus underfitting (Raschka, 2015). Adding complexity to a model (e.g. more features, interaction terms between features, squared features etc.) will most likely result in a better fit on the training set because the increased complexity will capture more of the variation in the training set - but it might also capture additional noise and spurious correlations, which means that the weights of the model will perform poorly on the test set. In this case, the model is said to be overfitted. On other hand, a model can be underfitted if it does not capture enough of the variation in the features, which means that model's prediction will be too simple. In Figure 5.5, a classic example of fitting a polynomial is displayed, which shows the challenges of constructing a balanced model, which fits the underlying structure of the variable of interest without being too sensitive to random fluctuations.

Figure 5.5: Underfitting versus overfitting

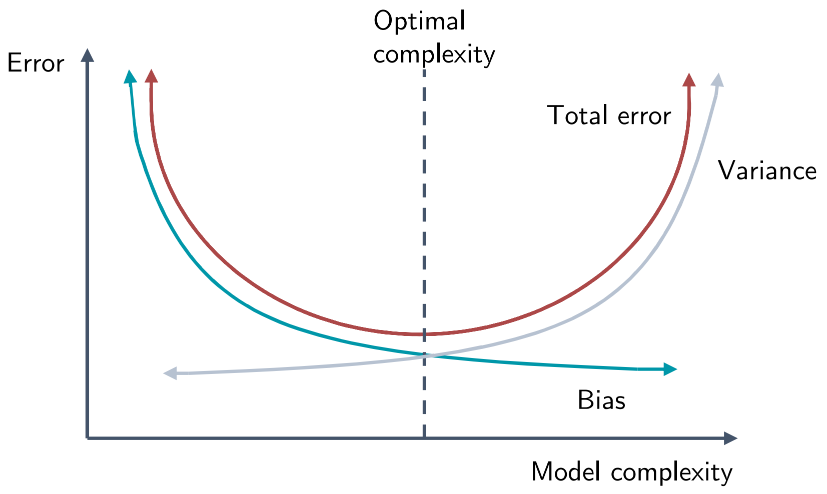


Source: Inspired by Bjerre-Nielsen (2018)

The first graph illustrates the case of a model with too low complexity - a simple linear model. The dark blue dots represent the actual data points and the grey line the underlying data generating process of the individual data points. Lastly the blue line shows the constructed model - i.e. the prediction of y based on x. By imposing this linear model of low complexity, the underlying structure of the data is not captured and the model is too simple to make sensible predictions. This clearly shows a case of underfitting. In the opposite case, where you employ a very complex model which capture all the variation in the data points, including noise, you end up in the case of overfitting. Here you also capture all noise in the training set, which again lead to poor performance out-of-sample. This is shown in the last example in Figure 5.5. The middle graph illustrates a model which is balanced with respect to capturing the underlying data generating process.

Underfitting versus overfitting a model is related to the classical *bias-variance* trade-off known from econometrics. An underfitted model will have a high bias in that it will consistently make inaccurate predictions out-of-sample. An overfitted model will have a high variance because it is too sensitive to certain data points, which means that the out-of-sample predictions will be widely spread around the true target. Ideally, we would like to a obtain model with low bias and low variance, but these two aspects often involve a trade-off.

Figure 5.6: Bias-variance trade-off



Source: Inspired by Raschka (2015)

Figure 5.6 illustrates the bias-variance trade-off in the context of model complexity. The more complex a model is, the lower the bias tends to be, but this is also associated with an increase in the variance. Complexity can be the number of features included in the model and any potential transformations to these features - such as interaction terms, squared terms etc.

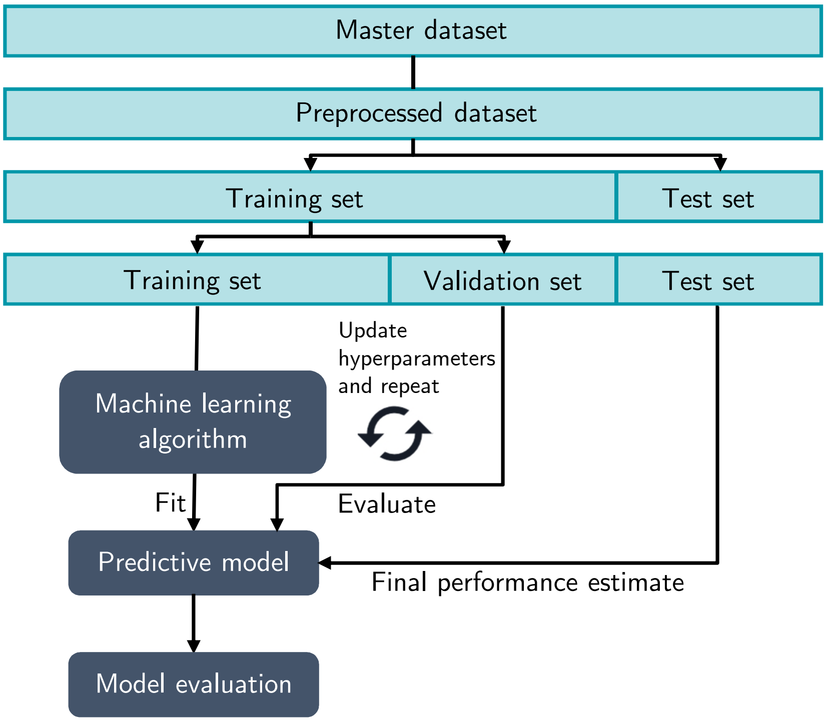
The bias-variance can be justified by returning to our example of nowcasting the unemployment rate. When nowcasting the unemployment rate, you can choose a very simple model by predicting the unemployment rate in the next period as the observed value of the unemployment rate of the last period. This will most likely result in large model bias as the underlying assumption of the unemployment rate staying constant relative to the last period is incorrect. However, the variance of the prediction will be low to the predictions not being sensitive to small disturbances in the underlying data. By increasing the complexity of the model by e.g. increasing the number of variables in the model, the bias will decrease as the underlying assumptions are more correctly specified, but this will also increase the variance in the model predictions. The optimal model complexity and error is represented by the dashed line in the centre of the figure where the both the error due to bias and variance are minimised simultaneously.

Balancing underfitting versus overfitting in order to reduce bias in models is easier said than done - especially when dealing with out-of-sample predictions. Many machine learning models have hyperparameteres specifically designed to reduce overfitting tendencies - we will return to these in more details in Section 6 and Appendix B.

5.2 Workflow of predictive machine learning

The previously described steps in machine learning can be summarised into a general workflow as shown in Figure 5.7. The specific workflow can change slightly depending on the model used, but overall, most machine learning prediction models involve the conceptual steps shown. This subsection will briefly summarise the general workflow of a standard prediction problem before moving on to explaining the extensions and corrections when introducing a time dimension in the standard prediction framework.

Figure 5.7: Machine learning workflow



Source: Inspired by Raschka (2015)

The overall workflow consists of five steps: *data preprocessing*, *data splits*, *machine learning model choice and hyperparameter tuning and training*, *the final predictive model* and lastly, *model evaluation* on the test set as shown in Figure 5.7.

Data preprocessing, which has not been explicitly mentioned so far, is one of the more important steps in machine learning. Data preprocessing include construction of variables, usually called feature engineering in machine learning, which includes how to encode categorical variables, interaction terms, handling of missing values etc. (Raschka, 2015). Preprocessing can also includes standardisation of features. We will return to the specifics for our data preprocessing approach in Section 6.1.

After the data has been preprocessed, the relevant random splits of the data are made with the hold-out approach to enable hyperparameter tuning[34] on the validation set as well as out-of-sample testing on the test set.

When the data has been split randomly, one must choose which machine learning model to implement. For many use-cases, more than one model can be trained and tuned in parallel with each model's respective hyperparameters and weights, since the input data is preprocessed in the same manner from the previous step across most models. The hyperparameters are tuned

[34] Where the hyperparameter tuning can be done with different approaches such as cross-validation with K-fold.

on the validation set.

When the hyperparameters have been tuned, the training set and validation set is merged once again, and the model is trained on the merged training with the tuned set of hyperparameters. This constitutes the final predictive model[35] in Figure 5.7.

Lastly, we evaluate the performance of the final predictive model by making predictions on the unseen test set. It is at this final step, where the actual predictive value of any machine learning model is tested as it must make predictions based on out-of-sample data. For machine learning models with a continuous target, we typically evaluate a machine learning model based on root mean squared error (RMSE) of the prediction, but there exists other scores on which to evaluate models (Raschka, 2015).

5.3 Predictive machine learning with a time dimension

The previous subsection described the general workflow of machine learning modelling - but for our purposes with a nowcasting setting, we need to take certain measures that are not usually taken with standard machine learning. Introducing an (explicit) time dimension to traditional supervised machine learning problems requires some more considered approaches to the conceptual framework. When working with a nowcasting prediction problem, the time dimension is very important in that we must avoid data leakage across time. This means that we cannot use data points in the future to train relative to test set points that are in the past. Concretely, it would be erroneous to train a nowcast model on data from 2010-2011 to make predictions about the unemployment rates of 2009. This might seem obvious - but in most applications of machine learning, the time dimension is implicitly ignored[36].

In terms of machine learning, the most common pitfall of time-related data leakage arises from the split of the data into the training and test set. If one takes the usual approach and makes a randomised train-test split of the data, you will most likely end up using some data points in the training set, which are in the future relative to the data points in the testing set. This means that you violate the underlying temporal dependence in the data.

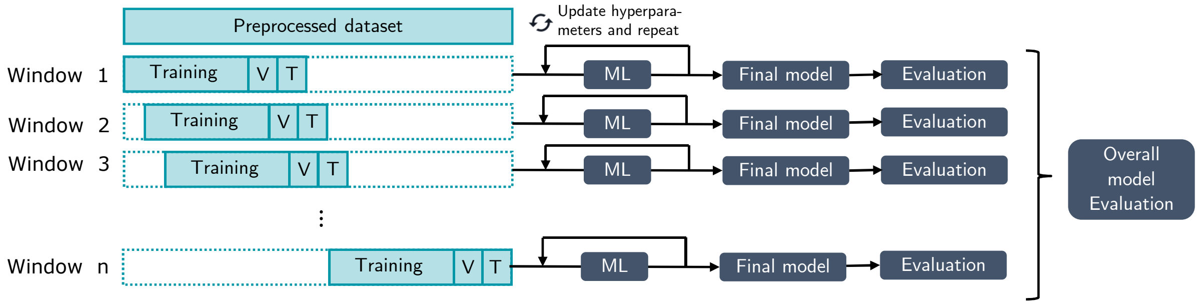
Another common pitfall is using some form of randomised cross-validation technique such as K-fold. These methods will, even if the train-test split has been done correctly with respect to the time dimension, inadvertently cause temporal data leakage because some of the training set will likely be in the future relative to the validation set. Figure 5.8 shows our general framework

[35] At this point, it is also possible to have multiple tuned machine learning models - the models are completely separate in this case and are evaluated in parallel.

[36] Usually, the time dimension is included as set of time period dummies or a single trend variable. These types of specification will still cause time dimension related data leakage if the data is split randomly into training and test set.

for nowcasting the Danish regional unemployment rates.

Figure 5.8: Machine learning workflow with time dimension



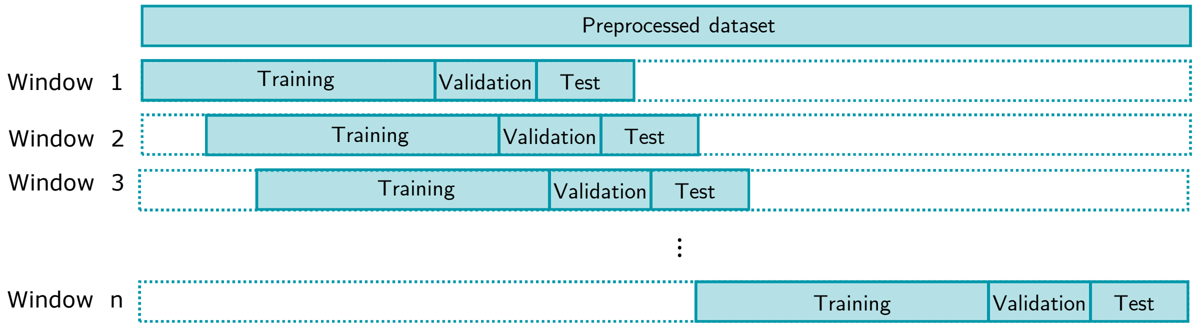
Note: The preprocessed data is split into a training set, validation set and test set represented by *Training, V* and *T* in the figure. *ML* refers to the term machine learning algorithm.

Data splits and model tuning

When splitting the data into training, validation and testing set, we need to explicitly adhere to the underlying temporal structure. We will mainly focus on one approach[37] which we will call *rolling windows* (Bergmeir and Benítez, 2012) as shown in Figure 5.9. In this approach, we will split the data explicitly according to the underlying time dimension - so it will *not* be a randomised split as described in Section 5.2. The data will be split many times into what we call *windows*. Each window will have a training, validation and test set. This aspect allows us to do nowcasts for each month in the data without having data leakage across time. The *rolling* in *rolling windows* refers to how we roll the origin point of a given window forward one month for each subsequent window while keeping the window size (i.e. number of months in a given window) constant. For our analysis, the windows will consist of 37 months of data, where 35 months will be training set, one month for validation and one month for testing. This method of splitting the data multiple times aligns with our goals of nowcasting the monthly unemployment rates - as it mimics the process of actually applying a nowcasting model in real-time, where each month you would nowcast the current unemployment rate during the publication lag as described in Section 3.4.

[37] Another approach would be *expanding windows*, where the start month is fixed and only the end month is rolled - this results in a larger and larger data set for each window.

Figure 5.9: Rolling windows



For each window, we follow the usual steps for machine learning as described in Figure 5.7 in Section 5.2. First, we tune the hyperparameters on the validation set (without any forms of cross-validation to preserve temporal structure) while fitting the weights on the training set, then we use the tuned hyperparameters on the merged training set to fit the final model weights and finally, we evaluate the model by comparing the prediction on the test set to the actual target that gives the final score for the given model in the given window. This process is then repeated for all windows in our full sample from 2007-2019.

Notable consequence of approach

Our approach to nowcasting with machine learning has some notable consequences that are important to keep in mind. First of all, there will be no single, final model with a set of hyperparameters that is applied all windows. Specifically, while we may deploy e.g. a Lasso regression machine learning model across all windows, the hyperparameters may differ across the windows as each window is treated as its own data period from which to split the data. This arises from the nowcasting approach, where it is necessary to mimic a more limited time-frame, where we cannot see the future values of the unemployment rates. As any given machine learning model must be trained across all windows in our data, this approach is quite computationally expensive when deployed to analyse years worth of data. In an applied perspective, you would only nowcast a single window at a time once every 30 days, the computational issues decrease tremendously.

6 Model overview

In the previous section, we described the overall framework of working with machine learning models in a nowcasting setting. This section describes specifics of our approach when applying machine learning models to the obtained data - as well as some of the technical aspects of how each model works. We will not delve too deeply into how each algorithms works in all of its detail - instead, we will outline the intuitive aspects of what the models do and what aspects they can account for.

6.1 Preprocessing

As mentioned in connection with Figure 5.7 in the previous section, preprocessing the input data is a prerequisite for any machine learning modelling. The process of transforming raw data into data that can be used for analysis is important and can require many decisions to be made. Many machine learning algorithms requires even more preprocessing steps than standard econometric models (Raschka, 2015) - but many of the preprocessing steps in machine learning involve less discretionary decision making[38] than those of standard econometrics.

One crucial step when modelling data for prediction with machine learning is to strive for simplicity and transparency. Simplicity refers to the argument of limiting the complexity of the models as to avoid overfitting. Additionally, one should be aware, that certain data transformation techniques can change or even remove important predictive information of the features. Thus, transformations of data should only be done if they can justified in terms of being necessary in order to run the model and/or improve predictions of a model. It is also vital that the preprocessing steps are described in full detail as any predictive model should be implementable and testable by others.

In the following, we will motivate and show the preprocessing of the raw data described in Section 4.

Construction of the target

As outlined in Section 4.1, our target is the unemployment rates across the five regions of Denmark. As in Pavlicek and Kristoufek (2015), we will transform the target into a within-region first-difference series: $\Delta UE = UE_t - UE_{t-1}$ for each region where UE is the unemployment rate. This allows us to make predictions across regions with different levels[39] of unemployment

[38] An example of this is whether or not to include interaction terms between features. In econometrics, this decision has to be made explicitly for each feature, but for some machine learning models such as Random forest and XGBoost, the algorithm itself can catch interaction terms between features without explicitly including interaction terms (Lundberg et al., 2019).

[39] See Figure 4.2 in Section 4.1.

rates with more ease and precision. The distribution of first-differences of the unemplyoment rates are shown in Figure 4.3 in Section 4.1. Predicting changes in a target is often more feasible than predicting the level of the target - and this approach still allows one to make predictions about the level of the target as one can simply add the predicted change to the lagged level of the target to get the predicted path of the target.

Construction of features

In practice, feature engineering procedures are often motivated by intuition and/or domain knowledge but often, not all proposed feature will have predictive power for prediction. To decide whether or not not include a feature, a trial-and-error motivated scheme is conducted by systematically including the new feature either independently or in certain combinations to investigate the predictive power of each new feature. As a result of this process, feature engineering for supervised learning is often time-consuming, and is also prone to bias and error as it it deeply dependent on human decision making (Dong and Liu, 2018). This is conundrum is shared across econometric and machine learning models - but many machine learning models are able to handle far more features than classical econometrics models. This significantly eases the decisions regarding inclusion of features and potential transformations in machine learning models compared to econometrics.

In order to include the raw data in our machine learning models, we have to define and encode each feature - the included features for our unemployment rate nowcasting models are shown in Table 6.1.

Table 6.1: Overview of included features

Source	Feature	Time component
Unemployment rate	Change in unemployment rate	One month lag 12 months lag
Google	Job openings Job centre Jobindex Jobnet Cash benefits Unemployment insurance Unemployment insurance fund ASE 3F KRIFA	Current period One month lag
Jobindex	Total job posts / labour force Information technology / labour force Engineering and technology / labour force Management and staff / labour force Trade and service / labour force Industry and craft / labour force Teaching / labour force Office and finance / labour force Social and health / labour force	Current period Three months lag
Controls	Population Labour force, % Higher education, % Urbanisation, % Time trend Monthly dummies (ref. January) Regional dummies (ref. Capital)	Current period

Note: The displayed Google search terms represent the label of the actual search terms. The actual search terms can be found in Appendix Table A.2. Note that some of control variables themselves are also lagged due to publication lags as described in Section 3.1 and Section 3.3.

The Google search term intensity features have been chosen and selected based on the criteria described in Section 4.2 - these features do not need any further adjustments as they are already indexed.

The job posts are transformed to job post rates, where the number of job posts is divided by the labour force for each region as described in Section 4.3. The same procedure is made for the sector job posts. The Jobindex features are included both as the current period and lagged three months, whereas the Google features are included as the current period and lagged one month.

We also include lagged interaction terms between the unemployment targets and regional dummies. This is done to allow the models to learn from the time series aspect, where the unemployment rates shows dependence over time. This also means that our baseline model's features (as described in Section 6.3) will be nested within the machine learning models.

6.2 Feature engineering

Certain additional transformations of the features are necessary for many machine learning algorithms. These transformations are usually reffered to as *feature engineering* (Raschka, 2015) and are described below.

Standardisation

Standardisation ensures that features are all measured on the same scale such that the underlying values of the raw features do not have any negative effects on the model. Standardisation ensures that a given feature will follow a standard normal distribution with mean 0 and standard deviation 1 (Raschka, 2015) by subtracting the sample mean and dividing by the sample standard deviation:

$$x' = \frac{x - \overline{x}}{\sigma} \tag{2}$$

Standardisation is a necessary for most machine learning models as many of the algorithms are sensitive to the scale of the input features - especially if they are measured on different scales. A concrete example would be to apply gradient descent to optimise the weights of a model as shown in Figure 5.2. If all weights are initialised with a standard normal distribution, but one feature is measured in USD and another features is measured in millions USD, then the weight update steps will be disproportionate between the two features. This will significantly slow the convergence process of the weights - and can at times lead to sub-optimal local minima. Since one very rarely loses anything in terms of model computation speed and/or model predictive performance, standardisation is almost always done for machine learning models.

Principal component analysis

One of the advantages of many machine learning models is that they can handle a large number of features compared to classical econometric models (Chakraborty and Joseph, 2017). But the regression-based machine learning models (as will be described below) are sensitive to highly correlated features[40]. Highly correlated features can cause both weight optimisation convergence issues and also poor out-of-sample performance for regression-based machine learning models. Principal Component Analysis (PCA) is a transformation method[41] that constructs uncorrelated features from a set of correlated features.

