Forecasting the Living Standard in Canada

by

Victor Popa Burca & Steven Chung

ECO1400 Final Research Paper

University of Toronto

December 2nd, 2025

Executive Summary

This paper explores the benefits of different forecast modelling implementations on the Canadian standard of living. In a context that is relevant to individuals, firms and policy makers alike, the motivation of our paper stems from several different pieces of macro-econometric forecasting literature. The analysis starts by introducing several important pieces of literature that introduce specific techniques which we later use, and build on, but within the separate context of the Canadian standard of living. The standard of living is summarized by our focus in three main variables: income per capita, inflation growth and growth in new housing prices. These variables are popular and most commonly associated with living standards in the news, in political rhetoric, and in economic theory. The literature provides several model formulations, but most importantly, a large database of Canadian macroeconomic data which has been deliberately constructed to be used for forecasting experiments. We use this time series data to train our models with the goal of compiling forecasts for a short term and long-term horizon, consisting of 1 – 3 months ahead, and 1 – 12 months ahead respectively.

Starting the analysis with a simple yet popular linear forecasting model, we continuously improve our results by adding additional layers of complexity. These layers first come in the form of more variables, systematically selected from the same database and rooted in economic theory. Then, we allow for non-linear relationships between our predictors, which makes our predictions more accurate and flexible from a mechanical standpoint. The main results of the paper serve as additional evidence for the claims of some of the research that is reviewed at the beginning and used as motivation. The part of the analysis that builds and extends on this previous literature comes from conducting these estimations on the most updated database, and introducing a newer flexible model known as BART into the picture.

Our main findings show that allowing for non-linear relationships will greatly improve the accuracy of macroeconomic forecasts that relate to the standard of living. More specifically, the complex models we implement will detect seasonal cycles in the data series and put more importance on current trends when making forecast predictions. Using these results, we graph our forecasts over the next 1 - 12 months ahead starting from August 2025. Across all of our models, our graphs display a soft landing for the Canadian living standard, with inflation growth expected to remain mild and roughly unchanged, income per capita growth to grow very slightly and the housing prices showing clear signs of an upward recovery.

1 Introduction

Forecasts are a crucial tool for the macroeconomic policy making of central banks and public institutions around the world. This comes from the forward-looking nature of these institutions, which results from the lagged effects that many policy instruments have on the economy. There are many different models used in forecasting, and different variables of interest depending on the institutions that conduct the analysis. When focusing on welfare, forecasting variables that have direct standard of living and policy implications has become an important focus in much of the research today.

However, there exists a multitude of different models, and variables of interest that could be implemented for standard of living forecasts. In our paper, we focus on three main variables related to the standard of living in Canada: GDP per capita, CPI inflation and national new housing price. The choice of these variables is motivated by income per capita measures being strongly correlated to average wellbeing (Deaton 2008), but also incomplete once factoring in a more complete measure of living standards which includes leisure and inequality (Jones and Klenow 2016). In line with the idea of broadening indicators that measure welfare, we include inflation due to its link to the welfare costs of inflation and purchasing power implications, and new housing price because of its link to affordability and financial vulnerability. More generally, these are popular variables related to the standard of living, that are frequently analyzed in current news and literature. The most recent OECD Economic Survey on Canada has emphasized that weak GDP per capita growth and increased housing prices are important challenges for Canada's living standards and future economic performance.[1] To further investigate these challenges, our paper focuses on Canadian macroeconomic data, in line with previous forecasting analysis from Stevanovic.

More specifically, we ask the question of how the forecasting accuracy of Canadian GDP per capita, CPI inflation and new housing price can be improved relative to a simple Vector Autoregressive (VAR) benchmark model, when including a richer set of predictors for these variables and allowing for non-linearity through Machine Learning (ML) methods. To answer this question, we construct and compare three classes of forecasting models at both short period (1-3 months) and long period (1-12 months) forecasts: (i) a simple VAR model that only contains our 3 target variables of interest, (ii) an extended baseline VAR model, which builds on the simple model with efficient variable selection, (iii) non-linear tree based ML models, Random Forest (RF) and Bayesian Additive Regression Trees (BART) using the same set of variables as in the baseline model. We rely on the most recent August 2025 version of the large Canadian database constructed in Fortin et al. (2022).

[1] OECD Economic Surveys: Canada 2025 (pg. 11)

We compare each model by focusing on the average Root Mean Squared Error (RMSE) over the short and long period horizons separately. In both time horizons, the simple model performs the most poorly and is dominated by a considerable improvement in the average RMSE in the extensive model. Interestingly, we find that the RF tree-based model dominates the more flexible BART model in both time periods and consistently provides the best out of sample RMSE over both time horizons. These results are consistent with Canadian macroeconomic forecasting literature on how ML methods provide large forecasting gains (Coulombe et al. 2022) but also provides a new angle of model comparison with the inclusion of BART. These results present important implications for forecast modelling decisions and can be useful in the context of central bank policy discussion. This is especially true for inflation targeting policy and discussions regarding affordability for Canadians, in the context of increasing welfare.

In the following sections of our paper, we discuss the methodology and data behind our results, as well as interpretations. In Section 2, we outline recent relevant literature that is related to our topic and techniques. Then in Section 3, we describe the dataset that we use and conducts stationarity tests to verify its validity. Section 4 describes the construction of each model separately and explains the variable selection technique. The interpretation of our results is done in Section 5, and finally Section 6 concludes our analysis.

2 Literature Review

In this section, we will discuss and summarize the most important findings from recent literature on macroeconomic forecasting techniques. Starting with older publications on VAR models, we will summarize the evolution of the relevant literature that inspired our analysis and explain how our methods and results differ along the way.

First, we start with the simple reduced-form VAR model, as described in (Stock & Watson 2001). This paper provides an overview of VAR models in the context of macro econometric implementation and builds a simple 3 variable VAR model for U.S inflation, the fed's interest rate and unemployment. The main results of the paper show that the simple VAR model either matches or improves on the univariate Auto Regressive and random walk models in pseudo out of sample RMSE for their variables and dataset. They support the use of simple reduced-form VAR models as natural benchmark comparisons against richer models with more variables. We make use of these findings by starting our analysis with a simple 3 variable VAR model as the benchmark, before building a richer VAR.

The next important result that is useful for our analysis is the Large Canadian Database for Macroeconomic Analysis (Fortin et al. 2022). This paper constructs a regularly updated large macro

dataset containing monthly observations of hundreds of Canadian macroeconomic indicators from 1981 to the present month.[2] The database itself was constructed using spliced StatsCan tables and retropolation, then most series were transformed in log first differences for stationarity. Further, this paper runs several forecasting experiments on the annualized growth rates of several indicators and find that in the specific context of a data rich macro data set like LCDMA, nonlinear models consistently perform better than simple benchmark Autoregressive Direct (ARD) and Autoregressive with Diffusion Index (ARDI) models. Our results and empirical implementation use the most updated version of this database (January 1981 - August 2025), and we find a similar general result, in that nonlinear models are more accurate than simple benchmarks. Where our analysis differs, is in the use of the most recent data to test our results as well as the use of VAR models as the baseline rather than ARD.

Our paper's analysis of nonlinear models applied to forecasting uses important concepts and results from Coulombe et al. (2022). This paper finds that applying non-linear methods to baseline AR models will consistently produce more accurate forecasts, especially in longer time horizons. The authors use RF and Kernel Ridge Regression to introduce non-linear estimation in their baseline models and find improvements in the pseudo-R^2 of up to 23% (Coulombe et al. 2022). A second big result that the paper finds is that K-fold cross validation is the best hyperparameter tuning practice. Following previous literature in (Bergmeir et al. 2018), they use a standard K-fold CV approach inside of their previously established "expanding window" pseudo out-of-sample forecasting set up. Our methodology uses a similar approach, in that we compare non-linear models to our baseline, but we extend the paper's work by also applying BART. We mainly use the result regarding the K-fold CV in our methodology but have a slightly different characterization in the number of folds.

Finally, more recent work in macroeconomic forecasting of Canadian variables focuses on conditional risk scenarios of outside shocks (Moran, Stevanovic et al. 2024). This paper builds a medium-sized linear VAR model on a large data set, then runs conditional forecasts on US oil price, domestic labor market, etc. to show how the baseline VAR model can be used as a tool to communicate risk from domestic and international shocks. In a non-Canadian context, Chen et al. (2025) show that deep learning NN models are more accurate than time series forecast models of GDP per capita in China. This paper shows the powerful benefits of NN models for single target forecasts but out work differs because we have multiple targets and we focus on tree-based ML models.

[2] August 2025 LCDMA

3 Data and Variable Construction

This section introduces our dataset and the three target variables and summarize some variable construction details and data consistency for our model assumptions.

In this paper, we adopt a large monthly Canadian dataset for macroeconomic analysis (LCDMA), which contains hundreds of Canadian macroeconomic indicators. These variables include real activity, labour markets, housing, credit, prices, external trade and commodities. This dataset has been constructed by Fortin-Gagnon, O., Leroux, M., Stevanovic, D. and S. Surprenant (2022), A Large Canadian Database for Macroeconomic Analysis.[3] The authors aggregate a broad selection of Canadian macroeconomic variables from StatCan, the Bank of Canada, and international sources. They also have monthly data to be updated regularly through StatCan database. The August 2025 dataset is the newest dataset release available. Therefore, we adopted the stationary and balanced panel data starting from 1981M01 to August 2025 in our analysis.

The reason we adopted the LCDMA is because it is methodologically consistent with our forecasting goals and the necessary time series transformations have already been applied to all the series. These transformations include logarithm growth-rate conversions of levels-data, deflation where appropriate, and missing-value imputation by EM algorithms. Further, the LCDMA balanced panel aligns all series over a common sample, eliminating ragged edges (Fortin-Gagnon et al., 2022). Such alignment is required for VAR estimation, which relies on complete lagged observations. As such the variables that we use in the database are stationary, and suitable for our VAR forecasting models

Within this rich dataset, we focus on a subset of variables directly tied to our economic question of interest: forecasting the growth of living standards in Canada. Since most of the data series are presented as first differences of logs ($\Delta\log$), they are interpreted as monthly growth rates and our construction of target variables is adjusted accordingly. Our first target variable is GDP per worker which is constructed by taking:

$$GDP\_\text{new}_t - EMP\_\text{CAN}_t = \Delta\log(Y_t) - \Delta\log(N_t) = \Delta\log\left(\frac{Y_t}{N_t}\right)$$

Yielding a practical proxy for the GDP per capita monthly growth rate. In macroeconomic analysis, GDP per capita remains the canonical measure of living standards because it summarizes the average level of

[3] Fortin-Gagnon, O., Leroux, M., Stevanovic, D. and Surprenant, S. (2022), A large Canadian database for macroeconomic analysis. Canadian Journal of Economics/Revue canadienne d'économique, 55: 1799-1833. https://doi.org/10.1111/caje.12618

resources available to individuals. GDP per worker slightly differs by capturing more of the labour productivity dimension of living standards. It would be reflecting more living standard of workers rather than average citizen. However, we believe that GDP per worker can still help measure improvements in economic welfare and therefore use it as a proxy for GDP per capita in our analysis.

The second target is inflation rate (CPI_ALL_CAN), Canada's headline consumer price inflation index, which reflects the purchasing power of households, are presented in monthly inflation rate in the dataset. Inflation heavily affects cost of living of Canadian citizen, which is a popular topic in recent years. The final target is housing price, modeled by a national house price index (NHOUSE_P_CAN) capturing housing affordability, cost of living, and economic development momentum, which is especially important in the Canadian context where real estate dynamics propagate strongly to consumption, credit and investment.

All three target variables are presented in monthly growth rates in the LCDMA dataset, providing stationary variables for VAR models. We also included variables that help forecast the three target variables in the large VAR model, with variable selections explained in Section 4. All of the variables in the model passed the Augmented Dickey–Fuller tests, satisfying stationarity assumptions. Table 1 in Appendix A shows that the p-value of this test is ~ 0 for all variables, meaning the null hypothesis (which states that the series is not stationary) is rejected.

4 Methodology

In this section, we will describe each of our models in detail. In particular, we start by discussing key features that we apply to each of our models, including our loss function, validation approach, and variable selection for our future extensions beyond the simple model.

4.1 Horizon Groups and RMSE

We focus on forecasts over two main horizon groups in each of our models. The short-term horizon group $H_{short} = \{1, 2, 3\}$ provides the monthly point forecasts of each of our target variables, up to three months ahead, and the long-term horizon group $H_{long} = \{1, \dots, 12\}$ for up to twelve months ahead. Each model in the following sections is based on a forecast evaluation using the root mean squared error, following standard macro-forecasting practice (Buturac 2022). For a given group $i \in \{short, long\}$, target variable set $j = \{\text{GDP per worker, CPI, new housing price}\}$ and forecast origin T_i, the forecast error is computed as the difference between the realized value and the model's forecast: $e_{t+h,j} = y_{t+h,j} - \hat{y}_{t+h,j}$ .These errors are generated with an expanding-window procedure: at each forecast origin $t \in T_i$ the model is estimated up to a given point t in the sample, then it is used to forecast the next observation at horizon h

($\hat{y}_{t+h,j}$) in the relevant group. The window is then expanded by one month, and the process repeats until the end of the sample. For each model and lag length, we collect all forecast errors across all forecast origins in our folds, across all three target variables and across all horizons in the group. The RMSE reported in Section 5 is the square root of the average squared forecast error over this entire set. This means that the errors are first squared, then averaged across all origins, across all horizons in the group, and across the three targets to produce one RMSE for the short-term group and one RMSE for the long-term group:

$$RMSE_i = \sqrt{\frac{1}{N_i}\sum_{t}\sum_{h\in H}\sum_{j=1}^{3} e_{t+h,j}^2} \quad (1)$$

Where N_i is the total number of forecast errors in horizon group i. The lag length that minimizes this average RMSE is selected as the model's optimal lag for that horizon group, which can be seen in table 18 of Appendix A.

4.2 K-fold Expanding-Window Cross-Validation

Each of our models are evaluated using an expanding-window cross-validation scheme that replicates forecasting techniques in Bergmeir et al. (2018) and Coulombe et al. (2022). At each evaluation date, the model is estimated only on data up to that point, which is then used to predict future observations. For a given lag length p and forecast horizon H, we select $K = 10$ forecast "origin dates". These dates are evenly spaced out near the end of our sample and for each date, the model is re-estimated, forecasts are generated, and the forecast errors are stored. Therefore, at each origin t_k, the model is estimated only on the data up to time t_k, and used to forecast all the observations following this date (a pseudo out-of-sample forecast). We are doing it in K-fold for the computational efficiency gains outlined in the literature and pick K = 10 for all our estimated models[4]. This evaluation procedure is done twice for the short term and long-term horizons separately. These errors are then combined into a single RMSE by squaring them, averaging across folds, averaging across the forecast horizons in the group (either short term or long term), and averaging across the three target variables. Then similar to what we previously outlined, the lag length that produces the lowest expanding-window RMSE is chosen as the optimal lag for that model and horizon group.

