CEWP 20-14

Forecasting Canadian GDP growth using XGBoost

Shafiullah Qureshi
Carleton University &
Ottawa-Carleton GSE

Ba M Chu
Carleton University &
Ottawa-Carleton GSE

Fanny S. Demers
Carleton University &
Ottawa-Carleton GSE

August 2020; revised August 24 2020

CARLETON ECONOMICS WORKING PAPERS

Department of Economics

1125 Colonel By Drive
Ottawa, Ontario, Canada
K1S 5B6

Forecasting Canadian GDP growth using XGBoost

Shafiullah Qureshi*
Carleton University & Ottawa-Carleton GSE
Ba M Chu†
Carleton University & Ottawa-Carleton GSE
Fanny S. Demers‡
Carleton University & Ottawa-Carleton GSE

August 24, 2020

Abstract

The objective of this paper is to apply state-of-the-art machine-learning (ML) algorithms to predict the monthly and quarterly real GDP growth of Canada using both Google Trends (GT) and Official data that are available ahead of the release of GDP data by Statistics Canada. This paper applies a novel approach for selecting features with Extreme Gradient Boosting (XGBoost) using the AutoML function of H2O. For this purpose, 5000 to 15000 XGBoost models are trained using this function. We use a very rigorous variable selection procedure, where only the best features are selected into the next stage to build a final learning model. Then pertinent features are introduced into XGBoost for forecasting real GDP growth rate. The forecasts are further improved by using Principal Component Analysis (PCA) to choose the best factors out of the predictors selected by XGBoost. The results indicate that there are gains in nowcasting accuracy from using XGBoost with this two-step strategy. We first find that XGBoost is a superior algorithm for forecasting relative to our baseline methods, such as autoregression and other standard boosting algorithms. We also find that Google Trends data provides a very viable source of information for predicting Canadian real GDP growth with XGBoost when Official data are not yet available due to publication lags. Therefore, we can forecast real GDP growth rate accurately ahead of the release of Official data. Moreover, we apply various techniques to make the machine learning model more interpretable.

*Department of Economics, A806 Loeb, 1125 Colonel By Dr., Carleton University, Ottawa, Canada. Email: shafiullah.qureshi@carleton.ca Tel: +1.613.520.2600 (ext 3778) and Department of Economics, NUML, Sector H-9, Islamabad - Pakistan

†Department of Economics, B-857 Loeb, 1125 Colonel By Dr., Carleton University, Ottawa, Canada. Email: ba.chu@carleton.ca Tel: +1 613-520-2600 x 1546

‡Department of Economics, B-854 Loeb, 1125 Colonel By Dr., Carleton University, Ottawa, Canada. Email: fanny.demers@carleton.ca Tel: +1 613-520-2600 x 3775

1 Introduction

Gross domestic product (GDP) is the primary measure for assessing the performance of an economy. It enables policymakers to judge whether the economy is expanding or contracting, and permits them to make appropriate monetary or fiscal policy decisions accordingly. In this respect, the accurate and timely forecast of GDP growth ahead of the release of the Official data is vital. While Statistics Canada releases GDP data on an annual, quarterly, and monthly basis, both quarterly and monthly GDP data are issued with a delay of around two months.

In this paper, we explore the means of obtaining more timely forecasts of Canadian GDP growth. To this end, we apply a state-of-the-art ML algorithms to nowcast the monthly and quarterly GDP of Canada, with both Google Trends (GT) and Official data that are available ahead of the release of GDP data by Statistics Canada.[1] Using a very recent machine learning algorithm (namely, XGBoost), we first find that this algorithm can provide a superior forecast performance relative to alternative procedures, such as autoregression and standard boosting methods. We then show that using XGBoost with GT data can forecast real GDP growth quite accurately when Official data are not yet released.

Advanced ML algorithms have not been explored extensively in economics applications.[2] Several papers have used standard ML approaches to forecasting the GDP of various countries and regions. For example, Biau and D'Elia (2009) used the Random Forest (RF) model to predict the Euro-area GDP. They show that combining the RF and a linear model can actually outperform the benchmark autoregressive model. Tiffin (2016) employed the RF and the Elastic Net algorithms to nowcast GDP growth in Lebanon. Similarly, Jung et al. (2018) used the Elastic Net, SuperLearner, and Recurring Neural Network algorithms on the data of seven advanced and emerging economies, and found that these algorithms produced better accuracy than the traditional statistical models. Papers focusing on forecasting Canadian GDP include Tkacz (2001), which adopts a neural network forecasting approach, and Chernis and Sekkel (2017), which employs dynamic factor models. Yet, despite the advancements in ML and the importance of forecasting GDP growth, to the best of our knowledge, there have not been studies that apply up-to-date ML techniques to predict GDP growth rates (in particular, the Canadian GDP growth). Also, modern requirements for machine learning models include both *high predictive performance* (which has been the main focus of many papers, including this paper) and *model interpretability* (which has not been much explored in recent economic applications, thus it is the focus of this paper).

The use of GT data for forecasting purposes became popular since the seminal work of Choi and Varian (2009) and Choi and Varian (2012). GT data have been used to predict economic variables, such as retail sales, automotive sales, home sales, and travel volume, among many others. GT data allows users to download the time series data for searches of a particular topic in an index form by selecting a specific geographic region and a specific time. Notably, Tkacz (2013) used GT data to predict the recession of 2008-09 with a probit model for Canada. He has found an enormous surge in the Google search of the word "recession" one month before the actual occurrence of the 2008-09 recession.

Previous studies using GT data to forecast GDP have made use of bridge equation models. Götz and Knetsch (2019) used Official data together with GT data to forecast German GDP. Their approach achieves a better forecasting accuracy when GT data was used *instead of* Official (soft)

[1] For example, Canada's monthly GDP is released with a lag of two months after the reference period. These delays occur mostly due to multiple revisions.

[2] Examples are XGBoost of Chen and Guestrin (2016), LightGBM, and stacked ensembles of many models which are prevalent in both applied ML and in Kaggle competitions.

survey data, suggesting that Google Trends data can successfully replace survey data. Ferrara and Simoni (2019) also used bridge equation models to nowcast Euro-area GDP with GT data. They pre-select variables from Google searches by using the Sure Independence Screening (SIS) approach of Fan and Lv (2008), which basically selects variables with the highest absolute correlation with the dependent variable. They then apply a ridge regression to reduce the dimension of GT data and study its effectiveness in making forecasts, both when Official data are not available and when Official data become available. They found that, while the nowcasting power of GT data can diminish as soon as survey data became available (usually by the 5th week of the quarter), using Google data can be a reliable forecasting tool when Official data are not available.[3] In this paper, by applying XGBoost, we find a similar evidence that GT data provides remarkable forecasting accuracy in the absence of Official data.

Our method of forecasting GDP growth consists of the two main steps. In the first step, we utilize automated machine learning (AutoML) to select most informative variables from a broad set of potential candidate variables provided by GT and Official data.[4] In the second step, we select the most relevant GT features through *the variable importance measure* [of XGBoost], which are then fed into XGBoost to construct efficient predictors for the Canadian GDP growth.[5] It is to be noted that variable selection is a crucial step to improve forecast performance. In fact, as Götz and Knetsch (2019) have pointed out, variable selection method to be used can determine the forecast performance of a model. Therefore, we rigorously fit several thousand models to get better out-of-sample accuracy. Final variables are selected from the model with the best out-of-sample forecast performance [by again applying the variable importance measure of XGBoost]. This two-step procedure employs XGBoost as a powerful tool to select the most relevant variables from a large pool of variables at each step, thus it can provide a better forecasting accuracy. We also found that XGBoost delivers a much better out-of-sample performance than any other available algorithm.

We use both the root mean squared error (RMSE) and graphical representation approaches in order to compare the performance of forecast methods, and to examine the nowcasting power of using GT together with Official data. Graphical representation approaches have not been used in the existing works on this topic. Although Götz and Knetsch (2019) and Ferrara and Simoni (2019) conducted a very thorough analysis of GDP forecasting with GT data, they only employed the RMSE to compare the performance of various forecast models. Tkacz (2001) also employed only the mean absolute error and the mean squared error. While comparison based on the RMSE is useful in ranking forecast models, it cannot evaluate the performance of a given model individually against the actual target variables. It is therefore difficult to adequately assess the model performance in predicting the actual outcome. As we shall show in subsection 2.4, a graphical representation of the actual versus predicted time-paths is very useful in gauging a model's performance with GT and Official data. It provides a clear picture of how GDP growth, predicted with both GT and Official data, behaves relative to the actual GDP growth. In this respect, we follow Friedman (2001, p.1219)

[3] Giannone et al. (2008) introduced the term nowcasting, a contraction of two words "now" and "forecasting," and it represents the short-term forecasting that is predicting the present.

[4] AutoML is an interface which automates the process of simultaneously training a large number of ML models such as Gradient Boosting Machines (GBMs), XGBoost, Generalized Linear Models (GLMs), Distributed Random Forest (DRFs, which consist of Random Forest and Extremely Randomized Trees models), Deep Learning and Stacked Ensembles. AutoML is available in H2O, an open-source platform for ML.

[5] XGBoost (eXtreme Gradient Boosting) is an algorithm that has recently been dominating in applied machine learning and Kaggle data science competitions. It is a more efficient version (both in terms of speed and performance) of "gradient boosting decision trees." New models are created that predict the residuals or errors of prior models. Gradient boosting is an approach that uses a gradient descent algorithm to minimize a loss function when adding these new models together to make the final prediction.

who emphasizes that "Visualization is one of the most powerful interpretational tools."

Friedman goes on to say that "Graphical renderings of [the predictive model] provide a comprehensive summary of its dependence on the joint values of the input variables" and greatly enhance our understanding of the model. In accordance with Friedman, we also use graphical analysis as an important tool to interpret our results. ML methods are sometimes criticized as black-box models that do not lend themselves to interpretation for policy-making purposes. In general, it is almost impossible to understand the exact contribution of each of the input variables to the final prediction. We try to address these issues by including tools that can make ML more interpretable. These tools include both global interpretation tools, such as variable importance plots and partial dependence plots as advocated by Friedman (2006), and local interpretation tools, such as SHAP values as advocated by Lundberg and Lee (2017) and Lundberg et al. (2018).[6]

Friedman (2006) emphasizes that, while graphical representations are very useful for interpreting forecasts in high dimension settings, one must resort to variable importance plots and partial dependence plots in order to visualize the contribution of a few inputs at a time to the overall prediction. Thus, variable importance plots show us which variables have the most predictive power on the model. In addition, partial dependence plots help to understand the marginal impact of a feature on the predicted outcome. We provide these plots for inputs from both GT and Official data. We also check variable importance through causal inference. This method of obtaining each variable's influence on the outcome is well known in biostatistics but has not been used in the mainstream of machine learning and economics. It uses semiparametric estimation, making no assumptions whatsoever about the functional form (linear or otherwise), and the predictors are merely ranked by order of their importance.

We also provide SHAP values in order to compute the value of the contribution of each feature to the overall prediction. SHAP is an acronym for SHapley Additive exPlanations and is a theoretic approach used to interpret the output of machine learning model. Given the current set of feature values, the contribution of a new feature to the difference between the actual prediction (which includes the feature) and the mean forecast (which excludes the feature) is the estimated SHAP value of that feature. Thus, using SHAP values is a method that permits us to evaluate quantitatively the contribution of each predictor. Rigorously derived from game theory, SHAP values are singled out among all other existing additive feature attribution methods by Lundberg et al. (2018) as providing the *only* consistent and locally accurate feature attribution values while the other existing methods (such as, for example, Saabas) lack consistency and make it impossible to compare attributed values across models.

The remainder of the paper is organized as follows: Section 2 introduces the data (Official as well as GT data) and describes the forecast evaluation methodology. Section 3 presents a brief description of XGBoost. Empirical results are given in Section 4. Section 5 provides a robustness test, and Section 6 discusses different methods to interpret our forecast results. Section 7 concludes. Some additional data tables, figures, and technical material are collected in two appendices at the end of the paper.

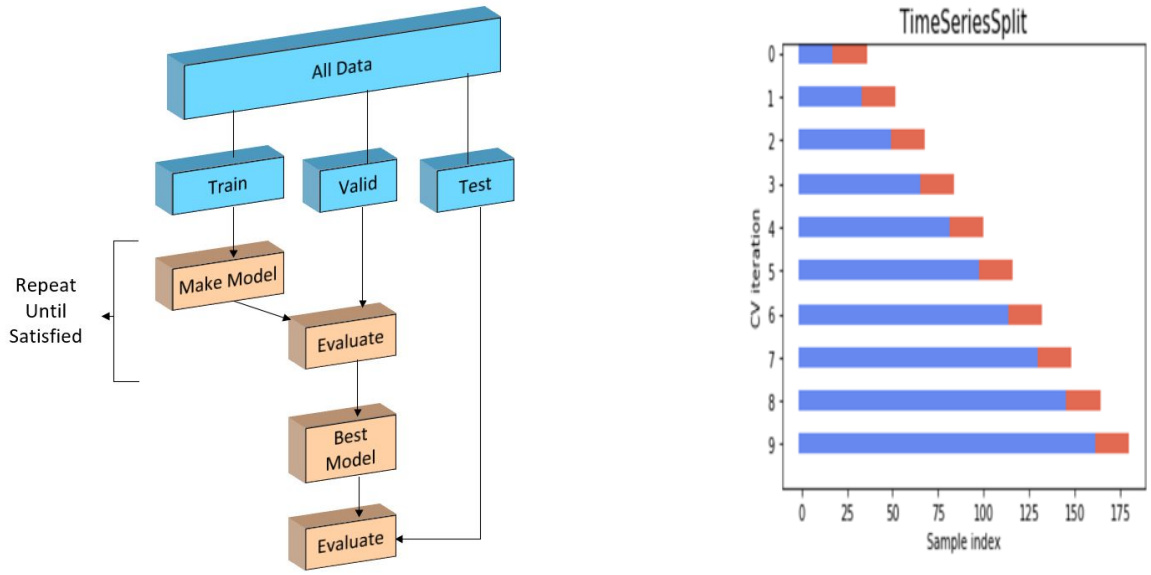
2 Data

Cross-validation (CV) is a widely used technique for tuning hyperparameters and to get robust measurements of model performance. The first CV for time series was proposed by Chu et al.