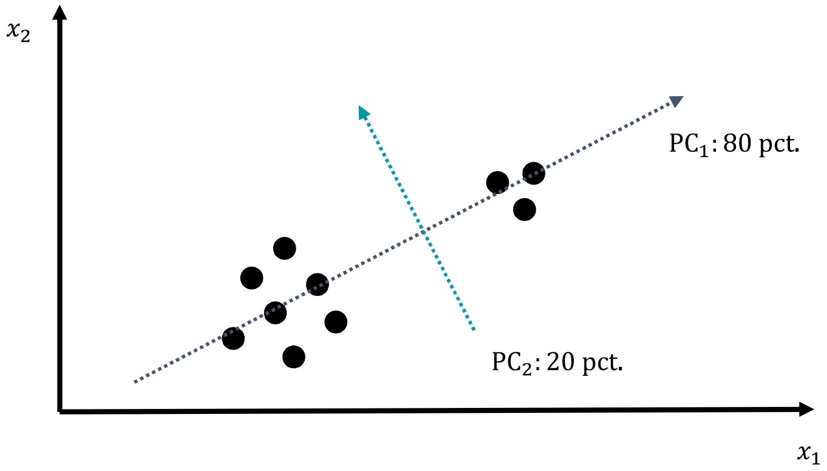
[40] The same logic applies standard regression analysis, where explanatory variables should not be too highly correlated (Wooldridge, 2018).

[41] PCA is an unsupervised machine learning method to compress a feature space into othorgonal (thereby uncorrelated) vectors.

Following Dong and Liu (2018), PCA constructs an orthogonal transformation of the features by converting a set of correlated features into a set of linearly uncorrelated features. These uncorrelated features are referred to as principal components. As a result, the number of principal components are less than or equal to the initial feature space. More formally, PCA is an orthogonal linear transformation of the data where each principal component is constructed by the argument of maximising the explained variance of the feature space (Raschka, 2015). Figure 6.1 illustrates the construction of each principal component in the feature space with two features, x_1 and x_2.

The first principal component, PC_1, will align with the direction that accounts for the largest variation in the data (80 percent). PC_2 account for the second-largest variation in the data (20 percent) PC_2 must be uncorrelated and by definition orthogonal to PC_1 as seen in the figure. For each new principal component, PC_n, it will be orthogonal to the first $n-1$ components and account for the $n^{th}-$ largest variation in the data.

Figure 6.1: Principal component analysis



PCA can be used to mitigate concerns about highly correlated features, but can also be used to reduce the number of features. This is done by only utilising a subset of the principal components as feature in the machine learning model - e.g. only using PC_1 from Figure 6.1. PCA also needs to standardisation of features much like many of other machine learning algorithms to avoid scale-related performance issues - thus, it is important to standardise the feature before constructing the principal components.

It is important to note that PCA involves certain trade-offs. It is possible that one might remove potential predictive aspects of the original feature space, which lowers the model's out-of-sample performance. How many principal components to include in the final model is also not obvious. The number of principal components can thus be considered a hyperparameter of any machine learning model, where you can then tune this aspect. In our models, we will also include the sum of the explained variance of the PC's as a hyperparameter. It is important to note that PCA is not necessary for the tree-based models - thus, sum of the explained variance is only a hyperparameter for the regression-based machine learnings models.

6.3 Regression-based models

This section describes the general concepts of the regression-based model in the analysis, including a autoregressive panel model, which will constitute the baseline model, and the regression-based machine learning models: Lasso, Ridge and Elastic Net.

6.3.1 Baseline: Autoregressive panel model

Many macroeconomic variables display a high degree of temporal dependence with persistence as past events often influence current events. Autoregressive models exploit this exact phenomenon by modelling a target variable as function of the past values of the target - thus, the baseline model will not include any of our additional features from the GT and Jobindex data.

The simplest case is referred to as an univariate time series model, where the target variable is a function of only its lagged value. The most simple example is illustrated below:

$$y_t = \beta_0 + \beta_1 y_{t-1} + \epsilon_t, \tag{3}$$

This represents a *first order autoregressive, AR(1)* model where y_t depends linearly on the most recent lag y_{t-1} (Wooldridge, 2018). This model can be estimated by OLS and will under certain conditions[42] yield a useful nowcast. An AR(1) model has been shown to be one of the better performing models when it comes to predicting unemployment rates (Tuhkuri, 2015) - especially considering how simple and easy to implement the model is. The simple AR(1) model is often chosen as the benchmark to beat in the nowcasting literature (see e.g. Son et al. (2010), Nagao et al. (2019), Tuhkuri (2015), Pavlicek and Kristoufek (2015)).

Since we are nowcasting the changes in the regional unemployment rates, we have to augment the simple AR(1) slightly to be able to nowcast the five regional unemployment rates simultaneously accounting for the cross-sectional dimension. We do this by introducing a full

[42] Certain assumptions like stationarity must be met in order to estimate the parameters and its standard errors - see Wooldridge (2018) for more details. As we are purely interested in the predictions of the baseline, we do not delve into these aspects in this thesis.

set of interaction terms with regional dummies as follows to get an autoregressive panel model:

$$\Delta y_{i,t} = \beta_0 + \sum_{i=1}^{5} \beta_i \Delta y_{i,t-1} \cdot region_i + \epsilon_{i,t} \tag{4}$$

Where $region_i$ represents a dummy for each of the Danish regions.

It is important to keep in mind that we have an out-of-sample nowcasting focus - this means that we will be estimating the baseline model for each nowcasting window and evaluate on out-of-sample data and rolling the origin forward as the described in Section 5.3.

One could wonder why our baseline model is not a linear regression, which includes all the features listed in Table 6.1. This is because such a linear regression with many highly correlated features would have very poor out-of-sample properties as it suffers from overfitting (Fornaro and Luomaranta, 2019). But there do exist certain regression-based machine learning methods that can account for this as outlined below.

6.3.2 Penalised regression models

Penalised regressions attempt to combat overfitting that stems from models with a high number of features and/or a model with highly correlated features. Penalised regressions include extra terms that allows one to regularise the coefficients/weights of a model such that they become small in absolute magnitude This can help prevent overfitting as the sensitivity to each feature will be minimised. For our purposes, we will be looking at three methods of regularisation: Ridge, Lasso and Elastic net.

To briefly reiterate the notation from Section 5: We have a target, y, and a set of features, x_i, along with the associated coefficient/weights, w_i:

$$y = XW + \epsilon = w_0 + w_1 x_1 + ... + w_m x_m$$

In the linear regression example, the associated cost-function that is minimised by gradient decent is $C(W) = \sum_{i=1}^{m}(y_i - \hat{y}_i)^2$. Regularisation methods change the cost-function such that the method of updating the weights changes.

Ridge regression

The approach of Ridge regression is to shrink the sizes of the feature weight towards 0 by augmenting the cost function with a penalisation term with the sum of the squared weights

(Raschka, 2015):

$$C_{Ridge}(W) = \sum_{i=1}^{m}(y_i - \hat{y}_i)^2 + \alpha_R \cdot \sum_{j=1}^{m} w_j^2 \tag{5}$$

Where $\alpha_R \geq 0$ is a hyperparameter of a Ridge model that determines how much to shrink the weights on the features towards zero. Large weights will be shrunk in a Ridge model, which can mitigate overfitting issues by not giving any features with too much importance, which can increase the predictive performance on out-of-sample unseen data.

Lasso regression

The approach of a Lasso regression is quite similar to that of Ridge, but instead of shrinking the coefficient towards zero, a Lasso regression will set some weights to exactly 0 for high enough regularisation (Raschka, 2015). The cost-function is augmented with the sum of absolute weights:

$$C_{Lasso}(W) = \sum_{i=1}^{m}(y_i - \hat{y}_i)^2 + \alpha_L \cdot \sum_{j=1}^{m} |w_j| \tag{6}$$

A Lasso model can still have large weights, but will completely remove some feature for a high enough $\alpha_L \geq 0$, which is a hyperparameter of the Lasso model. This can also reduce overfitting and may improve the out-of-sample predictions.

Elastic net regression

Elastic net simply combines the two methods of regularisation from Ridge and Lasso into a single cost function:

$$C_{Elastic}(W) = \sum_{i=1}^{m}(y_i - \hat{y}_i)^2 + \alpha_R \cdot \sum_{j=1}^{m} w_j^2 + \alpha_L \cdot \sum_{j=1}^{m} |w_j| \tag{7}$$

Thus, the Elastic net can limit both large weights and remove some features, which can render it more effective in combatting overfitting. The cost-function for the Elastic net can rewritten to align with our implementation:

$$C_{Elastic}(W) = \sum_{i=1}^{m}(y_i - \hat{y}_i)^2 + \alpha \cdot \sum_{j=1}^{m}[\lambda w_j^2 + (1-\lambda)|w_j|] \tag{8}$$

Where $\alpha \geq 0$ controls the degree of regularisation and λ indicates the weights between Ridge and Lasso regularisation. If e.g. $\lambda = 1$, then the Elastic net becomes equivalent to Ridge.

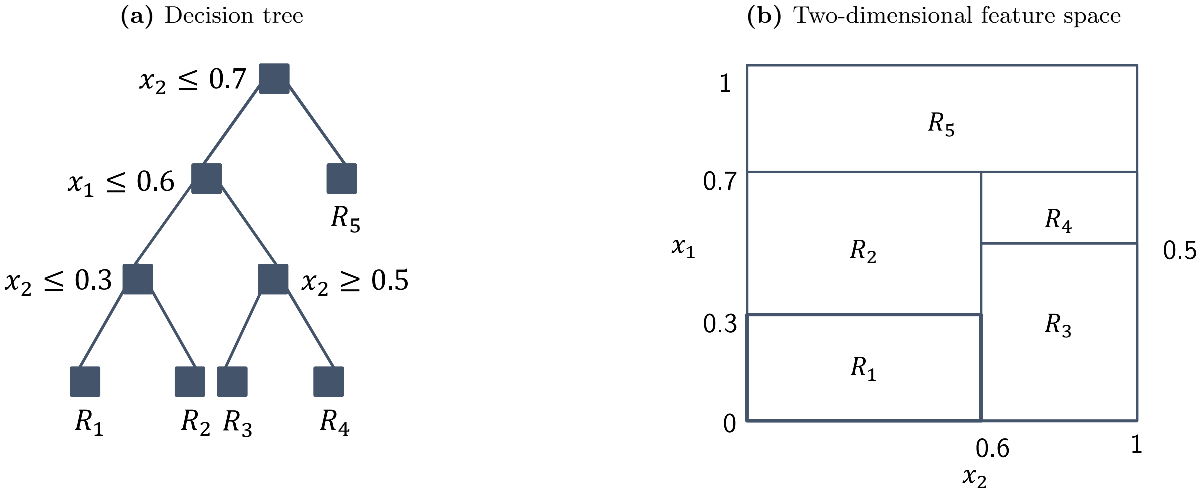
6.4 Tree-based models

A popular class of machine learning models that are not based on linear regression are the tree-based models. The tree-based machine learning technique have shown to be good at finding interaction effects in the input data, is easy to implement and have a great prediction ability when handling both unsupervised- and supervised machine learning problems. First, we will explain the general idea and intuition of a decision tree - and then expand to the tree-based methods that are applied in analysis: Random forest and XGBoost.

Decision trees

The concept of a decision tree is most easily explained by introducing a simple hypothetical example. Let us consider a problem with an continuous target, y and two features x_1 and x_2 where $x_1, x_2 \in [0:1]$. The general idea is to split the data sequentially according to values of the features, x_1, x_2, such that the conditional mean of y becomes as close as possible to the true value of y for the given splits (also called partitions) of the features. The process of sequentially splitting the feature space into multiple regions is displayed in Figure 6.2.

Figure 6.2: A decision tree and partitioning


(a) Decision tree **(b)** Two-dimensional feature space

Note: The left figure shows an example of a decision tree. The right figure shows the two-dimensional feature space corresponding to the decision tree on the right.

Source: Inspired by Hastie et al. (2009)

First, the feature space is split into two regions at what is called the root node (here $x_2 \leq 0.7$) as seen in Figure 6.2a. The first split is made such that the error between the two predicted values (The two conditional means, $\bar{y}$ for $x_2 \leq 0.7$ and $\bar{y}$ for $x_2 > 0.7$) are minimised[43]. This

[43] See Hastie et al. (2009) for more details on the exact algorithm.

process of further splitting the is continued further down the tree's branches until it is not possible to reduce the error further. The region for $x_2 \leq 0.7$ is split based on whether x_1 is greater than or less than 0.6. The region for $x_1 \leq 0.6$ is then again split at $x_2 = 0.3$ and the region for $x_1 \geq 0.6$ is split at $x_2 = 0.5$. Now all branches in the tree have reached its leaf node. The resulting process has divided the two-dimensional feature space into five different regions: $R_1, R_2, .., R_5$ as shown in Figure 6.2b. At each of the five regions, the prediction of y is the sample mean of y from the data in the region, i.e. $\bar{y}_{R_i}$. This logic extends to multiple features, which can both be discrete and continuous.

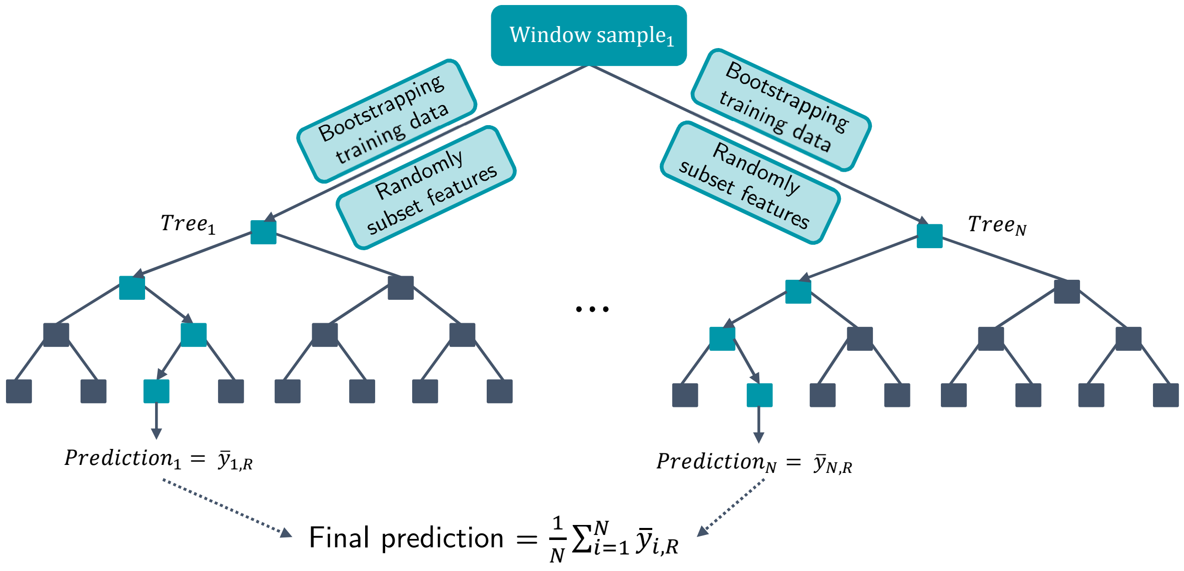
A single decision tree is characterised by being very good at fitting the features to the observed target, which means that a decision tree will be able to fit the training set very well - but a single decision tree will make very poor predictions once you apply its partitions to unseen data (i.e. test set). Thus, a single decision suffers from a very high degree of overfitting, which is why extensions to the framework are necessary in order to be able to use decision trees as the basis for predictions out-of-sample.

6.4.1 Random forest model

The idea behind the random forest algorithm is to apply many decision trees when making predictions instead of just a single tree in order to reduce the overfitting tendencies Hastie et al. (2009). The *random* in random forest stems from how the data of each tree is sampled. In order to get many different decision trees, the data must be different for each decision tree. This is achieved by bootstrapping the (training) data with replacement for each decision tree and by selecting a random subset of the features (can also be all the features). Thus, each decision tree will be trained on different data and will most likely lead to different predictions of the target.

Each decision tree works as described previously, but the final prediction are based on the mean of individual decision tree's prediction rather than just a single decision tree. The process is outlined in Figure 6.3 for a single window.

Figure 6.3: Random forest prediction for a single observation

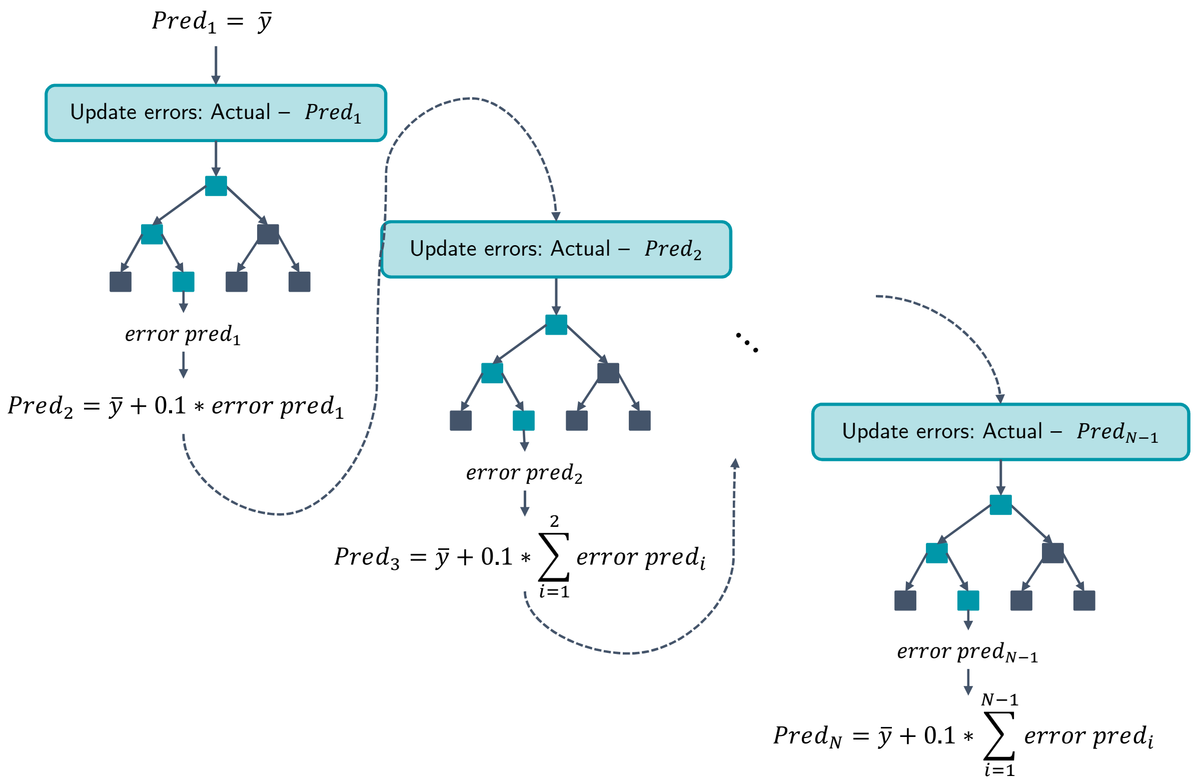

Window sample$_1$
Bootstrapping training data
Randomly subset features
Bootstrapping training data
Randomly subset features
$Tree_1$
$Tree_N$
...
$Prediction_1 = \bar{y}_{1,R}$
$Prediction_N = \bar{y}_{N,R}$
Final prediction $= \frac{1}{N}\sum_{i=1}^{N}\bar{y}_{i,R}$

A given observation may belong to a different partition for each tree, but there will be a prediction for each observation for each tree. The final prediction is then the simple average across the all the trees. The process of making different decision trees based on different features ensures a better out-of-sample prediction with lower overfitting tendencies compared to a single decision tree.

6.4.2 Extreme gradient boosting (XGBoost) model

Extreme gradient boosting (XGBoost, Chen and Guestrin (2016)) is another tree-based machine learning technique that can control for even more overfitting compared to random forest. As with random forest, XGBoost consists of multiple decision trees - but unlike random forest, the trees are not constructed independently. The idea is that the decision trees are constructed sequentially to improve upon the predictions of the previous tree, which is the *gradient boosting* aspect of XGBoost. First, you construct the simplest decision tree - without any features. This yields a prediction for all observations that is simply the mean of the target as shown in Figure 6.4.

Figure 6.4: Gradient boosting of XGBoost


$Pred_1 = \bar{y}$
Update errors: Actual − $Pred_1$
$error\ pred_1$
$Pred_2 = \bar{y} + 0.1 * error\ pred_1$
Update errors: Actual − $Pred_2$
$error\ pred_2$
$Pred_3 = \bar{y} + 0.1 * \sum_{i=1}^{2} error\ pred_i$
Update errors: Actual − $Pred_{N-1}$
$error\ pred_{N-1}$
$Pred_N = \bar{y} + 0.1 * \sum_{i=1}^{N-1} error\ pred_i$

You then calculate the errors (sometimes called psuedo residuals) for each observation. These errors then serve as the target for the next decision tree, where you then utilise all the features to attempt to predict the errors. The predicted errors from the first decision tree are then added[44] to each of the original predictions based on the mean of the target. The errors are then recalculated and moved down to a subsequent decision and so on. This process continues until the maximum number of decision trees are reached and/or until the errors do not become smaller for each decision tree. The final predictions are then the initial prediction plus the sum of all the predicted error from the individual decision trees. This process constitutes gradient boosting.

