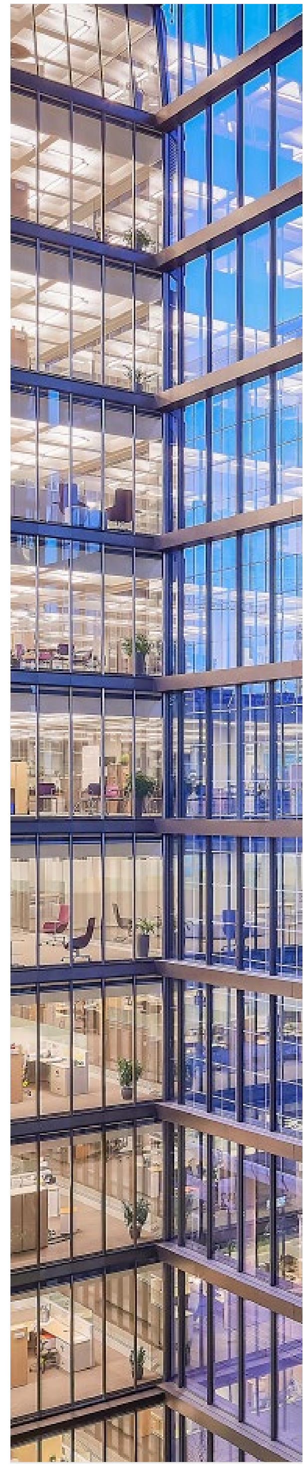
BANK OF CANADA
BANQUE DU CANADA

Staff Discussion Paper/Document d'analyse du personnel — 2023-21

Last updated: September 29, 2023



Predicting Changes in Canadian Housing Markets with Machine Learning

by Johan Brannlund, Helen Lao, Maureen MacIsaac and Jing Yang

Financial Stability Department
Bank of Canada, Ottawa, Ontario, Canada K1A 0G9

jbrannlund@bank-banque-canada.ca, HYLao@bank-banque-canada.ca, jyang@bank-banque-canada.ca

DOI: https://doi.org/10.34989/sdp-2023-21 | **ISSN** 1914-0568

Acknowledgements

We would like to thank Boyan Bejanov, Lin Chen, Thibaut Duprey, Stephanie Houle and Gias Uddin for their helpful comments and suggestions. Finally, we would also thank Ceciline Steyn and Jenna Rolland-Mills for their editorial assistance.

Abstract

This paper examines whether machine learning (ML) algorithms can outperform a linear model in predicting monthly growth in Canada of both house prices and existing home sales. The aim is to apply two widely used ML techniques (support vector regression and multilayer perceptron) in economic forecasting to understand their scopes and limitations. We find that the two ML algorithms can perform better than a linear model in forecasting house prices and resales. However, the improvement in forecast accuracy is not always statistically significant. Therefore, we cannot systematically conclude using traditional time-series data that the ML models outperform the linear model in a significant way. Future research should explore non-traditional data sets to fully take advantage of ML methods.

Topics: Econometric and statistical methods; Financial markets; Housing
JEL codes: A, C45, C53, R2, R3, D2

Résumé

Cette étude vise à établir si les algorithmes d'apprentissage automatique peuvent être plus efficaces qu'un modèle linéaire pour prévoir la croissance mensuelle des prix des logements et des ventes de logements existants au Canada. Notre objectif est d'utiliser deux techniques d'apprentissage automatique (la régression vectorielle de support et le perceptron multicouche), largement répandues dans le domaine des prévisions économiques, pour en comprendre la portée et les limites. Nous constatons que les deux techniques fonctionnent mieux qu'un modèle linéaire pour prédire les prix des logements et les reventes de logements. Cependant, l'amélioration de l'exactitude des prévisions n'est pas toujours statistiquement significative. Ainsi, nous ne pouvons pas conclure de manière systématique, en nous basant sur des séries chronologiques traditionnelles, que les modèles d'apprentissage automatique s'avèrent considérablement supérieurs au modèle linéaire. Des recherches devraient être effectuées avec des séries non traditionnelles pour tirer pleinement parti des méthodes d'apprentissage automatique.