[6]The Shapley value is a method derived from coalitional game theory and named after the Nobel-prize winning game-theorist and economist Lloyd Shapley (Shapley (1953)) who developed it. It provides a way of distributing the "payout" among the players equitably given their contribution to the coalition.

(1991). It is a modification of k-fold cross-validation for time-series data. However, the built-in CVs of many ML methods are not appropriate for time-series data. Therefore, to overcome this problem, we follow two steps. First, we divide both Official data and GT data into three groups, namely: "train," "validation," and "test". Note that, following Cook (2016), the validation and test data need not have the same length. We use GT data from January 2004 (the earliest date for which it is available) until March 2019 (182 months). We use training data from January 2004 to December 2015 (143 months), validation data from January 2016-December 2017 (24 months) and test data from January 2018 to March 2019 (15 months). We use the AutoML function for variables selection.The models are trained using the training data and scored using the validation data, as shown in the Figure 1. The validation is part of the training process to evaluate the tuning of the model parameters. We then apply the tuned model to the test data to assess the forecast performance of the model and obtain a final score of the model. Second, we apply time-series CV with expanding window as shown in Figure 1. To test whether the data are stationary, we multiply by 100 the first-difference of the logarithmic (log) data for both Official and Google Trends data, then apply the Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test.

Figure 1: Step 1: train,valid and test; Step 2: time-series CV with expanding window



2.1 Official data

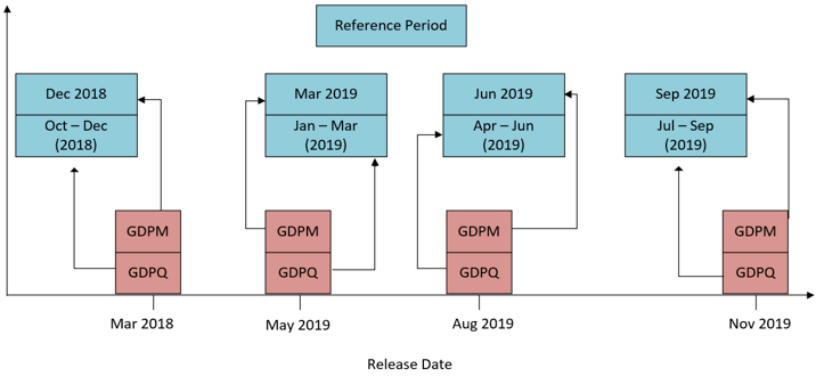
We use both Canadian and US economic variables to forecast the Canadian GDP growth.[7] We obtain data from Statistics Canada, Bloomberg, and the US Federal Reserve Board. As indicated in Table 6 that there are 15 Canadian economic variables and ten US variables. Our data is a mix of both "hard" (i.e., directly measurable) and "soft" (survey data) variables. The Official data variables used to predict the monthly real GDP growth rate are given in Table 6.The total duration for this pseudo timeline is 180 days, where the 0-30 day span represents any month of the given year, and the variables available during this period are as shown in Figure 3. There is no economic data available during the next month (30-60 days). Data available with lags of 36-54

[7]It has long been noted that some US variables have predictive value for Canadian GDP given the extensive Canada-US trading relations. A few of the variables we use have also been used in Chernis and Sekkel (2017) and Scotiabank's Nowcasting Model for the Canadian Economy.

days are shown in the 3rd month of this pseudo line (60-90 days). We can see that the monthly and quarterly GDP for Canada is released with around two months of delays.The actual release and reference periods for monthly and quarterly GDP is given in Figure 2. We have highlighted the months during which both quarterly and monthly GDP are released in the same month for the year 2018 and 2019. For example, in August 2019, monthly GDP data for June 2019 and the quarterly GDP for April-June 2019 are released simultaneously. Since we use variables from monthly GDP to nowcast quarterly GDP, this means that we can use monthly GDP data for April and May 2019 to nowcast the quarterly GDP for April-June 2019 (one month ahead of the official release date). (Quarterly GDP growth is given in Table 5.)

Our procedure is as follows. We select the best 18 variables out of 25, by using the *variable importance measure* of XGBoost. (This technique for variable selection is different from the traditional methods such as Principal Component Analysis (PCA), Partial Least Squares (PLS), Sure Independence Screening (SIS), and Least Absolute Shrinkage and Selection Operator (LASSO) used in the literature.)[8] For this purpose, we apply the AutoML option of XGBoost for training the models. We run 1000 models at a time.[9]Then, by using the variable importance measure, we drop variables with less than 4-5 % of importance.[10] The entire process is repeated until we get sufficient improvement in the out-of-sample RMSE. We typically run AutoML 5 to 15 times, which implies that we are training 5000 to 15000 models.

Figure 2: Release and reference period for monthly and quarterly GDP



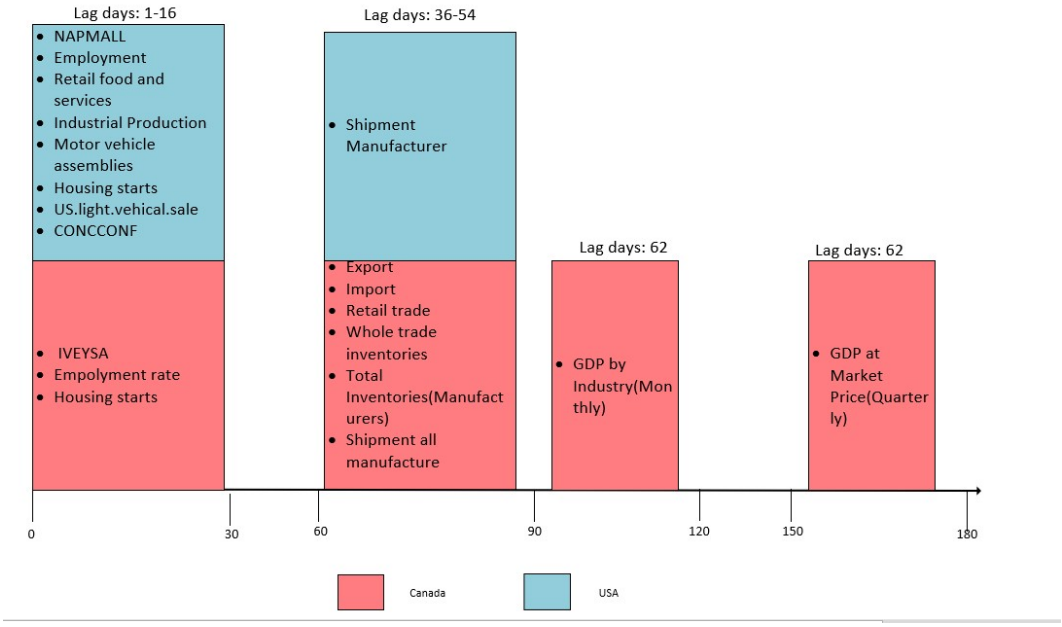
Note: GDPM stands for Monthly GDP and GDPQ for Quarterly GDP. The X-axis shows the release date, and on the top of the bar represents the reference months. It can be followed, for example, for August 2019, GDPM for June and GDPQ for April, May, and June are released at the same time.

[8]We have also tried various other measures for feature transformation and/or feature selection, including Regularized Autoencoders, PCA, Generalized Low Rank (GLRM) model, the Boruta algorithm, the Sure Independent Screening approach (SIS) proposed by Fan and Lv (2008) , SIS-Distance correlation (DC), SIS-partial distance correlation (PDC) proposed by Yousuf and Feng (2020), Distance correlation, and Lasso. However, we found that the model performed better in terms of accuracy when the features were selected with the variable importance measure of XGBoost.

[9]We found that out-of-sample performance was best when using 1000 models (instead of 800 or 1600 models). Training more models for the current AutoML project may be done by using the same project name with different random seeds. Each of these 1000 models has a different forecasting accuracy because the models mainly differ in the values of parameters such as "depth of the tree," "sample rate," type of "grow policy or booster," etc.. We choose the model with the smallest RMSE (called the leader model).

[10]We found that the best approach was to sequentially drop features with lower than 4-5% importance rather than choosing all at once the variables with the highest importance.

Figure 3: Pseudo time line for the release of Official data and GDP



Note: The calculation of number of lags days (end date included) are based release and reference period of Table 6 given in the appendix. The X-axis shows a pseudo timeline of 180 days, where 0-30 days represent any month of the given year, for example, March 2019. The monthly GDP for March 2019 should be available by the end of this month. However, the Official data for monthly GDP is released with a delay of approximately two months, i.e., May 31, 2019. In the same vein, the first-quarter GDP of 2019 should have been released by the end of March 2019 (i.e., by the end of 90 days of Figure 3). However, it is also released with around two months of delay. The days range 0-30 shows the data available in any month of the year both for the US and Canada. There is no Official data in preceding followed by the release of Official data in the third month, i.e., 60-90 days.

2.2 Google data

According to the Canadian Internet Use Survey in 2018, 94% of the population had internet access at their home, and the share of Canadians aged 15 and older who used the Internet was 91%. More importantly, approximately 84% of internet users bought $57.4 billion worth of goods or services online in 2018 compared to $18.9 billion in 2012. These figures imply that the frequency of Google searches for certain goods and services may have substantial informational content about GDP.

Google Trend (GT) data can be accessed through www.google.com/trends and are available from January 2004 to date. GT data indicate the percentage of web searches performed on www.google.com and thus reflect search trends. The data-set assesses how many searches were entered into Google's search engine for a particular term over a specified period. The data are reported as a query index rather than providing the raw number of queries Choi and Varian (2009). The index is normalized such that the highest search interest for the selected time and location is set to 100%, and every other observation calculated relative to that point. Hence, the resulting number, scaled in the range of 0 to 100, is based on the total number of searches for a specific category relative to all search queries.

We divide our GT data into two categories. The terms in the first category, which we call *general search terms*, are extracted from the R package called "gtrendsR". We download 26 categories and

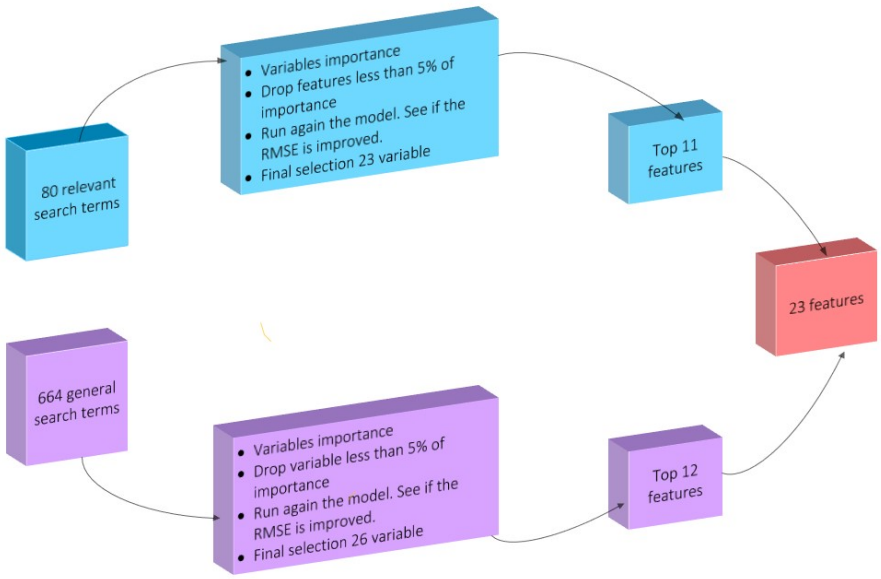
269 subcategories of GT data for a total of 1776. We are left with only 661 search terms after removing missing value columns and outliers. The second category, which we call the *relevant search terms*, takes the same variable names as those in official monthly and quarterly Canadian GDP. There are 80 search terms in this category. In the case of a column with up to two missing values for the second category, we have replaced the missing values with their mean. We use the X-13ARIMA-SEATS, a program developed by the US Census Bureau and the Bank of Spain to make seasonal adjustments to GT data-set since it is not seasonally adjusted.

2.3 Selection of Google Search terms

We use the AutoML option of XGBoost and follow the same procedure for the selection of GT terms as the one we described for the selection of Official data variables in Section 2.1 and illustrated in Figure 4. We first apply XGBoost on the *general search terms* category provided in Table 3 (in the appendix) and run 1000 such models.[11] The whole process is repeated until we obtain sufficient improvement in RMSE. Then, we use the variable importance measure and sequentially drop variables with less than 5-10 % of importance.[12] We stop running the process when we get 25 features.

Similarly, the same process is applied to the *relevant search terms* category given in Table 2. The process is stopped once we get 23 features. Finally, we combine the top features from both categories of the data and repeat the process once more. In the end, we are left with 24 features in all from both sets of data (see table 4) that are used for forecasting GDP.