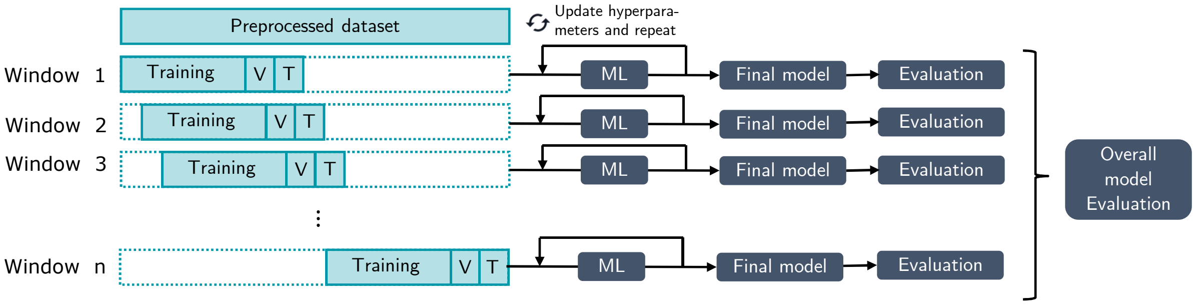
XGBoost has some additional aspects that makes this process much faster and more precise (Chen and Guestrin, 2016) - we will not go through these computational advantages. Another benefit of XGBoost is that it also includes regularisation akin to that of the regression-based models. This means that XGBoost has many hyperparameters that attempt to counter overfitting.

[44] The errors are multiplied with a learning rate that is between 0 and 1. This ensures that each corrective step of prediction will be small, which yields better convergence.

6.5 Putting it all together

Nowcasting with machine learning is somewhat complicated compared to the standard machine learning framework - in this subsection, we will briefly summarise the important aspects of Section 5 and Section 6. In Figure 6.5, we show how the data is split into different windows. Each window represents a nowcasting window consisting of 37 months of data, where the second-to-last month is reserved for validation set for hyperparameter tuning, and the last month is reserved as test set on which to make the actual nowcast prediction.

Figure 6.5: Machine learning workflow with time dimension



Note: The preprocessed data is split into a training set, validation set and test set represented by *Training, V* and *T* in the figure. *ML* refers to the term machine learning algorithm.

For each window, we train (on training set), tune[45] (on validation set) and test (on testing set) the five machine learning models outlined in Section 6. We also run the regressions for the baseline model for each window on the collapsed data of training and validation and test the predictions out-of-sample on the test set.

This results in 103 nowcasts of the regional unemployment rates from each of the five machine learning models as well as the baseline model. The nowcasts predictions are made from March 2011 up to and including September 2019.

Model evaluation and model selection

The performance of each window of the individual models are evaluated based on one main criteria, the root mean squared error ($RMSE$), which is the square root of the average of squared differences between prediction and target for each region.

$$RMSE_W = \sqrt{\frac{1}{R}\sum_{j=1}^{R}(y_j - \hat{y_j})^2} \quad \text{for} \quad W = 1, 2, ..., 103 \tag{9}$$

Where $\hat{y_j}$ is the prediction for region $j = 1, 2, 3, 4, 5$ in the given window, W, y_j is the

[45] See Appendix B for the tuning strategy for each of the machine learning models.

true change in the unemployment rate for region j in the the given window, W. $R = 5$ is the number of regions. An RMSE of 0 would mean that a model perfectly predicts the changes in the unemployment rates for all regions in a given window. A positive RMSE means that there is at least one region, where there is a prediction errors. An RMSE value can be interpreted as how wrong the model's predictions are on average across the regions. E.g. a value of 0.25 would mean that, on average, the model misses the the actual changes in the unemployment rate by 0.25 percentage points. Because the errors are squared, the RMSE does not change if a prediction is off by 0.01 or by -0.01 percentage points as it only the magnitude of the errors that is in focus. It is important to notice that each nowcasting window will have its own RMSE. Thus, when evaluating across windows this have to be taken into account.

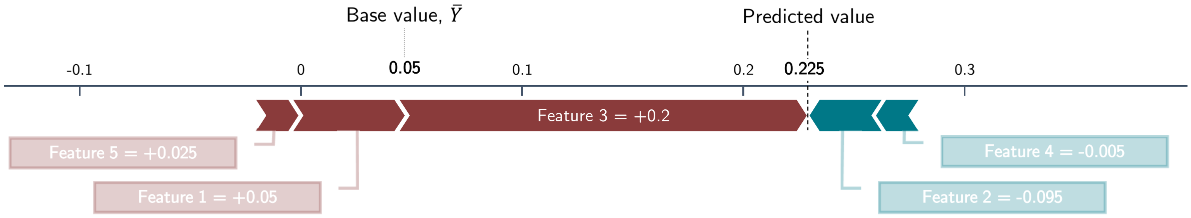
Feature importance

An aspect of machine learning that is often neglected is assessing the importance of the included features. Machine learning models are harder to interpret than traditional econometrical models and it is often tough to explain why a given prediction was made. In a nowcasting setting, we do not put too much emphasis on the feature importance as we do not prioritise explainability of the prediction - rather we prioritise the precision of predictions. Still, it is useful to quickly investigate the importance of the included features for our nowcasting model as it can point to possible ways of improving the model.

In order to do so, we will look at the *SHAP values*[46] (Lundberg et al., 2019) for the Google and Jobindex features. A SHAP value for a given feature and a given observation can be interpreted as the change in the prediction made when including the given feature relative to a baseline prediction that does not include the feature. Excluding and including all features iteratively in all combinations ensures a fair comparison of feature importance (Tseng, 2018). SHAP values are calculated for each individual prediction for each feature, which gives it a local interpretation as shown in Figure 6.6. In this example, *Feature 3* can be interpreted as the most important feature as the absolute value contributed by *Feature 3* is greater than all other SHAP values.

[46] We will not delve deeply into the theory and possibilities with the SHAP values as feature importance is not the main focus of this thesis.

Figure 6.6: Illustrated local interpretation of SHAP values



Note: The base value is 0.05 and each feature adds or subtract a certain amount relative to this base value, where the final prediction is the base value plus all feature contributions.

Source: Inspired by Lundberg et al. (2019).

A SHAP value can be both positive and negative, so in terms of feature importance, it is common to compare the mean absolute SHAP values across predictions/observations, which gives a global interpretation. A mean absolute SHAP value of e.g. 0.05 of a feature for a given window means that the given feature (in absolute terms on average across regions) adds 0.05 percentage points to the final prediction of the change in the unemployment rate.

Unlike many other feature importance measures in machine learning, SHAP values can be calculated for the test set - thus, the values can interpreted from the actual prediction and not extrapolated from the training set (Tseng, 2018).

7 Results

In this section, the overall results when nowcasting the unemployment rates across the five regions of Denmark are analysed. First, the results for the baseline model are displayed to understand the benchmark predictive performance. Secondly, we analyse the performance of the five machine learning models. Finally, we examine the results of the best-performing model in more depth before checking whether the overall results can be improved by introducing a weighted model.

In the following, all represented models are trained and tuned[47] on the same pre-processed data[48] across the same nowcasting windows. With 35 months reserved for training, one month for validation and one month for testing along with rolling windows, this yields a total of 103 nowcasting windows in total from March 2011 to September 2019.

7.1 A tough benchmark to beat

Our baseline panel autoregressive model as described in Section 6.3.1 is one of the simplest models to implement when it comes to predictions in a setting, where the temporal dimension is important. The pure time series autoregressive model is also a very powerful model and has proved to be very precise when applied to unemployment nowcasting and forecasting and it often chosen as the baseline model in the time series nowcasting literature (see e.g. Son et al. (2010), Nagao et al. (2019), Tuhkuri (2015), Pavlicek and Kristoufek (2015)).

A model with a higher degree of complexity should only be taken into consideration if it substantially improves upon the performance of the baseline model. This is important to keep in mind as machine learning models, especially for nowcasting, require a lot more coding and computational power compared to an autoregressive model, which can implemented with relative ease.

Figure 7.1 shows the results across the nowcasting windows for our baseline model for the Capital region[49]. It is important to note that Figure 7.1 is based on all 103 nowcasting windows - thus, each point in the graph represents a single nowcast prediction.

Figure 7.1a shows the predicted changes in the unemployment rate from the baseline model, where Figure 7.1b shows how these predicted changes would fit the actual path of the level of the unemployment rate. These figures shows how the autoregressive model most of the time predicts a change of very close 0 percentage points, which is equivalent to predicting that

[47] See Appendix B for exact tunning strategy for each of the five machine learning models

[48] The tree-based models do not include PCA in the feature engineering steps as it is not necessary unlike in the regression-based models unde under presence of highly correlated features (Raschka, 2015).

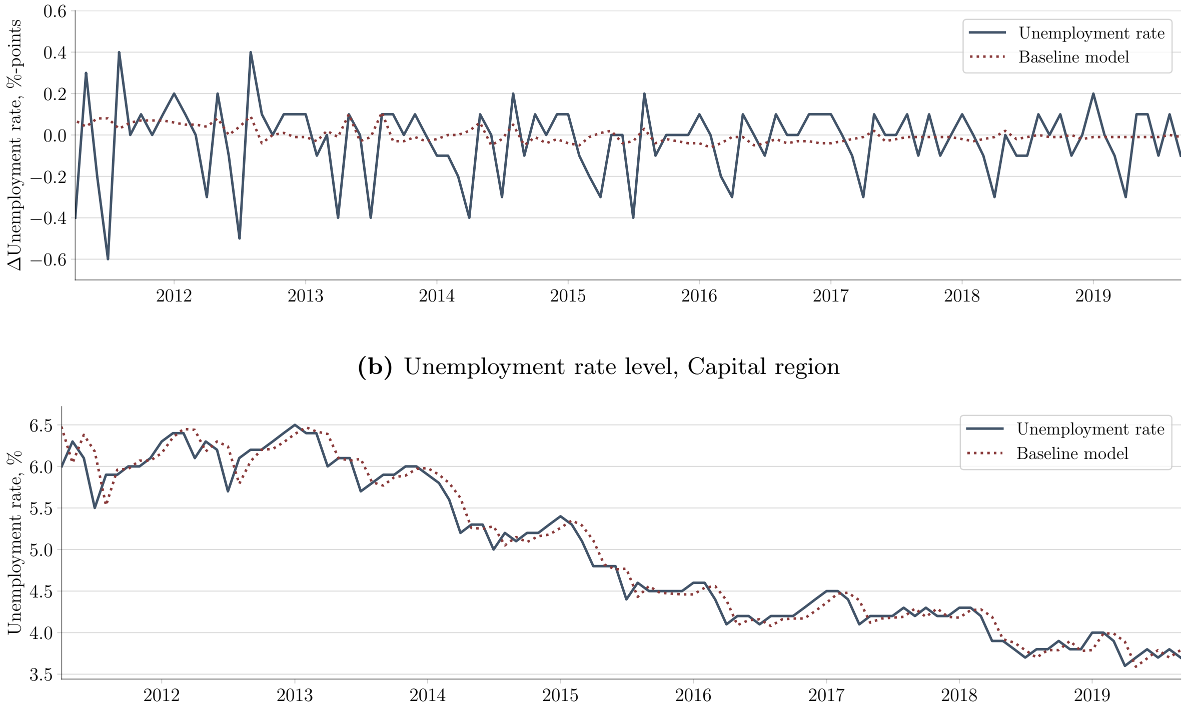
[49] The Capital region is chosen here as an example, but the same pattern and conclusions are reached for the other regions. See Figure A.2 in the Appendix

the unemployment rate will be equal to that of the previous window. Since unemployment rates in Denmark tend to be relatively stable month-to-month[50], predicting that the current unemployment is equal to or very close to the last period's unemployment actually yields a relatively a good prediction. Figure 7.1b also shows how an autoregressive model's predictions are to shift the entire series one period forward. Looking at the Capital region of Denmark, the mean absolute difference of the actual unemployment rate and the predicted unemployment rate is 0.13 percentage points and the largest difference is 0.7 percentage points, indicating that the baseline model performs quite well, but that there are also some relatively large deviation from the true target across the nowcasting windows.

Figure 7.1: Series of nowcasts, baseline model

(a) ΔUnemployment rate, Capital region

(b) Unemployment rate level, Capital region

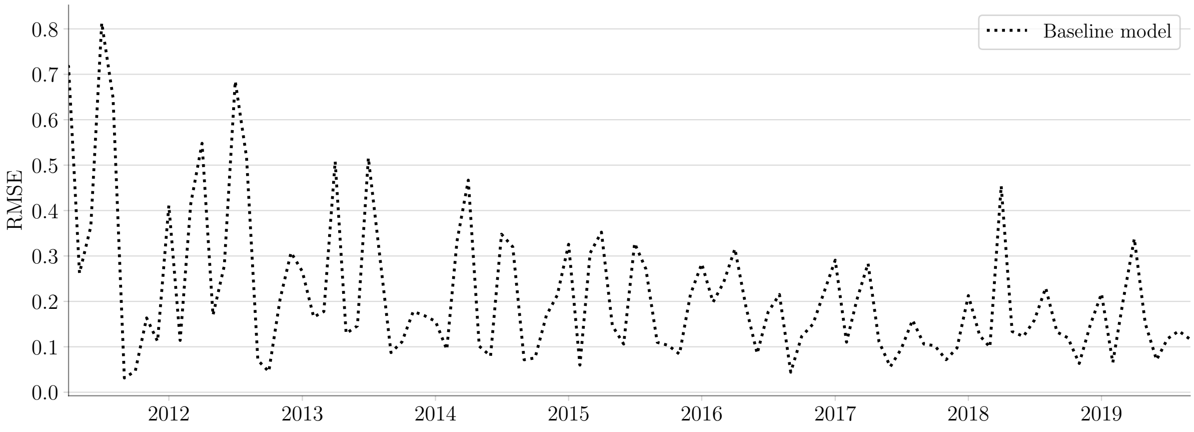


Note: The results for the four other Danish regions can be found Figure A.2 in Appendix.
Source Statistics Denmark

The next question is whether or not this is a sufficient prediction for nowcasting purposes for the five Danish regions. Figure 7.2 shows the predictive performance measured with the RMSE of the baseline model across the five regions within each nowcasting window.

[50] A change of 0 percentage points occurs with the second highest frequency - see Figure 4.3 in Section 4 for more.

Figure 7.2: RMSE across windows, baseline model



Source: Statistics Denmark, Baseline model output

The average RMSE across windows for the baseline model is 0.215 percentage points, which means that the baseline model, on average, misses the actual unemployment rates of each of the five Danish region with 0.215 percentage points across the nowcasting windows. The maximum RMSE for a single window is 0.813. We also notice that while there are certain periods with poor nowcasts, the baseline model has quite good predictive accuracy in terms of nowcasting the regional unemployment rates of Denmark.

Given the above results, it must be concluded that the baseline model is a tough benchmark to beat and might in several settings be a sufficient method to nowcast the unemployment rate.

7.2 Model performance

There are several aspects to consider when evaluating the performance machine learning models in a nowcasting setting. First of all, we are evaluating all the machine models against the baseline model - this ensures a common benchmark with which to compare predictive performance. Secondly, we have a nowcasting focus. This means that each window could be evaluated by itself as it pertains to a single nowcast prediction - but we are also interested in seeing how well any models does over multiple windows as if to mimic actual implementation. Thus, we are both focusing on the mean and the standard deviation of the RMSE across windows, but also the amount of times a given model performs better than the benchmark. By combining these aspects, we can evaluate the models across all nowcasting windows with more salience. The summary performance of the five machine learning models and the baseline model are shown in Table 7.1.

Table 7.1: Predictive performance across windows, all models

	Mean RMSE	**Std. RMSE**	**RMSE below benchmark, %**
Baseline	*0.2144*	*0.1560*	
Lasso	0.1373	0.0765	69
Elastic net	0.1358	0.0727	71
Ridge	0.1326	0.0744	73
XGBoost	0.1273	0.0685	76
Random forest	**0.1191**	**0.0652**	**81**

Source: Statistics Denmark, GT, Jobindex, Baseline-, Lasso-, Elastic net-, Ridge-, XGBoost-, Random forest model output

We see that all five machine learning models tend to be have a lower average RMSE with 0.12-0.14 percentage points compared to the baseline model's RMSE of 0.21 percentage points. All of the machine learning models have a lower standard deviation of the RMSE indicating more precision across windows. The tree-based machine learning models stand out as best performing models with a mean RMSE of 0.12-0.13 percentage points compared to 0.13-0.14 percentage points for the regression-based models.

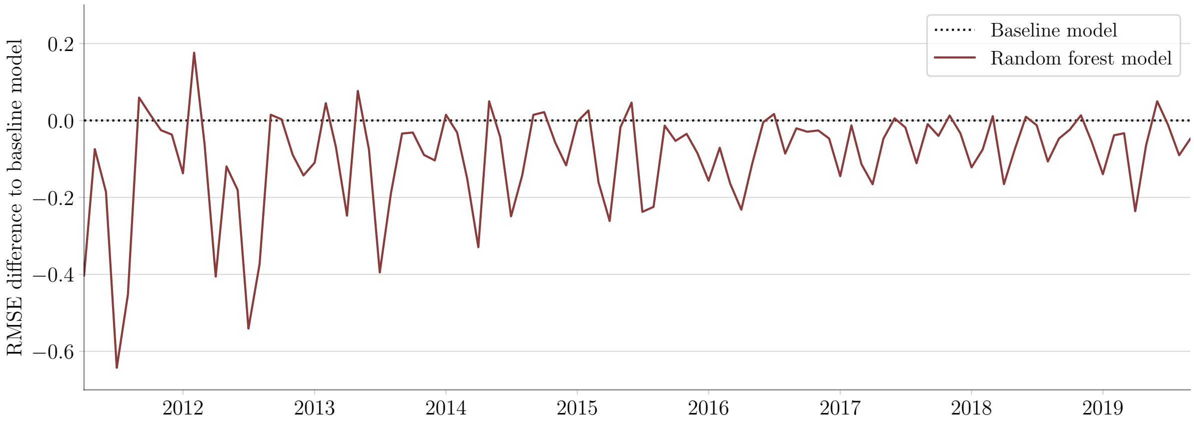
In terms of beating the baseline model, the machine learning models performs better than the baseline model around 69-81 percent of the 103 nowcasting windows. The best performing machine learning is the random forest, which has a lower RMSE than the baseline model in 81 percent of the nowcasting windows.

It must be noted that all the machine learning models perform quite similarly and that they offer, at best, a very modest improvement upon the baseline model. But it is also important to keep in mind that the baseline model itself is already quite precise in terms of nowcast predictions of the unemployment rates. There is no single standout machine learning model in terms of performance. In order to examine the results further, we choose to analyse the random forest model as it beats the baseline most frequently across the 103 nowcasting windows.

7.3 Examining the random forest model results

Figure 7.3 shows the performance of the random forest machine learning model compared to the baseline model across the nowcasting windows.

Figure 7.3: RMSE difference, baseline model vs. random forest



Note: RMSE difference to the baseline is calculated as the RMSE of the respective model subtracted from the baseline RMSE for each window - thus, a negative value indicates that the random forest model out-performed the baseline in the given window.

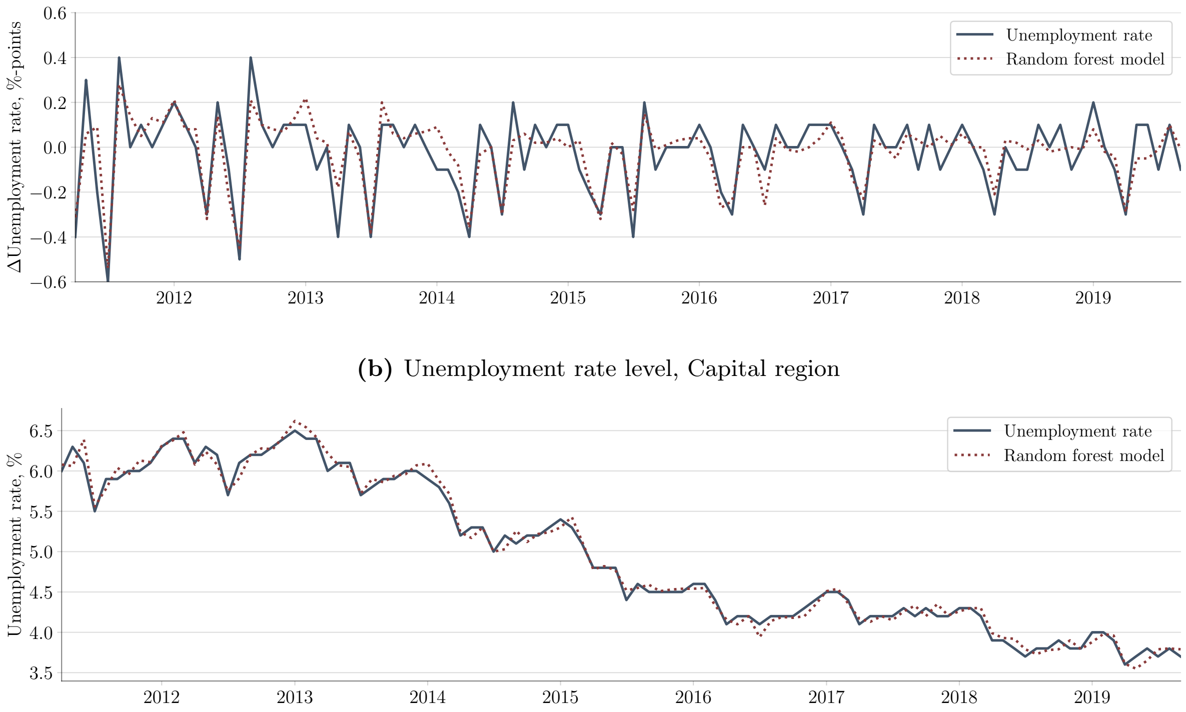
Source: Statistics Denmark, GT, Jobindex, Baseline model output, Random forest model output

Figure 7.3 shows that while the random forest model performs better for the majority of the nowcasting windows (81 percent), it does not appear to do so consistently as the performance difference fluctuates a lot. We also notice that the magnitude of the relative RMSE tend to be larger when random forest beats the baseline model compared to when it does not.