[4] Compared to K=5 in Coulombe et al. (2022)

4.2.1 Simple Benchmark Model: Baseline VAR

The basic forecasting logic is first introduced through a simple 3-variable VAR model. Under this framework, each target variable is regressed on its own lags, and the lags of the other two target variables, with the optimal number of lags selected by the K-fold expanding-window cross-validation scheme working to minimize RMSE mentioned in Section 4.1 and 4.2. This baseline provides a reference point for evaluating the value added in forecast accuracy of additional predictors or nonlinear interactions. Let $g_t = GDP\ per\ worker\ growth, \pi_t = CPI\ growth, hp_t = new\ housing\ price\ growth$. The model can be summarized in the following reduced form equation:

$$\boldsymbol{y}_t = \boldsymbol{c} + \boldsymbol{A}_1\boldsymbol{y}_{t-1} + \boldsymbol{A}_2\boldsymbol{y}_{t-2} + \cdots + \boldsymbol{A}_p\boldsymbol{y}_{t-p} + \boldsymbol{u}_t \tag{2}$$

Where $\boldsymbol{y}_t = \begin{pmatrix} g_t \\ \pi_t \\ hp_t \end{pmatrix}$, $\boldsymbol{A}_i$ is a 3x3 coefficient matrix for lag i and $\boldsymbol{u}_t = \begin{pmatrix} u_{g,t} \\ u_{\pi,t} \\ u_{hp,t} \end{pmatrix}$ is the vector of shocks.

Then, the forecasts of this model are given by:

$$\boldsymbol{y}_{T+j\,|\,T} = \boldsymbol{A_1}\boldsymbol{y}_{T+j-1\,|\,T} + \boldsymbol{A_2}\boldsymbol{y}_{T+j-2\,|\,T} + \cdots + \boldsymbol{A_p}\boldsymbol{y}_{T+j-p\,|\,T},$$

where any $\boldsymbol{y}_{T+1\,|\,T}$ is given by $\boldsymbol{y}_{T+1|T} = E[\boldsymbol{y}_{T+1} \mid \boldsymbol{y}_T, \boldsymbol{y}_{T-1}, \ldots]$.[5]

4.3 Variable Selection

Our target is to create an informative predictor set that is economically intuitive while minimizing multicollinearity and the possibility of overfitting. First, we began with our LCDMA dataset and retained only nationally aggregated Canadian data, particularly those greatly related to our three forecast targets real GDP per worker, CPI inflation, and the national house-price index). Next, we computed the maximum absolute correlation of each candidate variable with all three targets and kept the top forty series. This step aimed to eliminate variables that were obviously irrelevant.

Adopting the idea of block selection from literature (Li & Chen, 2014), the remaining variables are categorized into economic blocks: real activity, labour, housing activity, credit and money, interest rates, external trade, prices, and commodity/market indicators. Then, we select up to two representative predictors from each block based on: (i) highest correlation with our targets, and (ii) excluding highly

[5] ECO1400 Lecture 11a. VAR, Cointegration, VEC

collinear variables (pairwise |ρ|>0.90). The logic behind such design originates from our investigation of the dataset. We observed that most highly correlated variables are sub-categories of our target variables, which will lead to high out-of-sample errors in the estimation. Following popular techniques in the relevant literature, we first attempted to use packages like LASSO for variable selection. LASSO is based on in-sample errors and penalty, which ended up selecting mostly provincial-level or sub-category counterparts of our target variables. After trying estimation based on LASSO-selected variables, we saw that the resulting RMSE from that model is higher than our current one. Therefore, in terms of forecasting, highly correlated (pairwise) variables should not be selected. We believe selection in blocks helps to diversify the information our model captured by including variables that are less correlated but from a wider selection.

The process outlined above resulted in a selection of 17 variables, consisting of three of our target variables and fourteen economically interpretable predictors. To ensure consistency, we used this exact set for the large VAR, Random Forest, and BART models later. The following table summarizes the exact seventeen variables selected:

Table 1: Summary of Selected Variables for the Large VAR Model

Variable	Block	Short_Description
GDP_per_w	Target	GDP per worker, monthly growth
CPI_ALL_CAN	Target	Headline CPI inflation, monthly
NHOUSE_P_CAN	Target	National house price index, monthly
IP_new	Real activity	Industrial production index, monthly
BSI_new	Real activity	Business sentiment / survey index
UNEMP_CAN	Labour	Unemployment rate, Canada
EMP_CONS_CAN	Labour	Construction employment, Canada
hstart_CAN_new	Housing	Housing starts, Canada, monthly
CRED_HOUS	Credit	Household / housing credit stock
CRED_HOUS_MORT	Credit	Mortgage credit for housing
BANK_RATE_L	Interest	BoC policy rate (level)
TBILL_3M	Interest	3-month T-bill rate
Exp_BP_new	External	Exports of goods (BOP), monthly
Imp_BP_new	External	Imports of goods (BOP), monthly
IPPI_METAL_CAN	Prices	Metal producer price index, Canada
OILP_new	Commodities	Oil price index
WTISPLC	Commodities	WTI spot crude oil price (level)

4.3.1 Extended VAR

We estimated the extended VAR model based on similar assumptions made in the small VAR, only changing our variable set to include the seventeen variables we selected. This only changes our definition of $\boldsymbol{y_t}$, $\boldsymbol{A_i}$ and $\boldsymbol{u_t}$ to include our chosen number of total predictors in equation (2). We then estimate the model based on the optimal lag selected from the expanding cross-validation discussed above

and provide a forecast. Based on that, we develop a forecast of three of our target variables from both short-term and long-term, which are also compared with the forecast developed by machine learning models.

4.4 Machine Learning Models

The models in this subsection build on our extended VAR model by allowing for nonlinear relationships between the lags of each selected variable. We first estimate a Random Forest model, following some of what was estimated in Coulombe et al. (2022). Then we estimate our models using Bayesian Additive Regression Trees (BART), which takes a new angle on previous research in our context[6]. We then compare their results of average RMSE of short and long term and optimal lags with our previous linear VAR models.

4.4.1 Random Forest

We applied the Random Forest model based on the same variables set, validation method, and RMSE function as above. We fixed the parameter based on the literature and kept the above assumptions on the other cross-validation parameters. We take $m = 1/3$ of the variables in the tree building process in both the process of CV selection and final model building. Using the ranger implementation, we then estimate the model efficiently and cross-validate using an expanding window. In particular, we select the lag length by searching over lags p = 1 to 12 using a small forest of 100 trees first to allow computation efficiency. We select optimal lag p by computing expanding-window RMSE, similar to what we did for VAR, in both the short-term and long-term forecasting models. Then, the lag with the lowest RMSE is selected, and we proceed to model fitting and forecasting.

We realized, in particular for the Machine Learning model, that generating informative forecasts that exceed a simple one-step-ahead smooth result necessitates estimating more than one model. For our purpose, we would like to develop forecasts for one to three months ahead ($h = 1 - 3$) and one to twelve months ahead ($h = 1 - 12$), , which requires fitting 15 models in total. Balancing accuracy and computation efficiency, we decided to fit our model and forecast based on a final RF with 500 trees, estimating at the lag selected by cross-validation. The resulting nonlinear forecast is then compared to VAR benchmarks.

[6] Coulombe et al. (2022) does not implement BART, and in the specific context of a sparse forecasting model of Canadian macroeconomic variables, previous research has not explicitly used BART.

4.4.2 Bayesian Additive Regression Trees (BART)

The Bayesian design allows us to construct a tree randomly, which helps further reduce potential model selection bias in other models. Similarly, with a panel of lagged predictors, we underwent 10-fold expanding-window cross-validation for our 1 – 3 months and 1 – 12 months ahead forecasts. We use fewer BART trees (150), posterior draws kept (200), and burn-in draws (200) to allow computation feasibility. This is consistent with typical BART implementations, where the number of trees is smaller for cross-validation but is then increased for final forecast inferences (Sparapani et al. 2021). Consequently, we relied on a cloud-based computation platform, Kaggle, for computation feasibility as R studio may crash out. We then selected optimal lag for both short horizon (h=1-3) and long horizon (h=1-12) models by minimizing the out-of-sample RMSE as usual. Finally, we used the optimal lag result to estimate BART models and developed our forecast. We use a higher number of trees (500), posterior draws kept (2000), and burn-in draws (1000), to improve our model accuracy. Our forecast is made directly based on current model building, hoping to provide a more dynamic forecast than recursive (constant) prediction for our time period of concern.

5 Empirical Results

This section outlines the main important results following our methodology discussion. We start by comparing the forecast accuracy of each model and outlining their chosen lags, focusing on interpreting these results in our standard of living context. Afterwards, we display the graphical forecasts at both the short- and long-term horizons of each model and interpret what these results could mean for the general state of the economy in the near future.

5.1 Model Results and Comparison

The main output of each model is the optimal lag selection based on our pseudo out-of-sample K-fold cross-validation approach, and the resulting average RMSE over each horizon. These results are all summarized in the following table:

Table 2: Summary of Optimal Lags and Forecast Accuracy

Model	p (h = 1–3)	RMSE 1–3	p (h = 1–12)	RMSE 1–12
Small VAR (3 vars)	7	0.0039	7	0.0036
Large VAR (17 vars)	2	0.0030	1	0.0034
Random Forest	12	0.0024	5	0.0026
BART	11	0.0027	3	0.0027

In the baseline small VAR model, the cross-validation procedure selects $p^* = 7$ lags for both the short- and long-term horizons. With this selected specification, we obtain a short term average RMSE that is slightly above that of the long-term horizon. This means that our selection process finds that using 7 months of history for all three of our target variables exclusively, will produce the most accurate forecast. Intuitively, this means that any shocks to GDP per worker, inflation or new housing price is persistent, which is why we consider the data from up to 7 months ago as still being useful in predicting those of the future (pseudo out-of-sample). We can interpret the slightly lower average RMSE for the long-term horizon as the model having an easier time forecasting longer horizons because the data converges to a stable path with less volatility over time compared to a shorter term.

The large VAR model displays significant gains in accuracy, with lower average RMSEs, especially over the short-term horizon. At the short-term horizon, the extended VAR produces a substantially lower average RMSE of 0.0030 compared to 0.0039. The result reassures the validity of our variable selection method and shows that including more predictors does help increase our forecast accuracy. Also, this suggests that short-term movements in our three target variables are more strongly influenced by changes in the credit market, external trade market, labour market, etc. This makes intuitive sense from a theoretical standpoint, especially when thinking of temporary productivity shocks that affect the labour market in a business cycle framework. On the other hand, an important difference in the extensive model is the considerably smaller number of lags that are selected by our expanding window scheme at each horizon. We go from 7 to 2 lags, which means that the additional information brought to the model by our new predictors could have previously been misrepresented through longer lags in the simple model. Another potential explanation for this result is that the larger number of variables combined with long lags could cause an overfitting issue, since the number of parameters increases more intensively with the number of lags. The cross-validation scheme we are doing to select $p^*$is then penalizing additional lags by more than before because we are using a pseudo out-of-sample RMSE that will pick up on an overfit model.

The RF model clearly exhibits the best forecast accuracy out of all our models here. This result aligns with a similar one in Coulombe et al. (2022), in that the RF approach can outperform linear benchmarks. An interesting observation here is that the RF approach selects a larger number of lags over both horizons compared to the extended VAR model, even though it uses the same set of 17 variables as predictors. This result could be attributed to the greedy random splitting process in RF, which means that the algorithm will not select a candidate lag unless it reduces prediction error. Compared to the extended VAR, this approach ends up discarding more lags by construction rather than estimating coefficients on every lag. Further, the model's chosen length of lags differs between forecasts focusing on short-term vs.

long term horizons. Starting with the long term, we can say that older information is not useful for determining what GDP per worker, CPI and new housing price will be a year from now because distant lags bring old information to the model that is washed out by more recent shocks in the economy. The short-term model selects $p^* = 12$ lags, meaning the RF approach benefits from access to yearly trends in each of our variables to forecast short term shocks. Intuitively, this makes sense because some variables like unemployment or industrial production can exhibit seasonal patterns over a year, and knowing where we stand in this yearly pattern will give the model another layer of useful information that improves forecasts.

Finally, the BART model is very similar in lag length and average RMSE to RF but produces slightly more conservative lag selections and looser forecast accuracy. Similar to RF, BART puts more importance on older information in its short-term forecasts (same idea of capturing seasonal patterns) than it does in the long horizon. The crucial result regarding BART's output is the smoothing between average RMSEs of both short- and long-term horizons. The RF estimation was slightly more accurate in the short term, whereas BART produces a smoother forecast, coming at the cost of weaker trees producing a higher average RMSE.

In the context of the Canadian standard of living described by our 3-target series, our model results mostly agree on the persistence of these variables. This is especially true for the short term and means that near-sighted changes in the living standard are principally shaped by its history within the last year. When expanding our predictor set, this persistence is less crucial because of possible overfitting/over-parametrization in the extended VAR model, but the additional variables still end up bringing useful information to our forecasts. Allowing for nonlinearity with RF and BART vastly improves our model accuracy, which suggests that living standards in Canada depend on the growth of our selected variables in a complex way, rather than a simple linear one.

5.2 Graphical Forecast Results and Interpretation

Below, we will present our graphical forecast for three of the optimized models. Tables recording the exact forecast number are in the appendix. We begin with the commentary and interpretations of our models' forecasts. Then, we compare the forecast from two horizons and how it differs due to the difference in the optimal lag used.