Sujets : Méthodes économétriques et statistiques; Marchés financiers; Logement
Codes JEL : A, C45, C53, R2, R3, D2

1. Introduction

Machines are performing increasingly intelligent tasks, ranging from facial recognition to self-driving. As macroeconomic forecasters, we wonder if some of these new machine learning technologies can be used to forecast macroeconomic indicators. The appeal of machine learning techniques is that they can uncover nonlinear and generalizable patterns in the data, while conventional time-series econometric models (i.e., ARIMA, linear regression) typically use only historical linear relationships for forecasting. In this paper, we explore whether machine learning techniques can improve the accuracy of short-term forecasts of average house prices and housing resales compared with the benchmark conventional econometric model of the ordinary least squares (OLS) linear regression. This exercise allows us to understand the scope and limitations of machine learning models in the context of economic forecasting.

The rest of the paper is structured as follows. Section 2 provides an explanation of the machine learning algorithms and their application to time-series forecasting in the literature. Section 3 presents the data. Section 4 outlines our estimation and testing framework. Section 5 presents our forecast results. Section 6 reports the results of sensitivity testing that varied the parameters put into the models. Section 7 offers some concluding remarks and key takeaways.

2. Machine learning and time-series forecasting

The benchmark model we choose for this forecast comparison is the OLS linear regression (a parametric predictor). Linear regressions assume that the relationship between a continuous dependent variable and one or more explanatory variables is linear. This type of model is often used for macroeconomic forecasting because it is relatively easy to decompose the predicted dependent variable into the contributions of the explanatory variables. The two machine learning algorithms we choose for our exercise are multi-layer perceptron (MLP) and support vector regression (SVR). Both MLP and the SVR are widely used machine learning techniques; they are easy to implement and fast to compute.[1, 2]

MLP is an implementation of a feed-forward (i.e., containing no feedback loops) artificial neural network. The network consists of at least three layers: an input layer, a hidden layer and an output layer (**Figure 1**). Other than the nodes in the input layer, each node receives input from nodes in other layers. And each input has an associated weight, which is assigned based on its relative importance to other inputs. As the input propagates from layer to layer, a nonlinear *activation function* (i.e., sigmoid, logistic, etc.) is applied at each step. The network

[1] We have also tried a more-sophisticated machine learning neural network, the long short-term memory. Its forecast performance was not significantly better than those of MLP and SVR.

[2] We use the scikit-learn Python module in this paper. See Pedregosa et al. 2011.

learns by iteratively adjusting node weights to minimize a loss function that measures the difference between predictions and actual outcomes.

SVR is constructed by analogy with the support vector classification case. Effectively, we construct a hyperplane that minimizes a linear combination of the length squared of the normal vector and the distances from the data points to a tube of width ε (**Figure 2**). This corresponds to dealing with a so-called hinge loss function, which ascribes zero error to predictions that fall within the tube.

Figure 1: **A simple MLP network**

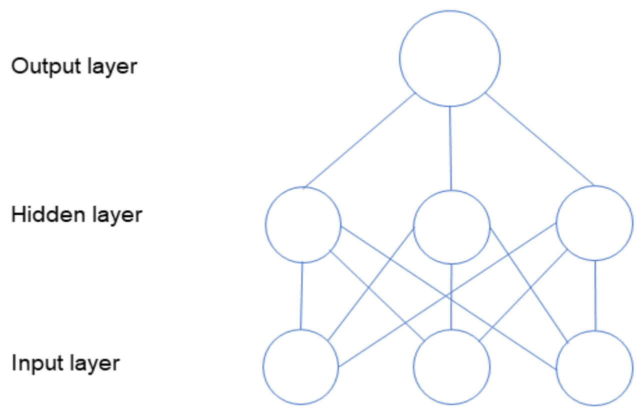
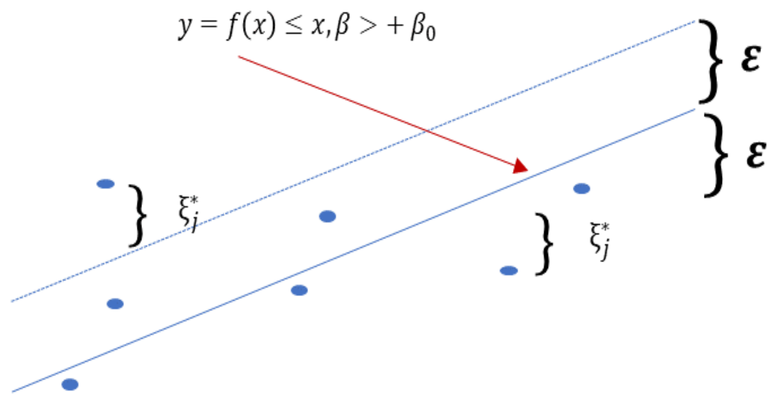


Figure 2: **Minimizing hyperplane of SVM**



Note: y = f(x) describes the data points.