Figure 7.4 shows the predicted unemployment rate for the Capital region when applying the random forest model and compare the results to the actual unemployment rate. A comparison between Figure 7.4a with Figure 7.1a highlights the difference in the behaviour of the baseline model and the random forest model.

The baseline model makes predictions of the changes in the unemployment rates that are close 0, which is somewhat accurate, but also results in some large errors for periods with fluctuating unemployment rates. The random forest model yields predictions that are relatively far away from 0 - meaning that the random forest might be able to capture signals from the noisy Google data and Jobindex data that may indicate large changes in the unemployment rates.

Figure 7.4: Series of nowcasts, random forest model



(a) ΔUnemployment rate, Capital region

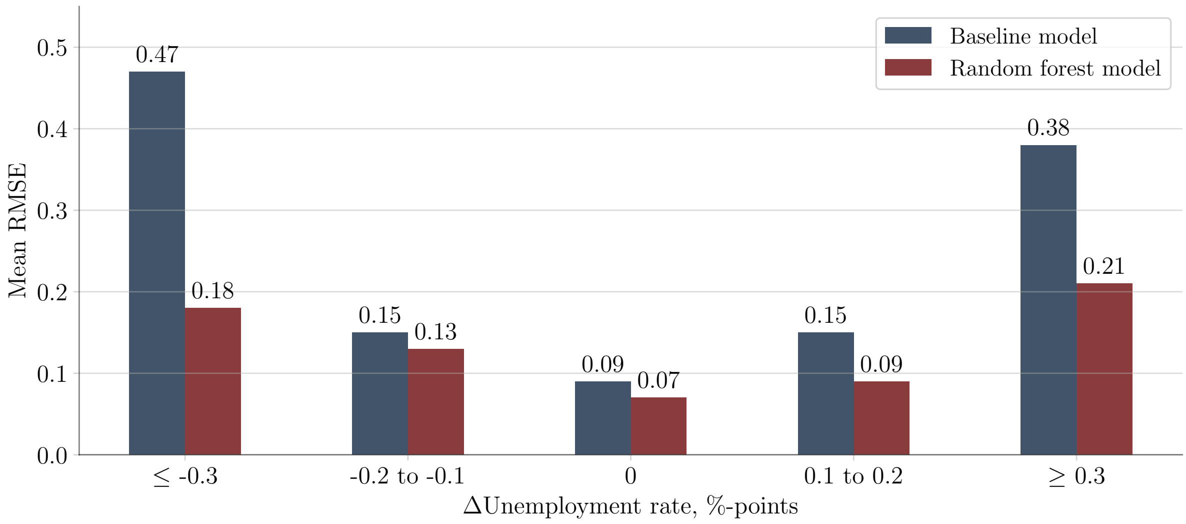
(b) Unemployment rate level, Capital region

Note The results for the four other Danish regions can be found Figure A.3 in Appendix.

Source: Statistics Denmark, GT, Jobindex

In order to investigate whether the predictive performance of the baseline and the random forest model differ across periods with relatively large changes in the actual regional unemployment rate, we calculate the mean RMSE across all predictions made for a given set of changes in the actual regional unemployment rate as shown in Figure 7.5.

Figure 7.5: Mean RMSE across actual Δunemployment rate, baseline model vs. random forest model



Source: Statistics Denmark, GT, Jobindex

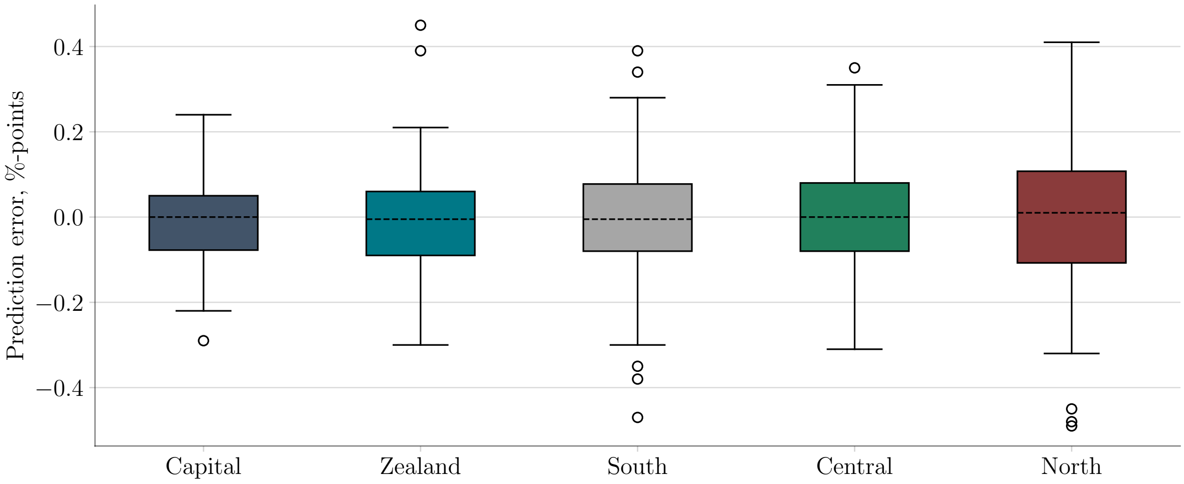
This shows the predictive performance for each model when the actual change in the unemployment rate is of high magnitude, which can be seen as a measure of how well each model performs during tough-to-predict periods. We see that the baseline model and the random forest model tends to perform similarly when the change in the unemployment rates are small - for changes in unemployment rates of -0.2 to 0.2 percentage points, the mean RMSE across the two models is similar. However, when the changes in the unemployment rates are $\geq |0.3|$, the random forest models tends to make better prediction than the baseline model. For changes in the unemployment rates $\geq |0.3|$, the mean RMSE of the baseline is 0.38-0.47 percentage points, whereas the mean RMSE is 0.18-0.21 percentage points for the random forest.

This highlights where the improvement in performance of the random forest model compared to baseline model stems from - the random forest is able to predict the high magnitude changes in the unemployment rates better than the autoregressive baseline model. This aspect is very attractive for a nowcasting model, where the discovery of high magnitude changes in the unemployment rates is more valuable to catch.

Regional errors

Figure 7.6 shows the distribution of the individual predictions error for each window across the five regions.

Figure 7.6: Distribution of prediction errors across regions, random forest model



Note: Prediction error is calculated as the true change in the unemployment rate subtracted the predicted change from the random forest model.

Source: Statistics Denmark, GT, Jobindex

We see that the median prediction error for all five regions is very close to 0 and that the distributions appear to quite similar across the five regions. The prediction errors appear to have a higher variance for North Denmark, but differences across the regions are very small. Thus, there does not appear to be any regional heterogenity in the prediction errors for the random forest model across the nowcasting windows. The same pattern holds for the baseline model.

Feature importance in the random forest model

The global interpretation of the mean absolute SHAP values for the Google and Jobindex data across all nowcasting windows are shown in Figure A.5 in the Appendix. A mean absolute SHAP value of e.g. 0.05 of a feature for a given window means that the given feature (in absolute terms on average) adds 0.05 percentage points to the final prediction. Interpreting SHAP values across many nowcasting windows is not very salient, but what is noticeable in Figure A.5 is that while there are some of the features that are consistently important in terms of the final predictions, many of the features have very low SHAP values across all windows (darkest areas of Figure A.5. There is also not a clear pattern where a feature has a very high SHAP value across all nowcasting windows.

All the SHAP values have a relatively low magnitude, usually between 0-0.075 percentage points. This indicates that while the Google and Jobindex features add some predictive value, it does not do so with consistency over time and that there might be possible to improve the features by e.g. collecting data on more search terms and/or filter out the search terms with consistently low SHAP values.

What is also noticeable is that the mean absolute SHAP value for the feature capturing the change in the unemployment rate 12 month lag[51] is by far the most important feature across almost all windows with mean absolute SHAP values around 10 times higher than the second most important feature, the GT search term *job openings.* This indicates that the most important feature for predicting the current change in the unemployment rate is the change in the unemployment for the same month from the previous year. So while the Google and Jobindex features do matter in the terms of the final predictions, their relative value is small compared to the lag of the actual target unemployment rate.

Weighted model

While all the machine learning models have somewhat similar summary performance statistics across the nowcasting windows, it may be that certain models perform better in some nowcasting windows compared to other windows - thus, it may be possible to combine all five models in a weighting scheme to get a better predictive model.

However, we cannot simply weight the results of a given window to get the best predictions without violating the principle of out-of-sample evaluation. In order to get a weighted model, we can use the information from the previous window to set the weights for the current window. Specifically, for a given window n, we find the optimal weights in window $n-1$ that yields the lowest RMSE for window $n-1$ and then we roll these weights forward to window n and apply the weights to window n's predictions[52]. This ensures that we are still using an out-of-sample approach[53]. The results are shown in Table 7.2.

Table 7.2: Summary results including a weighted model

	Mean RMSE	**Std. RMSE**	**RMSE below benchmark, %**
Baseline	*0.2144*	*0.1560*	
Lasso	0.1373	0.0765	69
Elastic net	0.1358	0.0727	71
Ridge	0.1326	0.0744	73
XGBoost	0.1273	0.0685	76
Random forest	**0.1191**	**0.0652**	**81**
Weighted	0.1184	0.0639	80

Source: Statistics Denmark, GT, Jobindex

The weighted model does not increase the predicted performance relative to the best machine learning model, random forest. This is most likely because the machine learning model's results

[51] This feature is not shown in A.5 as it would distort the colour scheme.

[52] This means that we will lose the first out of the 103 nowcasting windows.

[53] Chakraborty and Joseph (2017) uses an unweighted approach by taking the simple average of the machine learning models. This would also result in an out-of-sample approach, but would not be optimised.

are all highly correlated across the windows as shown in Figure A.4 in the Appendix. This goes against what others in the literature have found, where a weighted model resulted in modest improvements (Chakraborty and Joseph (2017), Hall (2018)). However, these papers do not weight according to our scheme with a premium on out-of-sample evaluation. In general, the weighted model is more difficult to interpret and hence is more valuable in cases where the only evaluation component is model precision and other components such as model simplicity and transparency is less critical.

8 Robustness check

In this section a number of robustness checks are conducted and the results are compared to the initial results found in Section 7. First, we explore whether the results are robust when extending the initial baseline model by including a one year lag of the target unemployment rate, which was the most important feature for the random forest model. Second, we repeat the analysis for the regions of Sweden to explore if the results for Denmark are robust across comparable countries.

8.1 Extending the baseline model

In this first robustness check, we explore whether the results are robust to an extended baseline, which includes an extra lag of the unemployment rate compared to the initial autoregressive panel model. This extra lag represent the one year lag of the unemployment rate as it was the most important feature for the predictions of the random forest model[54]. The estimated extended baseline model is stated in Equation 10:

$$\Delta y_{i,t} = \beta_0 + \sum_{i=1}^{5} \beta_i \Delta y_{i,t-1} \cdot region_i + \sum_{i=1}^{5} \delta_i \Delta y_{i,t-12} \cdot region_i + \epsilon_{i,t} \tag{10}$$

Where i refers to the region of interest and t is the time component. The extended baseline is still a rather simplistic model, which is easy to implement, but it includes more information about the previous unemployment rate for the predictions.

Table 8.1 shows the summary performance of the initial baseline, the extended baseline and Random forest model. Again, we focus on the mean and the standard deviation of the RMSE across windows, as well as the amount of times the RMSE of the random forest is below that of the extended baseline.

We see that by including the one year lag of the unemployment rate in the autoregressive panel model, the mean RMSE of the baseline model decreases with around 50 percent (from 0.2144 to 0.1056 percentage points) and the standard deviation with around 66 percent (0.1560 to 0.0509). We can clearly see how the extension to the baseline model vastly increases the predictive performance across the nowcasting windows.

[54] Tuhkuri (2015) also extends his baseline autoregressive model with a one year lag.

Table 8.1: Summary results including extended baseline

	Mean RMSE	**Std. RMSE**	**RMSE below benchmark, %**
Baseline	*0.2144*	*0.1560*	
Extended baseline	*0.1056*	*0.0509*	
Random forest	**0.1191**	**0.0652**	**42**

Note: RMSE refers to root mean squared error and std. RMSE refers to the standard deviation of the RMSE.
Source: Statistics Sweden, GT, Jobbsafari

Comparing the extended baseline and the random forest model, we see that the extended baseline also outperforms the random forest model for both the mean and standard deviation of the RMSE across windows. The random forest model only have more accurate predictions than the extended baseline in 42 percent of the windows. Given this, the extended baseline model has higher prediction accuracy than all the machine learning models.

Two particular characteristics of the input data can cause the machine learning model to perform modestly relative to the baseline models. First, the Danish data only include five regions such that only five predictions are made in each nowcasting period. Increasing the amount of targets in each period is likely to increase the performance of machine learning algorithms as it can discover more patterns in the input data. Secondly, the monthly regional unemployment rates of the Danish regions do not fluctuate much from month to month with the most frequent changes being 0 and -0.1 percentage points[55]. This again hinders the machine learning algorithm in discovering patterns in the data. In general, increasing both the number of targets in each period as well as having a fluctuating of the target variable might increase the performance of the machine learning algorithms relative to the autoregressive baseline models.

In summary, it is possible to achieve highly precise nowcasting results by applying pure autoregressive panel models when nowcasting the regional Danish unemployment rates. This is both simple to implement, transparent and constitute the most precise predictions both in terms of mean and standard deviation of RMSE. Including novel real-time data such as Google searches and online job posts and applying machine learning techniques cannot outperform the prediction accuracy of the extended baseline model when nowcasting the regional unemployment rates of Denmark.

8.2 Nowcasting the Swedish unemployment rates

In this subsection, we will investigate whether or not machine learning and novel real-time data can increase the performance of nowcasting in Sweden. In general, the labour market

[55] See Figure 4.3 for more details.

characteristics in Sweden is to a large extent similar to that of Denmark. Further, nowcasting the Swedish regional[56] unemployment rate holds at least two attractive aspects in the case of using machine learning in a nowcasting setting.

First, there are 21 regions in Sweden, which generates more validation and test set for each nowcasting window. This might increase the precision on the machine learning algorithms as the larger cross-sectional dimension enables the machine learning models to make discover more dependencies in the data. Second, the Swedish unemployment rate fluctuates more than the Danish, which also increase the possibility that our novel data sources have more predictive power in a nowcasting setting. This will be described in more depth in the data overview below.

Nowcasting of the Swedish unemployment rate will be conducted for 20 Swedish counties excluding Jämtland[57]. Even though Denmark and Sweden to a large extent are comparable with respect unemployment scheme, certain data deviations exist, which must be discussed before continuing.

In the following, the collection of the Swedish data and highlights of the similarities and differences between the Danish and Swedish data will be described, Next, the results for Sweden are displayed and compared to the results of Denmark.

8.2.1 Data overview

The Swedish data is constructed to be as similar as possible to that of Denmark. The data is retrieved from three primary data sources. The unemployment rates and the control variables are retrieved from Statistics Sweden. The Google searches are, as for the Danish data, collected using GT and the job posts are scraped from the Swedish version of Jobindex, called Jobbsafari. The detailed description of the collected Swedish data can be found in Table 8.2.

[56] Regions are referred to as *counties* on Statistics Sweden and *län* on Jobbsafari

[57] Jämtland is excluded due to insufficient Google search term data stemming from a low population

Table 8.2: Variable overview, Sweden

Variable	Source	Description	Lag	Frequency
Unemployment rate	Statistics Sweden Table AM0401VB	Share of unemployed inhabitants relative to the labour force	1.5 m.	Quarterly
Google searches	Google Trends	Google search terms which indicate labour market status**	1 d.	Quarterly
Job posts	Jobbsafari	Number of jobposts and jobposts within selected sectors*: *Information technology, Engineering technology, Industry craft* and *Office finance*	1 d.	Quarterly
Population	Statistics Sweden Table BE0101N1	Regional population	2 m.	Yearly
Labour force, %	Statistics Sweden Table AM0401VB	Share of population in labour force	1.5 m.	Quarterly
Higher education, %	Statistics Sweden Table UF0506A1	Share of population with a higher education Higher education includes: *post-graduate education, post-secondary education of 3 years or more* and *post-secondary education of less than 3 years*	4 m.	Yearly
Urbanisation, %	Statistics Sweden Table MI0810AO	Share of the population in a densely populated area or urban settlement. A hub of buildings is registered as an urban settlement if it is inhabited by at least 200 persons.	10 m.	Every 5^{th} year

Note: m. refers to months and d. to days. *Only sectors which are included in the models job posts are labelled in the table. **See Appendix Table A.4 for actual search terms

Source: Statistics Sweden, GT, Jobbsafari

There are some important distinctions between the Swedish and the Danish data. First, the unemployment rate of the Swedish regions, which constitutes the target, is measured with the labour force survey. The labour force survey is based on a stratified sample of the population, which is characterised as representative. It is conducted either online or by phone. The sample is later extrapolated to represent the full population of Sweden. The labour force survey is conducted for all European countries, which permits comparason across countries. This statistic is only available on a quarterly level, which results in only four nowcasting periods in a given year compared to twelve periods per year for the Danish model.

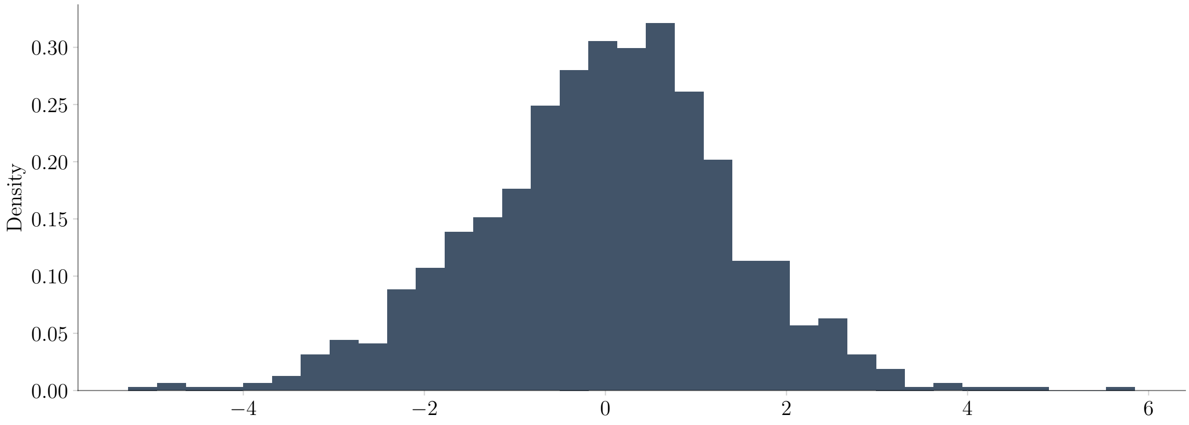
The mean level[58] of the unemployment rates for the Swedish regions is 7.54 percent with a standard deviation of 1.78 percentage points. For the Danish regions, this was 4.68 percent and 1.21 percentage points, respectively. Given this higher standard deviation, there are more potential that alternative data sources can improve nowcasting of the unemployment rate as the baseline autoregressive model per definition performs better if the target is relative stable across periods.

The distribution of the changes in the quarterly regional Swedish unemployment rate from 2007 to 2019 is displayed in Figure 8.1. The quarterly changes varies between -5.27 and 5.85 percentage points with a mean of -0.016 percentage point indicating some extreme outliers for

[58] Calculated as the simple mean across regions.

the changes of the unemployment rates for the Swedish regions. The quarterly Swedish unemployment rates shown in Figure 8.1 fluctuate far more than the monthly changes for Denmark as seen in Figure 4.3. One would expect that a quarterly unemployment rate will fluctuate more than a monthly unemployment rate in and of itself as a quarterly series contains three of the monthly series. However, the fluctuations in the Swedish labour market far exceeds that of the Danish labour market. The standard deviation of the changes in the unemployment rates of the Swedish regions is 1.41 percentage points, whereas it is only 0.3 percentage points for the Danish regions.

Figure 8.1: Histogram of changes in regional unemployment rates (%-points), 2007-2019



Source: Statistics Sweden

Second, as the target is retrieved on a quarterly level, the data for GT search terms and job posts are also retrieved on a quarterly level to secure alignment of the data[59]. The GT search terms are as far as possible constructed to mimic the Danish search terms, but some deviations exist due to different languages as well as differences between the two unemployment schemes. Both include search terms such as job sites, unemployment insurance funds and search terms indicating labour market status such as *jobs openings, unemployment insurance benefits* and job posting sites. We have narrowed down the included features in a similar manner as for the Danish data. The original list of search terms as well as the included terms in the model can be found in Appendix A.4. The included sectors are: *Information technology, Engineering technology, Industry craft* and *Office finance*[60].