5.2.1 Short-Horizon Models

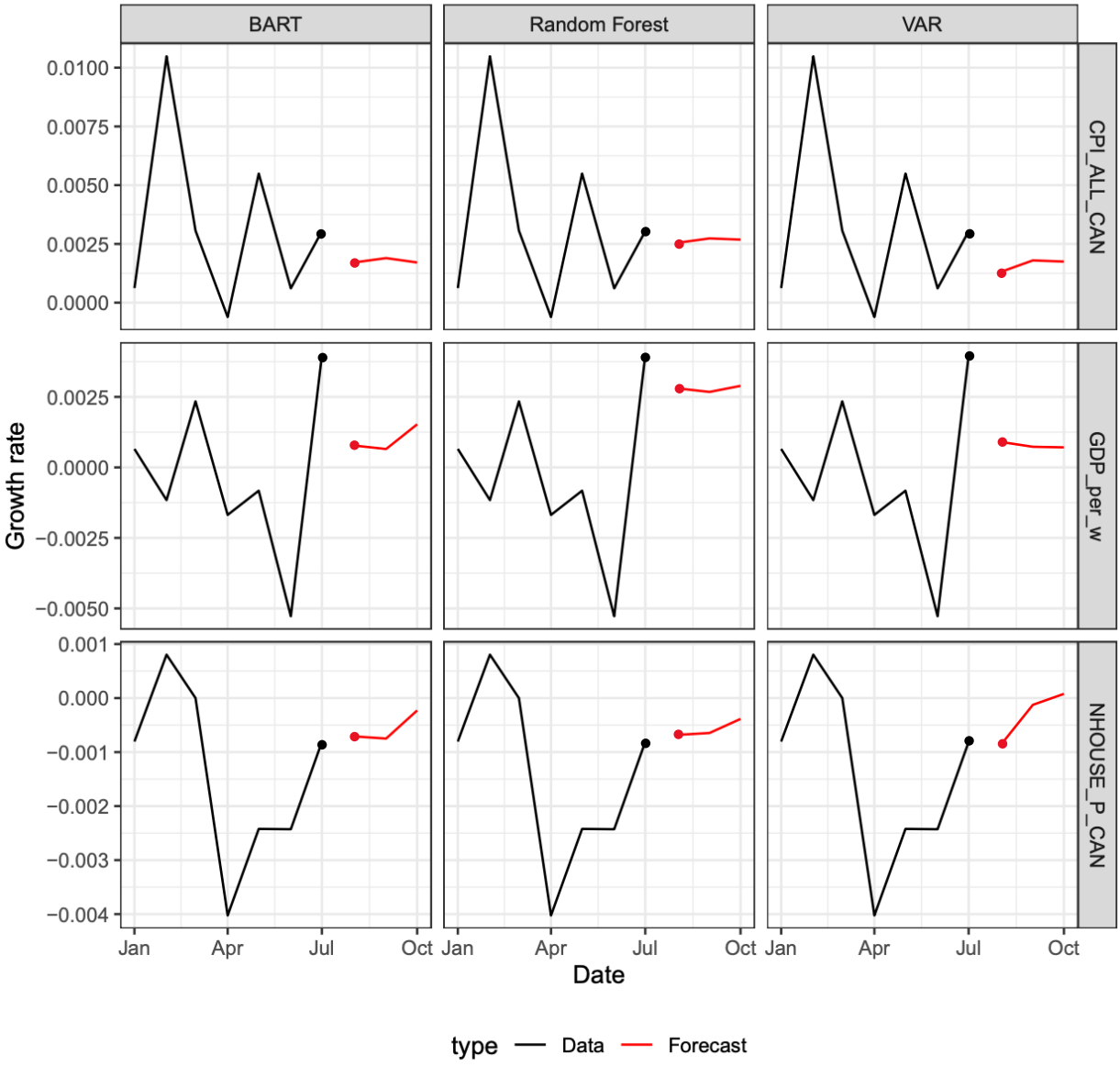
The short-horizon models are optimized to forecast one to three months (h = 1-3) ahead from the most updated available data, which is until August 2025. The following Graph 1 displays the models' Canadian economy forecast for our three target variables, with the black line showing historical data starting from January 2025 and the red line displaying the three-month forecast.

The first row provides a forecast for inflation, with models left to right being BART, Random Forest, and Large VAR. Overall, models show a relatively mild and stable monthly inflation compared to volatility in the first half of 2025. BART and RF give slightly higher CPI estimates than the VAR model. The machine learning duo performs better in this regard, as we can see Canada's monthly inflation bounce back in September and October 2025 (Trading Economics, 2025). VAR, however, providing a lower inflation forecast, might indicate model inefficiency in capturing that bounce-back. Also, all three models show the pattern that the monthly inflation rate will start low in September and increase in October. However, September inflation is higher than October in the newest release.

Next, the second row provides a forecast for the GDP per worker growth rate. The forecasts are all close to zero but with small positive growth. In general, the forecasts show a gradual economic "soft landing", consistent with what is expected from the news. GDP per worker can be interpreted as labour productivity, which shows no strong rebound and remains a concern in the last decades. Moreover, comparing models, RF is slightly more optimistic than the conservative BART and VAR here

The growth rate of the house price index shows a clearer increasing momentum/ trend for the one-to-three-month forecast. With the growth rate approaching zero, it implies a slowing down of the previous national house price falling trend. In particular, we observe that VAR shows the strongest upward trend, while there is a slight dip for BART's forecast. Two reasons might help explain. First, the Bank of Canada's decision to further lower the interest rate creates monetary stimulus, boosting asset prices. Secondly, the recent housing starts, and supply of houses fell significantly. People expecting a rebound of the housing market decided to "buy the dip" today. Such factors are captured by our model predictors.

Graph 1: Short−horizon forecasts (h = 1−3)



5.2.2 Long-Horizon Models

The long-horizon models are optimized to forecast one to twelve months (h = 1-12) ahead from the most updated available data. The following graph (Graph 2) displays the models' Canadian economy forecast for our three target variables, with the black line showing historical data starting from January 2024 and the red line presenting the 12-month forecast.

The first row provides a forecast for inflation. Overall, all models show a relatively mild and stable monthly inflation in the mid-term (h =1-12), which might reflect either the underperformance of the model in longer-term forecasts or a calming of the economy as a result of a soft landing. First, our

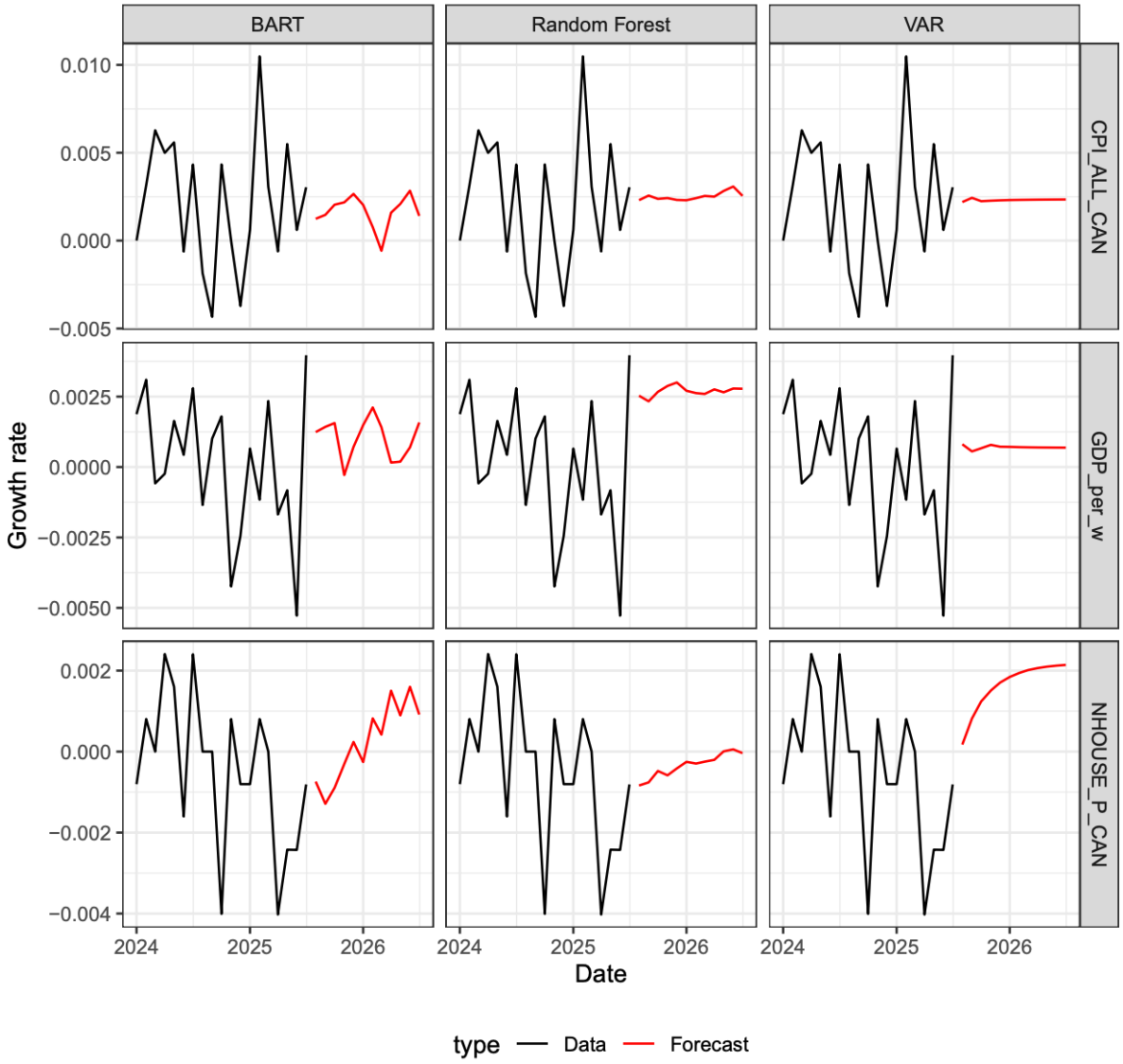
forecasting built from data today has decaying predictive power for later months (h = 7-12). We might better predict with a quarterly data model and change it back to monthly data. In addition, VAR and Random Forest produce a much flatter forecast for inflation in the long horizon. In comparison, BART is the most volatile with a negative dip around h=7-9. This divergence between two machine learning models reflects a structural difference in model building for these two models, despite having similar out-of-bag RMSE in CV.

The second row provides a forecast for the GDP per worker growth rate. All models' forecasts lie in a range of 0.1 to 0.2%, with RF providing the highest forecast, and VAR delivering the lowest. Such a monthly growth rate implies a roughly 1.2-to-1.5%-year growth rate, consistent with Canada's recent years' GDP growth struggle, especially in per capita/worker terms. Breaking down into each model, the random forest's optimism may be a result of the Bank of Canada's recent policy rate cut. Furthermore, VAR is smoother than the other two machine learning models. This might be a result of its linear structure failing to capture nonlinear relations. When Canada is facing a potential output "soft landing", it creates nonlinear feedback among variables that VAR cannot capture well. Here also show our forecast limitations. If time allowed, going beyond and decomposing that 12 RF models we fitted (h=1-12) might deliver better analysis on the difference between variables' explanatory power.

The growth rate of the house price index demonstrates an upward trend for the one-to-twelve-month forecast. All forecasts show a negative to positive trend that will persist. VAR, in particular, predicts the strongest but smooth upward recovery. We interpreted that as reflecting the mean reversion of the housing index in VAR. Traditionally, Canda housing market rebounded strongly after a downturn, due to a time mismatch between demand and supply (long construction period). Furthermore, RF is relatively pessimistic on the house price, barely crossing zero despite having an upward trend. BART's big upward trend is also worth mentioning. This might be a consequence of nonlinear sensitivity to recent turning points, such as reducing housing start/ construction, and monetary policy changes.

One point to note for long horizon forecast as a whole is that BART's forecast is more unstable than VAR and RF, given that BART is Bayesian and randomly constructed. It might imply instead that BART fits the model better and provides better nonlinear forecasts, especially for a short-horizon model.

Graph 2: Long−horizon forecasts (h = 1−12)



5.2.3 Comparison across Short and Long Horizon

To compare our short- and long-horizon forecast models, we interpret them from three perspectives: volatility differences, inflation risk, and economic prediction. First, in terms of the volatility of the forecast, model divergence grows in the long horizon, but short horizon models cluster closely. Especially for inflation and GDP per worker, short-horizon models do not have a huge difference in the forecast delivered. Long-horizon models' difference explodes with BART having the highest volatility and VAR being the most stable.

Secondly, short-horizon models' inflation forecasts all predict a soft landing, while long-horizon model diverges in their inflation trend. BART predict a turning point moving back up, while RF and VAR

suggest anchoring. Thirdly, long-horizon models' divergence in forecast might be a result of growing macro uncertainty in 2026, rather than the long-time frame considered. Stepping into 2026, when the US mid-term election will be held, there will be more risk and volatility not existed in the short-term we considered.

6 Conclusion

This paper aims to examine how forecasting models differ in their predictive power of the Canadian living standard. We adopted GDP per worker, CPI inflation rate, and the national housing price as the indicators, and applied consistent expanding-window cross-validation using the most updated 2025 LCDMA dataset. We vary our analysis by evaluating models for both short (1–3 months) and long (1–12 months) horizons.

Our key findings show that the simple VAR is a decent baseline, while the large VAR helps reduce RMSE significantly. Also, forecasting with machine learning models under a similar set of parameters and validation methods to the VAR suggests that ML models perform much better overall. In addition, we provided a Canadian economy forecast based on our models and the most updated data from the dataset. The models' forecasts diverge for the long horizon but agree on a short-horizon Canadian economic "soft landing." This implies that model performance and selection depend on the forecasting horizon.

Future directions might rely on more structured VAR models and other machine learning methods, such as RNNs. A mixed forecasting frequency might also help improve forecasting, especially in the long horizon.

References

Bergmeir, Christoph, Rob J. Hyndman, and Bonsoo Koo. 2018. "A note on the validity of cross-validation for evaluating autoregressive time series prediction." Computational Statistics and Data Analysis 120 (March): 70–83. https://doi.org/10.1016/j.csda.2017.11.003.

Buturac, Goran. 2022. "Measurement of Economic Forecast Accuracy: A Systematic Overview of the Empirical Literature." Journal of Risk and Financial Management 15, no. 1: 1. https://doi.org/10.3390/jrfm15010001.

Chen, Xiao-Shan, Min Gyeong Kim, Chi-Ho Lin, and Hyung Jong Na. 2025. "Development of per Capita GDP Forecasting Model Using Deep Learning: Including Consumer Goods Index and Unemployment Rate." Sustainability 17, no. 3: 843. https://doi.org/10.3390/su17030843.

Chudý, Marek, and Erhard Reschenhofer. 2019. "Macroeconomic Forecasting with Factor-Augmented Adjusted Band Regression." Econometrics 7, no. 4: 46. https://doi.org/10.3390/econometrics7040046.

Commission on the Measurement of Economic Performance and Social Progress (CMEPSP). 2009. Report by the Commission on the Measurement of Economic Performance and Social Progress. Accessed December 2, 2025. https://www.stat.si/doc/drzstat/stiglitz%20report.pdf.

Deaton, Angus. 2008. "Income, Health, and Well-Being around the World: Evidence from the Gallup World Poll." Journal of Economic Perspectives 22, no. 2 (Spring): 53–72.

Fortin-Gagnon, Olivier, Maxime Leroux, Dalibor Stevanovic, and Stéphane Surprenant. 2022. "A large Canadian database for macroeconomic analysis." Canadian Journal of Economics / Revue canadienne d'économique. https://doi.org/10.1111/caje.12618.

Jones, Charles I., and Peter J. Klenow. 2016. "Beyond GDP? Welfare across Countries and Time." Working paper, Stanford GSB and NBER (February 10). Accessed December 2, 2025. https://web.stanford.edu/~chadj/rawls.pdf.

Li, Jiahan, and Weiye Chen. 2014. "Forecasting macroeconomic time series: LASSO-based approaches and their forecast combinations with dynamic factor models." International Journal of Forecasting 30, no. 4: 996–1015. https://doi.org/10.1016/j.ijforecast.2014.03.016.

Organisation for Economic Co-operation and Development (OECD). 2025. OECD Economic Surveys: Canada 2025. Paris: OECD Publishing.

https://www.oecd.org/content/dam/oecd/en/publications/reports/2025/05/oecd-economic-surveys-canada-2025_ee18a269/28f9e02c-en.pdf.

Stock, James H., and Mark W. Watson. 2001. "Vector Autoregressions." Journal of Economic Perspectives 15, no. 4 (Fall): 101–15. https://www.princeton.edu/~mwatson/papers/Stock_Watson_JEP_2001.pdf.

Trading Economics. n.d. "Canada Inflation CPI." Accessed December 2, 2025. https://tradingeconomics.com/canada/inflation-cpi.