Our paper contributes to a growing number of studies that have applied machine learning to macroeconomic forecasting. Couloumbe et al. (2022) find that the standard factor model is the best regularization in their comparison of time-series forecasts by various methods. However, they note that machine learning is useful for macroeconomic forecasting by mostly capturing important nonlinearities that arise in the context of uncertainty and financial frictions. Other studies have specifically applied machine learning techniques to predict house prices and compare forecast performance across models. Milunovich (2020) employs 47 different algorithms to forecast Australian log real house prices and growth rates. These algorithms consist of traditional time-series models, machine learning procedures and deep learning neural networks. Although the ranking of performance depends on the length of the forecast horizon as well as on the choice of dependent variable (log price or growth rate), Milunovich finds that six of the eight top forecasts are generated by a linear SVR.

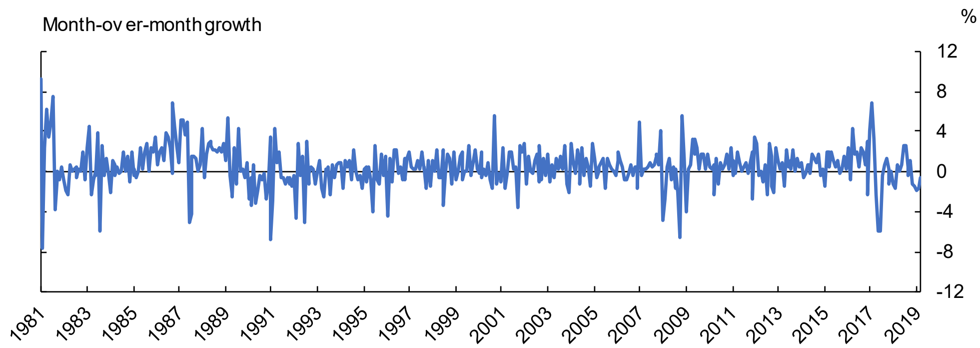
3. Data

3.1 Dependent and independent variables

Our two dependent variables are the monthly growth rate of the Multiple Listing Service average home price and monthly growth of existing home sales. The data are from the Canadian Real Estate Association (CREA) and are seasonally adjusted using an X-12 approach

(**Chart 1** and **Chart 2**).[3] The independent variables used are from two sources: housing indicators are from CREA, and macroeconomic indicators are from the Large Canadian Database for Macroeconomic Analysis (LCDMA) (Fortin-Gagnon et al. 2019), which includes 282 indicators (national and regional) related to the Canadian macroeconomy. We choose one more housing indicator in addition to the two target variables (price and resale growth)—the monthly difference in sales-to-new-listings ratio. This ratio is the number of existing home sales divided by the number of new listings entering the market and is generally considered to be a measure of tightness in the housing market.[4] All explanatory variables are available from 1981 to early 2019. As detailed in Fortin-Gagnon et al. (2019), the LCDMA variables are balanced and stationary.[5]

Chart 1: Monthly growth of house prices had been quite volatile



Note: The authors use X-12 to seasonally adjust the data.
Source: Canadian Real Estate Association

Last observation: March 2019

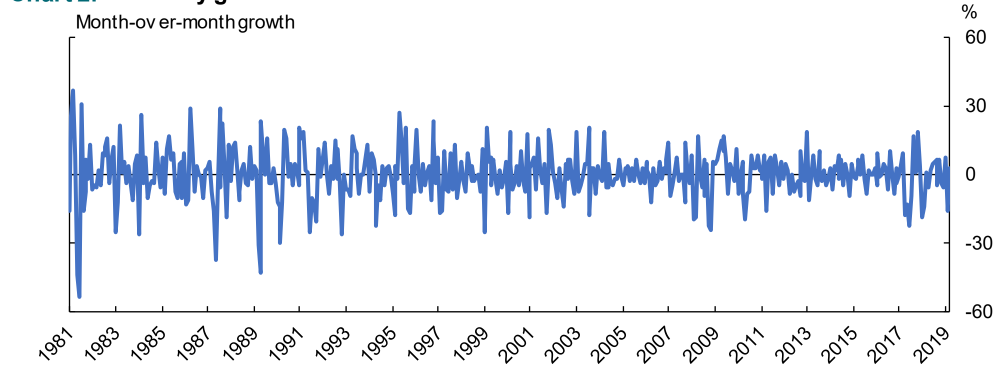
---

[3] Although CREA publishes the seasonally adjusted average home price, the series is available only from 1988, whereas the non-seasonally adjusted data go back to 1980. We find that the monthly growth from seasonally adjusting the non-seasonally adjusted data using X-12 does not differ significantly from using the seasonally adjusted average home price series from CREA, on average, over history.