Thirdly, the control variables differ slightly from the Danish data, but as the primary deviations consist of longer publishing lags, this is not a major concern, as we simply shift the data series accordingly to avoid data leakage. The average regional characteristics for the control

[59] It is possible to utilise the higher frequency of the GT and job data as will be discussed further in Section 9.
[60] See Appendix Table A.5

variables can be found in Table A.6 in the Appendix.

Again, a master data set is constructed before modelling the data and the same prepossessing steps as for the Danish data are implemented. The Swedish data is available from 2008Q1 - 2019Q3. With 11 quarters reserved for training, one quarter for validation and one quarter for testing along with rolling windows, this yields a total of 34 nowcasting windows in total from 2011Q2 to 2019Q3.

8.2.2 Results

As for the analysis of nowcasting the regional unemployment rates of Denmark, we construct the same two baseline models, which consist of an autoregressive panel model and an extended autoregressive panel model with an additional one year lag term - referred to as the baseline model and the extended baseline model, respectively. The same five machine learning models have been trained, tuned and tested with the the same model specifications as for the Danish models [61].

The summary results for Sweden is displayed in Table 8.3. As with the Danish data, the most accurate benchmark model is the extended baseline model, which have both a lower mean and standard deviation of the RMSE compared to the initial baseline. Given this result, all machine learning models are evaluated against the extended baseline model.

Table 8.3: Summary results, Sweden

	Mean RMSE	**Std. RMSE**	**RMSE below benchmark, %**
Baseline	*1.3220*	*0.4360*	
Extended baseline	*1.2296*	*0.3031*	
Ridge	1.2349	0.3536	53
Lasso	1.2144	0.3440	59
Elastic net	1.2187	0.3506	59
Random forest	1.1698	0.4509	68
XGBoost	**1.1159**	**0.3587**	**80**

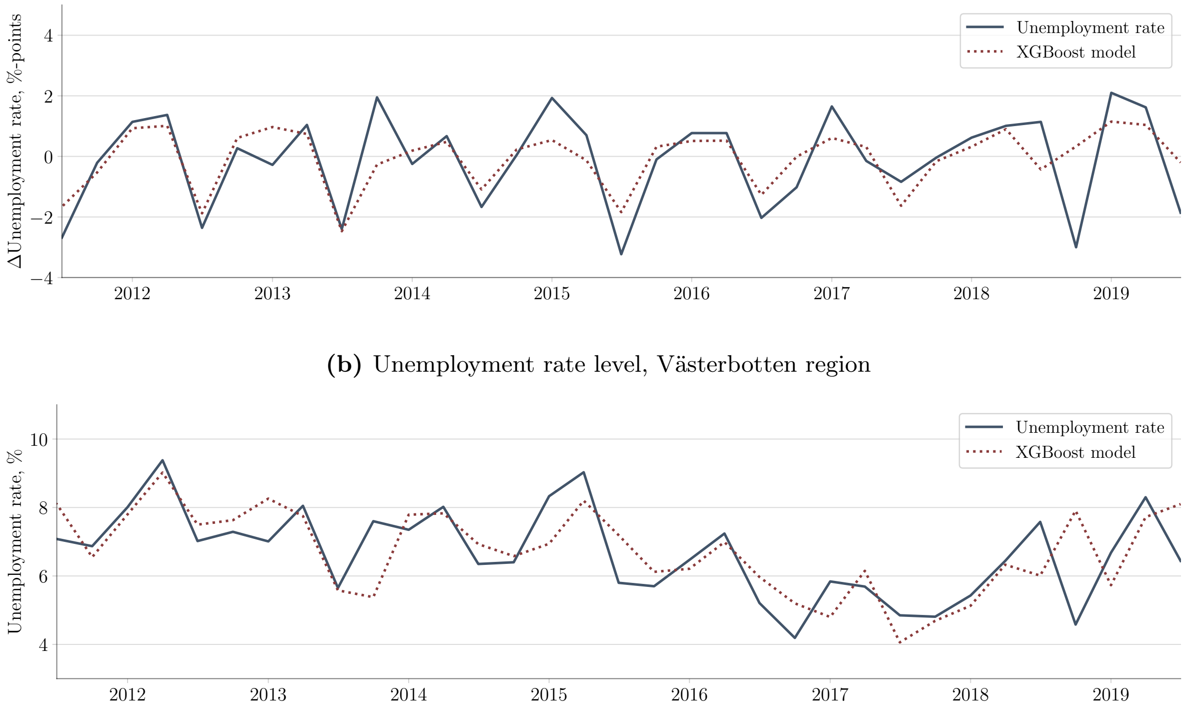
Note: The machine learning models are all evaluated relative to the extended baseline model.
Source: Statistics Sweden, GT, Jobbsafari

Looking at the mean RMSE of the machine learning models, the XGBoost model has the lowest mean RMSE across windows of 1.1159 percentage points, where it beats the benchmark model, the extended baseline model, for 80 percent of the nowcasting windows. Thus, unlike for the Danish regions, both the tree-based models for Swedish data can beat the extended baseline for the majority of nowcasting windows.

[61] See Appendix B for exact tuning strategy.

Figure 8.2 shows the predicted versus target plot for the XGboost model for the Västerbotten region[62].

Figure 8.2: Series of nowcasts, XGBoost model



(a) ΔUnemployment rate, Västerbotten region

(b) Unemployment rate level, Västerbotten region

Note The results for the 19 other Swedish regions can be found Figure A.6 in Appendix.

Source: Statistics Sweden, GT, Jobbsafari

From Figure 8.2a, we see that the XGBoost is able to capture the changes in the unemployment rates relatively well despite the large fluctuations. But there are also nowcast windows with quite poor predictions such as 2018Q4, where the predicted change in the unemployment rate is 0.08 percentage points, but the actual change is -3 percentage points. While the XGBoost appears to be able to make better predictions than the extended baseline model, it is still flawed and there may be potential for improvements. Unlike for the Danish regions, there are not found any tendency in the machine learning model making substantially better predictions in periods of large fluctuations in the target variable[63].

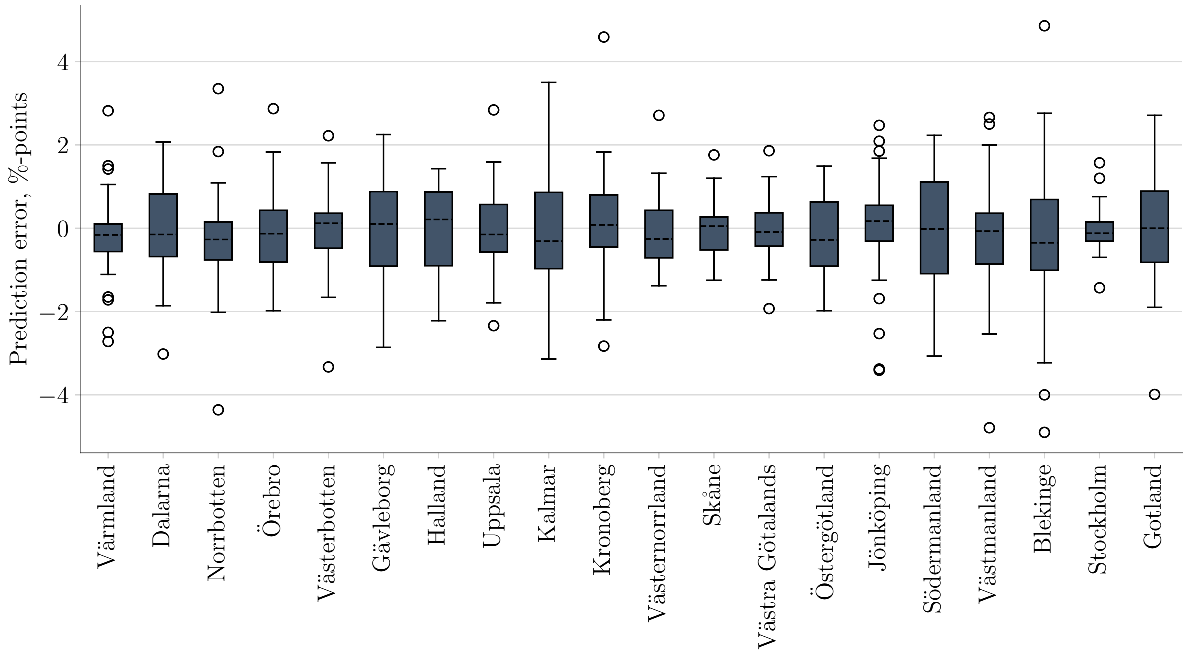
From Figure 8.3, we see that the median prediction error tend be centered around zero for most of the Swedish regions as was the case for Danish regions. However, for the Swedish regions, we do observe some regional heterogeneity in the variance of the prediction errors across regions, where e.g. Stockholm has a much smaller interquantile range compared to Södermanland. There

[62] Again, we have taken out a region to illustrate the results, but the patterns shown and discussed apply across all regions.

[63] See Appendix Figure A.8

are two possible explanations for this. First, the unemployment rate of Stockholm fluctuates far less compared to most of the smaller regions, which increases the predictive accuracy of both the baseline and machine learning models. Secondly, the quality of both the Google data and the Jobsafari data may be higher for more densely populated regions[64]. The two explanations are not mutually exclusive and both effects could explain the regional heterogeneity.

Figure 8.3: Distribution of prediction errors across regions, XGBoost model

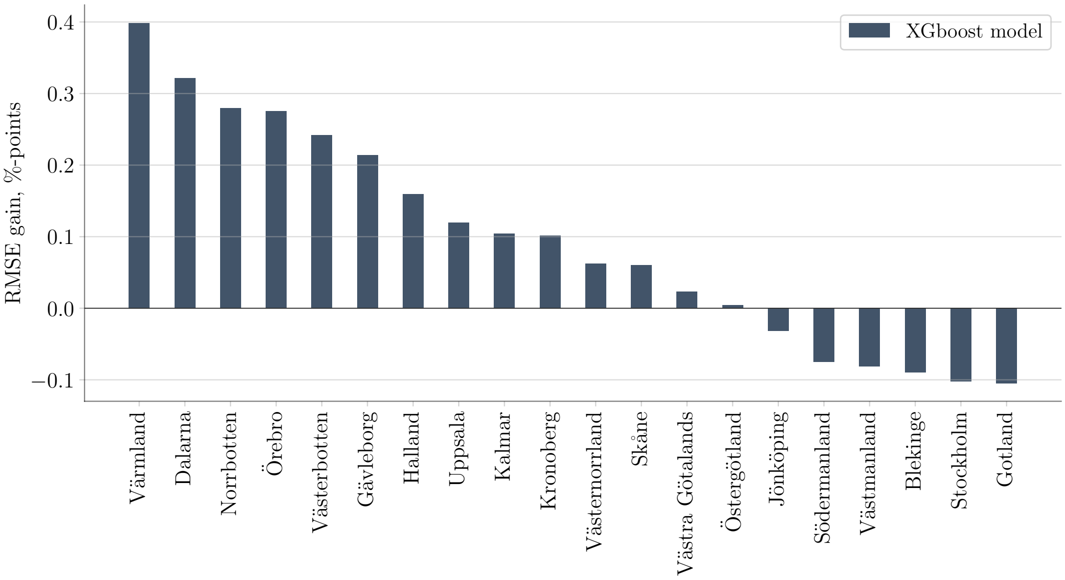


Source: Statistics Sweden, GT, Jobbsafari

Figure 8.4 shows the average gain in the RMSE across nowcasting windows for each of the Swedish regions. A positive value of e.g. 0.2 means that the mean RMSE of a given region is 0.2 percentage points lower than the mean RMSE from the extended baseline model. For 14 of the 20 regions, there a gain in the mean RMSE of the XGBoost over the baseline model. What is noticeable is that the regions, where the extended baseline model performs better than the XGBoost, are the regions with a more stable unemployment with smalle fluctuations. This shows how the effect of an unemployment rate with smaller fluctuations dominate any potential useful predictive value from Google and job data - since the extended baseline model is already quite precise.

[64] Unlike for the Danish regions, the Swedish regions vary much more in terms of population size with a factor difference upwards of 15-20.

Figure 8.4: RMSE gain across regions, extended baseline vs. XGBoost



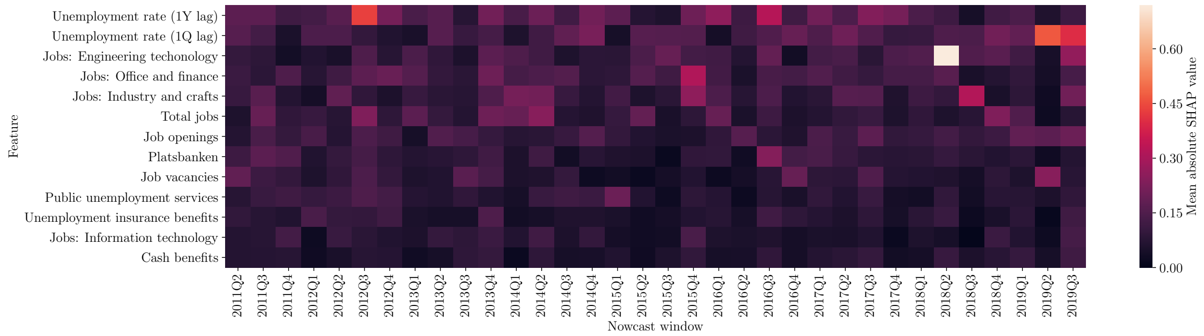
Note: A positive value means that the RMSE of XGBoost is lower on average for the given region.

Source: Statistics Sweden, GT, Jobbsafari

Feature importance

As with the Danish data, we can examine the feature importance for the Swedish data. The mean absolute SHAP values across the nowcasting windows are shown in Figure 8.5. We observe a similar pattern with the Danish data, where there is no clear patterns across the nowcasting windows. However, it appears that the job post features contribute more to the prediction that the Google variables as number 3-7 most important features are job post features. We have also included the lagged changes in the unemployment rates as features, and it is noticeable that the lagged target feature are not as important in terms of SHAP values as they were for the Danish example. This is in line with how the Danish machine learning models performed worse than the extended baseline, whereas the machine learning models outperforms the extended baseline for the Swedish data. Thus, it appears that the additional Google and job post features have relatively more predictive value in the Swedish setting as compared to the Danish setting.

Figure 8.5: Feature importance XGBoost across nowcasting windows, Sweden



Note: The mean absolute SHAP values for features that enters multiple times due to additional lag terms are summed as to keep the Figure relatively clean.

Source: Statistics Sweden, GT, Jobbsafari

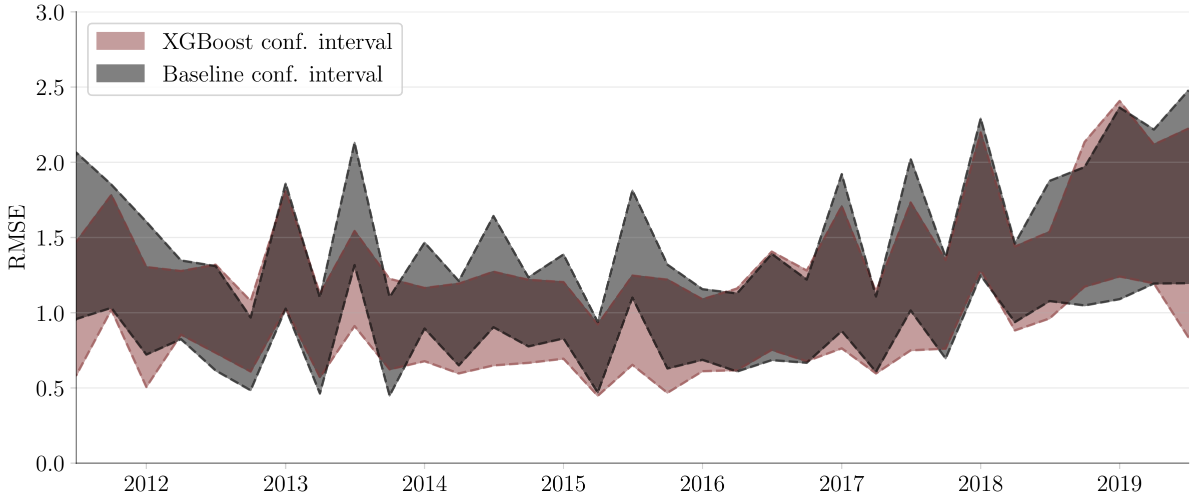
Significance of the results

Evaluating whether or not the improvements of the XGBoost over the extended baseline model are statistically significant is not straightforward. Unlike for the field of econometrics, inference for precision measures for machine learning models is very much an unexplored, or at the very least, under-utilised field. Assumptions regarding distributions, linearity and standard errors are most likely violated so many parametric approaches are usually dropped in favour of non-parametric approaches (James et al., 2013). In particular, bootstrapping is a popular non-parametric approach to estimating confidence intervals of machine learning models evaluation metrics such as the RMSE (Brownlee, 2018).

For our purposes, bootstrapping involves resampling the test set with replacement across the regional dimension for each nowcasting window. Then we will reevaluate the model with the same hyperparameters and weights on this new bootstrapped test set giving a bootstrapped RMS for a given window. Repeating this process many times, e.g. 2,500 times, for each window, it is possible to construct a bootstrapped confidence interval for both the baseline model's and the XGboost's RMSE for each window by simply looking at the percentiles of the bootstrapped RMSE's for each window.

In order to be able to bootstrap, it is necessary to have enough cross-sectional variation - thus, we cannot bootstrap the results for the Danish data, but we can do so for the Swedish data as shown in Figure 8.6.

Figure 8.6: 95% bootstrapped confidence interval, extended baseline model vs. XGBoost model



Note: 95 percent confidence interval is based on 2,500 bootstrap samples (with replacement) for each window.

Source: Statistics Sweden, GT, Jobbsafari

Figure 8.6 shows that our results are not statistically significant different when using a bootstrapped confidence interval as the interval for the extended baseline model and the XGBoost model overlap for almost all nowcasting windows. Thus, while the XGBoost has a lower RMSE for 80 percent of the nowcasting windows, it cannot be ruled out that this is a statistical anomaly. This is somewhat discouraging, but it is also noticeable that the results of the XGBoost is never significantly worse than the benchmark model - which in and of it self is already a very accurate nowcasting model. By e.g. collecting additional data on other search terms, it is possible to improve the XGBoost and thus there is the potential for the XGBoost to become significantly better than baseline model.

9 Discussion

In this section, the obtained results, methodological considerations and limitations are discussed. We initiate by discussing our findings and compare it to the literature. Secondly, we discuss our implementation of machine learning for nowcasting and then subsequently we look at the data and the potential limitations. Lastly, recommendations for further research are discussed.

9.1 Our findings and the literature

Our analysis and robustness check concludes that the combination of machine learning and novel data provides, at best, limited improvements to the nowcasting of regional unemployment rates over the benchmark autoregressive panel models, and that these improvements are contingent on certain conditions. Specifically, we find that nowcasting with machine learning shows modest improvements relative to the benchmarks for the Swedish regions, but not for the Danish regions. However, the improvements for the Swedish regions appear to be statistically insignificant when examining the significance with bootstrapped confidence intervals.

Comparing our obtained results to those of the current literature is not straightforward. Nowcasting macroeconomic variables is usually done with pure time series econometrics for country-level variables, so our results are not directly comparable with many results of the literature as we are nowcasting on a regional level with a cross-sectional dimension. Current contributions to this subject primarily constitutes of augmenting pure time series autoregressive models with the inclusion of GT search terms, and then predicting macroeconomic variables on an aggregate level for different countries (see e.g. Askitas and Zimmermann (2009), Hall (2018), Tuhkuri (2015), D'Amuri and Marcucci (2017), Nagao et al. (2019) and Chakraborty and Joseph (2017)).

To the best of our knowledge, Hall (2018) and Katris (2019) are some of the only studies, who have applied machine learning techniques to predict the unemployment rate - here in a forecasting setting on a national level. They find that the machine learning models improve forecasts relative to simple time series benchmark models. Fornaro and Luomaranta (2019) apply machine learning to nowcast GDP in Finland using traffic measurements and Pratap and Sengupta (2019) forecast consumer prices and inflation in India. Makridakis et al. (2018) finds that machine learning models performs poorly relative to time series benchmark models when examining the performance rigorously across many different time series.

Thus, machine learning is being applied more and more to macroeconomic forecasting issues, but the results are ambigious and not consistent across settings and articles. However, to the best of our knowledge, no study has examined the nowcasting of regional unemployment rates with machine learning algorithms in similar manner to what we have done.

The contribution of this thesis to the topic of nowcasting macroeconomic variables is to explore the potential of using machine learning in combination with alternative data sources to identify whether an already accurate benchmark can be improved. We also include a cross-sectional dimension to our analysis in the form of regions unlike most studies. As noted in Tuhkuri (2015), inclusion of a cross-sectional dimension has potential to improve nowcast that utilises Google data. Thus, it is important to keep the cross-sectional aspect of our analysis in mind when comparing the results to that of the literature, which usually deals with macroeconomic variables at the national level in a time series context.