Vincent, Nicolas. 2024. “Monetary Policy Decision-Making: Behind the Scenes.” Remarks before the Chambre de commerce et industrie de Sherbrooke, Sherbrooke, Quebec, September 19. Bank of Canada. https://www.bankofcanada.ca/2024/09/monetary-policy-decision-making-behind-the-scenes/.

Appendix A: Table and Graph

Table 1 – Augmented Dickey–Fuller test results

Variable	Statistic	p-value
GDP_new	-7.92	0.01
IP_new	-7.15	0.01
NDM_new	-7.52	0.01
DM_new	-7.38	0.01
CON_new	-7.88	0.01
RT_new	-7.55	0.01
WT_new	-8.53	0.01
PA_new	-6.73	0.01
FIN_new	-6.99	0.01
BSI_new	-7.56	0.01
GPI_new	-7.43	0.01
SPI_new	-8.65	0.01
TOT_HRS_CAN	-9.37	0.01
CPI_ALL_CAN	-6.96	0.01
IPPI_MACH_CAN	-7.13	0.01
IPPI_METAL_CAN	-5.90	0.01
OILP_new	-8.42	0.01
OIL_CAN_new	-12.40	0.01
BANK_RATE_L	-6.68	0.01
TBILL_3M	-6.47	0.01
TBILL_6M	-6.93	0.01
G_AVG_5.10.Bank_rate	-4.07	0.01
G_AVG_10p.TBILL_3M	-3.86	0.01586616
USDCAD_new	-7.82	0.01
GBPCAD_new	-9.08	0.01
WTISPLC	-8.47	0.01
Exp_BP_new	-8.52	0.01
Imp_BP_new	-7.75	0.01
FOR_SEC_NETFLOW	-5.43	0.01
hstart_CAN_new	-9.65	0.01
NHOUSE_P_CAN	-4.12	0.01
CRED_HOUS_MORT	-3.51	0.04136784
CRED_HOUS	-3.22	0.08423760
EMP_CONS_CAN	-8.09	0.01
EMP_FOR_OIL_CAN	-7.13	0.01
TSX_CLO	-7.94	0.01
TSX_HI	-7.48	0.01
EMP_CAN	-8.09	0.01
EMP_SERV_CAN	-8.70	0.01
EMP_FIN_CAN	-8.62	0.01
EMP_MANU_CAN	-6.49	0.01
EMP_PART_CAN	-10.55	0.01
EMP_SALES_CAN	-8.31	0.01
UNEMP_CAN	-7.10	0.01
UNEMP_DURA_1.4_CAN	-9.43	0.01
UNEMP_DURA_5.13_CAN	-8.56	0.01
UNEMP_DURAvg_CAN_new	-7.50	0.01
CLAIMS_CAN	-5.77	0.01

Table 2: Top 40 Variables Ranked by Maximum Correlation with Target Variables

Variable	Correlation Score
IP_new	0.488
GPI_new	0.474
BSI_new	0.447
GDP_new	0.436
NDM_new	0.367
DM_new	0.345
OILP_new	0.344
SPI_new	0.338
WT_new	0.329
RT_new	0.296
WTISPLC	0.281
Exp_BP_new	0.269
CRED_HOUS	0.239
CRED_HOUS_MORT	0.231
G_AVG_10p.TBILL_3M	0.226
BANK_RATE_L	0.218
Imp_BP_new	0.209
FIN_new	0.209
IPPI_METAL_CAN	0.204
G_AVG_5.10.Bank_rate	0.196
TBILL_3M	0.193
CON_new	0.185
TBILL_6M	0.184
UNEMP_CAN	0.175
EMP_CONS_CAN	0.166
EMP_PART_CAN	0.140
EMP_FOR_OIL_CAN	0.137
UNEMP_DURA_5.13_CAN	0.134
EMP_SERV_CAN	0.132
EMP_MANU_CAN	0.131
EMP_CAN	0.121
hstart_CAN_new	0.121
USDCAD_new	0.120
TSX_CLO	0.109
TOT_HRS_CAN	0.109
OIL_CAN_new	0.104
TSX_HI	0.102
EMP_SALES_CAN	0.100
PA_new	0.089
EMP_FIN_CAN	0.088

Table 3: Variables selected for the 17-variable VAR

Variable	**Block**	**Short_Description**
GDP_per_w	Target	GDP per worker, monthly growth
CPI_ALL_CAN	Target	Headline CPI inflation, monthly
NHOUSE_P_CAN	Target	National house price index, monthly
IP_new	Real activity	Industrial production index, monthly
BSI_new	Real activity	Business sentiment / survey index
UNEMP_CAN	Labour	Unemployment rate, Canada
EMP_CONS_CAN	Labour	Construction employment, Canada
hstart_CAN_new	Housing	Housing starts, Canada, monthly
CRED_HOUS	Credit	Household / housing credit stock
CRED_HOUS_MORT	Credit	Mortgage credit for housing
BANK_RATE_L	Interest	BoC policy rate (level)
TBILL_3M	Interest	3-month T-bill rate
Exp_BP_new	External	Exports of goods (BOP), monthly
Imp_BP_new	External	Imports of goods (BOP), monthly
IPPI_METAL_CAN	Prices	Metal producer price index, Canada
OILP_new	Commodities	Oil price index
WTISPLC	Commodities	WTI spot crude oil price (level)

Table 4: CV lag selection for small VAR, short horizon

Lag p	($h = 1$–3) **RMSE**
1	0.004199
2	0.004237
3	0.004218
4	0.004154
5	0.004339
6	0.004244
7	**0.003892**
8	0.004659
9	0.004638
10	0.004816
11	0.004833
12	0.004840

Table 5: CV lag selection for small VAR, long horizon

Lag p	($h = 1$–12) **RMSE**
1	0.004263
2	0.004383
3	0.004113
4	0.004147
5	0.004772
6	0.006166
7	**0.003624**
8	0.003738
9	0.003823
10	0.003777
11	0.003910
12	0.004035

Table 6: CV lag selection for large VAR, short horizon

Lag p	($h = 1$–3) **RMSE**
1	0.003155
2	**0.003001**
3	0.003143
4	0.003198
5	0.003495
6	0.004099
7	0.004484
8	0.004089
9	0.004150
10	0.004919
11	0.005630
12	0.005035

Table 7: CV lag selection for large VAR, long horizon

Lag p	($h = 1$–12) **RMSE**
1	**0.003424**
2	0.003443
3	0.003514
4	0.003691
5	0.003966
6	0.004225
7	0.004610
8	0.004269
9	0.004439
10	0.005668
11	0.006925
12	0.005855

Table 8: CV lag selection for Random Forest, short horizon

Lag p	($h = 1$–3) **RMSE**
1	0.002540
2	0.002568
3	0.002598
4	0.002667
5	0.002521
6	0.002418
7	0.002431
8	0.002520
9	0.002639
10	0.002610
11	0.002422
12	**0.002397**
Final CV RMSE with 500 trees:	0.002372

Table 9: CV lag selection for Random Forest, long horizon

Lag p	($h = 1$–12) **RMSE**
1	0.002676
2	0.002659
3	0.002554
4	0.002528
5	**0.002518**
6	0.002622
7	0.002591
8	0.002639
9	0.002668
10	0.002641
11	0.002642
12	0.002625
Final CV RMSE with 500 trees:	0.002554

Table 10: CV lag selection for BART, short horizon

Lag p	($h = 1$–3) **RMSE**
1	0.002717
2	0.002857
3	0.002830
4	0.003077
5	0.002858
6	0.002793
7	0.002748
8	0.002878
9	0.002929
10	0.002784
11	**0.002660**
12	0.002753

Table 11: CV lag selection for BART, long horizon

Lag p	($h = 1$–12) **RMSE**
1	0.002810
2	0.002822
3	**0.002717**
4	0.002777
5	0.002843
6	0.002826
7	0.002915
8	0.003010
9	0.003067
10	0.003093
11	0.003023
12	0.003055

Table 12: Short-horizon forecasts for GDP per worker (%)

h	VAR	Random Forest	BART
1	0.0903	0.2803	0.0774
2	0.0730	0.2676	0.0652
3	0.0710	0.2890	0.1528

Table 13: Long-horizon forecasts for GDP per worker (%)

h	VAR	Random Forest	BART
1	0.0808	0.2533	0.1240
2	0.0554	0.2332	0.1426
3	0.0667	0.2670	0.1560
4	0.0786	0.2877	-0.0286
5	0.0721	0.3001	0.0715
6	0.0716	0.2707	0.1487
7	0.0705	0.2624	0.2114
8	0.0700	0.2595	0.1411
9	0.0695	0.2757	0.0157
10	0.0692	0.2652	0.0190
11	0.0690	0.2786	0.0696
12	0.0688	0.2778	0.1582

Table 14: Short-horizon forecasts for CPI inflation (%)

h	VAR	Random Forest	BART
1	0.1309	0.2547	0.1711
2	0.1798	0.2734	0.1896
3	0.1748	0.2684	0.1710

Table 15: Long-horizon forecasts for CPI inflation (%)

h	VAR	Random Forest	BART
1	0.2195	0.2304	0.1243
2	0.2437	0.2565	0.1468
3	0.2244	0.2378	0.2045
4	0.2271	0.2423	0.2176
5	0.2293	0.2316	0.2658
6	0.2310	0.2298	0.2032
7	0.2319	0.2419	0.0769
8	0.2326	0.2545	-0.0578
9	0.2332	0.2505	0.1589
10	0.2336	0.2827	0.2089
11	0.2339	0.3079	0.2840
12	0.2342	0.2545	0.1409

Table 16: Short-horizon forecasts for house prices (%)

h	VAR	Random Forest	BART
1	-0.0867	-0.0681	-0.0709
2	-0.0125	-0.0646	-0.0749
3	0.0078	-0.0385	-0.0228

Table 17: Long-horizon forecasts for house prices (%)

h	VAR	Random Forest	BART
1	0.0174	-0.0837	-0.0740
2	0.0815	-0.0760	-0.1289
3	0.1240	-0.0479	-0.0893
4	0.1507	-0.0587	-0.0307
5	0.1705	-0.0415	0.0238
6	0.1844	-0.0253	-0.0258
7	0.1943	-0.0295	0.0820
8	0.2013	-0.0248	0.0419
9	0.2063	-0.0204	0.1504
10	0.2098	0.0006	0.0894
11	0.2123	0.0055	0.1600
12	0.2141	-0.0038	0.0916

Table 18: Summary of Optimal Lags and Forecast Accuracy

Model	p ($h = 1$–3)	RMSE 1–3	p ($h = 1$–12)	RMSE 1–12
Small VAR (3 vars)	7	0.0039	7	0.0036
Large VAR (17 vars)	2	0.0030	1	0.0034
Random Forest	12	0.0024	5	0.0026
BART	11	0.0027	3	0.0027

Appendix B: Code

Note: Too many models and CV, so I do it in a separate Notebook, Thus, their headers and first few sections duplicate. However, in order to keep it as it is, I would paste everything completely for reference..

B1. VAR notebook

````
---
title: "Baseline VAR – Canada GDP, Housing & Inflation"
output:
  pdf_document: default
  html_document: default
  word_document: default
---
# 0. Packages
```{r, message=FALSE, warning=FALSE, echo=TRUE, results='hide'}
R_packages <- c("readr", "dplyr", "tseries", "vars", "ggplot2", "lubridate", "scales", "tsDyn", "xts", "zoo", "tidyverse", "lubridate")

options(repos = c(CRAN="http://cran.rstudio.com"))
if (!requireNamespace("librarian", quietly = TRUE)) install.packages("librarian")
librarian::shelf(R_packages)

if (!requireNamespace("tinytex", quietly = TRUE)) install.packages("tinytex")
if (Sys.which("pdflatex")=="") {library(tinytex); tinytex::install_tinytex()}
```

# 1. Data
```{r}
df <- read_csv("balanced_can_md.csv", show_col_types = FALSE)

df_subset <- df %>%
  dplyr::select(
    Date,
    GDP_new,
    CPI_ALL_CAN,
    NHOUSE_P_CAN,
    EMP_CAN
)

head(df_subset)
```
## 1.1 Variable: National-level variables from Fortin
```{r}
## initial screening out provincial variables and some dubplicate one.

vars_all_blocks <- c(
  "GDP_new","IP_new","NDM_new","DM_new","CON_new","RT_new","WT_new",
  "PA_new","FIN_new","BSI_new","GPI_new","SPI_new","TOT_HRS_CAN",
  "CPI_ALL_CAN","IPPI_MACH_CAN","IPPI_METAL_CAN",
  "OILP_new","OIL_CAN_new",
  "BANK_RATE_L","TBILL_3M","TBILL_6M",
  "G_AVG_5.10.Bank_rate","G_AVG_10p.TBILL_3M",
  "USDCAD_new","GBPCAD_new","OILP_new","OIL_CAN_new","WTISPLC",
  "Exp_BP_new","Imp_BP_new","FOR_SEC_NETFLOW",
  "hstart_CAN_new","NHOUSE_P_CAN","CRED_HOUS_MORT","CRED_HOUS",
  "FIN_new","EMP_CONS_CAN","EMP_FOR_OIL_CAN","TSX_CLO","TSX_HI",
  "EMP_CAN","EMP_SERV_CAN","EMP_FIN_CAN","EMP_MANU_CAN",
  "EMP_CONS_CAN","EMP_PART_CAN","EMP_FOR_OIL_CAN","EMP_SALES_CAN",
  "UNEMP_CAN","UNEMP_DURA_1.4_CAN","UNEMP_DURA_5.13_CAN",
  "UNEMP_DURAvg_CAN_new","CLAIMS_CAN"
)

df_large <- df %>% dplyr::select(Date, all_of(vars_all_blocks))
head(df_large)
```

## 1.2 STATIONARITY TEST
```{r}
vars <- names(df_large)
results <- data.frame(
  variable  = character(),
  statistic = numeric(),
````
````

```
  p_value   = numeric(),
  stringsAsFactors = FALSE
)

for (v in vars) {
  x <- df_large[[v]]
  if (is.numeric(x)) {

    test <- withCallingHandlers(
      adf.test(na.omit(x)),
      warning = function(w) {
        if (grepl("p-value smaller than printed p-value", w$message)) {
          invokeRestart("muffleWarning")
        }
      }
    )

    results <- rbind(
      results,
      data.frame(
        variable  = v,
        statistic = as.numeric(test$statistic),
        p_value   = as.numeric(test$p.value)
      )
    )
  }
}

print(results)
```

1.3 Contruct GDP per worker growth rate estimate
```{r}
df_subset$GDP_per_w = df_subset$GDP_new - df_subset$EMP_CAN
adf.test(df_subset$GDP_per_w)
df_large$GDP_per_w = df_large$GDP_new - df_large$EMP_CAN
```

2. Three variable baseline
```{r}
df_subset$Date <- as.Date(df_subset$Date)

start_year  <- as.numeric(format(min(df_subset$Date), "%Y"))
start_month <- as.numeric(format(min(df_subset$Date), "%m"))
vars_3 <- c("GDP_per_w", "NHOUSE_P_CAN", "CPI_ALL_CAN")
```