Figure 4: Google Variable Selection Method



[11] Please note that we only provide a few GT terms in this table out of 1776 search terms in this category

[12] As in the case of Official data, we found that the better strategy was to drop the less essential variables in multiple stages rather than selecting the top features in one step. Note that we use a stricter measure of importance in eliminating variables in the case of GT data (5-10% importance) than in the case of Official data (4-5% importance) because GT data contain a much larger number of features.

2.4 Forecast evaluation

Mean absolute error (MAE) and root mean squared error (RMSE) are two of the most common metrics used to measure the accuracy of the predictions for continuous variables. The RMSE is the square root of the average of squared differences between prediction and actual observation. This measure gives *more weight* to large deviations such as outliers.

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y_j})^2} \tag{2.1}$$

MAE is the average over the test sample of the absolute differences between prediction and actual observation. It gives *less weight* to outliers.

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y_j}| \tag{2.2}$$

For each of these metrics, our goal is to minimize the squared deviations between the predictions and the observations, and obtain the smallest value of these metrics with our choice of predictors, $\hat{y_j}, j = 1, \ldots, n$ in the case of RMSE and MAE. Thus, these metrics provide us with a measure of the success of our predictions.

2.4.1 SHAP values

As we mentioned in the introduction, attributing SHAP values is a method that permits us to evaluate *quantitatively* the contribution of each predictor. Lundberg and Lee (2017) and Lundberg et al. (2018) prove that, grounded in game theory, SHAP values are the *only* consistent and locally accurate feature attribution values. SHAP is an additive feature attribution method, where an explanation model, $g(z')$ of the complex original prediction model f, is depicted as a linear function of simplified features. Thus, a SHAP explanation model is given by a linear function of binary variables:

$$g(z') = \phi_0 + \sum_{j=1}^{M}\phi_j z'_j \tag{2.3}$$

where $z' \in \{0,1\}^M$,with M being the number of features. (Note that z' is an indicator function such that $z' = 1$ indicates that the feature is present while $z' = 0$ indicates that the feature is absent.). The term $\phi_i \in \mathbb{R}$ is the "feature attribution" (or weight) for feature i and captures the extent to which the presence of feature i contributes to the final output. [13]The SHAP value of a feature is estimated by comparing the model's prediction with and without that feature. Mathematically, the Shapley value for a particular feature i is given as

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(M - |S| - 1)!}{M!}\left[f_x(S \cup \{i\}) - f_x(S)\right] \tag{2.4}$$

where S is a subset of the set of all features (N) excluding i ,$f_x(S \cup \{i\})$ and $f_x(S)$ are the model's predicton of x with and without feature i respectively. Thus, ϕ_i is a weighted average of these differences over all possible subsets of S in N. Since calculating SHAP values is in practice

[13]The value ϕ_0 captures represents the output of the model when all of the simplified inputs are removed.

quite complex Lundberg et al. (2018) proposed the TreeSHAP model [14] to estimate the SHAP values for tree-based machine learning models such as decision tree, Random Forest, XGBoost, and LightGBM models. The TreeSHAP algorithm is fast and computes the exact Shapely Values for decision tree-based models.(See Lundberg et al. (2018) for a detailed algorithm.) TreeSHAP computes the correct conditional expectation by $f_x(S) = E[f(x) \mid x_S]$. Thus , $f_x(S \cup \{i\}) - f_x(S)$ is the contribution made by feature i to the prediction. In section 6.1 we provide SHAP values to evaluate the contribution of GT and Official data variables to the prediction of the real GDP growth rate.

2.4.2 BreakDown Plots

Aside from using the SHAP values, we also apply the "breakDown" process to find out the contribution of each predictor present in the model.[15] The primary goal of this process is to decompose the model prediction into different parts to compute localized variable importance scores (for an algorithm, see Staniak and Biecek (2018)). It uses the expectation of the prediction function conditional on independent features, as in SHAP values. However, it does not use an explanation model as in the SHAP value calculations but, instead, it directly estimates the variable attribution for the selected data. Moreover, it applies Friedman (2001)'s greedy function approximation approach at each step to add features with maximum contributions to prediction and iterate until all the features are added. The breakDown process is also faster than the SHAP values method.

2.4.3 Partial Dependence Plots (PDP)

As we indicated in the introduction, Partial Dependence Plots (PDP) are advocated by Friedman (2001) and Friedman (2005). The plot shows the marginal relationship between the variable of interest (e.g., the Canadian GDP) and a single predictor from the model. It can show us whether this relationship is linear, monotonic or more complex. We use the pdp R package introduced by Greenwell (2017). Following Greenwell (2017) and Molnar (2019), let $x = \{x_1, x_2, \cdots, x_p\}$ be the set of features of a given model whose prediction function is $\hat{f}(x)$. We separate the set of features x into two subsets, a subset of "interest" z_s and its complement, z_c. The partial dependence function for regression is defined as:

$$\hat{f}_s(z_s) = \int \hat{f}(z_s, z_c)\, dP(z_c) \tag{2.5}$$

where $dP(z_c)$ denotes the marginal probability density of z_c which is estimated on the basis of the training data.The subset z_s contains the features for which PDP are to be made while subset z_c contains remaining features that are used for training the model. Partial dependence are found by averaging out the effects of others feature in z_c. Eq (6) can be computed by finding the average of the training data of size, n, as follows:

$$\hat{f}_s(z_s) = \frac{1}{n} \sum_{i=1}^{n} \hat{f}\left(z_s, z_c^{(i)}\right) \tag{2.6}$$

[14] We find TreeSHAP values for XGBoost from H2O. The *shapper* package in R and *shap* in Python can also be used to get SHAP values.

[15] This process has been implemented into the R package called breakDown.

2.4.4 Variable Importance Plots

One of the advantages of using gradient boosting is to get importance scores for each feature after the boosted trees are constructed. Variables with high importance are the primary drivers of the outcome, and their values have a significant impact on explaining the model. We use the DALEX R packages and H2O to determine variable importance. In H2O, the variable importance scores are computed by finding the relative influence of each variable and whether that variable was selected to split during the construction of the decision trees, in other words, according to the extent to which the feature contributed to reducing mean squared error. In the regression, the relevant improvement is the difference in the sum of squared errors between that node and its child nodes. The DALEX package uses the permutation-based variable importance approach proposed by Fisher et al. (2018) to compute the variable importance scores. The model-agnostic variable importance score is calculated by permutation after the model has been fitted. We randomly shuffle one of the columns of the validation data or of the test data while keeping the other columns and the target variables unchanged. The prediction is made using the resulting data, and then this prediction is used to calculate target values. A feature is considered important if shuffling changes the error of the model because this indicates that the model depended more heavily upon this feature for prediction. If the error of the model remains unchanged after the shuffling, then the feature is deemed as unimportant since the model does not take into account this feature for prediction.

3 Extreme Gradient Boosting (XGboost)

As mentioned above, XGBoost is a short name for the eXtreme Gradient Boosting algorithm developed by Chen and Guestrin (2016) which has recently been dominating in applied machine learning and data science competitions at Kaggle. It is designed for speed and performance, and is a more efficient version of gradient boosting decision trees. We provide an example in Figure 16 (in the appendix).[16] Boosting is an ensemble meta-algorithm that combines the outputs of many "weak" learners to turn them into strong learners. The premise of boosting is to train weak learners sequentially such that each subsequent tree aims to reduce the errors of the previous trees. Models are added sequentially until no further improvements are made. The predictions are combined through a weighted average of regressions to produce the final prediction.[17]Boosting employs a nonparametric additive model in which each additive component function is constructed with decision trees.

The main idea of the XGBoost can be briefly described as follows. For a given data set of size T with N features, say $(y_t^*, \boldsymbol{x}_t^*)_{t=1}^T$ with $\boldsymbol{x}_t^*$ being a vector of N features, the model predicting the output y_t^* using the features $\boldsymbol{x}_t^*$ as predictors is a weighted additive model of the form: $y_t^* = \sum_{k=1}^K \alpha_k f_k(\boldsymbol{x}_t^*) + \epsilon_t$, where $f_k(\cdot)$ for $k = 1, \ldots, K$ are independent base learners (often characterized by regression trees), α_k are weights, and ϵ_t is a random error term. The XGBoost estimates both

[16] In Figure 16 we provide a decision tree using H2O's GBM as an example. It builds successive trees sequentially with reduced error in each subsequent iteration.

[17] Gradient boosting is an approach where new models are created that predict the residuals or errors of prior models and then are added together to make the final prediction. It is called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models. Gradient descent is a method that consists of obtaining a vector of coefficients that minimize a loss function. The coefficients represent local minima. The general idea of gradient descent is to tweak parameters iteratively to minimize a loss function. Extreme gradient boosting builds a forest of trees in an additive manner. The algorithm iteratively produces trees that minimize the prediction error, and ultimately provides an optimal set of predictive trees. If a new model is not satisfactory, new regression trees are added sequentially to reduce the error in predictions. The trees grow in this sequential manner, such that each tree is produced using information from previously grown trees. Each tree is fit on a modified version of the original data. The XGBoost library implements the gradient boosting decision tree algorithm.

the weights α_k, $k = 1, \ldots, K$, and the associated base learners $f_k(\cdot)$ by *sequentially* minimizing a penalized differentiable convex loss function of $y_t^* - \sum_{k=1}^{K} \alpha_k f_k(\boldsymbol{x}_t^*)$ (with respect to both α_k and $f_k(\cdot)$, $k = 1, \ldots, K$) over K boosting iterations. We can choose equal weights, say $\alpha_k = 1$ for $k = 1, \ldots, K$. But, in this case, we minimize the penalized convex loss function with respect to $f_k(\cdot)$, $k = 1, \ldots, K$. The purpose of penalizing the complexity of the model (constructed by regression trees) is to avoid overfitting so that the algorithm is more likely to select a simple model with good prediction power. A technical description of XGBoost is provided in Appendix B.

4 Empirical Results

We present our empirical results in Table 1. In the top panel of the table, for comparison purposes, we provide the RMSE, MSE, and MAE for the benchmark forecasts of monthly real GDP growth rates using Official data, obtained from three different methods: the first-order auto-regressive model, the original boosting model, and the gradient boosting model (GBM) respectively. In the bottom panel, we present the summary statistics of the out-of-sample forecast performance of XGBoost [18] for GT data, Official data, and for GT data combined with Official data that are used to predict the monthly GDP growth, as well as for the Official data that are used to predict the quarterly GDP growth rate. As indicated in the first column, we use the RMSE and the MSE, as well as the MAE as performance metrics. Comparing the metric values obtained for all three models in the top panel of Table 1 with the metric values obtained in the third column (pertaining to Official data) of the bottom panel, we can see that XGBoost performed much better than the base model in terms of forecasting accuracy.

Table 1: Out-of-sample forecast evaluation summary statistics for the real GDP (RGDP) growth rate

Monthly GDP growth with Official data			
Metric	AR(1)	Boosting	GBM
$RMSE$	0.09823	1.86999	0.05362
MSE	0.00964	3.49689	0.00287
MAE	0.08067	1.869997	0.04578

	Monthly GDP growth			Quarterly GDP growth
	with GT data	with Official data	with GT+Official data	with Official data
Metric	XGBoost			XGBoost
$RMSE$	0.028757	0.019763	0.026128	0.019234
MSE	0.000827	0.000391	0.000682	0.000369
MAE	0.024030	0.016845	0.021812	0.016082

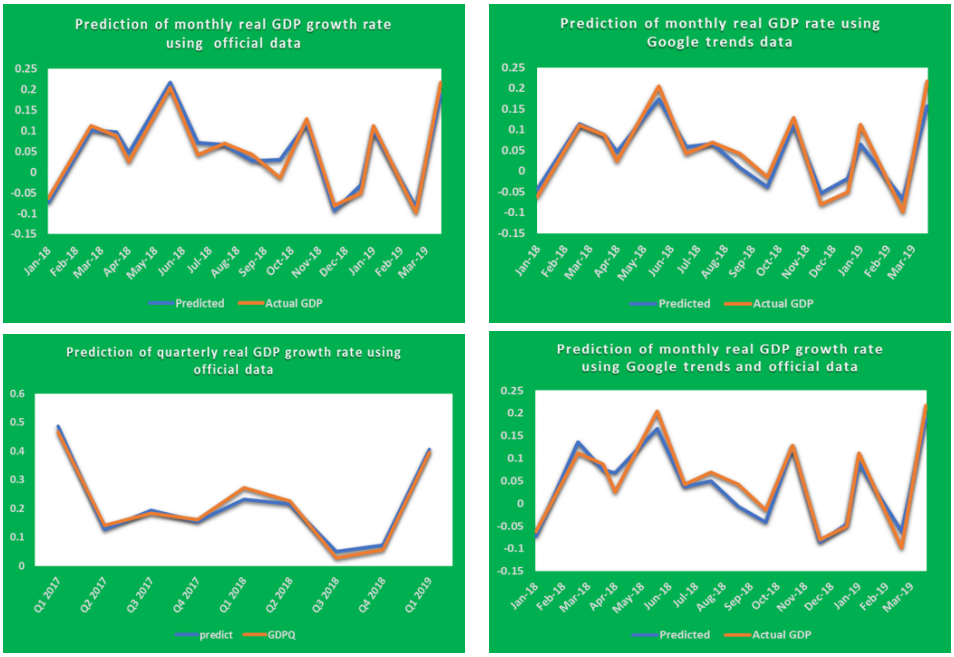
We can see [by comparing the first column of the bottom panel of Table 1 with all three columns of the top panel] that there is an improvement in the performance of XGBoost relative to the baseline methods even when we use GT data alone (when Official data are not available). Comparing the metrics in the first and third columns of the bottom panel, we also observe that when GT and Official data are used together, forecasting accuracy is further improved relative to using only GT

[18] We use R pacakge of H2O version 3.26.0.10 for XGBoost. H2O does not yet support CV for time-series. Hence we the *nfolds* option (which specifies the number of folds to use for CV) equal to zero to indicate the absence of CV.

data, as can be expected. We observe a slight improvement in the RMSE, the MSE, and the MAE when both GT data and Official data are used over the case when GT data is used alone. The reason behind these results is that we have a much better forecasting accuracy with the Official data than with GT data. Finally, the metrics obtained in the second column compared to those of both the first and the third columns indicate that forecasting accuracy using Official data by itself clearly dominates the accuracy of using only GT data. These results are reminiscent of Ferrara and Simoni (2019) and Götz and Knetsch (2019).