The obtained improvement, though insignificant, in prediction accuracy of the Swedish regions is still only modest compared to the simple autoregressive panel baseline models. This result is in line with literature that also utilises Google searches for nowcasting unemployment rates in the US (Tuhkuri, 2015). If there is an improvement, it is very small and the vast majority of the nowcasting prediction precision stems from time series aspect of the models - and not the addition of the Google data. This is driven by the already high accuracy by applying basic time series models such as an autoregressive model (Chakraborty and Joseph (2017) and Coulombe et al. (2019)).

Given these aspects of the time series, where the autoregressive model well in stable environments, the additional precision of including alternative data sources in a nowcasting setting can be expected/hoped to be largest around periods with large fluctuations such as in the financial crisis as seen in Tuhkuri (2015). This relationship was found in our analysis when comparing the results of the random forest to the initial baseline model on the Danish data, but not supported when extending the baseline model nor in the Swedish setting. This is somewhat discouraging, but as the data for neither Denmark nor Sweden includes the financial crisis (the nowcasting period covers 2011-2019), it is not possible to write off this expected link, where predictions from machine learning models with novel data are better under periods of economic downturn/upturn. Ideally, we would have conducted our analysis over a time period that covered both a boom and a recession, but this was unfortunately not possible due to data availability limitations. It is reasonable to postulate that greatest potential in applying alternative data with machine learning for nowcasting is early detection of large changes in the unemployment rates.

The overall implications of our results and the indications from the literature is that in many settings, an autoregressive based approach to nowcasting is sufficient to obtain precise nowcast predictions. This is also important to keep in mind for future research with nowcasting with machine learning, where the models must be evaluated fairly and clearly relative to an econometrics based nowcasting model as stressed in this thesis as well as Makridakis et al. (2018).

9.2 Methodological considerations and limitations

In this section the methodological considerations and limitations of applying machine learning in a nowcasting setting are discussed. The improvement found in both the Danish and Swedish nowcast models when applying machine learning were modest, at best, and not robust.

We did find that the nowcast predictions from our model framework worked better for the Swedish regions compared to the Danish regions. There are several, non-exclusive possible explanations for this. First, the increased number of regions for the Swedish data relative to the Danish data means that the number of targets in each nowcasting period increases from 5 to 20. This can affect the performance of the individual machine learning models as they can capture patterns with higher precision in the input data as well as identify noise. Second, as the regional unemployment rates of Sweden fluctuate more than those of Danish regions, again the potential of applying machine learning is higher. In such periods, an autoregressive model will perform poorly. Lastly, it may be that the input data (Google search terms and the job posts) have better predictive value for Sweden as compared to Denmark.

To summarise, we find indications towards that machine learning is more relevant for macroeconomic nowcasting if there exists a poor baseline (poor benchmark is usually associated with large fluctuations) and in settings, where there are potential, alternative data sources (usually available almost real-time and of higher frequency). Thus, for the current data definitions and availability, nowcasting monthly unemployment rates for Denmark is most likely not improvable with machine learning. To obtain better nowcasting results for some form of nowcasting in Denmark, it is required to change the target variable to another macroeconomic variable and/or nowcast a variable at a more granular target level such as the unemployment rate at a municipality level. However at this point, the Google search terms from Google trends are only available on a regional level and not at a municipality level - thus, this extension was not possible for this thesis.

Machine learning, in general, is rather complex to implement compared to an autoregressive panel model. This should be taking into account before concluding on the need for machine learning models nowcasting. The complexity arises from the great number of hyperparameters that exist for most algorithms, which can yield a very large hyperparameter space for model tuning - which in turn also slows the process of tuning a given machine learning model. On top of this, the time dimension in this setting requires multiple models to be calculated for each window before evaluating the overall performance.

One force of the machine learning workflow as displayed in Figure 5.7 is that the selected model is tested out-of-sample to evaluate the performance. This strong evaluation step is crucial for determining the overall performance and robustness of the results and should always be included the forecasting and nowcasting purposes. Among current literature, out-of-sample is not always stressed and tested (D'Amuri and Marcucci, 2017). To test the model performance out-of-sample, the historical data is carefully subset to constitute a training set, validation set and test set. As stated in Hall (2018), no general rule for subsetting the data in these sets exist as the optimal ratio of splitting depends on the signal-to-noise ratio in the data, and the amount of data it self. As we wish to nowcast the unemployment rates, the validation set and test set only constitute one period. This has limited our tuning process for the hyperparameters, as we have not been able to do any form of cross-validation[65] without introducing data leakage. However, one possibility, which we have not explored, is to do bootstrapping of the validation set across the cross-sectional dimension in order to fine-tune the hyperparameters. This might improve the out-of-sample precision of the machine learning models (Kim, 2009). Computationally, this has not been feasible in our coding setup as bootstrapping of each window's validation set along with hyperparameter tuning would simply take too long when we are looking the entire period of analysis. However, in an applied nowcast setting, where only a single windows is trained and tuned at a time, bootstrapping the validation set in order to fine-tune hyperparameters is certainly a worthwhile option.

To obtain the individual nowcast a rolling window was employed such that test prediction was constructed after training the model on three years of historical data. This approach using a rolling window is also present in Hall (2018) and D'Amuri and Marcucci (2017). Though one could test the result of using all historical data to nowcast a given period instead of only using the three most recent years. This is referred to as expanding window. The resulting nowcasting model is not a finished product, but must be maintained and checked each month to include the most current information in the following period of nowcasting. This is the case no matter what nowcasting model is constructed as long as it constitutes either a rolling or expanding window this only one period of nowcasting.

9.3 Novel real-time data as features

Utilising alternative, novel data sources stemming from online sources has both a large potential, but also certain limitaions when applied to a nowcasting setting.

[65] See Section 5 for more details

In general, the rise of the internet has lead to vastly more available, potential data sources. One method of retrieving this data is by web scraping. This technique holds certain drawbacks as stated in Section 3.2 including limitations of securing privacy rights. In an applied nowcast setting, one should get an agreement with the owner of the data in order to obtain the data and not simply web scrape it every month as done in this thesis. One such agreement could constitute of an access to an API, which allows the user to fetch the data more reasonably or simply data dumps from the data provider. Such an agreement can secure that the interest of both the data user and the data provider are aligned. It is also possible to get data that is cleaned and/or processed in a manner such that the data is better suited for the purpose at hand.

Looking more specifically at the Google searches, the data cannot be expected to be a random sample or representative of the population labour force. This is the case even though the Google data is extensive as Internet use in general are correlated with income level and place of living looking at the period of interest (Statistics Denmark (2019b)). These issues can lead to less accurate results for the population of the labour force within each region. Though, in recent years this tendency has shown to be decreasing across groups as the internet searches become more and more prevalent, which improves the general data quality of using Google searches in research.

The included Google search terms are not a exhaustive list of possible searches but driven by relevant terms form current literature (Askitas and Zimmermann (2009), D'Amuri and Marcucci (2017) and Nagao et al. (2019)), which has been applied in nowcasting of unemployment rates in other countries. Further, we included additional search terms to capture the institutional features of the Danish unemployment system. This last process has mainly been hypothesis driven and each feature has been strongly motivated before being included. As the included Google search terms are an indicator of unemployment, it is crucial that the individual searches are made by the relevant individual from the labour force. This cannot be guaranteed in the GT data, so it is important to keep this in mind.

One feature of the GT is that it gives suggestions for other relevant topics to search upon, which can give rise to noise in the data, where Google implicitly affects the search term intensities. There does not exist a method to control for such noise why this must be kept in mind when interpreting the results.

One last concern regarding the Google search terms is that the algorithm to collecting and analysing the individual searches by the population keeps on updating as Google the as mentioned in Tuhkuri (2015). This may affect the reproducibility of the results found in this thesis and, in general, affect the confidence that the results can be expected to hold in the future.

The features reflecting job opening scraped from Jobindex also contain some limitations. There is not controlled for duplicates in the retrieved data, which may result in some job openings counting multiple times in different sectors. Furthermore, it has not been possible to account for why a job post is removed from the job posting site. Here, we are only interested in job post only being removed due to the fact that the position has been occupied by an unemployed individual as it is this aspect that actually affects the target of the unemployment rate. To our knowledge, no current literature has explored the use of including job posts in a nowcasting setting. As the results from for Denmark and Sweden are ambiguous, the predictive power of including this information should be explored more thoroughly before making any conclusions about its predictive power.

9.4 Recommendations for future research

The results from the analysis shows the, at best, modest improvement to nowcasting when including both novel real-time data and machine learning. However, as the models and the data included in this thesis are only preliminary prototype models, it shows that there might also be a potential for nowcasting with machine learning. There is a need for further research to investigate in which macroeconomic contexts nowcasting with machine learning is more relevant.

First, it is likely that other real time data sources, which identify the individual labour market status, could improve the results further. This could e.g. include publicly available statistics from the web by including the individuals actions on social media, and identify posts, which indicate labour market status. Other potential sources could be to include traffic data as in Fornaro and Luomaranta (2019). As a large portion of the employed travel by car to work, which potentially renders this data source as an indicator of labour market status.

Secondly, the result from previous literature such as Tuhkuri (2015) suggest that the relative performance of applying alternative data sources is higher in periods with large fluctuations (i.e. a boom or bust period) when examining the US. This should be further investigated before making final conclusions about whether or not it is also the case for both Denmark and Sweden.

Thirdly, there exists many more machine learning algorithms than those explored in this thesis. It may be that another machine learning algorithm is more suited to nowcasting purposes. It is also a possibility to change the entire nowcasting setting to a machine learning classification problem, where instead of making exact unemployment rate predictions, the nowcasting model would instead make predictions as classifications - e.g. *large decrease*, *no change*, *large increase.* This might be more useful if one cares more about the change in outlook rather the exact predicted unemployment rate.

Lastly, this thesis only analyse one macroeconomic variable, the unemployment rate. As described previously, nowcasting the unemployment rate with an autoregressive based approach is already quite accurate, but this does not necessarily apply to all macroeconomic variables. Another possibility could e.g. be to nowcast household consumption patterns at the municipal level, where the are also a vast potential of alternative data sources, but where the target itself might be exhibit more fluctuations and instability. These alternative data sources include social media data, transaction data, sales data from large firms and so on.

Gil et al. (2018) nowcasts the quarterly private consumption of Spain by applying monthly soft indicators, payment card transaction data as well as GT data in a time series based model - it could be possible to extend this analysis with a focus on a more granular level data such as the municipal level (if data permit). Machine learning offers an opportunity to incorporate many different variables in a complex prediction setting with a large cross-sectional dimension, so there is a great potential to investigate these opportunities to get better nowcast prediction. It is also possible to nowcast and/or measure variables that have previously been harder to measure such as the mood of a nation (Lansdall-Welfare et al., 2012).

10 Conclusion

This thesis analyses whether or not novel real-time data, such as Google searches and online job posts, combined with a machine learning model framework can improve nowcasting of regional unemployment rates of Denmark and Sweden, respectively.

A successful result would have several implications. First, constructing an accurate nowcast is crucial for correctly evaluating the current state of the economy. From this, correct actions can be taken from decision makers in the economy. Second, being able to utilise alternative data sources in a nowcasting setting to measure fluctuations in macroeconomic variables, which has previously been hard to measure is a strong result in itself. It illustrates the potential of including such data sources in other nowcasting settings.

For the Danish regions, we find that the imposed model framework produce more accurate results than the baseline model, which constitutes an autoregressive panel model. This results is highly concentrated to periods with larger fluctuations in the actual unemployment rate.

The results are not robust when extending the baseline model to include an additional one year lag term as seen in Tuhkuri (2015). With this extension, the model outperforms the preferred machine learning model for the Danish regions. To investigate whether these discouraging results for Denmark is generated due to small fluctuations in the target variable, and/or too few observations in each nowcasting period, the analysis is also performed for the 20 Regions of Sweden. The results indicate that machine learning and real-time data can outperform the extended baseline model in 80 percent of the nowcasting windows. Though, the difference in accuracy between the two models is not significant when constructing a bootstrapped confidence interval.

The overall results indicate that there is a potential for using novel real-time data and machine learning to improve nowcasting of the unemployment rates - and also other macroeconomic variables in general. However, the potential is contingent on the the presence of certain characteristics of the data. First, the target of interest should have sufficient cross-sectional variation such that there are enough targets for each nowcasting window. What constitutes *enough* is a much tougher issue to figure out. Secondly, the target itself should exhibit characteristics that renders pure time series modelling difficult. This could be a series that exhibits high degree of fluctuations and/or unstable variance in the data. Thirdly, one needs alternative data sources to feed into any potential machine learning algorithm. The advantages of machine learning models (especially the tree-based) is that it is much easier use many features and finding non-linear and interaction effect of the features.

Thus, the areas of macroeconomic nowcasting with the most potential for machine learning to improve is most likely not series such as the unemployment rates as these series exhibit characteristics that renders time series nowcasting very accurate. Future nowcasting research that utilise machine learning in combination with novel data sources must take this into consideration - as it is important that any nowcasting model is evaluated relative to the, at times, well-performing and easy-to-implement autoregressive nowcasting models.

References

99firms (2019). 2019's Google Search Statistics. https://99firms.com/blog/google-search-statistics/. Accessed on November 22, 2019.

Askitas, N. and Zimmermann, K. F. (2009). Google Econometrics and Unemployment Forecasting. *Applied Economics Quarterly; Berlin*, 55(2):107–120.

Athey, S. and Imbens, G. W. (2019). Machine Learning Methods Economists Should Know About. *Annual Review of Economics*, 11.

Bergmeir, C. and Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191:192–213.

Bjerre-Nielsen, A. (2018). *Supervised learning, part 1, Social Data Science, Lecture*. University of Copenhagen.

Bok, B., Caratelli, D., Giannone, D., Sbordone, A. M., and Tambalotti, A. (2018). Macroeconomic Nowcasting and Forecasting with Big Data. *Annual Review of Economics*, 10(1):615–643.

Borger (2019). Kontanthjælp - hvis du er 30 år eller derover. https://www.borger.dk/arbejde-dagpenge-ferie/Dagpenge-kontanthjaelp-og-sygedagpenge/Kontanthjaelp/Kontanthjaelp-30-eller-derover. Accessed on December 15, 2019.

Brownlee, J. (2018). Confidence Intervals for Machine Learning. https://machinelearningmastery.com/confidence-intervals-for-machine-learning/. Accessed on August 9, 2019.

Cedefop (2019). The online job vacancy market in the EU: Driving forces and emerging trends. *Luxembourg: Publications Office.*, Cedefop Research Paper No. 72.

Chakraborty, C. and Joseph, A. (2017). Machine Learning at Central Banks. *Bank of England*, Staff Working Paper No. 674.

Chen, T. and Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM.

Coulombe, P. G., Leroux, M., Stevanovic, D., Surprenant, S., et al. (2019). How is Machine Learning Useful for Macroeconomic Forecasting? Technical report, CIRANO.

Dilmaghani, M. (2019). The racial 'digital divide'in the predictive power of Google trends data for forecasting the unemployment rate. *Journal of Economic and Social Measurement*, pages 1–24.

Dong, G. and Liu, H. (2018). *Feature engineering for machine learning and data analytics*. CRC Press.

D'Amuri, F. and Marcucci, J. (2017). The predictive power of Google searches in forecasting US unemployment. *International Journal of Forecasting*, 33(4):801–816.

Elkins, E. and Sonnek, P. (2019). pytrends: Unofficial api for google trends. https://github.com/GeneralMills/pytrends. Accessed on September 2, 2019.

Fornaro, P. and Luomaranta, H. (2019). Nowcasting Finnish Real Economic Activity: a Machine Learning Approach. *Empirical Economics*, pages 1–17.

Foster, I., Ghani, R., Jarmin, R. S., Kreuter, F., and Lane, J. (2016). *Big data and social science: A practical guide to methods and tools*. Chapman and Hall/CRC.

Giannone, D., Reichlin, L., and Small, D. (2008). Nowcasting: The real-time informational content of macroeconomic data. *Journal of Monetary Economics*, 55(4):665–676.

Gil, M., Pérez, J. J., Sanchez Fuentes, A. J., and Urtasun, A. (2018). Nowcasting Private Consumption: Traditional Indicators, Uncertainty Measures, Credit Cards and Some Internet Data. SSRN Scholarly Paper ID 3299575, Social Science Research Network, Rochester, NY.

Ginsberg, J., Mohebbi, M. H., Patel, R. S., Brammer, L., Smolinski, M. S., and Brilliant, L. (2009). Detecting influenza epidemics using search engine query data. *Nature*, 457(7232):1012.

Hall, A. S. (2018). Machine Learning Approaches to Macroeconomic Forecasting. *Economic Review-Federal Reserve Bank of Kansas City*, 103(4):63.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Heydt, M. and Zeng, J. (2018). *Python Web Scraping Cookbook*. Packt Publishing, Limited.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). Resampling Methods. In *An Introduction to Statistical Learning: with Applications in R*, Springer Texts in Statistics, pages 175–201. Springer, New York, NY.

Jobnet (2019). About Jobnet. https://info.jobnet.dk/om-jobnet. Accessed on November 22, 2019.

Katris, C. (2019). Prediction of Unemployment Rates with Time Series and Machine Learning Techniques. *Computational Economics*.

Kim, J.-H. (2009). Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Computational Statistics & Data Analysis*, 53(11):3735–3745.

Lansdall-Welfare, T., Lampos, V., and Cristianini, N. (2012). Nowcasting the mood of the nation. *Significance*, 9(4):26–28.

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., and Lee, S.-I. (2019). Explainable AI for Trees: From Local Explanations to Global Understanding. *arXiv preprint arXiv:1905.04610*.

Makridakis, S., Spiliotis, E., and Assimakopoulos, V. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PLOS ONE*, 13(3):e0194889.

Mankiw, N. and Taylor, M. (2014). *Economics*. Cengage Learning.

Monsell, B. (2017). X-13arima-SEATS Seasonal Adjustment Program.

Nagao, S., Takeda, F., and Tanaka, R. (2019). Nowcasting of the U.S. unemployment rate using Google Trends. *Finance Research Letters*, 30:103–109.

Pavlicek, J. and Kristoufek, L. (2015). Nowcasting Unemployment Rates with Google Searches: Evidence from the Visegrad Group Countries. *PloS one*, 10(5):e0127084.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pescatori, A. and Zaman, S. (2011). Macroeconomic Models, Forecasting, and Policymaking. *Economic Commentary*.

Pratap, B. and Sengupta, S. (2019). Macroeconomic Forecasting in India: Does Machine Learning Hold the Key to Better Forecasts? *Reserve Bank of India*.

Raschka, S. (2015). *Python machine learning*. Packt Publishing Ltd.

Salganik, M. (2019). *Bit by bit: Social research in the digital age*. Princeton University Press.

Son, L., Carica, G. G., Ciuca, V., and Paşnicu, D. (2010). An autoregressive short-run forecasting model for unemployment rates in romania and the european union. In *Proceedings of 11th WSEAS International Conference on Mathematics and Computers in Business and Economics (MCBE'10), ISSN*, volume 2769, pages 193–198.

STAR (2019a). Arbejdsløshedsdagpenge. https://star.dk/ydelser/ledighed/arbejdsloes-hedsdagpenge/. Accessed on November 20, 2019.

STAR (2019b). Oversigt over a-kasserne i Danmark. https://star.dk/tilsyn-kontrol-og-klager-over-a-kassernes-afgoerelser/tilsyn-og-kontrol-med-a-kasser/oversigt-over-a-kasserne/. Accessed on November 20, 2019.

Statistics Denmark (2014). *Nationalregnskab og offentlige finanser - ESA 2010, hovedrevision 2014*. TemaPubl 2014:2. Statistics Denmark.

Statistics Denmark (2019a). AUS08: Unemployed persons (seasonally adjusted) by region and seasonal adjustment and actual figures. https://statistikbanken.dk/aus08. Accessed on November 1, 2019.

Statistics Denmark (2019b). FABRIT01: Access to computer and internet in by household type by access, type and time. https://www.statistikbanken.dk/FABRIT01. Accessed on November 1, 2019.

Statistics Denmark (2019c). LSK01: Job vacancies by industry, unit and size. https://www.statistikbanken.dk/lsk01. Accessed on November 1, 2019.

Stephens-Davidowitz, S. (2014). The cost of racial animus on a black candidate: Evidence using Google search data. *Journal of Public Economics*, 118:26–40.

Tainer, E. M. (2006). *Using Economic Indicators to Improve Investment Analysis*. Wiley Finance Series. John Wiley & Sons Inc, 3rd ed. edition.

Tseng, G. (2018). Interpreting complex models with SHAP values. https://medium.com/@gabrieltseng/interpreting-complex-models-with-shap-values-1c187db6ec83. Accessed on December 6, 2019.

Tuhkuri, J. (2014). Big Data: Google Searches Predict Unemployment in Finland. ETLA Reports 31, The Research Institute of the Finnish Economy.

Tuhkuri, J. (2015). Big Data: Do Google Searches Predict Unemployment? Master's thesis, Helsingfors universitet.

Tuhkuri, J. (2016). Forecasting Unemployment with Google Searches. ETLA Working Paper 35, The Research Institute of the Finnish Economy.