2.1 K-Fold Expanding Window Cross Validation
```{r}
X <- df_subset %>%
  dplyr::select(all_of(vars_3)) %>%
  as.matrix() %>%
  stats::na.omit()

n <- nrow(X)
## define a CV function for our purpose: optimize forecast for h= 1-3 and for h= 1-12, respectively
ts_cv_rmse_for_p <- function(X, p, K, max_h) {
  n <- nrow(X)
  if (n <= p + K + 10) return(NA_real_)

  cutpoints <- floor(seq(from = p + 20, to = n - max_h, length.out = K))

  err_list <- vector("list", max_h)

  for (t0 in cutpoints) {
    train <- X[1:t0, , drop = FALSE]

    model <- try(VAR(train, p = p, type = "const"), silent = TRUE)
    if (inherits(model, "try-error")) next

    # 1 to max_h-step-ahead forecasts from t0
    fc_all <- predict(model, n.ahead = max_h)$fcst

    for (h in 1:max_h) {
      # forecast at horizon h for all variables
```

```
    fc_h <- sapply(fc_all, function(m) m[h, "fcst"])

    # actual at t0 + h
    actual_h <- X[t0 + h, ]

    err_list[[h]] <- rbind(err_list[[h]], actual_h - fc_h)
   }
  }

  # if no errors were collected, return NA
  if (all(vapply(err_list, is.null, logical(1L)))) return(NA_real_)

  # RMSE per horizon (averaged across variables),
  rmse_h <- numeric(max_h)
  for (h in 1:max_h) {
   if (!is.null(err_list[[h]]) && nrow(err_list[[h]]) > 0L) {
    rmse_vec <- sqrt(colMeans(err_list[[h]]^2, na.rm = TRUE))
    rmse_h[h] <- mean(rmse_vec)
   } else {
    rmse_h[h] <- NA_real_
   }
  }

  mean(rmse_h[is.finite(rmse_h)])
}
```

2.2 Small VAR: Short Term Model (h=1-3)

```{r}
min_p  <- 1
max_p  <- 12
step_p <- 1

p_grid <- seq(min_p, max_p, by = step_p)

results_p <- data.frame(p = integer(), avg_RMSE = numeric())

for (p in p_grid) {
  avg_rmse <- ts_cv_rmse_for_p(X, p, K = 10, max_h = 3)  # CV over horizons 1–3
  results_p <- rbind(results_p, data.frame(p = p, avg_RMSE = avg_rmse))
  cat("Done CV: p =", p, "avg_RMSE =", round(avg_rmse, 6), "\n")
}

print(results_p)

valid      <- which(is.finite(results_p$avg_RMSE))
best_id    <- valid[ which.min(results_p$avg_RMSE[valid]) ]
best_p_row <- results_p[best_id, ]
best_p_row

p_opt <- best_p_row$p

var_ts <- df_subset %>%
  dplyr::select(all_of(vars_3)) %>%
  ts(start = c(start_year, start_month), frequency = 12) %>%
  stats::na.omit()

var_base <- VAR(y = var_ts, p = p_opt, type = "const")
summary(var_base)
```

2.3 Small VAR: Long term Model (h=1-12)

```{r}
results_pl <- data.frame(p = integer(), avg_RMSE = numeric())

for (p in p_grid) {
  avg_rmsel <- ts_cv_rmse_for_p(X, p, K = 10, max_h = 12)  # CV over horizons 1–3
  results_pl <- rbind(results_pl, data.frame(p = p, avg_RMSE = avg_rmsel))
  cat("Done CV: p =", p, "avg_RMSE =", round(avg_rmsel, 6), "\n")
}

print(results_pl)

validl      <- which(is.finite(results_pl$avg_RMSE))
best_idl    <- validl[ which.min(results_pl$avg_RMSE[validl]) ]
best_p_rowl <- results_pl[best_idl, ]
```

```
best_p_rowl

p_optl <- best_p_rowl$p

var_base <- VAR(y = var_ts, p = p_optl, type = "const")
summary(var_base)
```

3. Large model Variable selection: How we select the FINAL 17
```{r}
targets <- c("GDP_per_w", "CPI_ALL_CAN", "NHOUSE_P_CAN")

all_numeric <- names(df_large)[sapply(df_large, is.numeric)]
all_numeric <- setdiff(all_numeric, "Date")

## 1. Correlation scores vs targets
corr_score <- function(vname) {
  x <- df_large[[vname]]
  sapply(targets, function(tn) {
    y <- df_large[[tn]]
    abs(cor(x, y, use = "complete.obs"))
  }) |>
    max(na.rm = TRUE)
}

# score for variable
scores_vec <- sapply(all_numeric, corr_score)

corr_table <- data.frame(
  variable = all_numeric,
  score    = as.numeric(scores_vec)
) |>
  dplyr::arrange(dplyr::desc(score))

# Keep top 40  by corr scores
corr_table <- corr_table[!corr_table$variable %in% targets, ]
top40_vars <- head(corr_table$variable, 40)

print(head(corr_table, 40))

## ----- separation line
## 2. Define economic blocks

block_real <- c(
  "GDP_new", "BSI_new", "GPI_new", "SPI_new",
  "IP_new", "NDM_new", "DM_new", "CON_new",
  "RT_new", "WT_new", "PA_new", "FIN_new",
  "TOT_HRS_CAN", "GOOD_HRS_CAN"
)

block_labour <- c(
  "EMP_CAN", "EMP_SERV_CAN", "EMP_FOR_OIL_CAN",
  "EMP_CONS_CAN", "EMP_SALES_CAN", "EMP_FIN_CAN",
  "EMP_MANU_CAN", "EMP_PART_CAN",
  "UNEMP_CAN",
  "UNEMP_DURA_1-4_CAN", "UNEMP_DURA_5-13_CAN",
  "UNEMP_DURA_14-25_CAN", "UNEMP_DURA_27+_CAN",
  "UNEMP_DURAvg_CAN_new",
  "CLAIMS_CAN",
  "TOT_HRS_CAN", "GOOD_OVT_HRS_CAN"
)

block_house_price <- c(
  "NHOUSE_P_CAN",
  "NHOUSE_P_NF", "NHOUSE_P_PEI", "NHOUSE_P_NS", "NHOUSE_P_NB",
  "NHOUSE_P_QC", "NHOUSE_P_ONT", "NHOUSE_P_MAN", "NHOUSE_P_SAS",
  "NHOUSE_P_ALB", "NHOUSE_P_BC"
)

block_house_activity <- c(
  "hstart_CAN_new",
  "hstart_NF_new", "hstart_PEI_new", "hstart_NS_new",
  "hstart_NB_new", "hstart_QC_new", "hstart_ONT_new",
  "hstart_MAN_new", "hstart_SAS_new", "hstart_ALB_new",
  "hstart_BC_new",
```

```
  "build_Total_CAN_new", "build_Ind_CAN_new", "build_Comm_CAN_new",
  "build_Total_NF_new", "build_Total_PEI_new", "build_Total_NS_new",
  "build_Total_NB_new", "build_Total_QC_new", "build_Total_ONT_new",
  "build_Total_MAN_new", "build_Total_SAS_new", "build_Total_ALB_new",
  "build_Total_BC_new"
)

block_manu <- c(
  "MANU_N_ORD_new", "MANU_UNFIL_new", "MANU_TOT_INV_new",
  "MANU_INV_RAT_new", "N_DUR_INV_RAT_new",
  "DUR_N_ORD_new", "DUR_UNFIL_new",
  "DUR_TOT_INV_new", "DUR_INV_RAT_new"
)

block_money_credit <- c(
  "M3", "M2p", "M_BASE1",
  "CRED_BUS_cb", "CRED_HOUS_cb", "CRED_MORT_HOUSE_cb", "CRED_T_cb",
  "CRED_HOUS_non_MORT", "CRED_HOUS_MORT", "CRED_HOUS", "CRED_BUS"
)

block_rates <- c(
  "BANK_RATE_L",
  "GOV_AVG_1_3Y", "GOV_AVG_3_5Y", "GOV_AVG_5_10Y", "GOV_AVG_10pY",
  "MORTG_1Y", "MORTG_5Y",
  "TBILL_3M", "TBILL_6M",
  "G_AVG_1-3-Bank_rate", "G_AVG_3-5-Bank_rate",
  "G_AVG_5-10-Bank_rate", "TBILL_6M-Bank_rate",
  "G_AVG_10p-TBILL_3M"
)

block_external <- c(
  "RES_TOT", "RES_USD", "RES_IMF",
  "Imp_BP_new", "IOIL_BP_new", "Exp_BP_new", "EOIL_BP_new",
  "EX_ENER_BP_new", "EX_MINER_BP_new", "EX_METAL_BP_new",
  "EX_IND_EQUIP_BP_new", "EX_TRANSP_BP_new", "EX_CONS_BP_new",
  "IMP_METAL_BP_new", "IMP_IND_EQUIP_BP_new",
  "IMP_TRANSP_BP_new", "IMP_CONS_BP_new",
  "USDCAD_new", "JPYCAD_new", "GBPCAD_new",
  "CAN_EQTY_NETFLOW", "CAN_SEC_NETFLOW",
  "FOR_SEC_NETFLOW", "CAN_US_SEC_NETFLOW"
)

block_prices <- c(
  "CPI_ALL_CAN", "CPI_SHEL_CAN", "CPI_CLOT_CAN", "CPI_HEA_CAN",
  "CPI_MINUS_FOO_CAN", "CPI_MINUS_FEN_CAN",
  "CPI_GOO_CAN", "CPI_DUR_CAN", "CPI_SERV_CAN",
  "IPPI_CAN", "IPPI_ENER_CAN", "IPPI_WOOD_CAN",
  "IPPI_METAL_CAN", "IPPI_MOTOR_CAN", "IPPI_MACH_CAN"
)

block_markets <- c(
  "OILP_new", "OIL_CAN_new", "WTISPLC",
  "TSX_HI", "TSX_LO", "TSX_CLO"
)

blocks <- list(
  real         = block_real,
  labour       = block_labour,
  house_price  = block_house_price,
  house_activity = block_house_activity,
  manu         = block_manu,
  money_credit = block_money_credit,
  rates        = block_rates,
  external     = block_external,
  prices       = block_prices,
  markets      = block_markets
)

## 3. Helper: score lookup + within-block selection

score_lookup <- function(v) {
  out <- corr_table$score[match(v, corr_table$variable)]
  ifelse(is.na(out), 0, out)
}
```

```
pick_block_reps <- function(block_vars,
                            top_vars,
                            df,
                            n_per_block = 2,
                            corr_cutoff = 0.90) {
  cand <- intersect(block_vars, intersect(top_vars, names(df)))
  if (length(cand) == 0) return(character(0))

  cand_scores <- score_lookup(cand)
  ord         <- order(cand_scores, decreasing = TRUE)
  cand        <- cand[ord]

  picked <- character(0)

  for (v in cand) {
    if (length(picked) == 0) {
      picked <- c(picked, v)
    } else {
      too_close <- FALSE
      for (u in picked) {
        cval <- cor(df[[v]], df[[u]], use = "complete.obs")
        if (is.finite(cval) && abs(cval) > corr_cutoff) {
          too_close <- TRUE
          break
        }
      }
      if (!too_close) {
        picked <- c(picked, v)
      }
    }
    if (length(picked) >= n_per_block) break
  }

  picked
}

## 4. Apply: choose up to 2 per block (variety)

set.seed(12345)

selected_by_block <- lapply(names(blocks), function(bn) {
  vars <- pick_block_reps(
    block_vars   = blocks[[bn]],
    top_vars     = top40_vars,
    df           = df_large,
    n_per_block  = 2,
    corr_cutoff  = 0.90
  )

  if (length(vars) == 0) return(NULL)

  data.frame(
    block    = bn,
    variable = vars,
    stringsAsFactors = FALSE
  )
})


selected_by_block <- do.call(rbind, selected_by_block)
selected_by_block

## 5. Final large-model variable set

vars_large_final <- unique(c(
  targets,
  selected_by_block$variable
))

length(vars_large_final)
vars_large_final

```
4. Large Model: Short Horizon
```{r}
```

```
X_large <- df_large %>%
  dplyr::select(all_of(vars_large_final)) %>%
  as.matrix() %>% na.omit()

p_grid_short <- 1:12
ts_cv_rmse_TARGETS_only <- function(X, p, K = 10, max_h = 3, target_idx) {
  n <- nrow(X)
  start_cv <- floor(n * 0.4)
  cut_points <- round(seq(start_cv, n - max_h, length.out = K))

  err_list <- list()

  for (t0 in cut_points) {
    train <- X[1:t0, , drop = FALSE]
    actual <- X[(t0+1):(t0+max_h), target_idx, drop = FALSE]

    model <- try(VAR(train, p = p, type = "const"), silent = TRUE)
    if (inherits(model, "try-error")) next

    pred_list <- predict(model, n.ahead = max_h)$fcst
    pred_mat <- sapply(target_idx, function(j) pred_list[[j]][,"fcst"])

    err <- actual - pred_mat
    err_list[[length(err_list)+1]] <- err
  }

  all_err <- do.call(rbind, err_list)
  sqrt(mean(all_err^2, na.rm = TRUE))
}

target_idx <- match(
  c("GDP_per_w","CPI_ALL_CAN","NHOUSE_P_CAN"),
  vars_large_final
)

results_large_short_opt2 <- data.frame(p = integer(), avg_RMSE = numeric())

for (p in p_grid_short) {
  rmse_p <- ts_cv_rmse_TARGETS_only(
    X = X_large,
    p = p,
    K = 10,
    max_h = 3,
    target_idx = target_idx
  )
  results_large_short_opt2 <- rbind(
    results_large_short_opt2,
    data.frame(p = p, avg_RMSE = rmse_p)
  )
  cat("Large VAR Short CV: p =", p, "| RMSE =", rmse_p, "\n")
}

## it give CV table p=[1:12] lol

best_id2 <- which.min(results_large_short_opt2$avg_RMSE)
best_p_large_short_opt2 <- results_large_short_opt2$p[best_id2]
best_p_large_short_opt2
```

5. Large: model: Long Horizon

```{r}
p_grid_long <- 1:12
results_large_long_opt2 <- data.frame(p = integer(), avg_RMSE = numeric())

for (p in p_grid_long) {
  rmse_p <- ts_cv_rmse_TARGETS_only(
    X = X_large,
    p = p,
    K = 10,
    max_h = 12,
    target_idx = target_idx
  )
  results_large_long_opt2 <- rbind(
    results_large_long_opt2,
    data.frame(p = p, avg_RMSE = rmse_p)
```

```
 )
 cat("Large VAR Long CV: p =", p, "| RMSE =", rmse_p, "\n")
}

## it give CV table p=[1:12] lol
best_id2L <- which.min(results_large_long_opt2$avg_RMSE)
best_p_large_long_opt2 <- results_large_long_opt2$p[best_id2L]
best_p_large_long_opt2
```