In contrast to other studies, we also plot the predictive power of XGBoost's out-of-sample forecasts for the period January 2018-March 2019. As is clear from Figure 5, the model provides a prediction close to the the actual GDP growth rate, tracking well both the ups and downs of the real GDP growth rate.

Figure 5: Predicted vs. actual RGDP growth rate

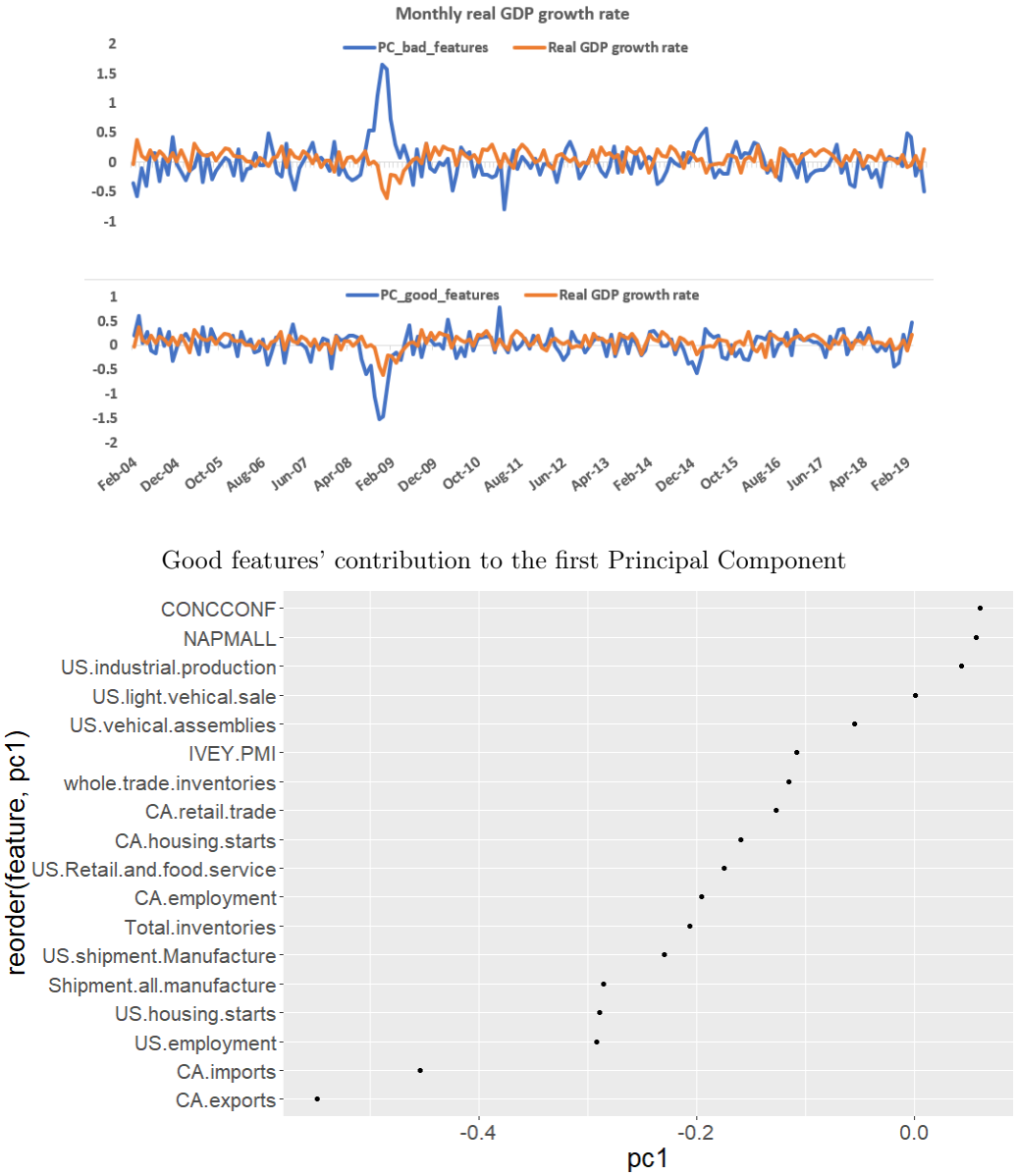


Götz and Knetsch (2019) have emphasized that feature selection is of paramount importance in determining the results. We therefore combined the feature selection method of XGBoost with PCA to choose the best factors out of the predictors selected by XGBoost. We select the best GT features using by using the *variable importance measure* of XGBoost. We further apply PCA to construct principal components (PCs) for these features.[19] Moreover, we further add three lags of GDP growth and two lags of the PCs. The performance of the model is then sufficiently improved for forecasting the monthly real GDP growth rate with Official data. We adopt the same strategy for forecasting quarterly GDP growth with Official data, but we only add one PC, and its once-lagged value, and we do not lag GDP. In the case of GT data, we also use the same strategy, but since forecasting performance worsens when we add lags of PCs or real GDP growth rates, we do not add these lags. As shown in the top panel of Figure 6, we find that the PCs can track well real GDP growth, including the financial crisis of 2008-09 and the oil-price shock of 2014, *only* when

[19] We apply the PCA in a unsupervised manner to avoid any data leakage.

the "good" features (i.e., the ones selected by XGBoost) are used, but fail to track the real GDP growth rate with the "bad" features (i.e, the non-relevant ones, those *not* selected by XGBoost). For illustrative purposes and as an attempt to "open up the black box," we provide a list of "good" features and their contribution to the first principal component (PC-1) in Figure 6.

Figure 6: Forecast of the RGDP growth rate using PCs with 'good' and 'bad' features for Official data



Note: The two top panels plot the actual GDP growths versus their forecasts using PCs (a) with features selected by the *variable importance measure* of XGBoost ('good' features) and (b) the features not selected by XGBoost ('bad' features) respectively. In the bottom panel graph, we open up the 'black box' of the PCA, and show the contribution of each 'good' feature to the first PC (PC1)

5 Robustness test with Driverless H20 AI

Driverless H2O AI is an automatic ML platform that fully automates feature engineering, model validation, hyper-parameter tuning, and model selection.[20] To provide a point of comparison for our application of XGBoost, we used the automatic algorithm selection feature provided by the Driverless H2O AI platform to choose a model for predicting the real GDP growth rate using Official data. The automatically selected algorithm was the LightGBM , introduced by Microsoft, which is faster than XGBoost.[21] In our case, LightGBM uses 25 original features and 22,483 engineered features. It trained and ranked 3552 models to evaluate the engineered features further, a process that took 16 hours and 38 minutes to complete. The model yields a RMSE value of 0.08 on the test data, which is inferior to the RMSE obtained by XGBoost. We also applied LightGBM to the same data using the ten-fold time-series cross-validation (TSCV), which produces a better forecast accuracy with an RMSE value of 0.065, but still inferior to our earlier results.[22] The TSCV is outlined in Figure 1 above (i.e., the model selects a sample of data for training, then uses the remaining of the sample for validation). The data are divided into sub-samples ten times. One sub-sample is selected as the test sample and the other sub-sample as the training sample. The first-fold training sample comprises observations between 0-18 (historical), and the 'future' 19-34 observations are used for testing. Then the second-fold training sample selected is (say) 0-50 (historical), and the 51-66 (future) observations are used for testing. This process continues until all the data is exhausted. In the same vein, we also used the TSCV with XGBoost, and obtained a RMSE value of 0.06580. In other words, its performance is worse than XGBoost used earlier on. Hence, we conclude that, for our given data set, XGBoost with the three-step "train, valid, and test" strategy presented in the previous section of this paper is a more powerful forecast procedure.

6 Interpretable machine learning (model interpretability)

We apply both global and local interpretation techniques to interpret (understand) out forecast models. The *variable importance measure* and *partial dependence* plots provide a global perspective. They identify the variables of the most significant importance and the predictor-response relation across the observations. On the other hand, the *SHAP* value and the *breakDown* process are tools for local interpretation. They highlight the contribution of each predictor to the predicted value.

6.1 Local interpretation: SHAP values

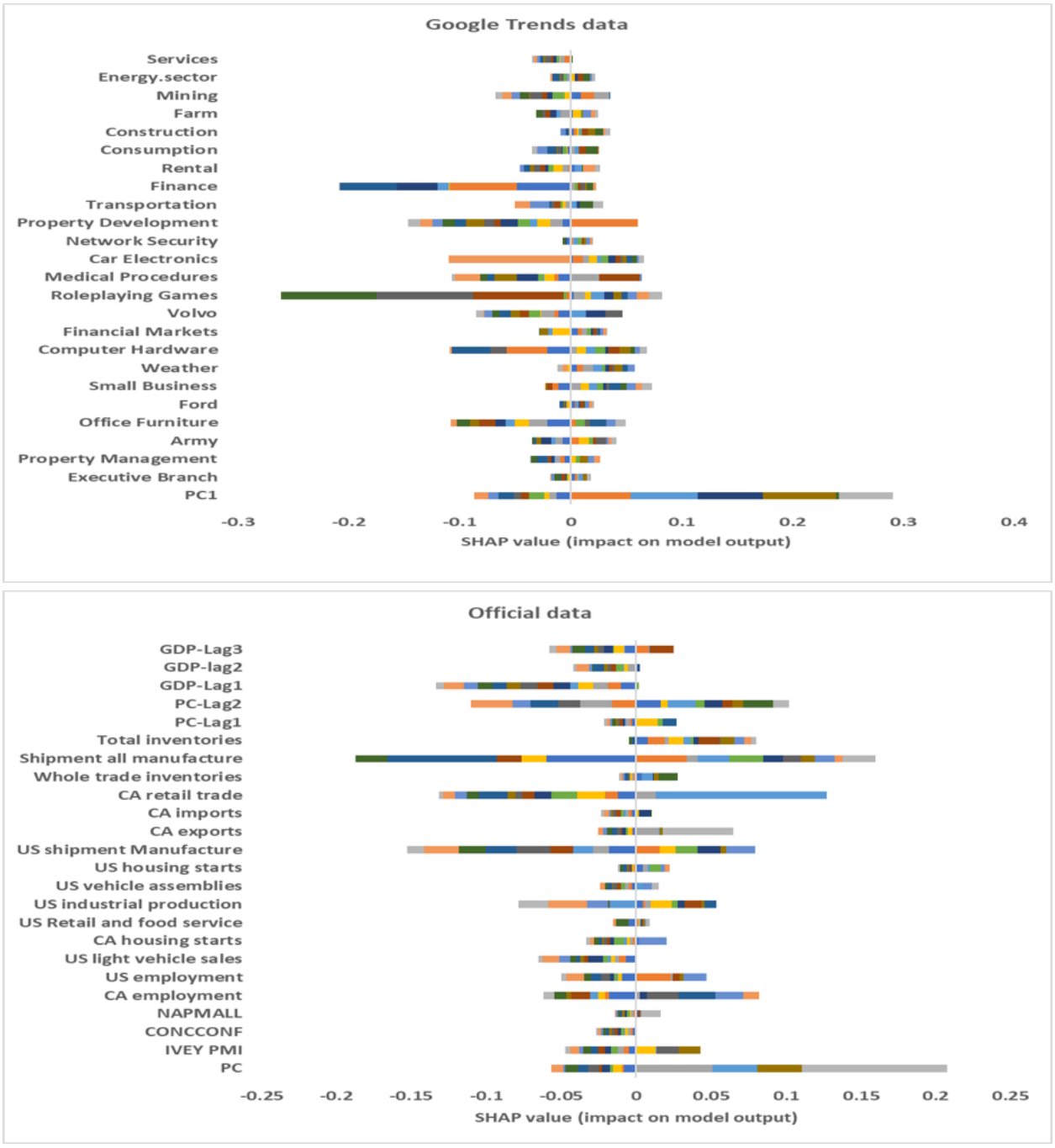
As we discussed in section 2.4.1, SHAP values help us to understand which features contribute most toward a better accuracy of a model. It computes the importance of a feature by comparing the model's forecast *with* and *without* that feature. Figure 7 shows the impact of each feature (Google search terms) on the model output variable (the real GDP growth rate). For example, as seen in the top panel of this figure, the GT data on Finance, Property Development, and Roleplaying games

[20] Driverless H2O AI provides an Automatic Machine Learning Platform, where we can run many algorithms such as XGBoost, LightGBM, GLRM, Isolation Forest, and decision trees. It is suitable for time series data and provides a rolling-window based prediction

[21] LightGBM differs from XGBoost in that it grows the trees vertically, i.e., leaf-wise, whereas XGBoost grows trees level-wise. It selects and splits the leaf that contributes the most to reducing the loss. Moreover, it uses a technique called Gradient-based One-Side Sampling (GOSS) that filters out the data instances for figuring out a split value. At the same time, XGBoost utilizes a pre-sorted algorithm and a histogram-based algorithm for finding the best split.

[22] We use the scikit learn library in Python for TSCV.