[4] According to the Canada Mortgage and Housing Corporation (CMHC), new listings are a gauge of the supply of existing homes coming onto the market, while sales are a proxy for demand. A sales-to-new-listings ratio above 55% is associated with conditions where inflation-adjusted home prices are generally rising. A sales-to-new-listings ratio below 40% has historically accompanied inflation-adjusted prices that are falling, a situation known as a buyer's market. When the sales-to-new-listings ratio is between 40% and 55%, the market is said to be balanced. For more information, see CMHC, "Methodologies for Housing Market Assessment."

[5] Missing values (for example, some variables are not available before 1981) are imputed using the expectation-maximization algorithm (see Fortin-Gagnon et al. 2019).

Chart 2: Monthly growth of resales had been even more volatile



Note: The authors use X-12 to seasonally adjust the data.
Source: Canadian Real Estate Association

Last observation: March 2019

3.2 Data processing and explanatory variable selection

The **pre-processing** for variable selection has three stages: standardization, lagging the variables, and variable selection using the elastic net method. First, the 285 variables are **standardized** so that the magnitude of the variables does not bias the coefficients in the variable selection stage. Second, the dataset is **expanded to include lags** for each of the time series in it. The number of lags is set to six to cover indicators in the past two quarters. However, including just one additional lag of the data will result in a dataset where the number of predictors (*p*—in this case 570) exceeds the total number of observations (*n*—450 monthly observations). This creates a complication in the next stage: variable selection.

Given the large number of possible explanatory indicators to use in the models (285 from CREA and the LCDMA), we employ a **variable selection method** before inputting variables in our linear model or machine learning algorithms. Least absolute shrinkage selection operator (LASSO) or ridge regression are popular choices for variable selection with high-dimension data. Both methods employ a penalization technique to improve the OLS minimization of the residual sum of squares through either penalizing the sum of squared coefficients in the case of ridge regression (Hoerl and Kennard 1970) or minimizing the sum of the absolute values of the coefficients (Tibshirani 1996). However, both of these methods have disadvantages given the nature of our data. Ridge regression simply shrinks coefficients on variables and thus does not produce a sparse set of coefficients. The large number of predictors (*p*) relative to the number of observations (*n*) in our dataset suggests that LASSO would have the limitation of choosing at most *n* variables before it saturates due to the nature of the optimization. The LASSO method has an additional limitation when pairwise correlation is high (as is likely the case in our grouping of macroeconomic indicators) because LASSO is likely to select relatively arbitrarily just one variable from the group. These considerations point us toward the elastic net as an alternative method of sparse variable selection (Zou and Hastie 2005). The elastic net combines the regularization methods of both ridge regression and LASSO and allows us to retain some variables with correlation while also delivering a sparse set of covariates for model training. The elastic net regression can be expressed as the minimization over β of the following function:

$$L(\lambda_1, \lambda_2, \boldsymbol{\beta}) = \frac{1}{2n}|\boldsymbol{y} - \boldsymbol{X\beta}|^2 + \lambda_1|\boldsymbol{\beta}|_1 + \lambda_2|\boldsymbol{\beta}|^2$$

where n is the number of samples, $|\boldsymbol{\beta}|_1 = \sum_{j=1}^{p}|\beta_j|$ is the LASSO penalty term, and $\lambda_1 = \alpha * \lambda$ and $\lambda_2 = (1-\alpha) * \frac{1}{2} * \lambda$).