6. RMSE Comparison: Small VAR vs Large VAR

```{r}
rmse_targets_only <- function(X, p, K = 10, max_h = 3, target_idx) {
 n <- nrow(X)
 start_cv  <- floor(n * 0.4)
 cut_pts   <- round(seq(start_cv, n - max_h, length.out = K))

 err_list <- list()

 for (t0 in cut_pts) {
  train  <- X[1:t0, , drop = FALSE]
  actual <- X[(t0 + 1):(t0 + max_h), target_idx, drop = FALSE]

  model <- try(VAR(train, p = p, type = "const"), silent = TRUE)
  if (inherits(model, "try-error")) next

  pred_obj <- predict(model, n.ahead = max_h)$fcst
  pred_mat <- sapply(target_idx, function(j) pred_obj[[j]][, "fcst"])

  err_list[[length(err_list) + 1]] <- actual - pred_mat
 }

 all_err <- do.call(rbind, err_list)
 sqrt(mean(all_err^2, na.rm = TRUE))
}

## Short-term forecast performance (h = 1–3)
# Small model: RMSE from earlier K-fold CV (3-var VAR).
rmse_small_short <- min(results_p$avg_RMSE, na.rm = TRUE)

# Large model: RMSE for the same 3 target series
rmse_large_short_opt2 <- rmse_targets_only(
 X          = X_large,
 p          = best_p_large_short_opt2,
 K          = 10,
 max_h      = 3,
 target_idx = target_idx
)

short_term_table <- data.frame(
 Model     = c("Small VAR (3 variables)",
               "Large VAR (17 variables)"),
 Lag_p     = c(p_opt,
               best_p_large_short_opt2),
 RMSE_1to3 = c(rmse_small_short,
               rmse_large_short_opt2)
)

short_term_table


## Long-term forecast performance (h = 1–12)

# Small model: RMSE from small K-fold CV (3-variable VAR, long horizon)
rmse_small_long <- min(results_pl$avg_RMSE, na.rm = TRUE)

# Large model: RMSE of 3 target series
rmse_large_long_opt2 <- rmse_targets_only(
 X          = X_large,
 p          = best_p_large_long_opt2,
 K          = 10,
 max_h      = 12,
 target_idx = target_idx
)
```

```
long_term_table <- data.frame(
  Model      = c("Small VAR (3 variables)",
                 "Large VAR (17 variables)"),
  Lag_p      = c(p_optl,
                 best_p_large_long_opt2),
  RMSE_1to12 = c(rmse_small_long,
                 rmse_large_long_opt2)
)

long_term_table
```
7. Final baseline VAR models
```{r}
var_ts_small <- ts(
  df_subset %>%
    dplyr::select(all_of(vars_3)),
  start     = c(start_year, start_month),
  frequency = 12
)
# Short-Horizon small VAR
var_small_short <- VAR(
  y    = var_ts_small,
  p    = p_opt,
  type = "const"
)

# Long-Horizon small VAR
var_small_long <- VAR(
  y    = var_ts_small,
  p    = p_optl,
  type = "const"
)

var_ts_large <- ts(
  df_large %>%
    dplyr::select(all_of(vars_large_final)),
  start     = c(start_year, start_month),
  frequency = 12
)

# Short-term large VAR
var_large_short <- VAR(
  y    = var_ts_large,
  p    = best_p_large_short_opt2,
  type = "const"
)

# Long-term large VAR
var_large_long <- VAR(
  y    = var_ts_large,
  p    = best_p_large_long_opt2,
  type = "const"
)

## Too long hide it
#summary(var_small_short)
#summary(var_small_long)
#summary(var_large_short)
#summary(var_large_long)
```


#. 8.Forecast

```{r}
H_short <- 3
H_long  <- 12

#  VAR forecasts
fc_large_short <- predict(var_large_short, n.ahead = H_short)$fcst
fc_large_long  <- predict(var_large_long,  n.ahead = H_long)$fcst

# Last observed date --> so no gap
last_date <- max(df_subset$Date)
```

```
dates_short <- seq(last_date %m+% months(1), by = "1 month", length.out = H_short)
dates_long  <- seq(last_date %m+% months(1), by = "1 month", length.out = H_long)

make_var_long <- function(fc_obj, H, dates_vec) {
  tibble(
    variable = rep(c("GDP_per_w", "CPI_ALL_CAN", "NHOUSE_P_CAN"), each = H),
    h        = rep(1:H, times = 3),
    date     = rep(dates_vec, times = 3),
    forecast = c(
      fc_obj$GDP_per_w[, "fcst"],
      fc_obj$CPI_ALL_CAN[, "fcst"],
      fc_obj$NHOUSE_P_CAN[, "fcst"]
    )
  )
}

var_large_short_long <- make_var_long(fc_large_short, H_short, dates_short)
var_large_long_long  <- make_var_long(fc_large_long,  H_long,  dates_long)

# CSV for the "empirical_results" file to merge other noteboaks.
write.csv(var_large_short_long,
          "VAR_large_all_variables_short.csv",
          row.names = FALSE)

write.csv(var_large_long_long,
          "VAR_large_all_variables_long.csv",
          row.names = FALSE)
```

B2. RF CV Notebook

```
---
title: 'RF Cross validation'
output:
  pdf_document: default
  html_notebook: default
---

## 0.Packages
```{r}
## This note book take 15-18 mins to run, be caseful lol.
R_packages <- c("readr","dplyr","lubridate",
                "randomForest","quantregForest","BART", "tidyr", "ranger", "knitr")
options(repos=c(CRAN="http://cran.rstudio.com"))

## Use Burda style package
if(!requireNamespace("librarian", quietly=TRUE)) install.packages("librarian")
librarian::shelf(R_packages)
```

1. Data
```{r}
# copy from VAR, same thing
df_ml <- read_csv("balanced_can_md.csv", show_col_types = FALSE) %>%
  mutate(
    Date     = as.Date(Date),
    GDP_per_w = GDP_new - EMP_CAN
  ) %>%
  arrange(Date)

targets <- c("GDP_per_w", "CPI_ALL_CAN", "NHOUSE_P_CAN")
features_large <- c(
  "GDP_per_w", "CPI_ALL_CAN", "NHOUSE_P_CAN", "IP_new",
  "BSI_new", "UNEMP_CAN", "EMP_CONS_CAN", "hstart_CAN_new",
  "CRED_HOUS", "CRED_HOUS_MORT", "BANK_RATE_L", "TBILL_3M",
  "Exp_BP_new", "Imp_BP_new", "IPPI_METAL_CAN", "OILP_new",
  "WTISPLC"
)

setdiff(features_large, names(df_ml))

```
## 2. HELPER: build laged panels
```

```r
build_lagged_panel <- function(df, predictors, targets, p, h) {
  dat <- df %>%
    dplyr::select(Date, dplyr::all_of(unique(c(predictors, targets)))) %>%
    arrange(Date)

  for (lag in 1:p) {
    for (v in predictors) {
      new_name <- paste0("L", lag, "_", v)
      dat[[new_name]] <- dplyr::lag(dat[[v]], lag)
    }
  }

  for (v in targets) {
    lead_name <- paste0("F", h, "_", v)
    dat[[lead_name]] <- dplyr::lead(dat[[v]], h)
  }

  lag_cols <- grep("^L[0-9]+_", names(dat), value = TRUE)
  y_cols   <- paste0("F", h, "_", targets)

  dat_final <- dat %>%
    dplyr::select(dplyr::all_of(c(lag_cols, y_cols))) %>%
    tidyr::drop_na()

  list(
    X = as.matrix(dat_final[, lag_cols, drop = FALSE]),
    Y = as.matrix(dat_final[, y_cols,  drop = FALSE])  # columns in same order as targets
  )
}
```

3a. Time Series K-fold CV

```r, cache=TRUE}
rf_ts_cv_rmse_fast <- function(df, predictors, targets,
                               p,
                               K         = 10,
                               max_h,
                               num.trees = 200,
                               mtry_frac = 1/3) {

  se_all <- list()

  for (h in 1:max_h) {
    panel <- build_lagged_panel(df, predictors, targets, p, h)
    X <- panel$X
    Y <- panel$Y
    n <- nrow(X)
    n_y <- ncol(Y)   # 3 targets

    if (n < K + 5) stop("Too few obs after lags/leads for this p / h.")


    start_idx <- floor(0.6 * n)
    fold_ends <- floor(seq(from = start_idx, to = n - 1, length.out = K))

    se_h <- matrix(NA_real_, nrow = 0, ncol = n_y)

    for (end_idx in fold_ends) {
      train_idx <- 1:end_idx
      test_idx  <- end_idx + 1
      if (test_idx > n) break

      x_train <- X[train_idx, , drop = FALSE]
      y_train <- Y[train_idx, , drop = FALSE]
      x_test  <- X[test_idx, , drop = FALSE]

      mtry_val <- max(1L, floor(ncol(x_train) * mtry_frac))

      preds <- numeric(n_y)


      for (j in seq_len(n_y)) {
        rf_fit <- ranger::ranger(
```

```
        dependent.variable.name = "y",
        data       = data.frame(y = y_train[, j], x_train),
        num.trees  = num.trees,
        mtry       = mtry_val,
        write.forest = TRUE,
        respect.unordered.factors = "order"
       )

       preds[j] <- predict(rf_fit, data = as.data.frame(x_test))$predictions
      }


     se_h <- rbind(se_h, (Y[test_idx, ] - preds)^2)
    }

    se_all[[h]] <- se_h
  }

  se_cat <- do.call(rbind, se_all)
  sqrt(mean(se_cat, na.rm = TRUE))
}
```

3B. Helper: search over p with cheap RF (100)

```{r, cache=TRUE}
rf_search_p <- function(df, predictors, targets,
                        p_grid,
                        K,
                        max_h,
                        num.trees_search = 100) {

  res <- data.frame(p = integer(), RMSE = numeric())

  for (p in p_grid) {
    rmse_p <- rf_ts_cv_rmse_fast(
      df         = df,
      predictors = predictors,
      targets    = targets,
      p          = p,
      K          = K,
      max_h      = max_h,
      num.trees  = num.trees_search
    )

    res <- rbind(res, data.frame(p = p, RMSE = rmse_p))
    cat("RF search: p =", p, "RMSE =", round(rmse_p, 6),
        " (max_h =", max_h, ")\n")
  }

  res
}
```

4A. Short horizon

```{r}
## it take 5 minutes
p_grid <- 1:12  # Victor your wish: check all lags

results_rf_short_search <- rf_search_p(
  df         = df_ml,
  predictors = features_large,
  targets    = targets,
  p_grid     = p_grid,
  K          = 10,
  max_h      = 3,
  num.trees_search = 100
)

results_rf_short_search

best_row_short <- results_rf_short_search[
  which.min(results_rf_short_search$RMSE), ]
best_p_rf_short <- best_row_short$p
best_row_short
```

```
rmse_rf_short_final <- rf_ts_cv_rmse_fast(
  df         = df_ml,
  predictors = features_large,
  targets    = targets,
  p          = best_p_rf_short,
  K          = 10,
  max_h      = 3,
  num.trees  = 500  # fancy lol
)

short_term_rf_table <- data.frame(
  Model      = "RF (17 variables)",
  Lag_p      = best_p_rf_short,
  RMSE_1to3  = rmse_rf_short_final
)

short_term_rf_table

```

4B. Long Dimension
```{r}
## it take 15 minutes, no joke
results_rf_long_search <- rf_search_p(
  df         = df_ml,
  predictors = features_large,
  targets    = targets,
  p_grid     = p_grid,
  K          = 10,
  max_h      = 12,
  num.trees_search = 100
)

results_rf_long_search

best_row_long <- results_rf_long_search[
  which.min(results_rf_long_search$RMSE), ]
best_p_rf_long <- best_row_long$p
best_row_long

rmse_rf_long_final <- rf_ts_cv_rmse_fast(
  df         = df_ml,
  predictors = features_large,
  targets    = targets,
  p          = best_p_rf_long,
  K          = 10,
  max_h      = 12,
  num.trees  = 500
)

long_term_rf_table <- data.frame(
  Model      = "RF (17 variables)",
  Lag_p      = best_p_rf_long,
  RMSE_1to12  = rmse_rf_long_final
)

long_term_rf_table
```

B3. RF Application Notebook

```
---
title: 'Random Forest – Application'
output:
  pdf_document: default
---


## 0.Packages
```{r}
# Start a new file, so dont need to do the cross validation (15 minutes) once again lol
R_packages <- c(
  "readr", "dplyr", "lubridate",
  "ranger", "tidyr", "tibble", "knitr"
```
```

```
)

options(repos = c(CRAN = "http://cran.rstudio.com"))

if (!requireNamespace("librarian", quietly = TRUE)) {
  install.packages("librarian")
}

librarian::shelf(R_packages)

set.seed(12345)
```

1. Data
```{r}
df_ml <- read_csv("balanced_can_md.csv", show_col_types = FALSE) %>%
  mutate(
    Date      = as.Date(Date),
    GDP_per_w = GDP_new - EMP_CAN
  ) %>%
  arrange(Date)

targets <- c("GDP_per_w", "CPI_ALL_CAN", "NHOUSE_P_CAN")

features_large <- c(
  "GDP_per_w", "CPI_ALL_CAN", "NHOUSE_P_CAN", "IP_new",
  "BSI_new", "UNEMP_CAN", "EMP_CONS_CAN", "hstart_CAN_new",
  "CRED_HOUS", "CRED_HOUS_MORT", "BANK_RATE_L", "TBILL_3M",
  "Exp_BP_new", "Imp_BP_new", "IPPI_METAL_CAN", "OILP_new",
  "WTISPLC"
)

setdiff(features_large, names(df_ml))

```

2. HELPER
```{r}
build_lagged_panel <- function(df, predictors, targets, p, h) {
  dat <- df %>%
    dplyr::select(Date, dplyr::all_of(unique(c(predictors, targets)))) %>%
    arrange(Date)

  for (lag in 1:p) {
    for (v in predictors) {
      new_name <- paste0("L", lag, "_", v)
      dat[[new_name]] <- dplyr::lag(dat[[v]], lag)
    }
  }

  for (v in targets) {
    lead_name <- paste0("F", h, "_", v)
    dat[[lead_name]] <- dplyr::lead(dat[[v]], h)
  }

  lag_cols <- grep("^L[0-9]+_", names(dat), value = TRUE)
  y_cols   <- paste0("F", h, "_", targets)

  dat_final <- dat %>%
    dplyr::select(dplyr::all_of(c(lag_cols, y_cols))) %>%
    tidyr::drop_na()

  list(
    X = as.matrix(dat_final[, lag_cols, drop = FALSE]),
    Y = as.matrix(dat_final[, y_cols,  drop = FALSE])
  )
}
```


```{r}
## from that RF_final rmd file
p_rf_short <- 12
p_rf_long  <- 5
```

```
num_trees_final <- 500
```

```{r}
rf_direct_h <- function(df, predictors, target, p, h,
                        num.trees = num_trees_final) {

  panel <- build_lagged_panel(df, predictors, target, p, h)
  X <- panel$X
  y <- panel$Y[, 1]

  mtry_val <- max(1L, floor(ncol(X) / 3))

  train_df <- data.frame(y = y, X)

  rf_fit <- ranger::ranger(
    dependent.variable.name   = "y",
    data                      = train_df,
    num.trees                 = num.trees,
    mtry                      = mtry_val,
    write.forest              = TRUE,
    respect.unordered.factors = "order"
  )

  X_last <- tail(X, 1)
  pred   <- predict(rf_fit, data = as.data.frame(X_last))$predictions

  as.numeric(pred)
}

targets_to_run <- c("GDP_per_w", "CPI_ALL_CAN", "NHOUSE_P_CAN")

all_short_rf <- list()
all_long_rf  <- list()

for (tt in targets_to_run) {
  fc_s <- sapply(1:3, function(h) {
    rf_direct_h(df_ml, features_large, tt, p_rf_short, h)
  })
  all_short_rf[[tt]] <- fc_s

  fc_l <- sapply(1:12, function(h) {
    rf_direct_h(df_ml, features_large, tt, p_rf_long, h)
  })
  all_long_rf[[tt]] <- fc_l
}

last_date <- max(df_ml$Date)

rf_short_all <- tibble(
  variable = rep(targets_to_run, each = 3),
  h        = rep(1:3, times = length(targets_to_run)),
  date     = rep(seq(last_date %m+% months(1),
                     by   = "1 month",
                     length.out = 3),
                 times = length(targets_to_run)),
  forecast = unlist(all_short_rf)
)

rf_long_all <- tibble(
  variable = rep(targets_to_run, each = 12),
  h        = rep(1:12, times = length(targets_to_run)),
  date     = rep(seq(last_date %m+% months(1),
                     by   = "1 month",
                     length.out = 12),
                 times = length(targets_to_run)),
  forecast = unlist(all_long_rf)
)

knitr::kable(
  rf_short_all,
  digits  = 4,
  caption = "Direct RF forecasts (ALL VARIABLES, h = 1–3)"
```

```
)

knitr::kable(
  rf_long_all,
  digits  = 4,
  caption = "Direct RF forecasts (ALL VARIABLES, h = 1–12)"
)

# export CSVs for later comparison
write.csv(rf_short_all, "RF_all_variables_short.csv", row.names = FALSE)
write.csv(rf_long_all,  "RF_all_variables_long.csv",  row.names = FALSE)
```