Vermorken, M., Gendebien, M., Vermorken, A., and Schröder, T. (2013). Skilled monkey or unlucky manager? *Journal of Asset Management; London*, 14(5):267–277.

Wooldridge, J. M. (2018). *Introductory Econometrics*. Cengage Learning, Inc, 7 edition.

Appendix

A Figures and tables

Table A.1: Variable overview, Denmark

Variable	**Source**	**Description**	**Lag**	**Frequency**
Unemployment rate	Statistics Denmark Table AUS08	Share of unemployed inhabitants relative to the labour force	1 m.	Monthly
Google searches	Google Trends	Google search terms which indicate labour market status*	1 d.	Monthly
Job posts	Jobindex	Number of jobposts and jobposts within selected sectors: *Information Technology, Engineering and technology, Management and staff, Trade and service, Industry and craft, Sales and communication, Teaching, Office and finance, Social and health and Other positions*	1 d.	Monthly
Population	Statistics Denmark Table FOLK1A	Regional population	-1.5 m.**	Quarterly
Labour force, %	Statistics Denmark Table AUS08	Share of population in labour force	1 m.	Quarterly
Higher education, %	Statistics Denmark Table HFUDD10	Share of population with a higher education Higher education includes: *Vocational bachelors educations,* *Bachelors programmes* *Master programmes* and *PhD programmes*	6.5 m.	Yearly
Urbanisation, %	Statistics Denmark Table: BEF1A, BEF1A07, FOLK1, BY1, BEF4A and BEF44***	Share of the population in a densely populated area or urban settlement. A hub of buildings is registered as an urban settlement if it is inhabited by at least 200 persons.	4 m.	Yearly

Note: m. refers to months and d. to days. *See Appendix Table A.2 for actual search terms. ** -1.5 m. refers to medio current quarter. ***The actual statistic is retrieved from the Ministry of Social Affairs and Interiors platform for key municipality figures

Source: Statistics Denmark, GT, Jobindex

Table A.2: Google search terms overview, Denmark

Label	Description	Actual search terms	Correlation with target	Included
Jobnet	Danish job posting site	jobnet+"jobnet.dk"+"jobnet cv"	0.72	Yes
Jobindex	Danish job posting site	jobindex+"job index"	0.63	Yes
KRIFA	Danish unemployment insurance fund	krifa+"krifa a kasse"+"krifa a-kasse"	0.63	Yes
Job centre	Job centre	jobcenter+jobcentre+"job center" +"job centre"	0.59	Yes
Unemployment insurance fund	Unemployment insurance fund	akasse+akasser+"a-kasse"+ "a-kasser"+"a kasse"	0.51	Yes
3F	Danish unemployment insurance fund	faglig fælles a-kasse+"3f"+"3f a-kasse" +"3f a kasse"	0.49	Yes
Cash benefits	Cash benefits and cash benefit rates	Kontanthjælp+"kontanthjælp sats"+ "kontanthjælp satser"	0.46	Yes
Unemployment insurance rate	Unemployment insurance cash benefits and unemployment insurance rates	dagpengesats+"dagpenge sats"+ dagpengesatser+"dagpenge satser"	0.46	Yes
Job openings	Job positions	job+jobopslag+"job opslag"	-0.41	Yes
ASE	Danish unemployment insurance fund	ase+"ase a-kasse"+"ase akasse"	0.39	Yes
Unemployment	Unemployed and unemployment	arbejdsløs+arbejdsløshed	0.29	No
Lærernes a-kasse	Danish unemployment insurance fund	"lærernes a-kasse"+"lærernes a kasse"	0.27	No
Akademikernes	Danish unemployment insurance fund	akademikernes+"akademikernes a-kasse"+ "akademikernes akasse"+ "ingeniørernes akasse"+ iak	0.26	No
Resume	Resume and examples of such	cv+"cv eksempel"+"cv skabelon"	0.25	No
BUPL	Danish unemployment insurance fund	bupl+"Børne- og Ungdomspædagogernes a-kasse"+"Børne- og Ungdomspædagogernes Landsforbund"	-0.24	No
Unemployment insurance	Unemployment insurance cash benefits and unemployment insurance rules	dagpenge+"dagpenge regler" +dagpengeregler	0.19	No
Open positions	Open positions	"ledige job"+"ledige jobs"+ "ledig stilling"+"ledige stillinger"	-0.16	No
HK	Danish unemployment insurance fund	hk+hk a-kasse+"hk a kasse" +"hk danmark"	0.15	No
FOA	Danish unemployment insurance fund	foa+"fag og arbejde"+ "fag og arbejde a-kasse"+"fag og arbejde a kasse"	-0.12	No
Unemployed	Unemployed	ledig+ledighed	-0.07	No
Fired	Fired	fyret	0.04	No
Ofir	Danish job posting site	ofir+ofir.dk+"ofir jobportal"	0.0	No

Note: By inserting "" you restrict to search for the entire combination of words in the specified order. + refers to the case of one labelled search term including multiple actual search terms. A Google search term is included if a given sectors correlation with target is $\geq |0.35|$.

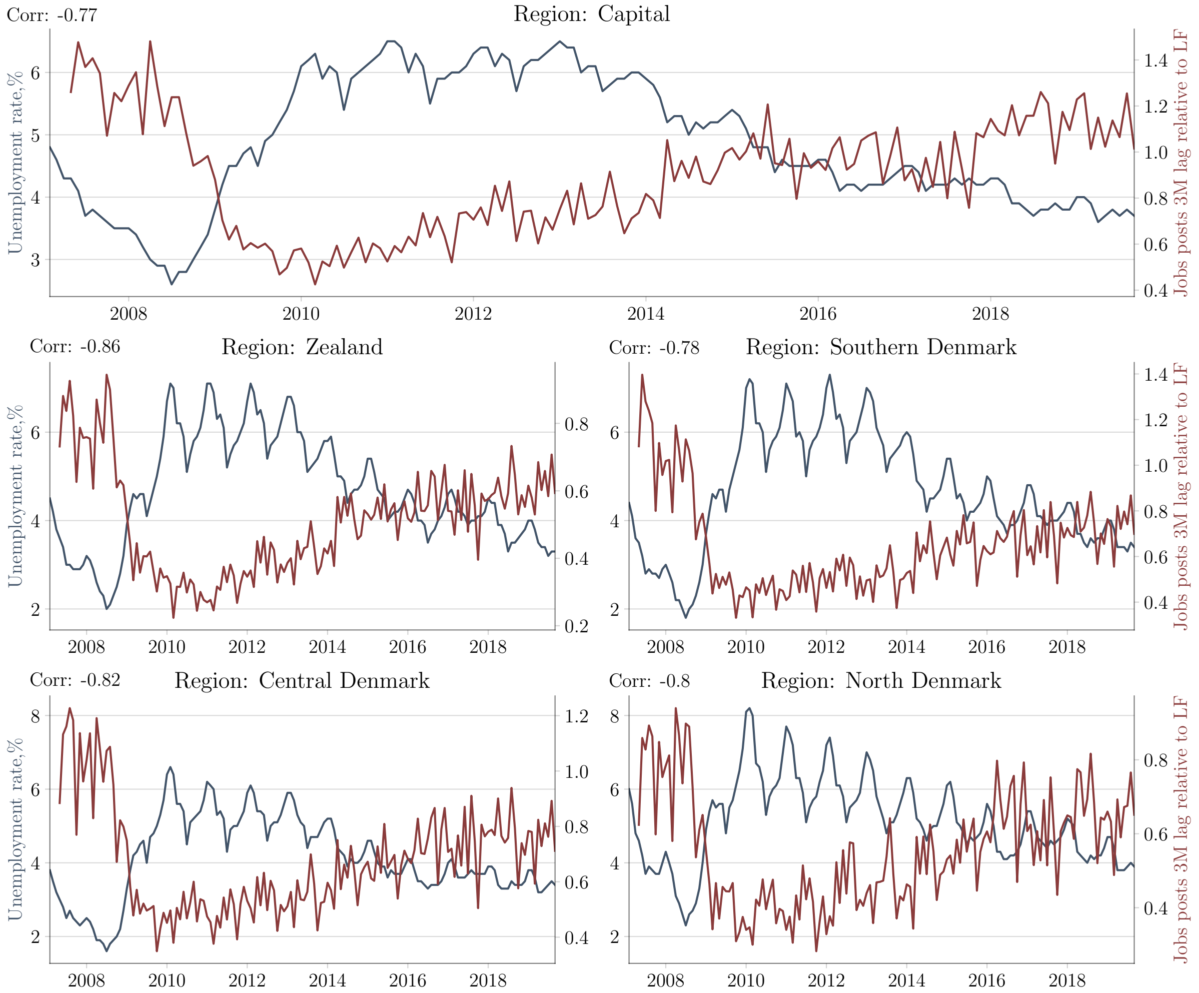
Source: Statistics Denmark, GT

Table A.3: Sectors overview, Denmark

Label	Correlation with target	Included
Management and staff	-0.66	Yes
Teaching	-0.64	Yes
Sales and communication	-0.62	Yes
Office and finance	-0.59	Yes
Social and health	-0.57	Yes
Industry and craft	-0.5	Yes
Information Technology	-0.47	Yes
Engineering and technology	-0.46	Yes
Trade and service	-0.3	No
Other positions	-0.28	No

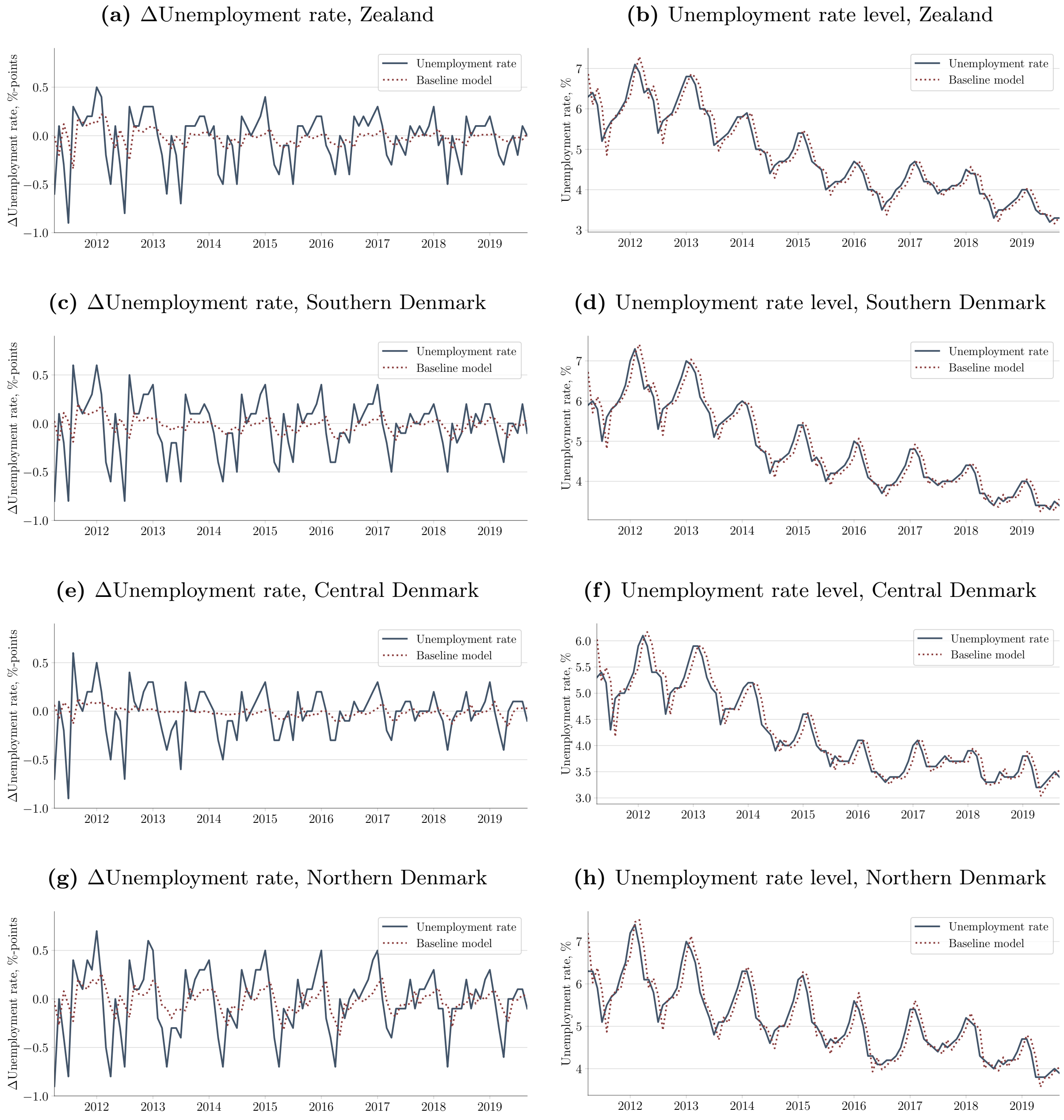
Note: A sector is included if a given sectors correlation with target is ≥ |0.35|.
Source: Statistics Denmark, Jobindex

Figure A.1: Regional sample correlation, Unemployment rates vs. 3M lag of jobposts relative to labour force



Source: Statistics Denmark, Jobindex

Figure A.2: Series of nowcasts, baseline model, Denmark



(a) ΔUnemployment rate, Zealand

(b) Unemployment rate level, Zealand

(c) ΔUnemployment rate, Southern Denmark

(d) Unemployment rate level, Southern Denmark

(e) ΔUnemployment rate, Central Denmark

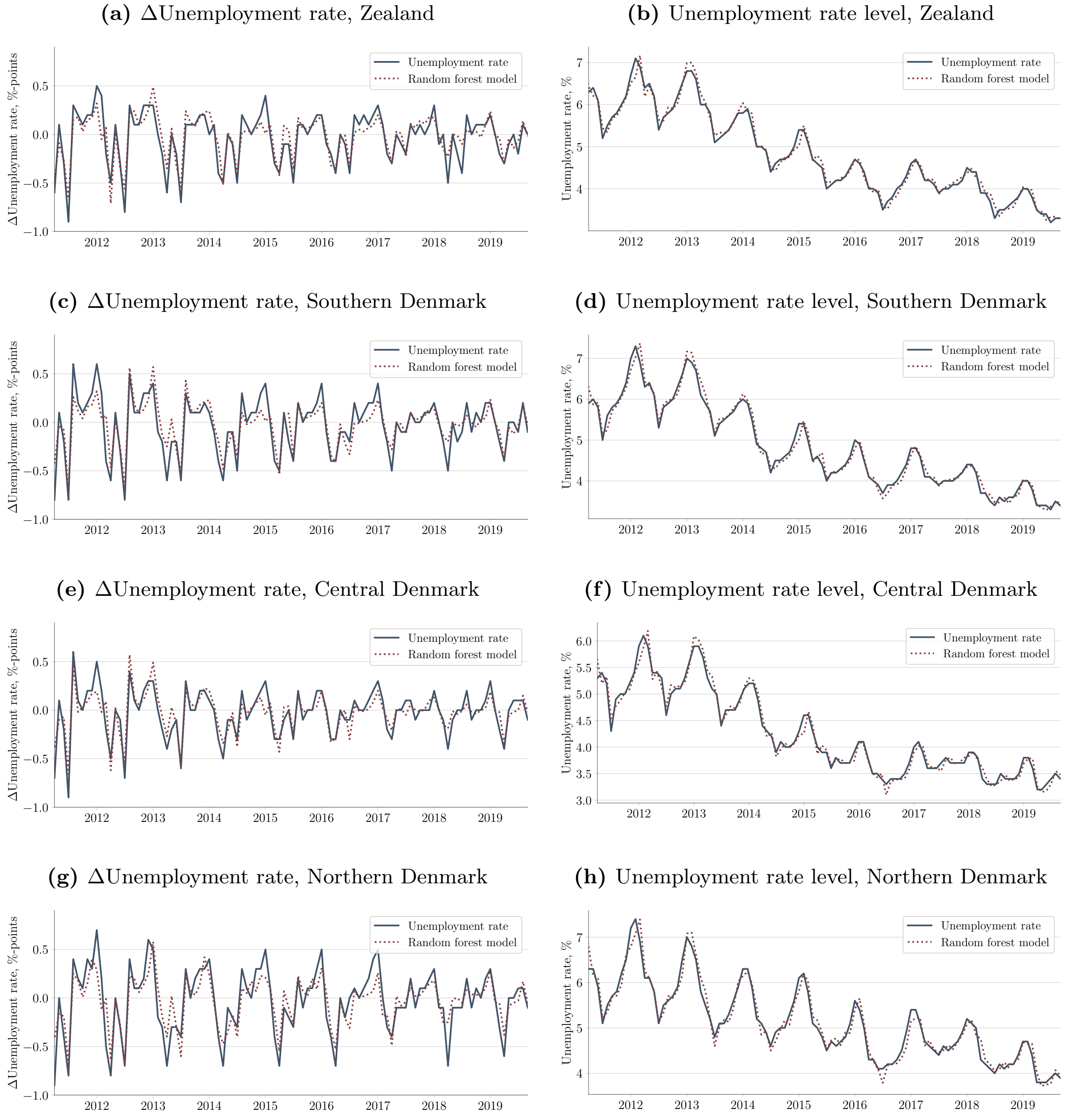
(f) Unemployment rate level, Central Denmark

(g) ΔUnemployment rate, Northern Denmark

(h) Unemployment rate level, Northern Denmark

Source: Statistics Denmark

Figure A.3: Series of nowcasts, Random forest model, Denmark



(a) ΔUnemployment rate, Zealand

(b) Unemployment rate level, Zealand

(c) ΔUnemployment rate, Southern Denmark

(d) Unemployment rate level, Southern Denmark

(e) ΔUnemployment rate, Central Denmark

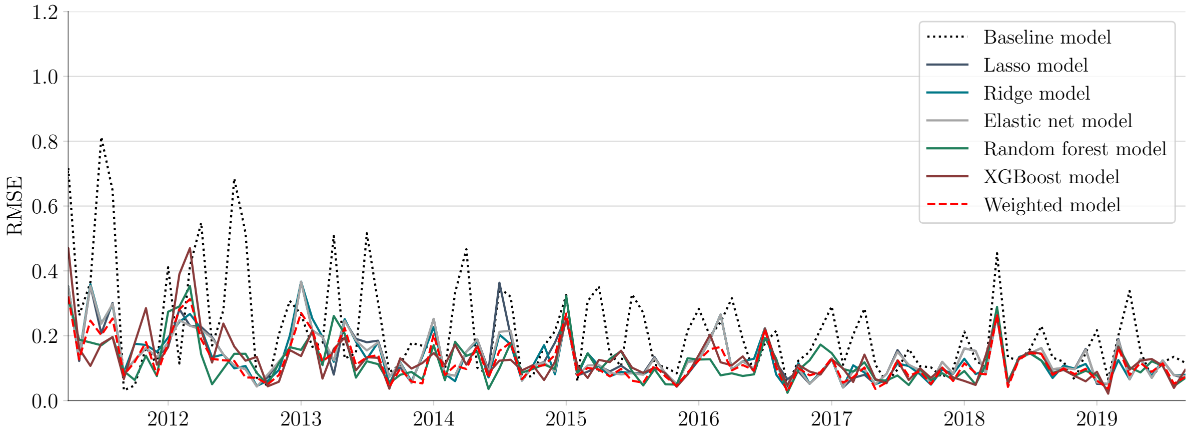
(f) Unemployment rate level, Central Denmark

(g) ΔUnemployment rate, Northern Denmark

(h) Unemployment rate level, Northern Denmark

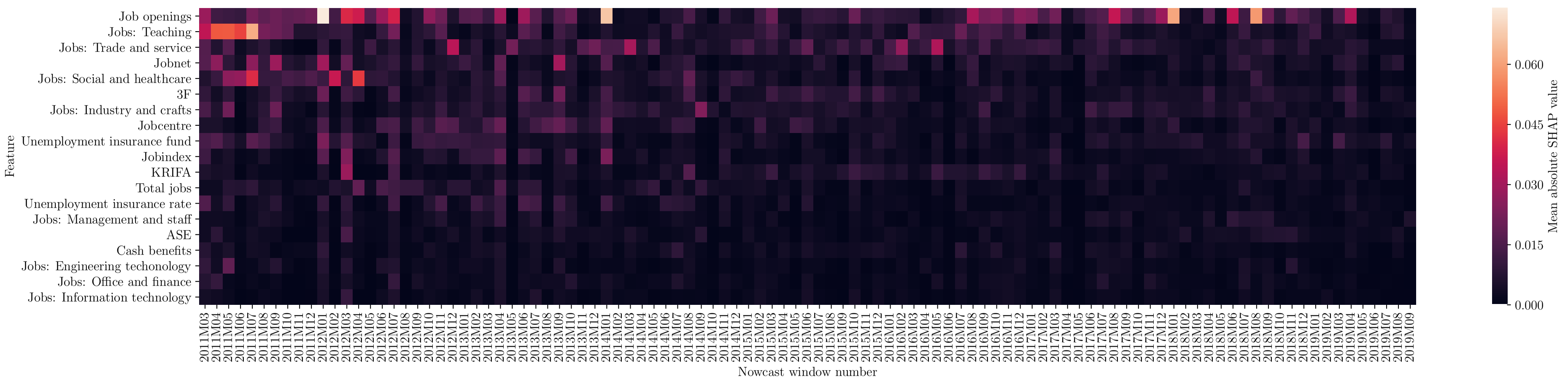
Source: Statistics Denmark, GT, Jobindex

Figure A.4: RMSE across windows, Denmark



Source: Statistics Denmark, GT, Jobindex

Figure A.5: Feature importance for random forest across nowcasting windows, Denmark



Note: The mean absolute SHAP values for features that enters multiple times due to additional lag terms are summed as to keep the figure relatively clean.