Figure 7: Contribution of GT and Official data to the prediction of real GDP growth rate



are the features that contribute the most to the prediction of the real GDP growth rate.[23] The importance of these features is also evident from Official data sources. In fact, according to Statistics Canada, the two sectors (Finance & Insurance and Real-estate, Rental & Leasing) contribute $131.781 billions and $254.048 billions, respectively to the GDP for November 2019. Also, according to the reports of the Entertainment Software Association of Canada (ESAC), the video game industry contributed a total of $4.5 billions to the Canadian GDP in 2019 (a 20 percent increase compared to 2017).

The bottom panel of Figure 7 indicates that the weather, Executive Branch, Ford, and Network Security made weaker contributions to the model prediction. Similarly, for Official data used for the forecast of the real GDP growth, the variables "shipment of all manufacture," "US shipment manufacturer," "CA retail trade," "US industrial production," "CA employment" and the "Ivey PMI," are important variables, each making a significant contribution to the prediction of the real GDP growth rate.[24] On the other hand, variables such as the Conference Board Consumer Confidence (CONCCONF) CA[25] imports, US retail, and food service and NAPMALL[26] are less important.

6.2 Local interpretation: Model predictions via breakDown

We use the breakDown process to look at the further contribution of each feature, as shown in Figure 8. The contribution of Weather, Network security, and Transportation on the predicted values of the real GDP growth rate are zero. The Finance, Roleplaying games, Mining, and Services contribute significantly to the prediction of real GDP growth rate using GT data. In the case of Official data, we see that CONCCONF and CA imports make zero contribution to the predicted values of the real GDP growth. At the same time, the latter is influenced negatively by shipment of all manufacture, US shipment manufacturer, CA retail trade, and CA employment, and positively by US industrial production and US housing starts. The contribution of these features is consistent with the results obtained with SHAP values.

[23] It is not possible to search composite terms such as "professional, scientific and technical services" in Google Trends. (It gives a "not enough data to show" message for the search.) In the case of searches involving terms with fewer words such as "finance and insurance," Google Trends will fetch the data, but the strength of the data will be weaker compared to searching for the terms "finance" and "insurance" separately. Therefore, we have chosen mostly single words such as "finance" as our search terms.

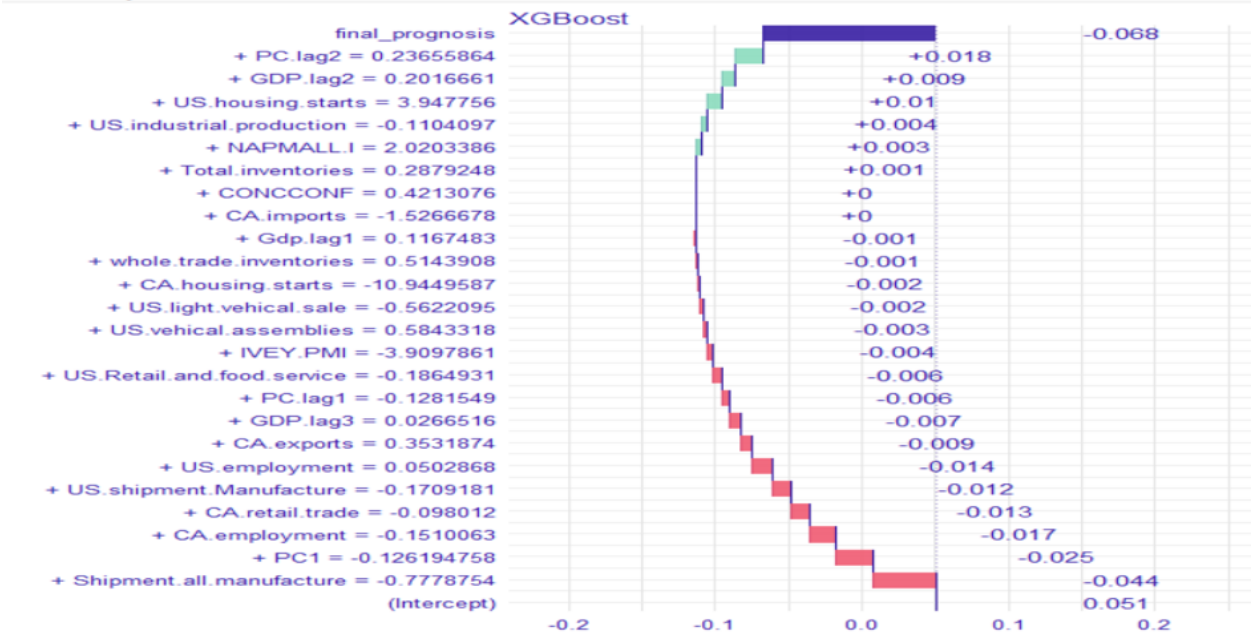
[24] The Ivey PMI is the Ivey Purchasing Managers Index released by the Richard Ivey School of Business is the leading economic indicator for forecasting business conditions in Canada. An Ivey PMI above 50 indicates an expansion of the manufacturing sector, while an index below 50 is deemed to be an indication of a contraction.

[25] The CONCCONF index provides details about consumer sentiments and buying intentions in the US. The Conference Board publishes it.

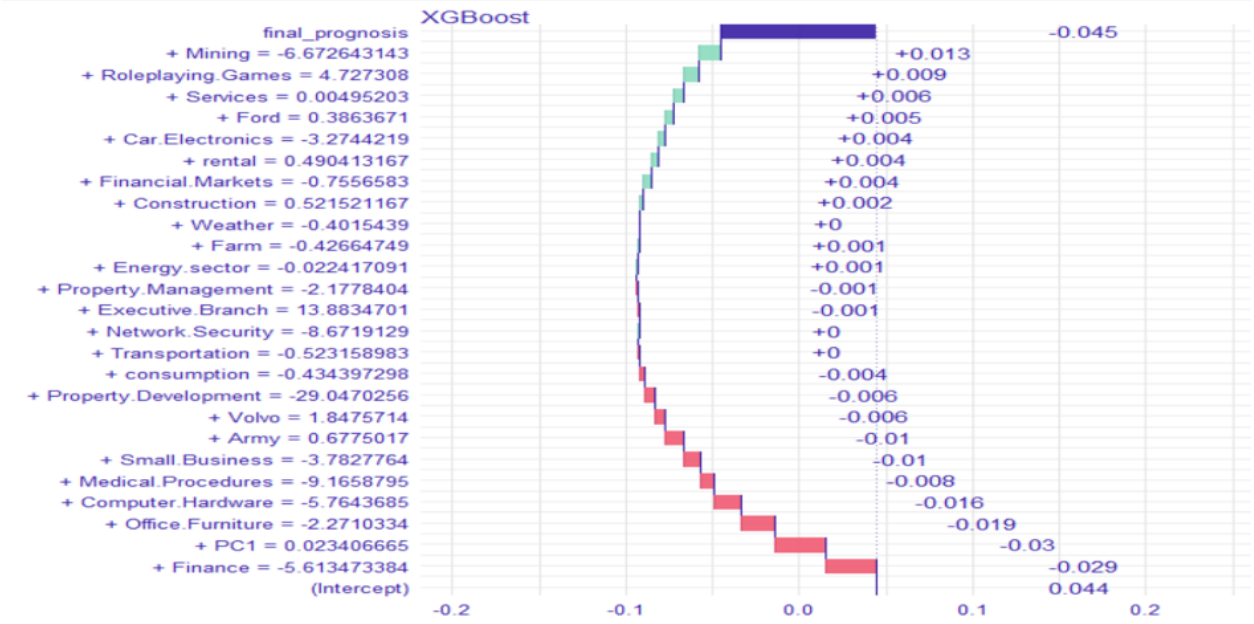
[26] The value of economy weighted manufacturing and non-manufacturing for the US provided by the Institute of Supply Management.

Figure 8: Contribution of each predictor to the real GDP growth rate forecast (generated by *breakDown*)

Official data



Google Trends

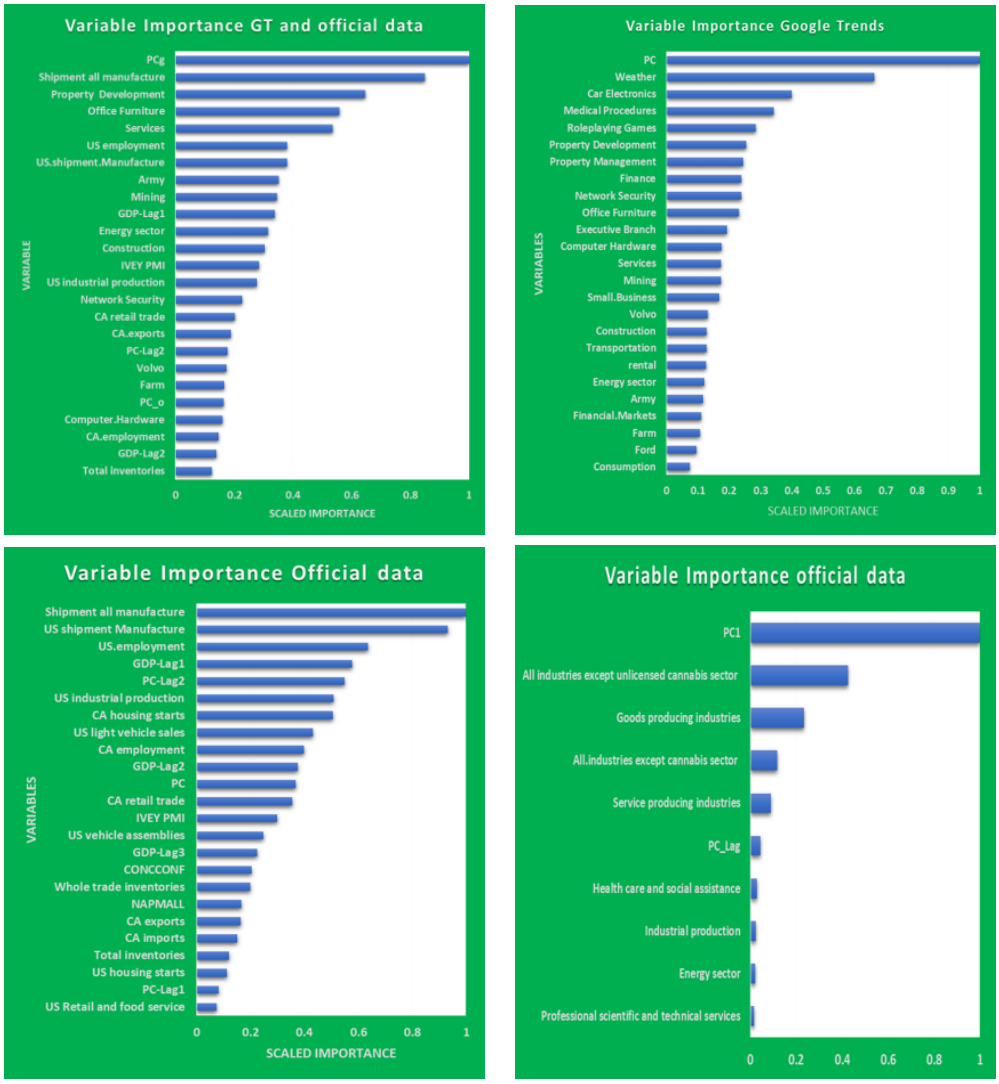


6.3 Global interpretation: Variable importance plot

A crucial task in ML is to know which variables can determine the predicted outcome. For predicting the real GDP growth rate with Official data, the shipment of all manufacture, US shipment manufacturer, US employment, property development, and services are all important predictors

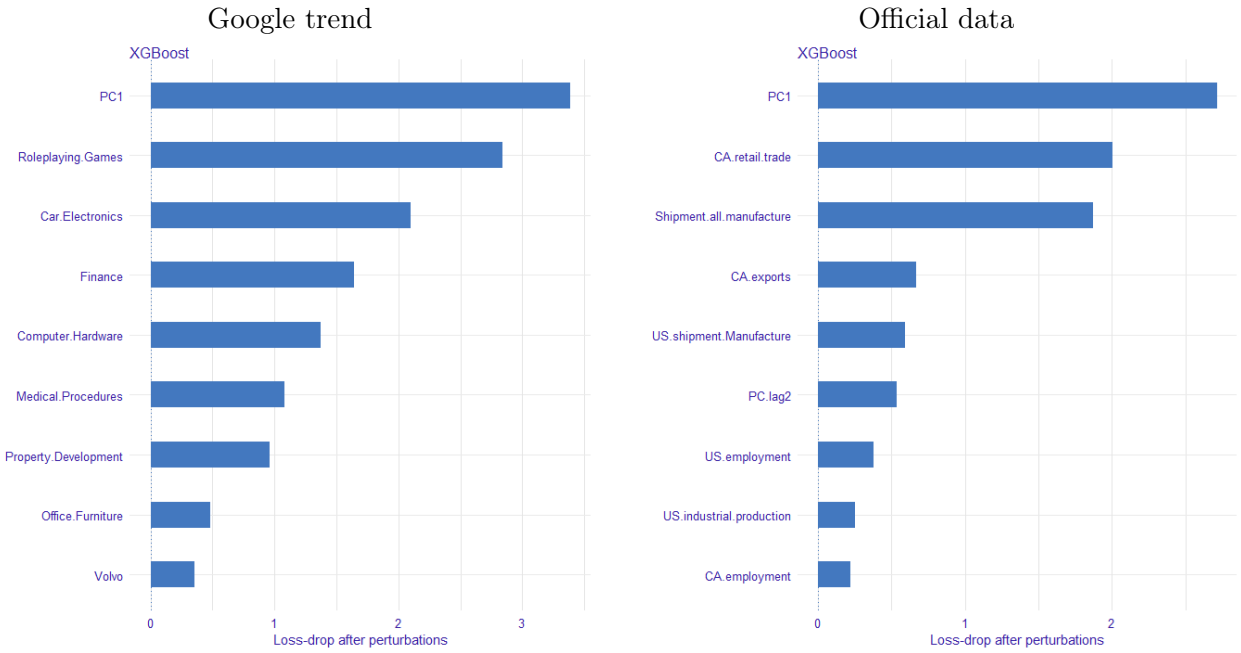
(as shown in Figure 9). The variable importance plot with the *DALEX* package is provided in Figure 10.[27] For Official data, it indicates that CA retail trade, the shipment of all manufacture, CA exports, US shipment manufacturer, US employment, US industrial production, and CA employment are the most prominent features. The important predictors of the real GDP growth rate selected by H2O are weather, Car electronics, Property development, and Roleplaying games. However, we have found that 'weather' has zero contribution. Note that 'weather' only has a minimal contribution when we evaluate it with breakDown and with SHAP values. As for GT data, the crucial features selected by the DALEX package are Roleplaying games, Car electronics, Finance, and others as shown in the bottom panel of Figure 10.