B4. BART CV Kaggle Notebook

```
---
title: 'Project: KAGGLE BART CODE'
output:
  pdf_document: default
  html_notebook: default
---
In this file, it captured the code i used on KAGGLE that takes 2 hours to run. In R studio, my computer overheat and will crash.
# 0.Packages
```{r}

library(tidyverse)

list.files(path = "../input")
# Burda style
R_packages <- c("dbarts", "tidyverse", "lubridate", "knitr")

options(repos = c(CRAN = "https://cloud.r-project.org"))

if (!requireNamespace("librarian", quietly = TRUE))
  install.packages("librarian")

librarian::shelf(R_packages)
## KAGGLE code
df_ml <- read.csv("/kaggle/input/dataset/balanced_can_md.csv")
df_ml$GDP_per_w <- df_ml$GDP_new - df_ml$EMP_CAN
df_ml %>%
  select(Date, GDP_new, EMP_CAN, GDP_per_w) %>%
  head(10)

targets <- c("GDP_per_w", "CPI_ALL_CAN", "NHOUSE_P_CAN")

features_large <- c(
  "GDP_per_w", "CPI_ALL_CAN", "NHOUSE_P_CAN", "IP_new",
  "BSI_new", "UNEMP_CAN", "EMP_CONS_CAN", "hstart_CAN_new",
  "CRED_HOUS", "CRED_HOUS_MORT", "BANK_RATE_L", "TBILL_3M",
  "Exp_BP_new", "Imp_BP_new", "IPPI_METAL_CAN", "OILP_new",
  "WTISPLC"
)
```

1. Helper 1: lag panel
```{r}
build_lagged_panel <- function(df_ml, predictors, targets, p, h) {
  dat <- df_ml %>%
    dplyr::select(Date, dplyr::all_of(unique(c(predictors, targets)))) %>%
    arrange(Date)

  for (lag in 1:p) {
    for (v in predictors) {
      new_name <- paste0("L", lag, "_", v)
      dat[[new_name]] <- dplyr::lag(dat[[v]], lag)
    }
  }

  for (v in targets) {
    lead_name <- paste0("F", h, "_", v)
    dat[[lead_name]] <- dplyr::lead(dat[[v]], h)
  }

  lag_cols <- grep("^L[0-9]+_", names(dat), value = TRUE)
```
```

```
 y_cols   <- paste0("F", h, "_", targets)

 dat_final <- dat %>%
  dplyr::select(dplyr::all_of(c(lag_cols, y_cols))) %>%
  tidyr::drop_na()

 list(
  X = as.matrix(dat_final[, lag_cols, drop = FALSE]),
  Y = as.matrix(dat_final[, y_cols,  drop = FALSE])   # columns match targets
 )
}
```

2. BART_time_series_CV

```{r}
bart_ts_cv_rmse <- function(df_ml, predictors, targets,
                            p,
                            K       = 10,
                            max_h,
                            ntree    = 150,
                            ndpost   = 200,
                            nskip    = 200) {

 se_all <- list()

 for (h in 1:max_h) {
  panel <- build_lagged_panel(df_ml, predictors, targets, p, h)
  X <- panel$X
  Y <- panel$Y

  n   <- nrow(X)
  n_y <- ncol(Y)

  if (n < K + 5) stop("Too few obs after lags/leads for this p / h.")

  start_idx <- floor(0.6 * n)
  fold_ends <- floor(seq(from = start_idx, to = n - 1, length.out = K))

  se_h <- matrix(NA_real_, nrow = 0, ncol = n_y)

  for (end_idx in fold_ends) {
   train_idx <- 1:end_idx
   test_idx  <- end_idx + 1
   if (test_idx > n) break

   x_train <- X[train_idx, , drop = FALSE]
   y_train <- Y[train_idx, , drop = FALSE]
   x_test  <- X[test_idx, , drop = FALSE]

   preds <- numeric(n_y)


   for (j in seq_len(n_y)) {
    fit_bart <- dbarts::bart(
     x.train  = x_train,
     y.train  = y_train[, j],
     x.test   = x_test,
     ntree    = ntree,
     ndpost   = ndpost,
     nskip    = nskip,
     keeptrees = FALSE,
     verbose   = FALSE
    )

    preds[j] <- fit_bart$yhat.test.mean
   }

   se_h <- rbind(se_h, (Y[test_idx, ] - preds)^2)
  }

  se_all[[h]] <- se_h
 }

 se_cat <- do.call(rbind, se_all)
 sqrt(mean(se_cat, na.rm = TRUE))
```

```
}
```

3. SHORT-HORIZON BART, h=1-3. (Output release end of this block)
```{r}
## Please run carefully this block as it would definitely crash!!!
p_grid <- 1:12

results_bart_short <- data.frame(p = integer(), RMSE_1to3 = numeric())

for (p in p_grid) {
  rmse_p <- bart_ts_cv_rmse(
    df_ml     = df_ml,
    predictors = features_large,
    targets    = targets,
    p          = p,
    K          = 10,
    max_h      = 3
  )

  results_bart_short <- rbind(
    results_bart_short,
    data.frame(p = p, RMSE_1to3 = rmse_p)
  )

  cat("BART short: p =", p, "RMSE_1to3 =", round(rmse_p, 6), "\n")
}

results_bart_short

best_row_bart_short  <- results_bart_short[which.min(results_bart_short$RMSE_1to3), ]
best_p_bart_short    <- best_row_bart_short$p
rmse_bart_short_best <- best_row_bart_short$RMSE_1to3

short_term_bart_table <- data.frame(
  Model    = "BART ",
  Lag_p    = best_p_bart_short,
  RMSE_1to3 = rmse_bart_short_best
)

short_term_bart_table
```


4. Long horizon BART , h =1-12
```{r}
results_bart_long <- data.frame(p = integer(), RMSE_1to12 = numeric())

for (p in p_grid) {
  rmse_p <- bart_ts_cv_rmse(
    df_ml     = df_ml,
    predictors = features_large,
    targets    = targets,
    p          = p,
    K          = 10,
    max_h      = 12
  )

  results_bart_long <- rbind(
    results_bart_long,
    data.frame(p = p, RMSE_1to12 = rmse_p)
  )

  cat("BART long: p =", p, "RMSE_1to12 =", round(rmse_p, 6), "\n")
}

results_bart_long

best_row_bart_long  <- results_bart_long[which.min(results_bart_long$RMSE_1to12), ]
best_p_bart_long    <- best_row_bart_long$p
rmse_bart_long_best <- best_row_bart_long$RMSE_1to12

long_term_bart_table <- data.frame(
  Model     = "BART",
```

```
  Lag_p     = best_p_bart_long,
  RMSE_1to12 = rmse_bart_long_best
)

long_term_bart_table
```

5. Result to export
```{r}
# -> data back to local csv and combine everything
write.csv(short_term_bart_table, "/kaggle/working/bart_short.csv", row.names = FALSE)
write.csv(long_term_bart_table, "/kaggle/working/bart_long.csv", row.names = FALSE)

knitr::kable(short_term_bart_table)
knitr::kable(long_term_bart_table)
```

```{r}
#All I do in this shot file is to jsut get the opt_p, nth else are saved actually,
```

##

B5. BART Application Notebook

```
---
title: "BART Application"
output:
  pdf_document: default
  html_notebook: default
  word_document: default
---
This file: we already had optimal lag solved in Kaggle, based on that opt_p, and all other parameter assumed, here target to fit in specific models and forecast for our varaibles locally

```{r, message=FALSE, warning=FALSE, echo=TRUE, results='hide'}
R_packages <- c("dbarts", "tidyverse", "lubridate", "knitr", "tibble")

options(repos = c(CRAN = "https://cloud.r-project.org"))

if (!requireNamespace("librarian", quietly = TRUE))
  install.packages("librarian")

librarian::shelf(R_packages)
set.seed(12345)
```
1. DATA as usual
```{r}
df_ml <- read.csv("balanced_can_md.csv")
df_ml$GDP_per_w <- df_ml$GDP_new - df_ml$EMP_CAN
df_ml %>%
  select(Date, GDP_new, EMP_CAN, GDP_per_w) %>%
  head(10)

targets <- c("GDP_per_w", "CPI_ALL_CAN", "NHOUSE_P_CAN")

features_large <- c(
  "GDP_per_w", "CPI_ALL_CAN", "NHOUSE_P_CAN", "IP_new",
  "BSI_new", "UNEMP_CAN", "EMP_CONS_CAN", "hstart_CAN_new",
  "CRED_HOUS", "CRED_HOUS_MORT", "BANK_RATE_L", "TBILL_3M",
  "Exp_BP_new", "Imp_BP_new", "IPPI_METAL_CAN", "OILP_new",
  "WTISPLC"
)
```

2. Helper
```{r}
build_lagged_panel <- function(df, predictors, target, p, h) {
  dat <- df %>%
    dplyr::select(Date, all_of(unique(c(predictors, target)))) %>%
    arrange(Date)

  for (lag in 1:p) {
```
```

```
   for (v in predictors) {
     dat[[paste0("L", lag, "_", v)]] <- dplyr::lag(dat[[v]], lag)
   }
  }

  dat[[paste0("F", h, "_", target)]] <- dplyr::lead(dat[[target]], h)

  dat_final <- dat %>% drop_na()

  X <- as.matrix(dat_final %>% select(starts_with("L")))
  Y <- as.matrix(dat_final %>% select(starts_with("F")))

  list(X = X, Y = Y)
}

target     <- "CPI_ALL_CAN"
predictors <- features_large
p_short <- 11
p_long  <- 3
```

```{r}
ntrees_direct  <- 500
nskips_direct  <- 1000
ndposts_direct <- 2000

bart_direct_h <- function(df, predictors, target, p, h) {

  panel <- build_lagged_panel(df, predictors, target, p, h)
  X <- panel$X
  y <- panel$Y[, 1]

  fit <- dbarts::bart(
    x.train   = X,
    y.train   = y,
    ntree     = ntrees_direct,
    nskip     = nskips_direct,
    ndpost    = ndposts_direct,
    keeptrees = TRUE,
    verbose   = FALSE
  )

  X_last <- tail(X, 1)
  pred   <- predict(fit, X_last)

  as.numeric(mean(pred))
}
```

```{r}
horizons_short <- 1:3

bart_short_fc_direct <- sapply(horizons_short, function(h) {
  cat("Direct BART short, h =", h, "\n")
  bart_direct_h(df_ml, predictors, target, p_short, h)
})

horizons_long <- 1:12

bart_long_fc_direct <- sapply(horizons_long, function(h) {
  cat("Direct BART long, h =", h, "\n")
  bart_direct_h(df_ml, predictors, target, p_long, h)
})


last_date <- as.Date(tail(df_ml$Date, 1))

bart_short_tbl <- tibble(
  h        = horizons_short,
  date     = seq(last_date %m+% months(1), by = "1 month", length.out = 3),
  forecast = bart_short_fc_direct
)
```

```
bart_long_tbl <- tibble(
  h        = horizons_long,
  date     = seq(last_date %m+% months(1), by = "1 month", length.out = 12),
  forecast = bart_long_fc_direct
)

knitr::kable(bart_short_tbl, digits = 4,
             caption = "Direct BART CPI forecasts (h = 1–3)")
knitr::kable(bart_long_tbl, digits = 4,
             caption = "Direct BART CPI forecasts (h = 1–12)")
```

```{r}
targets_to_run <- c("GDP_per_w", "CPI_ALL_CAN", "NHOUSE_P_CAN")

all_short <- list()
all_long  <- list()

for (tt in targets_to_run) {
  fc_s <- sapply(1:3, function(h) {
    bart_direct_h(df_ml, predictors, tt, p_short, h)
  })
  all_short[[tt]] <- fc_s

  fc_l <- sapply(1:12, function(h) {
    bart_direct_h(df_ml, predictors, tt, p_long, h)
  })
  all_long[[tt]] <- fc_l
}

last_date <- as.Date(tail(df_ml$Date, 1))

bart_short_all <- tibble(
  variable = rep(targets_to_run, each = 3),
  h        = rep(1:3, times = length(targets_to_run)),
  date     = rep(seq(last_date %m+% months(1), by = "1 month", length.out = 3),
                 times = length(targets_to_run)),
  forecast = unlist(all_short)
)

bart_long_all <- tibble(
  variable = rep(targets_to_run, each = 12),
  h        = rep(1:12, times = length(targets_to_run)),
  date     = rep(seq(last_date %m+% months(1), by = "1 month", length.out = 12),
                 times = length(targets_to_run)),
  forecast = unlist(all_long)
)

knitr::kable(bart_short_all, digits = 4,
             caption = "Direct BART Forecasts (ALL VARIABLES, h = 1–3)")
knitr::kable(bart_long_all, digits = 4,
             caption = "Direct BART Forecasts (ALL VARIABLES, h = 1–12)")

write.csv(bart_short_all, "BART_all_variables_short.csv", row.names = FALSE)
write.csv(bart_long_all,  "BART_all_variables_long.csv",  row.names = FALSE)
```


ignore right now, old code not delete yet
```{r}
## hard code gen nice table lol
comparison <- data.frame(
  Model      = c("Small VAR (3 vars)",
                 "Large VAR (17 vars)",
                 "Random Forest",
                 "BART"),
  RMSE_1to3  = c(0.003892, 0.0030008, 0.002372, 0.00266),
  RMSE_1to12 = c(0.003624, 0.0034235, 0.002554, 0.002717)
)

knitr::kable(comparison, digits=6,
             caption = "Model Out-of-Sample-CV RMSE: VAR vs RF vs BART")
```

```r
write.csv(
  bart_short_tbl,
  "BART_forecasts_short.csv",
  row.names = FALSE
)

write.csv(
  bart_long_tbl,
  "BART_forecasts_long.csv",
  row.names = FALSE
)

write.csv(
  comparison,
  "Model_Comparison_RMSE.csv",
  row.names = FALSE
)
```