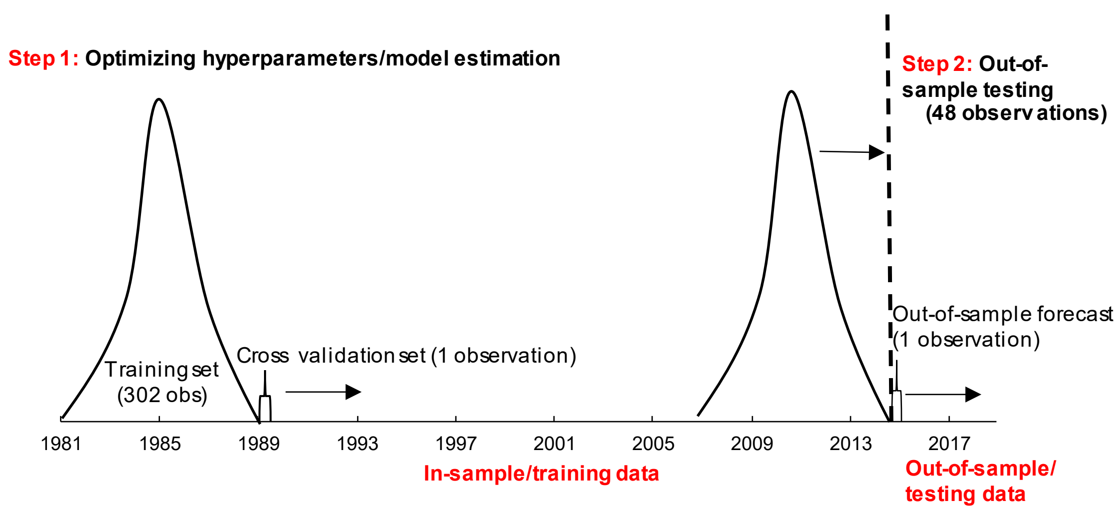
The alpha (α) parameter multiplies the penalty terms in the optimization problem, and a lower value of alpha corresponds to a higher number of covariates with non-zero coefficients in the elastic net estimation. α = 0 corresponds to ridge regression and α = 1 to LASSO regression. Therefore, we can choose an α between 0 and 1 to optimize the elastic net. The penalty parameter presented in the baseline estimates is 0.1, but alternative values of alpha will result in a different number and composition of variables. We therefore present estimation results for different values of alpha in the sensitivity analysis.

4. Training and testing framework

4.1 Model estimation and validation

Having a training and testing framework allows us to estimate the forecast errors of each model in the same manner and compare their forecast performance on equal footing. The framework is summarized in **Figure 3**. It consists of optimizing the hyperparameters and estimating the model as well as out-of-sample testing.

Figure 3: Illustration of the training and testing framework



The first step of the framework is to fit the model on the training data. To establish the training data, we must first establish our testing data, which are the proportion of the dataset used to evaluate the out-of-sample forecast of the models (**Figure 3**). Between 2015 to 2018, the Canadian housing market was impacted by national, provincial and regional regulatory changes (Khan and Webley 2019). To properly evaluate out-of-sample performance between the algorithms and our OLS estimates, we retain the last four years (48 months) of our dataset,[6] which cover this window. This leaves the remaining data as the training data. The size of the training sample is 402 observations.

To train the model, we first separate the training data into a training set, which is 75% of the training data or 302 observations. This data set is used to fit the model and establish its hyperparameters. The model produces a one-month-ahead prediction of the target variable (monthly growth of house prices or resales) based on the training data. This is similar to performing a pseudo out-of-sample forecast. In machine learning, this is referred to as cross-validation, which is a common procedure used to evaluate the prediction skills of machine learning models on a limited data sample. This one-month-ahead predicted value is then compared with the actual value of the target variable, and we estimate the root mean squared error (RMSE). We then roll over the training set (302 observations) one month ahead and produce another forecast and estimate the RMSE until we reach the last observation in the training sample. As the training process progresses through to the last observation of the training sample, the model hyperparameters are selected based on the combination that produces the lowest in-sample relative RMSE ratio for each model class (using only the training data). The relative RMSE ratio is the RMSE from the machine learning models divided by the RMSE from the linear model. Therefore, a relative RMSE ratio smaller than 1 implies that the RMSE produced by the machine learning model is smaller than that from the linear model, signifying a decrease in forecast error. After the hyperparameters of the models are optimized when the training reaches the last observation of the training sample, the hyperparameters from each model class remain constant, and we no longer allow the hyperparameters to change as the process moves into the data testing.

4.2 Out-of-sample testing

After the hyperparameters of the models are finalized based on the training data, we test the models with the fixed hyperparameters on the testing sample. In this out-of-sample testing phase, we allow the models to change only based on model weights. Similar to the in-sample training and testing phase, we apply the models to estimate a one-month-ahead forecast of the target variable. We also apply the same one-month rolling window in the testing data, where we roll over the data and re-estimate the models based on model weights one month ahead with each step to produce a one-month-ahead forecast. Therefore, in the out-of-sample testing phase, there are 48 out-of-sample forecasts in which we have calculated the out-of-sample RMSE.