Figure 9: Variable importance plot using *the H2O AI platform*



[27] DALEX is an R package providing a set of tools for descriptive machine learning explanation ranging from global to local interpretation methods.

Figure 10: Variable importance plot using *DALEX*



6.4 Global interpretation: Partial dependence plots

We provide partial dependence plots (PDP) for a few variables for Official and GT data by using a calibration option. We first use the R package *RemixAutoML* to generate PDPs. We apply the aggregate mean based on summary statistics. These plots depict the value of the target variable compared to the predicted value. These plots permit us to see if our predicted values match those of the target variable, for all values of the independent variable. The two lines would be very close to each other for a well fitted model, while there would be a more considerable gap between these lines for a poorly fitted model. We found that, for both Official and GT data, the model performs remarkably well in predicting the real GDP growth rate, as shown by Figure 11 and Figure 12. Moreover, the PDP based on the DALEX package for the nine variables used to predict the real GDP growth rate (Figure 13) shows the mean response of predicted outcomes to each feature separately. As seen, shipment of all manufacturing, CA exports, US shipment manufacturer, and CA retail trade are important predictors of the Canadian real GDP growth rate. Similarly, PDPs for GT data in Figure 14 show that finance, car electronics, and property development move up with the predicted response of the real GDP growth rate.

6.5 Variable importance through causal inference

We apply *causal inference statistics* to calculate the variable importance indices for the Official data. Causal inference is estimated semiparametrically, making no assumptions on the functional form of the relationship, and the predictors are ranked according to their orders of importance (see, e.g., Hubbard et al. (2018)). Figure 15 shows that the shipment of all manufacture, CA retail trade, US employment, CA employment, CA exports, US housing starts, and US shipment manufacturer all have significantly positive impact on the real GDP growth rate. In contrast, CA imports and CONCCONF have a negative effect. The impact of CONCCONF is minimal. Hence, overall the

important variables identified by causal inference closely matche the ones singled out by the methods used above.

Figure 11: Partial dependence plot for Official data (actual vs. predicted)

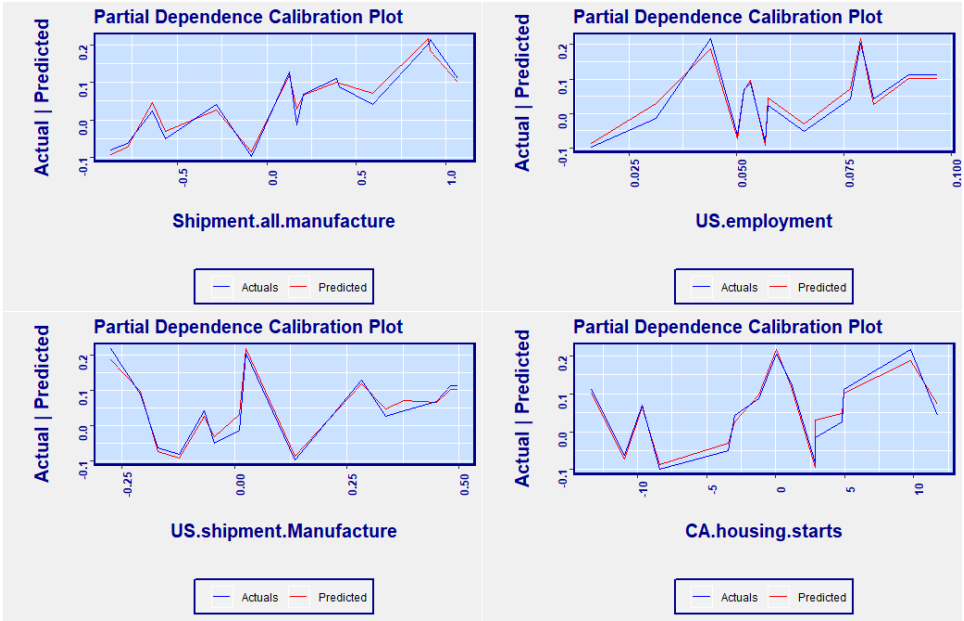


Figure 12: Partial dependence plot for GT data (actual vs. predicted)

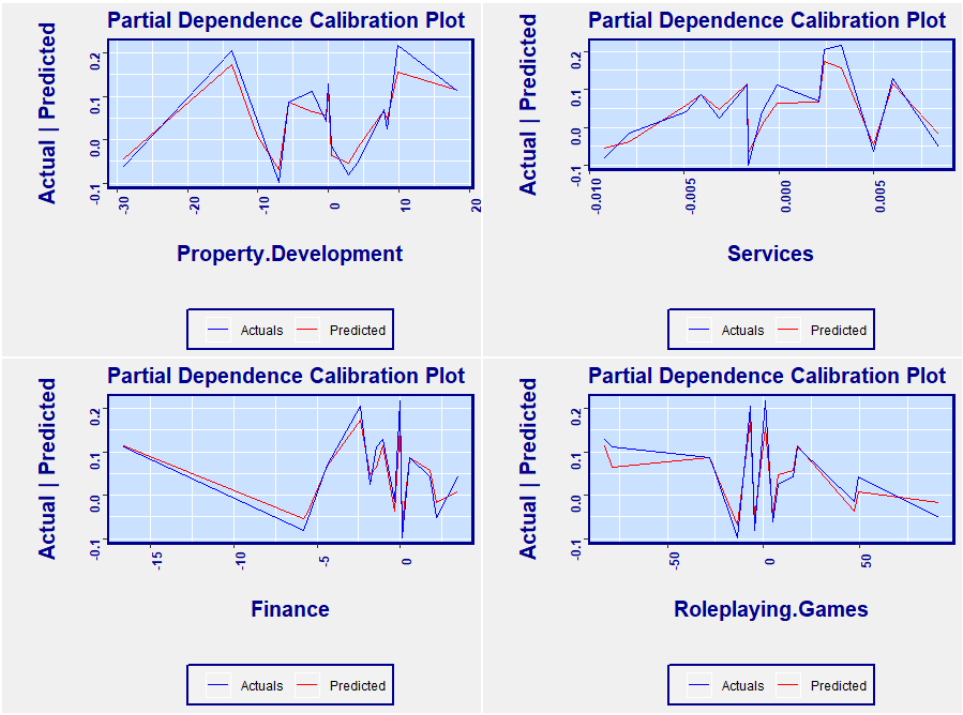


Figure 13: Partial dependence plot for Official data (mean response of the prediction to each feature)

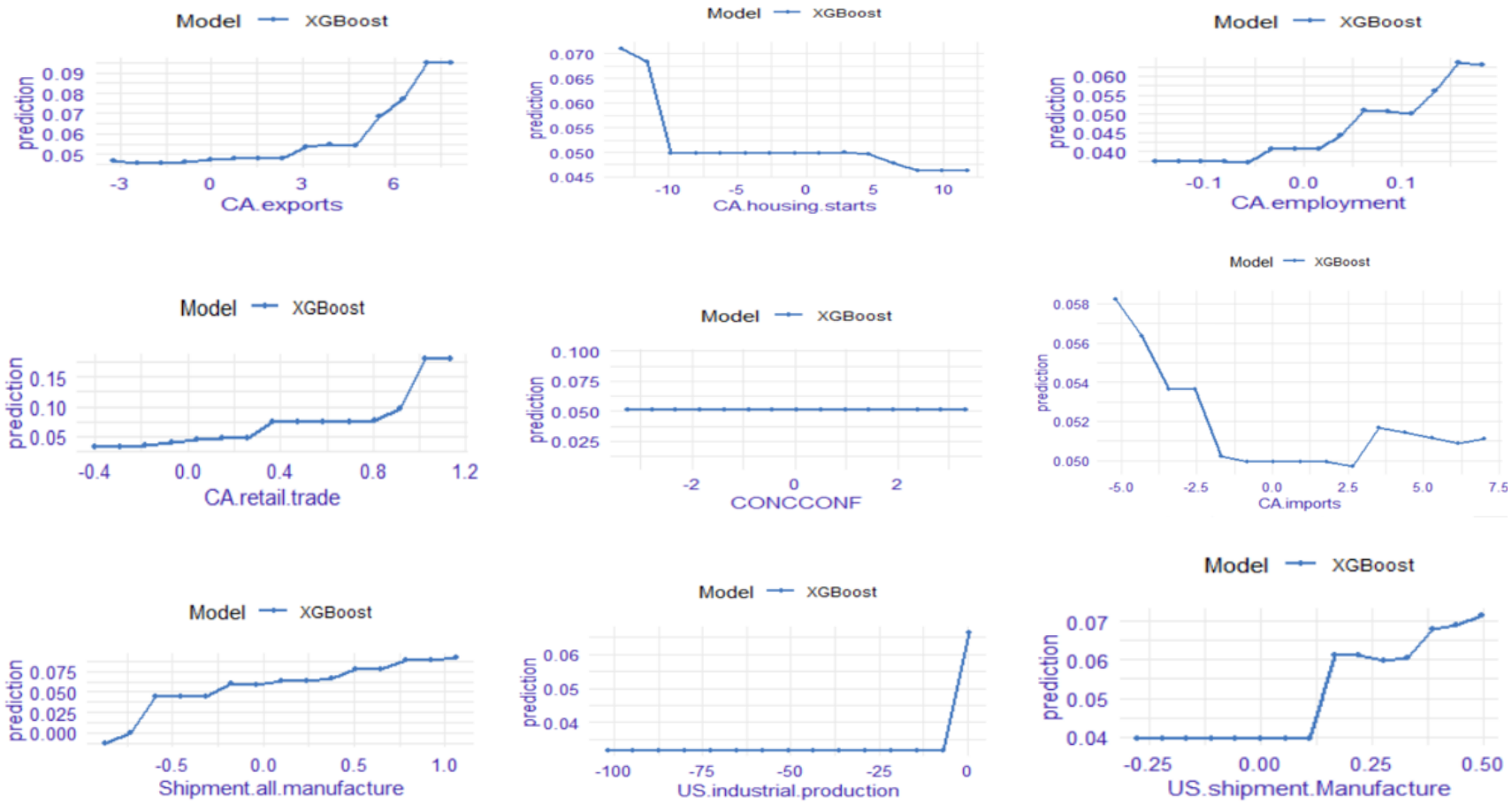


Figure 14: Partial dependence plot for GT data (mean response of the prediction to each feature)