B6. Forecasting and Graph Notebook

```
---
title: "Forecast Comparison VAR vs RF vs BART"
output:
  pdf_document:
    number_sections: false
  html_notebook: default
  word_document: default
df_print: paged
---
```

```r
R_packages <- c("readr", "dplyr", "tidyr", "tibble",
                "ggplot2", "lubridate", "knitr", "zoo")

options(repos = c(CRAN = "https://cloud.r-project.org"))

if (!requireNamespace("librarian", quietly = TRUE)) {
  install.packages("librarian")
}

librarian::shelf(R_packages)
if (!requireNamespace("tinytex", quietly = TRUE)) install.packages("tinytex")
if (Sys.which("pdflatex")=="") {library(tinytex); tinytex::install_tinytex()}
theme_set(theme_bw())
knitr::opts_knit$set(root.dir = getwd())
options(knitr.kable.NA = "", knitr.kable.auto_format = FALSE)
knitr::opts_knit$set(kable.auto_number = FALSE)
```

```r
### Read csv for previous notebook: VAR_final_with_aplication, RF_application, etc
var_short <- read_csv("VAR_large_all_variables_short.csv", show_col_types = FALSE) |>
  mutate(model = "VAR", horizon_group = "short")

var_long  <- read_csv("VAR_large_all_variables_long.csv",  show_col_types = FALSE) |>
  mutate(model = "VAR", horizon_group = "long")


rf_short <- read_csv("RF_all_variables_short.csv", show_col_types = FALSE) |>
  mutate(model = "Random Forest", horizon_group = "short")

rf_long  <- read_csv("RF_all_variables_long.csv",  show_col_types = FALSE) |>
  mutate(model = "Random Forest", horizon_group = "long")


bart_short <- read_csv("BART_all_variables_short.csv", show_col_types = FALSE) |>
  mutate(model = "BART", horizon_group = "short")

bart_long  <- read_csv("BART_all_variables_long.csv",  show_col_types = FALSE) |>
  mutate(model = "BART", horizon_group = "long")
```

```{r, message = FALSE, warning = FALSE}
get_fc <- function(df, var_name, horizons) {
  df %>%
    filter(variable == var_name, h %in% horizons) %>%   # use h column
    arrange(h) %>%
    pull(forecast)
}

h_short <- 1:3
h_long  <- 1:12

tbl_GDP_short <- tibble(
  h             = h_short,
  `VAR `= get_fc(var_short,  "GDP_per_w",   h_short),
  `Random Forest`= get_fc(rf_short,   "GDP_per_w",   h_short),
  BART          = get_fc(bart_short, "GDP_per_w",   h_short)
)

tbl_GDP_long <- tibble(
  h             = h_long,
  `VAR`= get_fc(var_long,   "GDP_per_w",   h_long),
  `Random Forest`= get_fc(rf_long,    "GDP_per_w",   h_long),
  BART          = get_fc(bart_long,  "GDP_per_w",   h_long)
)

tbl_CPI_short <- tibble(
  h             = h_short,
  `VAR `= get_fc(var_short,  "CPI_ALL_CAN",   h_short),
  `Random Forest`= get_fc(rf_short,   "CPI_ALL_CAN",   h_short),
  BART          = get_fc(bart_short, "CPI_ALL_CAN",   h_short)
)

tbl_CPI_long <- tibble(
  h             = h_long,
  `VAR`= get_fc(var_long,   "CPI_ALL_CAN",   h_long),
  `Random Forest`= get_fc(rf_long,    "CPI_ALL_CAN",   h_long),
  BART          = get_fc(bart_long,  "CPI_ALL_CAN",   h_long)
)

tbl_HOUSE_short <- tibble(
  h             = h_short,
  `VAR`= get_fc(var_short,  "NHOUSE_P_CAN",   h_short),
  `Random Forest`= get_fc(rf_short,   "NHOUSE_P_CAN",   h_short),
  BART          = get_fc(bart_short, "NHOUSE_P_CAN",   h_short)
)

tbl_HOUSE_long <- tibble(
  h             = h_long,
  `VAR `= get_fc(var_long,   "NHOUSE_P_CAN",   h_long),
  `Random Forest`= get_fc(rf_long,    "NHOUSE_P_CAN",   h_long),
  BART          = get_fc(bart_long,  "NHOUSE_P_CAN",   h_long)
)

scale_to_pct <- function(tbl) {
  tbl %>%
    dplyr::mutate(dplyr::across(-h, ~ . * 100))
}

tbl_GDP_short  <- scale_to_pct(tbl_GDP_short)
tbl_GDP_long   <- scale_to_pct(tbl_GDP_long)
tbl_CPI_short  <- scale_to_pct(tbl_CPI_short)
tbl_CPI_long   <- scale_to_pct(tbl_CPI_long)
tbl_HOUSE_short <- scale_to_pct(tbl_HOUSE_short)
tbl_HOUSE_long  <- scale_to_pct(tbl_HOUSE_long)

knitr::kable(
  tbl_GDP_short,
  digits  = 4,
  caption = "Short-horizon forecasts for GDP per worker (h = 1–3, pct.)"
)

knitr::kable(
  tbl_GDP_long,
  digits  = 4,
  caption = "Long-horizon forecasts for GDP per worker (h = 1–12, pct.)"
```

```
)

knitr::kable(
  tbl_CPI_short,
  digits  = 4,
  caption = "Short-horizon forecasts for CPI inflation (h = 1–3, pct.)"
)

knitr::kable(
  tbl_CPI_long,
  digits  = 4,
  caption = "Long-horizon forecasts for CPI inflation (h = 1–12, pct.)"
)

knitr::kable(
  tbl_HOUSE_short,
  digits  = 4,
  caption = "Short-horizon forecasts for house prices (h = 1–3, pct.)"
)

knitr::kable(
  tbl_HOUSE_long,
  digits  = 4,
  caption = "Long-horizon forecasts for house prices (h = 1–12, pct.)"
)
```

```
#. Forecast!
```{r, echo=FALSE, message=FALSE, warning=FALSE, fig.show='hold', fig.width=7, fig.height=7}

forecasts_all <- bind_rows(
  var_short, var_long,
  rf_short,  rf_long,
  bart_short, bart_long
) %>%
  filter(variable %in% c("GDP_per_w", "CPI_ALL_CAN", "NHOUSE_P_CAN")) %>%
  mutate(date = as.Date(date))

models_vec <- sort(unique(forecasts_all$model))


data_full <- read.csv("balanced_can_md.csv")
data_full$Date <- as.Date(data_full$Date)
data_full$GDP_per_w <- data_full$GDP_new - data_full$EMP_CAN

last_date <- max(data_full$Date)

Dat_all <- zoo(
  data_full[, c("GDP_per_w", "CPI_ALL_CAN", "NHOUSE_P_CAN")],
  order.by = data_full$Date
)

start_plot_short <- last_date %m-% months(6)
end_plot_short   <- last_date %m+% months(3)

start_plot_long  <- as.Date("2024-01-01")
end_plot_long    <- last_date %m+% months(12)

Dat_short <- window(Dat_all, start = start_plot_short, end = last_date)
Dat_long  <- window(Dat_all, start = start_plot_long,  end = last_date)

make_hist_long <- function(Dat_zoo, horiz_label) {
  fortify.zoo(Dat_zoo) %>%
    rename(date = Index) %>%
    pivot_longer(
      cols      = c(GDP_per_w, CPI_ALL_CAN, NHOUSE_P_CAN),
      names_to  = "variable",
      values_to = "value"
    ) %>%
    mutate(horizon_group = horiz_label)
}

hist_short <- make_hist_long(Dat_short, "short")
hist_long  <- make_hist_long(Dat_long,  "long")
```
```

```
hist_all <- bind_rows(
  tidyr::crossing(model = models_vec, hist_short),
  tidyr::crossing(model = models_vec, hist_long)
) %>%
  mutate(type = "Data")

fc_all <- forecasts_all %>%
  select(model, horizon_group, variable, date, forecast) %>%
  rename(value = forecast) %>%
  mutate(type = "Forecast")

plot_all <- bind_rows(hist_all, fc_all)


p_short <- plot_all %>%
  filter(horizon_group == "short") %>%
  ggplot(aes(x = date, y = value, colour = type)) +
  geom_line() +
  facet_grid(variable ~ model, scales = "free_y") +
  labs(
    title = "Short-horizon forecasts (h = 1–3)",
    x = "Date",
    y = "Growth rate"
  ) +
  coord_cartesian(xlim = c(start_plot_short, end_plot_short)) +
  scale_colour_manual(values = c("Data" = "black", "Forecast" = "red")) +
  theme_bw() +
  theme(legend.position = "bottom")

print(p_short)


p_long <- plot_all %>%
  filter(horizon_group == "long") %>%
  ggplot(aes(x = date, y = value, colour = type)) +
  geom_line() +
  facet_grid(variable ~ model, scales = "free_y") +
  labs(
    title = "Long-horizon forecasts (h = 1–12)",
    x = "Date",
    y = "Growth rate"
  ) +
  coord_cartesian(xlim = c(start_plot_long, end_plot_long)) +
  scale_colour_manual(values = c("Data" = "black", "Forecast" = "red")) +
  theme_bw() +
  theme(legend.position = "bottom")

print(p_long)
```



```{r}
comparison <- tribble(
  ~Model,              ~RMSE_1to3,   ~RMSE_1to12,
  "Small VAR (3 vars)",   0.003892,     0.003624,
  "Large VAR (17 vars)",  0.0030008,    0.0034235,
  "Random Forest",        0.002372,     0.002554,
  "BART",                 0.00266,      0.002717
)

lag_info <- tribble(
  ~Model,              ~p_short, ~p_long,
  "Small VAR (3 vars)",      7,        7,
  "Large VAR (17 vars)",     2,        1,
  "Random Forest",          12,        5,
  "BART",                   11,        3
)

summary_table <- comparison %>%
  left_join(lag_info, by = "Model") %>%
  select(Model, p_short, RMSE_1to3, p_long, RMSE_1to12)

summary_print <- summary_table
```

```
colnames(summary_print) <- c(
  "Model",
  "p (h = 1–3)",
  "RMSE 1–3",
  "p (h = 1–12)",
  "RMSE 1–12"
)

cat("\\setcounter{table}{1}")
kable(
  summary_print,
  digits  = 4,
  caption = "Summary of optimal lags and forecast accuracy (RMSE)"
)
```

```{r}
vars_large_17 <- tibble::tribble(
  ~Variable,       ~Block,          ~Short_Description,
  "GDP_per_w",     "Target",        "GDP per worker, monthly growth",
  "CPI_ALL_CAN",   "Target",        "Headline CPI inflation, monthly",
  "NHOUSE_P_CAN",  "Target",        "National house price index, monthly",
  "IP_new",        "Real activity", "Industrial production index, monthly",
  "BSI_new",       "Real activity", "Business sentiment / survey index",
  "UNEMP_CAN",     "Labour",        "Unemployment rate, Canada",
  "EMP_CONS_CAN",  "Labour",        "Construction employment, Canada",
  "hstart_CAN_new","Housing",       "Housing starts, Canada, monthly",
  "CRED_HOUS",     "Credit",        "Household / housing credit stock",
  "CRED_HOUS_MORT","Credit",        "Mortgage credit for housing",
  "BANK_RATE_L",   "Interest",      "BoC policy rate (level)",
  "TBILL_3M",      "Interest",      "3-month T-bill rate",
  "Exp_BP_new",    "External",      "Exports of goods (BOP), monthly",
  "Imp_BP_new",    "External",      "Imports of goods (BOP), monthly",
  "IPPI_METAL_CAN","Prices",        "Metal producer price index, Canada",
  "OILP_new",      "Commodities",   "Oil price index",
  "WTISPLC",       "Commodities",   "WTI spot crude oil price (level)"
)
```

```{r}
knitr::kable(
  vars_large_17,
  booktabs = TRUE,
  caption = " The 17 variables selected for the large VAR model"
)
write.csv(vars_large_17, "vars_17_large_model.csv", row.names = FALSE)

```

B7, BVAR Notebook (extra)

```
---
title: "BVAR Forecasting Three Variaible"
output:
  pdf_document: default
  html_document: default
---


```{r}
## We do not have time and space to merge this into our analysis. However, some solid result still worth mentioning and recorded.

options(repos = c(CRAN = "https://cloud.r-project.org"))

if (!requireNamespace("librarian", quietly = TRUE)) install.packages("librarian")
librarian::shelf(readr, dplyr, lubridate, tidyr, BVAR)


df <- readr::read_csv("balanced_can_md.csv", show_col_types = FALSE) |>
  dplyr::mutate(
    Date      = as.Date(Date),
    GDP_per_w = GDP_new - EMP_CAN
  ) |>
  dplyr::arrange(Date)
```
```

```
df_small <- df |>
  dplyr::select(
    Date,
    infl_can      = CPI_ALL_CAN,
    gdp_pc_proxy  = GDP_per_w,
    house_price   = NHOUSE_P_CAN
  ) |>
  tidyr::drop_na()

Y_bvar  <- as.matrix(df_small[, c("infl_can", "gdp_pc_proxy", "house_price")])
n_total <- nrow(Y_bvar)


h_out   <- 24L
train_n <- n_total - h_out

Y_train <- Y_bvar[1:train_n, , drop = FALSE]
Y_test  <- Y_bvar[(train_n + 1):n_total, , drop = FALSE]


p_bvar      <- 2L
priors_bvar <- BVAR::bv_priors()

set.seed(123)
bvar_model <- BVAR::bvar(
  data   = Y_train,
  lags   = p_bvar,
  n_draw = 8000L,
  n_burn = 4000L,
  priors = priors_bvar,
  verbose = FALSE
)


rmse_vec <- BVAR::rmse(bvar_model, holdout = Y_test)

bvar_rmse_by_var <- data.frame(
  Variable = colnames(Y_bvar),
  RMSE     = as.numeric(rmse_vec)
)


rmse_avg <- mean(bvar_rmse_by_var$RMSE)


readr::write_csv(bvar_rmse_by_var, "bvar_rmse_by_var.csv")

bvar_summary <- data.frame(
  Model      = "BVAR (3 variables, p=2)",
  Lag_p      = 2L,
  RMSE_1to12 = rmse_avg
)

readr::write_csv(bvar_summary, "bvar_holdout_summary.csv")

print("======== BVAR RESULTS ========")
print(bvar_rmse_by_var)
print(bvar_summary)
print("CSV files saved:")
print(" - bvar_rmse_by_var.csv")
print(" - bvar_holdout_summary.csv")


## it should lead to result saying that 3 variables models with BVAR beat VAR with 17 var. However, we do not have time and space to merge this into our analysis.

```
```
```