Source: Statistics Denmark, GT, Jobindex

Table A.4: Google search terms overview, Sweden

Label	**Description**	**Actual search terms**	**Correlation with target**	**Included**
Platsbanken	Swedish job posting site	platsbanken+ "platsbanken arbetsförmedlingen"	0.54	Yes
The Swedish Public Employment Service	The Swedish Public Employment Service	arbetsförmedlingen	-0.43	Yes
Job vacancies	Job vacancies	lediga platser"+"lediga jobb"+ vakans	-0.34	Yes
Job openings	Job and job openings	jobb+platsannonser+platsannons	-0.29	Yes
Unemployment insurance benefits	Unemployment insurance benefits	arbetslöshetsersättningen+ ersättning+ersättningsperiode	-0.23	Yes
Cash benefits	Cash benefits	försörjningsstöd+socialbidrag+" socialbidrag krav"+ "ekonomiskt bistånd"	0.18	Yes
Unemployment insurance fund	Unemployment insurance fund	a-kassa+akassa+"a kassa"	0.12	No
Jobbsafari	Swedish job posting site	jobbsafari	0.08	No
Named unemployment insurance funds	Swedish unemployment insurance fund	unionen+htf+sif+ tjänstemannaförbundet+ "Svenska industritjänstemannaförbundet"	0.02	No
LO	Swedish unemployment insurance fund	lo + landsorganisationen+ "landsorganisationen sverige"	0.0	No

Note: By inserting "" you restrict to search for the entire combination of words in the specified order. + refers to the case of one labelled search term including multiple actual search terms. A Google search term is included if a given sectors correlation with target is ≥ |0.2|.

Source: Statistics Sweden, GT

Table A.5: Sectors overview, Sweden

Label	**Correlation with target**	**Included**
Information Technology	-0.31	Yes
Industry and craft	-0.24	Yes
Engineering and technology	-0.21	Yes
Office and finance	-0.2	Yes
Social and health	-0.13	No
Sales and communication	-0.1	No
Trade and service	-0.08	No
Management and staff	-0.06	No
Other positions	-0.05	No
Teaching	-0.01	No

Note: A sector is included if a given sectors correlation with target is ≥ |0.2|.

Source: Statistics Sweden, Jobbsafari

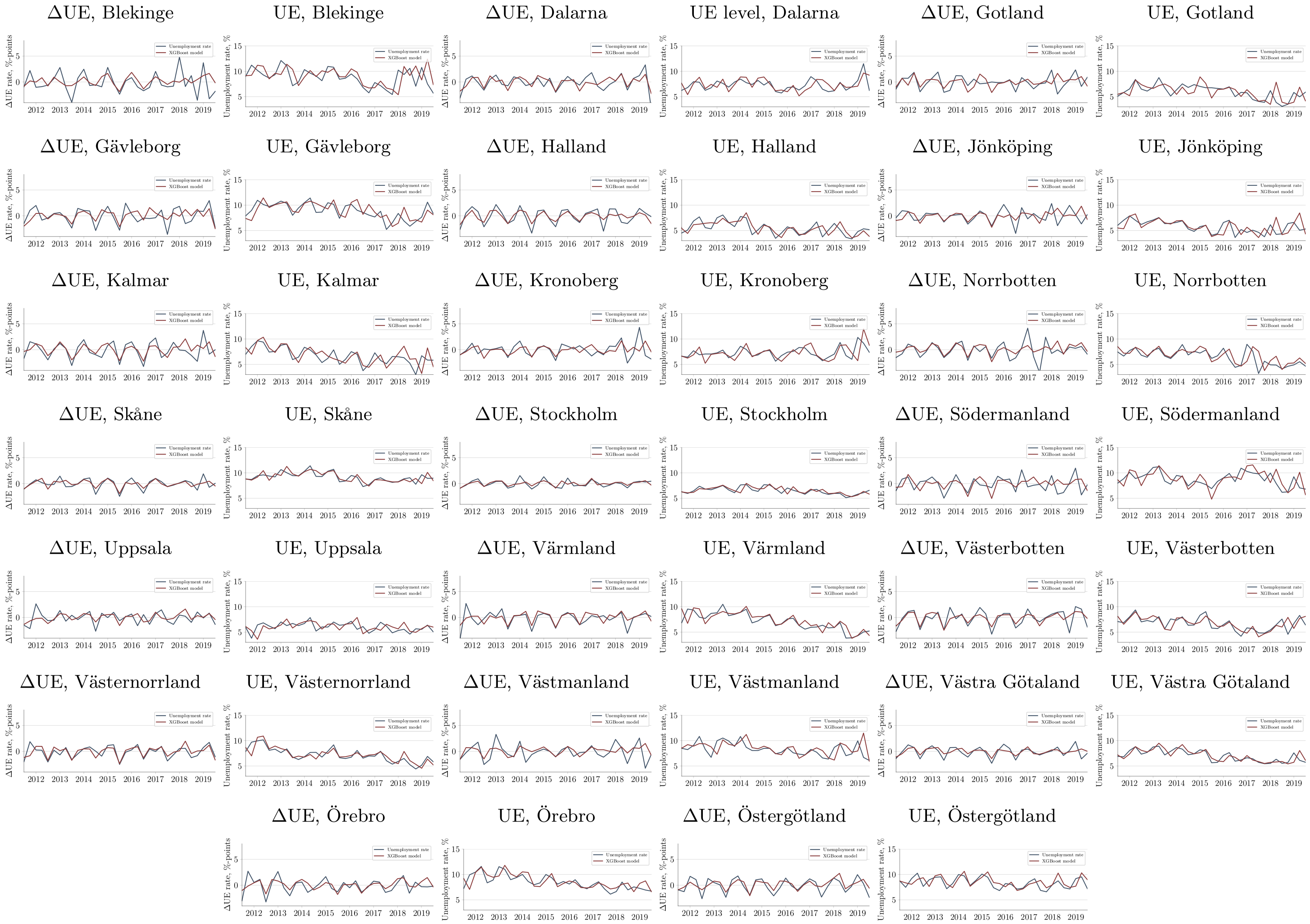
Table A.6: Average regional characterics, Sweden

Region	**Unemployment rate, %**	**Population**	**Higher education, %**	**Labour force, %**	**Urbanisation, %**
Blekinge	8.6	15,413	21.4	71.1	80.2
Dalarna	7.4	278,771	18.3	71.7	80.3
Gotland	6.4	57,480	19.8	72.8	59.0
Gävleborg	8.6	278,814	18.1	70.6	78.5
Halland	5.8	305,064	21.9	74.7	80.3
Jönköping	5.9	341,693	19.2	74.8	82.8
Kalmar	6.9	235,971	19.1	70.8	77.8
Kronoberg	7.0	187,031	21.1	74.5	77.7
Norrbotten	7.3	249,896	20.8	70.3	82.2
Skåne	8.8	1,262,614	25.0	73.3	88.7
Stockholm	6.4	2,114,131	30.7	77.1	95.9
Södermanland	8.4	275,851	19.0	71.3	82.7
Uppsala	6.5	342,115	28.9	73.5	80.4
Värmland	7.5	274,999	20.0	70.7	74.5
Västerbotten	6.9	261,286	25.7	72.0	77.6
Västernorrland	7.7	243,523	19.9	70.6	76.9
Västmanland	8.4	257,907	20.7	72.0	87.7
Västra Götalands	7.2	1,606,265	24.2	74.1	84.3
Örebro	8.5	284,760	20.9	71.6	82.7
Östergötland	8.5	435,579	23.3	71.9	84.3

Note: The listed characteristics are the averages of each monthly data series for each region. For the exact definitions and sources of each metric, see Table 8.2.

Source: Statistics Sweden

Figure A.6: Series of nowcasts, XGBoost model, Sweden



ΔUE, Blekinge

UE, Blekinge

ΔUE, Dalarna

UE level, Dalarna

ΔUE, Gotland

UE, Gotland

ΔUE, Gävleborg

UE, Gävleborg

ΔUE, Halland

UE, Halland

ΔUE, Jönköping

UE, Jönköping

ΔUE, Kalmar

UE, Kalmar

ΔUE, Kronoberg

UE, Kronoberg

ΔUE, Norrbotten

UE, Norrbotten

ΔUE, Skåne

UE, Skåne

ΔUE, Stockholm

UE, Stockholm

ΔUE, Södermanland

UE, Södermanland

ΔUE, Uppsala

UE, Uppsala

ΔUE, Värmland

UE, Värmland

ΔUE, Västerbotten

UE, Västerbotten

ΔUE, Västernorrland

UE, Västernorrland

ΔUE, Västmanland

UE, Västmanland

ΔUE, Västra Götaland

UE, Västra Götaland

ΔUE, Örebro

UE, Örebro

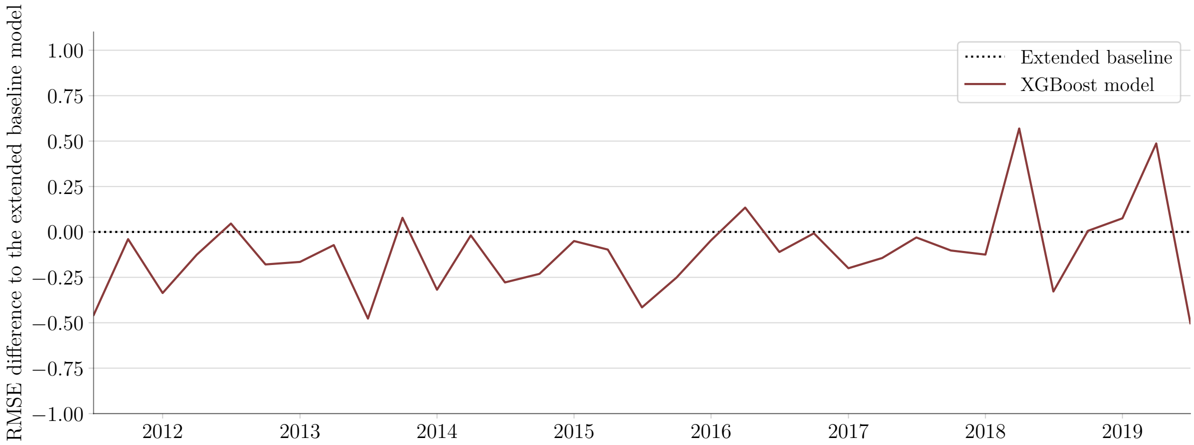
ΔUE, Östergötland

UE, Östergötland

Source: Statistics Sweden, GT, Jobbsafari

[H]

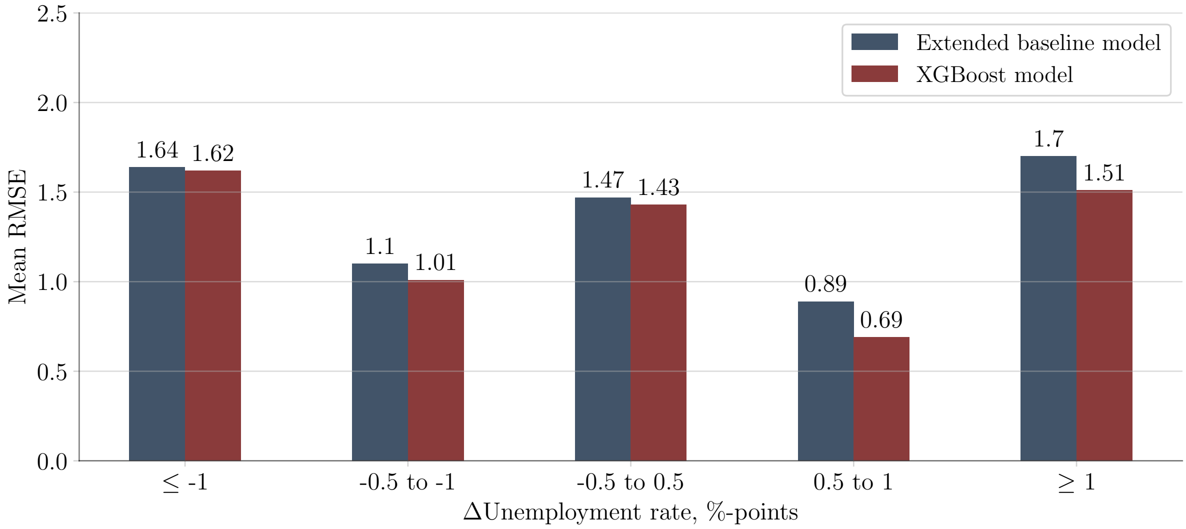
Figure A.7: RMSE difference, Extended baseline model vs. XGBoost, Sweden



Note: RMSE difference to the baseline is calculated as the RMSE of the respective model subtracted from the extended baseline RMSE for each window - thus, a negative value indicates that the XGBoost model outperformed the extended baseline in the given window.

Source: Statistics Sweden, GT, Jobbsafari

Figure A.8: Mean RMSE across actual ∆unemployment rate, baseline model vs. weighted model, Sweden



Source: Statistics Sweden, GT, Jobbsafari

B Tuning strategy of machine learning models

As stated in Section 5.1, the hyperparameters of the machine learning models must be tuned in order to identify which hyperparameter values lead to the best predictive performance. Each model allows for multiple restrictions with respect to hyperparameter values. The process of identifying the optimal hyperparameter spaces for each machine learning model is often based on a trail-and-error based engineering as no superior guideline applies to all input data and problems. For the tree-based models many more hyperparameters exists than tested for the purpose of this thesis. examining all possible values for each model is very time consuming why only the most important hyperparameters are varied for each model. The excess hyperparameters are set to their default value.

In practice, we are performing each machine learning model with all the possible combinations of the defined hyperparameter spaces for each window. This is also referred to as grid search. By this strategy we secure that the most optimal value is obtained. One could also have chosen to do a random search if the time frame for obtaining the results are more sparse.

This section states the overall tuning strategy for the penalised regressions and the tree-based models including both random forest and XGBoost.

Hyperparameter tuning of the penalised regressions

For the regression-based machine learning models, we will tune the sum of the explained variance of the included principal components as well as the regularisation parameters. The primer is included to control for overfitting. For example, setting the sum of the explained variance to 0.6 means that we include the first x principal components whose sum of the explained variance exceeds 0.6. Setting the sum of the explained variance to 1 means that one would include all the principal components. The overall tuning strategy is summarised in Table B.1.

For the regulation parameter α, 10.000 evenly spaced numbers are drawn from a log scale between -8 and 8. This parameter space is included in all three models to control for overfitting and optimise the bias-variance trade-off.

Furthermore, as the Elastic net regression include both the penalisation of the Lasso regression and the Ridge regression an extra parameter is tuned to optimise this weight. This parameter, λ, is varied between 0.01 and 0.99 with increments of 0.05.

Table B.1: Hyperparameter tuning of the penalised regressions

Hyperparameter	**Description**	**Parameter space**
Principal components	The explained variance by PC	0.6 to 0.9 with increments of 0.05
α_R, α_L, α	The degree of regularisation	10.000 numbers evenly spaced on a log scale between -8 and 8
λ	Weight between α_R and α_L (Only included in Elastic net)	0.01 to 0.99 with increments of 0.05

Note: The cost function of the Ridge, Lasso and Elastic net regression are stated in Equation 5, 6 and 8 respectively. The λ hyperparameter is only included in the Elastic net regression.

Hyperparameter tuning of the random forest

To optimise the performance of random forest four hyperparameters which all restrict for overfitting has been tuned for this project. The different parameter values is described in Table B.2.

The first hyper parameter is the number of times the training set is sampled and thereby the number of trees in the forest, $n\_estimators$. We allow for between 50 and 500 trees. Increasing the number of tree increases the performance and the stability of the prediction in sample but you might be subject to overfitting and it hampers the speed of the model.

The second hyper parameter is the maximum depth of each grown tree, max_depth. Naturally a deeper three, must consider more splits and therefore affects the run time. Here we allow for the individual tree to be both very simple but also very deep.

Table B.2: Hyperparameter tuning of random forest

Hyperparameter	**Description**	**Parameter space**
n_estimators	The number of trees in the forest	50 to 500 with increments of 10
max_depth	The maximum depth of the tree	3 to 11 with increments of 1 and 20 to 100 with increments of 20
max_features	The number of features to consider when looking for the best split	The square root of number of features and all features
min_sample_leaf	Minimum number of observations required at each leaf node	1 to 5 with increments of 2

Note: The label of each hyperparameter refers to the actual label in the model specification of the *RandomforestRegressor* in the python library *sklearn* by Pedregosa et al. (2011).

The third hyper parameter is refers to the number of considered features when looking for the best split based on explained variance, $max\_features$. When increasing the number of featured considered, you generally increase the model performance, as a higher number of options for each node is present. However, a large number of features decreases the diversity of the individual trees. We both consider the case where no restrictions on the individual tree are made and the case where we randomly sample the number of features corresponding to the square root of the total number of features.

The last parameter is the minimum number of observations required at each leaf node. Here a smaller number of observations in a given leaf node makes the model more sensitive to capturing noise in train data. A the data set is quite sparse in the number of observations this is set to an interval between one and five.

Hyperparameter tuning in XGBoost

The parameters in XGBoost are in general divided into three categories; general parameters, booster parameters and learning task parameters. Many of which has the purpose to reduce overfitting. For this project we have decided to tune only boosting parameters including among others the number of trees in the forest, $n\_estimators$ and the maximum depth of a single tree, max_depth with the same restrictions as in the hyperparameter tuning of random forest seen in Table B.2. The full specification of hyperparameter tuning of the XGBoost is displayed in Table B.3.

To further restrict for overfitting are three extra hyperparameters are considered and tuned. First we only include a subsample of columns when constructing each tree This parameter is named $colsample\_bytree$. Here we allow for a ratio of 0.3 to 0.9. Second, only a subsample of all observations are included in each tree. This ratio is restricted to including half of the observations to including the full data set. Lastly, as for the random forest the minimum number of observations of a leaf node is restricted to be between one and five by the hyperparameter min_child_weight.

Table B.3: Hyperparameter tuning of XGBoost

Hyperparameter	**Description**	**Parameter space**
n_estimators	The number of trees in the forest	50 to 500 with increments of 10
max_depth	The maximum depth of the tree	3 to 11 with increments of 1 and 20 to 100 with increments of 20
colsample_bytree	The subsample ratio of columns when constructing each tree	0.3 to 0.9 with increments of 0.2 and 1
subsample	The fraction of observations to be randomly sampled for each tree	0.5 to 1 with increments of 0.25
min_child_weight	Minimum number of observations required at each leaf node	1 to 5 with increments of 2

Note: The label of each hyperparameter refers to the actual label in the model specification of the *XGBRegressor* in the python library *xgboost* by Chen and Guestrin (2016).

C Applied programming language and packages

For conducting the analysis including both data collection, preprocessing and modelling we have used the programming language *Python* and the interface *Jupyter lab*. Through the open source environment we have used several packages which includes specific algorithms and functions to ease the work stream of the individual processes. For this thesis the most central used packages are highlighted below grouped by overall purpose.

Data collection

- **requests** [*https://realpython.com/python-requests/*] is today the preferred package for making HTTP requests in python. We have used the GET request, *get()* for retrieving online job post data from Jobindex and Jobbsafari respectively.
- **bs4** [*https://www.crummy.com/software/BeautifulSoup/bs4/doc/*] is a library for navigating and subsetting parsed data from the web fetched by using the package requests. We have used the package *BeautifulSoup*.
- **pytrends** [*https://github.com/GeneralMills/pytrends*] is a pseudo (or unofficial) API for Google trends. It has a simple interface for automating the process of downloading data series from Google Trends. We have used the package *TrendReq*.

Data preprocessing and descriptives

- **NumPy** [*https://numpy.org/*] contains fundamental data handling functions and scientific functions from linear algebra including a simple mean, *mean()* and a sophisticated array object, *np.array* which we have widely used for data structuring. We have also used the function *concatenate()* for concatenating the train and validation set before calibrating a model on the test set.
- **pandas** [*https://pandas.pydata.org/*] includes functions for structuring and preprocessing data. We have heavily used of the following packages for constructing the master data: *DataFrame(), groupby(), filter(), dropna(), sort_ index()* and *get_ dummies().*

Model calibration

- **scikit-learn** [*https://scikit-learn.org/stable/*] includes packages for model calibration of among others machine learning models. We have used the following packages: *PCA, LinearRegression, Lasso, Ridge, ElasticNet* and *RandomforestRegressor.*

- **XGBoost** [*https://xgboost.readthedocs.io/en/latest/*] includes functions for calibration of the XGBoost model. We have used the package *XGBRegressor*.

Other central packages includes: *matplotlib* for constructing graphs, *datetime* for handling date variables, *pickle* for saving model objects and *multiprocessing* for calibrating the models on multiple processors.