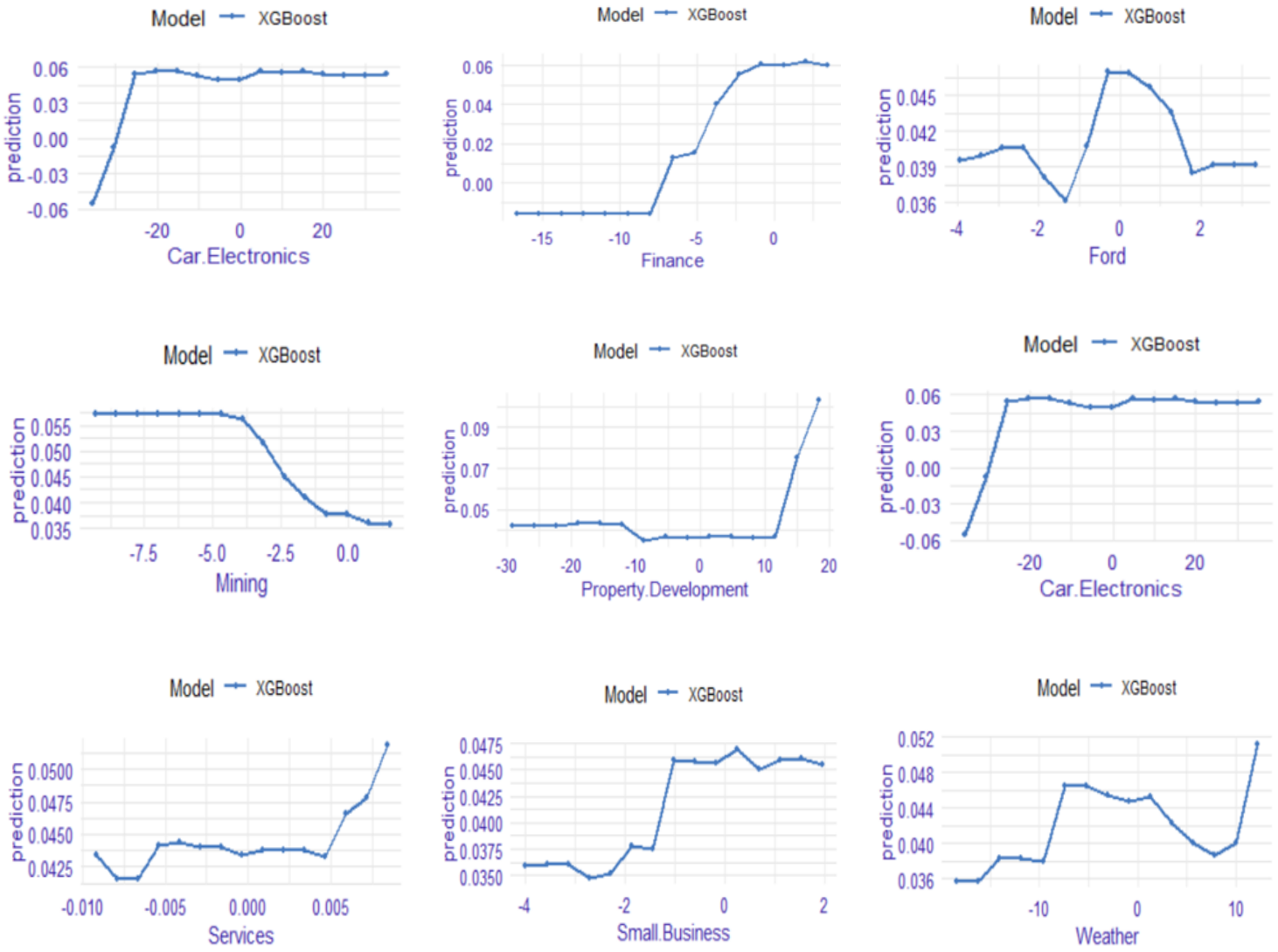


Figure 15: Variable importance for Official data

7 Conclusion

This paper presents a new machine-learning (ML) approach to forecasting the Canadian GDP growth. In particular, we use automated machine learning (*AutoML*) together with *XGBoost* to select most relevant variables from a broad set of potential candidate variables obtained from Google Trends and Official data. We select the best features to construct forecasts by using the *variable importance measure* function of XGBoost, which is based on the out-of-sample forecast performance of a model. We run 1000 XGBoost models at a time, and run AutoML 5 to 15 times until the *RMSE* is sufficiently improved. We drop variables with less than 5-10 percent of variable importance. As Collins (2009) put it in his book *Good to Great*, 'getting the right people on board is the key to success.' In our case, selecting the right features is the key to good forecasts. We conclude that XGBoost is superior to other algorithms in this respect.[28] We also find that dividing data into three groups (train, valid, and test) is a better strategy to achieve a good forecasting accuracy.

We also evaluate the real-time forecasting accuracy of XGBoost by using it to nowcast monthly and quarterly real GDP growth in Canada. We use Google Trends data together with Official data to forecast monthly GDP growth rates and only Official data for quarterly GDP forecast. Our results

[28] We have also experimented with other algorithms, such as Distributed Random Forest, GBM, Adaboost, and Light-GBM on the selected features to forecast the GDP. However, we have found that XGBoost performs the best. One advantage of AutoML is that it can be executed with only a few lines of codes, maintained and updated easily. However, it could be a time-consuming process as it takes an average of 1.5 hours to run the 1000 models in our case, although the duration may vary, depending on the data. By contrast, XGBoost used to select features with hyper-tuned parameters takes only 30 seconds to run.

indicate that XGBoost is a useful tool for macroeconomic prediction, and that Google Trends data can be a suitable alternative to Official data to predict monthly Canadian real GDP growth rates. We argue that, while GT data cannot replace Official data, it allows us to predict the monthly and quarterly GDP growth ahead of the release of Official data with a substantial degree of accuracy.

ML methods are sometimes criticized as 'black box' models as they do not lend themselves to good interpretation for policy-making purposes. We have addressed these issues by including tools that can make ML more interpretable. To this end, we provide partial dependence plots, variable importance plots, and SHAP values. We show that these interpretable machine learning methods all select similar important features for forecasting the Canadian real GDP growth rate. With Official data, they all select CA employment, CA export, CA retail, trade, the shipment of all manufacturing, US shipment manufacturer, and US industrial production as the most important predictors of the Canadian GDP growth. With Google Trends data, they all select finance, roleplaying games, car electronics, and property development as the important predictors. Interestingly, we also find that PCA can track the target variables quite well when using only 'good' features. We will address the issue of constructing forecast intervals using XGBoost in a future research.

References

Biau, O. and A. D'Elia (2009). Euro area GDP forecasting using large survey datasets. In *A random forest approach. Available via: http://unstats. un. org/unsd/nationalaccount/workshops/2010/moscow/AC223-S73Bk4. PDF*. 2

Chen, T. and C. Guestrin (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794. ACM. 2, 11, 33

Chernis, T. and R. Sekkel (2017). A dynamic factor model for nowcasting Canadian GDP growth. *Empirical Economics 53*(1), 217–234. 2, 5

Choi, H. and H. Varian (2009). Predicting initial claims for unemployment benefits. *Google Inc*, 1–5. 2, 7

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

Chu, C.-K., J. S. Marron, et al. (1991). Choosing a kernel regression estimator. *Statistical Science 6*(4), 404–419. 4

Collins, J. (2009). *Good to Great (Why some companies make the leap and others don't)*. SAGE Publications Sage India: New Delhi, India. 24

Cook, D. (2016). *Practical machine learning with H2O: powerful, scalable techniques for deep learning and AI*. " O'Reilly Media, Inc.". 5

Fan, J. and J. Lv (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 70*(5), 849–911. 3, 6

Ferrara, L. and A. Simoni (2019). When are Google data useful to nowcast GDP? an approach via pre-selection and shrinkage. 3, 13

Fisher, A., C. Rudin, and F. Dominici (2018). All models are wrong but many are useful: Variable importance for black-box, proprietary, or misspecified prediction models, using model class reliance. *arXiv preprint arXiv:1801.01489*. 11

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232. 3, 10

Friedman, J. H. (2006). Separating signal from background using ensembles of rules. In *Statistical Problems in Particle Physics, Astrophysics and Cosmology*, pp. 127–136. World Scientific. 4

Giannone, D., L. Reichlin, and D. Small (2008). Nowcasting: The real-time informational content of macroeconomic data. *Journal of Monetary Economics 55*(4), 665–676. 3

Götz, T. B. and T. A. Knetsch (2019). Google data in bridge equation models for German GDP. *International Journal of Forecasting 35*(1), 45–66. 2, 3, 13

Greenwell, B. M. (2017). pdp: An R package for constructing partial dependence plots. *The R Journal 9*(1), 421–436. 10

Hubbard, A. E., C. J. Kennedy, and M. J. van der Laan (2018). Data-adaptive target parameters. In *Targeted Learning in Data Science*, pp. 125–142. Springer. 20

Jung, J.-K., M. Patnam, and A. Ter-Martirosyan (2018). *An Algorithmic Crystal Ball: Forecasts-based on Machine Learning*. International Monetary Fund. 2

Lundberg, S. M., G. G. Erion, and S.-I. Lee (2018). Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*. 4, 9, 10

Lundberg, S. M. and S.-I. Lee (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pp. 4765–4774. 4, 9

Molnar, C. (2019). *Interpretable Machine Learning*. Lulu. com. 10

Shapley, L. S. (1953). A value for n-person games. *Contributions to the Theory of Games 2*(28), 307–317. 4

Staniak, M. and P. Biecek (2018). Explanations of model predictions with live and breakdown packages. *arXiv preprint arXiv:1804.01955*. 10

Tiffin, A. (2016). Seeing in the dark: a machine-learning approach to nowcasting in Lebanon. 2

Tkacz, G. (2001). Neural network forecasting of Canadian GDP growth. *International Journal of Forecasting 17*(1), 57–69. 2, 3

Tkacz, G. (2013). Predicting recessions in real-time: Mining Google Trends and electronic payments data for clues. *CD Howe Institute Commentary* (387). 2

Yousuf, K. and Y. Feng (2020). Partial distance correlation screening for high dimensional time series. *Working Paper*. 6

Appendices

A. Additional tables and figures

Table 2: Google Trends (Relevant search terms)

Energy sector	Content	Construction	Rental
Business sector	Media	Manufacturing	Leasing
Business industries	Cannabis	Wholesale trade	Professional
Industrial production	Agriculture	Retail trade	Scientific
Manufacturing industries	Forestry	Transportation	Technical services
Durable good	fishing and hunting	Warehousing	Management companies
Information	Mining	cultural industries	Management enterprises
communication technology	Quarrying	Cultural	Administrative support
Technology sector	Oil extraction	Finance	Waste management
Public Sector	Utilities	Real estate	Health care
Social assistance	Arts	Entertainment	Recreation
Accommodation	Food services	Public administration	Perishable goods
Export	Import	Trade balance	US industrial
US Shipment	Building Permits	Consumer confidence index	Inflation
Employment	US employment	GDP growth	Price Index
Housing starts	US Retail	Shipment	US Shipment
US pmi	global pmi	Inventories	Retail trade
US retail	US housing	Consumption	Expenditure
Services	Investment	Residential	Machinery
Equipment	Intellectual.property	Farm	

Table 3: Google Trends general search terms

Porsche	Aquaculture	Office.Supplies	Import.Export
Rolls.Royce	Food.Production	Office.Furniture	Maritime.Transport
Saab	Forestry	Printers	Packaging
Saturn	Horticulture	Scanners	Parking
Subaru	Livestock	Outsourcing	Public.Storage
Suzuki	Business.Education	Signage	Rail.Transport
Toyota	Business.Finance	Civil.Engineering	Computer.Hardware
Scion	Investment.Banking	Electricity	Computer.Components
Volkswagen	Risk.Management	Nuclear.Energy	Computer.Memory
Volvo	Venture.Capital	Oil...Gas	Hard.Drives
Auto.Interior	Financial.Markets	Waste.Management	Network.Storage
Car.Electronics	Business.Operations	Recycling	Copiers
Car.Audio	Human.Resources	Data.Management	Desktop.Computers
Car.Video	Management	Hospitality.Industry	Computer.Security
Body.Art	Business.Process	Event.Planning	Network.Security
Cosmetic.Surgery	Project.Management	Food.Service	Audio.Equipment
Fitness	Project.Management.Software	Restaurant.Supply	Headphones
Bodybuilding	Strategic.Planning	Generators	Speakers
Hair.Care	Supply.Chain.Management	Heavy.Machinery	Camera.Lenses
Hair.Loss	Business.Services	Manufacturing	Cameras
Massage.Therapy	Consulting	Small.Business	Nintendo
Weight.Loss	Corporate.Events	Home.Office	Sony.PlayStation
Marketing.Services	Knowledge.Management	Aviation	Infectious.Diseases

Table 4: Final XGboost-selected Google Trend data

Consumption	Small Business	Car Electronics
Ford	Mining	Weather
Farm	Services	
Financial Markets	Executive Branch	
Rental	Office Furniture	
Energy sector	Network Security	
Army	Property Management	
Volvo	Finance	
Construction	Property Development	
Transportation	Roleplaying Games	
Computer Hardware	Medical Procedures	

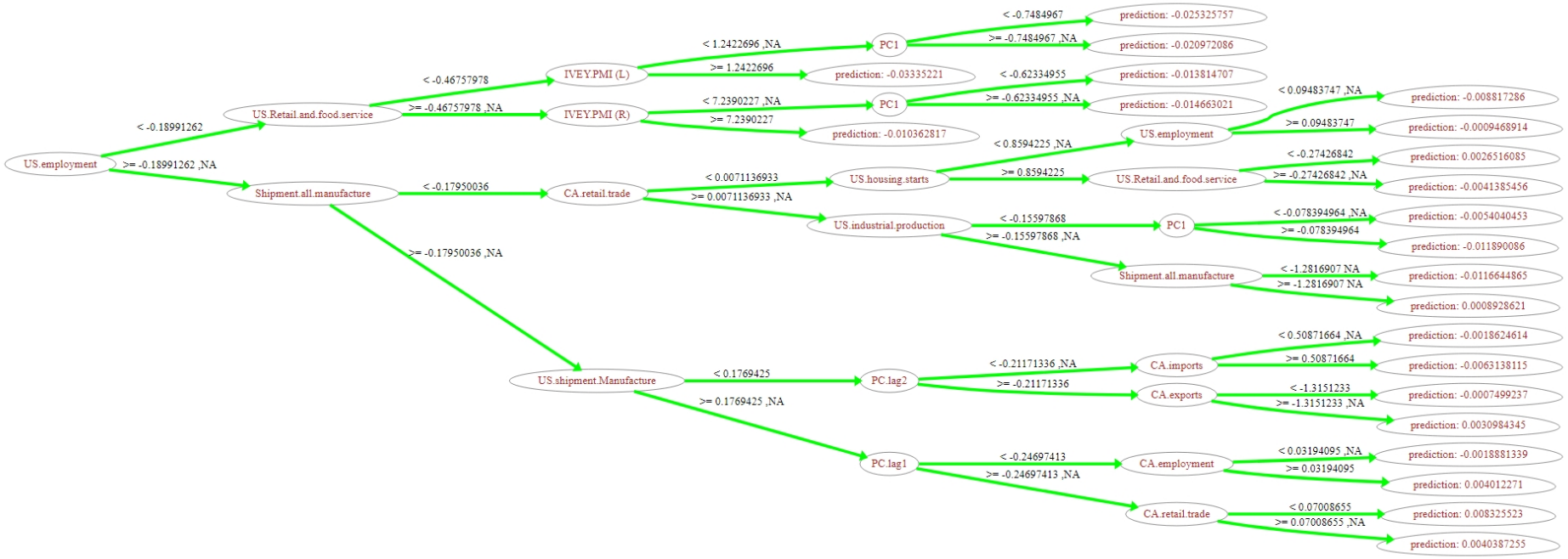
Table 5: Official data variables used for forecasting quarterly GDP

Public administration	Finance and insurance
Other services except public administration.	Manufacturing
Accommodation and food services	Transportation and warehousing
Durable manufacturing industries	Administrative and support waste management and remediation services
Educational.services	Retail trade
Mining quarrying and oil and.gas extraction..	Real estate and rental and.leasing
Arts entertainment and recreation..	Non durable manufacturing industries
Wholesale trade	Agriculture forestry fishing and hunting.
Utilities	Industrial production
Information and cultural industries	Health care and social assistance
Construction	Service producing industries
Professional scientific and technical services	All.industries except cannabis sector
Energy sector	All industries except unlicensed cannabis sector
Goods producing industries	

Table 6: Official data variables from Canada and the US used for forecasting monthly GDP

Variable	Ref Time	Release date	Pub Lag(days)	Freq
US ISM Manufacturing PMI	March 2019	1st April,2019	01	M
WTI Oil Price	March 2019	1st April,2019	01	M
US:NAPMALL	-	3rd April, 2019	03	M
CONCCONF	-	1st April,2019	01	M
IVEYSA	-	4th April,2019	04	M
Empolyment rate	March 2019	5 April,2019	06	M
US:Employment	-	3 April,2019	04	M
US: Light weight vehicle sale	-	7 June,2019	08	M
Housing starts	-	8 April,2019	09	M
US: Retail and food services	-	15 April,2019	16	M
Price Index	-	17 April,2019	18	M
US:Industrial Production	-	14 June,2019	15	M
US: Motor vehicl assemblies	-	14 June,2019	15	M
Business outlook	Quarter 1,2019	15 April,2019	16	Q
US:Housing starts	March,2019	19 April,2019	20	M
US: Shipment Manuafacture	March,2019	24 April,2019	25	M
US: Non-defense Manuafcture	April,2019	04 June,2019	36	M
Export	March,2019	09 May, 2019	40	M
Import	March,2019	09 May, 2019	40	M
Building Permits	March, 2019	10 May,2019	41	M
New motor vehicle sales	March, 2019	14 May,2019	45	M
Shipment all manufacture	March,2019	16 May, 2019	47	M
Total Inventories	March,2019	16 May, 2019	47	M
Retail Sales	March, 2019	22 May, 2019	53	M
Whole trade inventories	March,2019	23 May, 2019	54	M
GDP by Industry	March, 2019	May 31, 2019	62	M
GDP	1st Quarter, 2019	May 31, 2019	62	Q

Figure 16: An example of the decision tree (plotted for *GBM*)



Note: We plot tree number one by using the "data.tree" R package with H2O. It splits on column US employment (It is called root node). When the split threshold is < -0.18991262, it goes to the left node and further splits on US retail and foodservice. Similarly, when the threshold is >= -0.18991262, then moves to the right node and select shipment of all manufacture for splitting. The split threshold for US retail and foodservice is < -0.46757978 to the left and >= -0.46757978 to the right node. Similarly, other nodes can be interpreted likewise. The CA retail trade, CA employment, CA exports, CA imports, Shipment of all manufacture, PC1, US retail and foodservice, and US employment are the terminal node. Finally, we get prediction (in RMSE) associated with each terminal node.

B. Technical description of XGBoost

We shall follow the presentation of Chen and Guestrin (2016). Assuming equal weights, the final prediction is a linear combination of the score of each tree, f_k, as follows:

$$\hat{y}_t = \phi(x_t) = \sum_{k=1}^{K} f_k(x_t), f_k \in \mathcal{F} \tag{A.1}$$

$$\mathcal{F} = \{f(x) = w_q(x)\}, q : \mathbb{R}^N \longrightarrow \mathcal{T}, w \in \mathbb{R}^{\mathcal{T}}$$

A tree is described by two parameters: q and w: The parameter q determines the structure of the tree, t.e., it maps the sample instances to leaves, and the parameter w determines the weight attached to the leaves. $\mathcal{F}$ denotes the space of regression trees, K the number of trees, $\mathcal{T}$ is number of leaves in a tree, n is the sample size , and N represents the number of features. Let $l(y_t, \hat{y}_t)$ be a continuous twice-differentiable convex function. The learning objective consist of a loss function $\mathcal{L}(\phi)$ and a regularization term as in (9).

$$\mathcal{L}(\phi) = \sum_{t=1}^{n} l(y_t, \hat{y}_t) + \sum_{k=1}^{K} \Omega(f_k) \tag{A.2}$$

$$\Omega(f_k) = \gamma\mathcal{T} + \frac{1}{2}\lambda\|w\|^2 \tag{A.3}$$

The loss function reduces bias and measures how well the model fits the training data. The regularization term, $\Omega(f_k)$ controls the complexity of the model and prevents overfitting by penalizing the complex tree. The parameter λ controls the degree of the regularization of each tree and $\|w\|^2$ is L^2-*norm* of leaf scores. The parameter γ penalizes increased tree-complexity (as also shown in eq.(22) below.) As Chen and Guestrin (2016) indicate, the loss function $\mathcal{L}(\phi)$ in (9) is a tree-ensemble model which has functions as parameters. Hence, it cannot be optimized with usual methods in Euclidean space. Therefore we need to train the model additively in a sequential manner (*Boosting*) to minimize the loss function in (9). To do so, we add the f_ℓ(the tree) that most improves the model in equation (9) to the forecast $\hat{y}_t^{(\ell-1)}$, such that the new, ℓ^{th} iteration forecast is $\hat{y}_t^{(\ell)} = \hat{y}_t^{(\ell-1)} + f_\ell$. [29]Denoting the loss function at the ℓ^{th} iteration by $\mathcal{L}^{(\ell)}$

$$\mathcal{L}^{(\ell)} = \sum_{t=1}^{n} l\left(y_t, \hat{y}_t^{(\ell)}\right) + \Omega(f_k) \tag{A.4}$$

$$\mathcal{L}^{(\ell)} = \sum_{t=1}^{n} l\left(y_t, \hat{y}_t^{(\ell-1)} + f_\ell(x_t)\right) + \Omega(f_\ell) \tag{A.5}$$

[29]That is, the prediction $\hat{y}_t^{\ell}$ is obtained iteratively as follows:
$\hat{y}_t^{(0)} = 0$
$\hat{y}_t^{(1)} = f_1(x_t) = \hat{y}_t^{(0)} + f_1(x_t)$
$\hat{y}_t^{(2)} = f_1(x_t) + f_2(x_t) = \hat{y}_t^{(1)} + f_2(x_t)$
$\hat{y}_t^{(\ell)} = \sum_{k=1}^{\ell} f_k(x_t) = \hat{y}_t^{(\ell-1)} + f_1(x_t)$

where, $\hat{y}_t^{(\ell-1)}$ is the prediction from the previous iteration. Then by applying a second-order Taylor expansion to $l\left(y_t, \hat{y}_t^{(\ell-1)} + f_\ell(x_t)\right)$ around $\hat{y}_t^{(\ell-1)}$ we obtain:[30]

$$\mathcal{L}^{(\ell)} \approx \sum_{t=1}^{n} l\left(y_t, \hat{y}_t^{(\ell-1)}\right) + \frac{\partial l\left(y_t, \hat{y}_t^{(\ell-1)}\right)}{\partial \hat{y}_t^{(\ell-1)}} f_\ell(x_t) + \frac{1}{2} \frac{\partial^2 l\left(y_t, \hat{y}_t^{(\ell-1)}\right)}{\partial \hat{y}_t^{(\ell-1)^2}} f_\ell(x_t)^2 + \Omega(f_\ell) \tag{A.7}$$

where $l\left(y_t, \hat{y}_t^{(\ell-1)}\right)$ is constant. Letting $g_t \equiv \frac{\partial l\left(y_t, \hat{y}_t^{(\ell-1)}\right)}{\partial \hat{y}_t^{(\ell-1)}}$ and $h_t \equiv \frac{\partial^2 l\left(y_t, \hat{y}_t^{(\ell-1)}\right)}{\partial \hat{y}_t^{(\ell-1)^2}}$, we have

$$\mathcal{L}^{(\ell)} = \sum_{t=1}^{n} \left[g_t f_\ell(x_t) + \frac{1}{2} h_t f_\ell(x_t)^2\right] + \Omega(f_\ell) + constant \tag{A.8}$$

or, after omitting the constant, we write the objective function as:

$$\tilde{\mathcal{L}^{(\ell)}} = \sum_{t=1}^{n} \left[g_t f_\ell(x_t) + \frac{1}{2} h_t f_\ell(x_t)^2\right] + \Omega(f_\ell) \tag{A.9}$$

We substitute (10) into (16), letting $\|w\|^2 = \sum_{j=1}^{\mathcal{T}} w_j^2$ (since w is a $\mathcal{T}$-dimensional vector) to obtain

$$\tilde{\mathcal{L}^{(\ell)}} = \sum_{t=1}^{n} \left[g_t f_\ell(x_t) + \frac{1}{2} h_t f_\ell(x_t)^2\right] + \gamma\mathcal{T} + \frac{1}{2}\lambda \sum_{j=1}^{\mathcal{T}} w_j^2 \tag{A.10}$$

Our goal is to minimize $\tilde{\mathcal{L}^{(\ell)}}$ with respect to f_ℓ. Recall also that each tree has an independent tree structure q and leaf weights w . Let I_j be the set of instances when data point t belongs to lead node j. I_j is defined as $I_j = \{t \mid q(x_t) = j\}$,where $q(x_t)$ depicts the tree structure from the root to node j in the decision tree. Then we can rewrite

$$\tilde{\mathcal{L}^{(\ell)}} = \sum_{j=1}^{\mathcal{T}} \left[\left(\sum_{t\in I_j} g_t\right) w_j + \frac{1}{2}\left(\sum_{t\in I_j} h_t + \lambda\right) w_j^2\right] + \gamma\mathcal{T} \tag{A.11}$$

We want to find optimized weight w_j^* for node j for a given structure q by minimizing (17) with respect to w_j. The first-order condition is:

$$\frac{\partial \tilde{\mathcal{L}^{(\ell)}}}{\partial w_j} = \sum_{j=1}^{\mathcal{T}} \left[\left(\sum_{t\in I_j} g_t\right) + \left(\sum_{t\in I_j} h_t + \lambda\right) w_j\right] = 0 \tag{A.12}$$

Solving for w_j^* yields:

$$w_j^* = -\frac{\sum_{t\in I_j} g_t}{\sum_{t\in I_j} h_t + \lambda} \tag{A.13}$$

[30] That is, the Taylor expansion of $\theta(z)$ around a yields:

$$\theta(z) = \theta(a) + \frac{\theta'(a)}{1!}(z-a) + \frac{\theta''(a)}{2!}(z-a)^2 + \frac{\theta'''(a)}{3!}(z-a)^3 + \ldots \tag{A.6}$$

Using a second-order expansion by ingoring higher order terms, letting $\theta(z) \equiv l\left(y_t, \hat{y}_t^{(\ell-1)} + f_\ell(x_t)\right)$, $z = \hat{y}_t^{(\ell-1)} + f_\ell(x_t)$, and $a = \hat{y}_t^{(\ell-1)}$,we obtain the result in equation (14) .

By substituting (20) into (18), we obtain:

$$\tilde{\mathcal{L}}^{(\ell)}(q) = -\frac{1}{2}\sum_{j=1}^{\mathcal{T}} \frac{\left(\sum\limits_{t\in I_j} g_t\right)^2}{\sum\limits_{t\in I_j} h_t + \lambda} + \gamma\mathcal{T} \tag{A.14}$$

the first term indicating that the loss has been reduced by the new forecast obtained after adding $f_\ell(x_t)$,and the second term indicating the cost $\gamma\mathcal{T}$ of increasing the complexity of the tree. Thus, equation (21) provides a measure of how good the tree structure $q(x_t)$ is by calculating the *change* in the loss function $\mathcal{L}^{(\ell)}$ given by $\tilde{\mathcal{L}}^{(\ell)}(q)$. It denotes the score of the tree. If it were possible to know all possible tree structures, we could choose the tree with the highest score. However, since in practice we cannot know all possible structures q we can proceed sequentially, and calculate the *gain* (t.e., the *negative* of the loss) of a split as we add levels to the tree. Suppose that we split the leaf at node j into two leaves. The net gain of adding an additional leaf at node j is given by:

$$gain = \frac{1}{2}\left[\frac{\left(\sum\limits_{t\in I_j} g_t\right)^2}{\sum\limits_{t\in I_L} h_t + \lambda} + \frac{\left(\sum\limits_{t\in I_j} g_t\right)^2}{\sum\limits_{t\in I_R} h_t + \lambda} - \frac{\left(\sum\limits_{t\in I_j} g_t\right)^2}{\sum\limits_{t\in I} h_t + \lambda}\right] - \gamma \tag{A.15}$$

where I_{jL} and I_{jR} are the set of instances that a data point t belongs to the left and right child respectively after splitting the leaf at node j and $I_j \cup \{I_{jL}, I_{jR}\}$ is, as before, the set of instances that a data point t belongs to the leaf at node j. The first two terms in brackets in equation (22) represent the scores of the left and right child while the last term is the score before the split. Here γ represents the cost (or penalty) of increasing complexity by introducing an additional leaf.