[6] We use final data for our estimation and evaluation (as of April 2019) because real-time data are not available for our indicators.

5. Results

In this section, we present the one-step-ahead forecast results for national monthly average house price growth and national monthly resales growth. The results are presented in relative RMSE ratios, which, as noted above, are the RMSE of the machine learning models relative to that of the benchmark linear model. The results are based on a penalty parameter of 0.1, which includes 149 variables picked up by the elastic net for forecasting growth in national house prices and resales growth. The baseline specification also includes up to six lags of the variables.

Table 1 presents the results for monthly national house price growth. The in-sample relative RMSE ratio for MLP is 0.77 and 0.77 for SVR, which implies that the in-sample RMSEs are smaller for the machine learning models than for the linear benchmark model. This result also holds in the out-of-sample case. The out-of-sample relative RMSE ratios are 0.906 for the MLP and 0.892 for the SVR. However, the Kolmogorov-Smirnov (K-S) test[7] shows that we cannot reject the null hypothesis that the distributions of the forecast results from the linear model and the machine learning models are the same. Therefore, although the two machine learning algorithms can perform marginally better than the linear model in forecasting one-step-ahead monthly house price growth, we cannot conclude that the machine learning models perform significantly differently than the linear benchmark model.[8]

Table 1: Canadian national house price growth forecast (one-step-ahead)

Model	In-sample relative RMSE ratio	Out-of-sample relative RMSE ratio	K-S p-value
Multi-layer perceptron	0.767	0.906	0.44
Support vector regression	0.768	0.892	0.72

Note: RMSE is root mean squared error. K-S is Kolmogorov-Smirnov.

The results for forecasting one-step-ahead monthly resales growth are similar. We find that both the in-sample and out-of-sample relative RMSE ratios of the machine learning models are below 1, signifying that the RMSEs of the MLP and SVR models are smaller than those of

[7] The Kolmogorov-Smirnov (K-S) test tries to determine whether two data samples differ significantly. It is a two-sided test with the null hypothesis that two independent samples are drawn from the same continuous distribution. In this case, we compare the distribution of forecasts produced by the machine learning models with the distribution of forecasts produced by the linear benchmark model. Therefore, when the K-S statistics are small, or the p-value is high, we cannot reject the hypothesis that the distributions of the two samples are the same.

[8] One may wonder whether there is any structure at all in the evolution or whether it is simply a random walk. A random walk would imply that the correlation between the dependent variable and its one-period lag is zero. With a standard statistical test on the correlation, we are able to rule out the random walk hypothesis at the 95% confidence level.

the OLS linear model. However, we are also unable to reject the null hypothesis of the machine learning distribution being the same as the linear distribution (**Table 2**).

Table 2: Canadian national home resales growth forecast (one-step-ahead)

Model	In-sample relative RMSE ratio	Out-of-sample relative RMSE ratio	K-S p-value
Multi-layer perceptron	0.650	0.655	0.11
Support vector regression	0.585	0.680	0.08

Note: RMSE is root mean squared error. K-S is Kolmogorov-Smirnov.

To test whether the national finding is also consistent with findings at the local level, we conduct the forecasting exercise with Toronto data. **Table 3** and **Table 4** present the results for forecasting one-step-ahead growth in house prices and resales in Toronto. Similar to the national findings, the relative RMSE ratios produced by the machine learning models are lower than 1, signifying that the RMSEs of the machine learning models are lower than those produced by the linear benchmark model. However, the results are not statistically significant because we fail to reject the K-S test null hypothesis that the distributions of the forecast results are the same.

Table 3: Toronto house price growth forecast (one-step-ahead)

Model	In-sample relative RMSE ratio	Out-of-sample relative RMSE ratio	K-S p-value
Multi-layer perceptron	0.574	0.878	0.06
Support vector regression	0.596	0.837	0.03

Note: RMSE is root mean squared error. K-S is Kolmogorov-Smirnov.

Table 4: Toronto home resales growth forecast (one-step-ahead)

Model	In-sample relative RMSE ratio	Out-of-sample relative RMSE ratio	K-S p-value
Multi-layer perceptron	0.606	0.705	0.02
Support vector regression	0.566	0.696	0.04

Note: RMSE is root mean squared error. K-S is Kolmogorov-Smirnov.

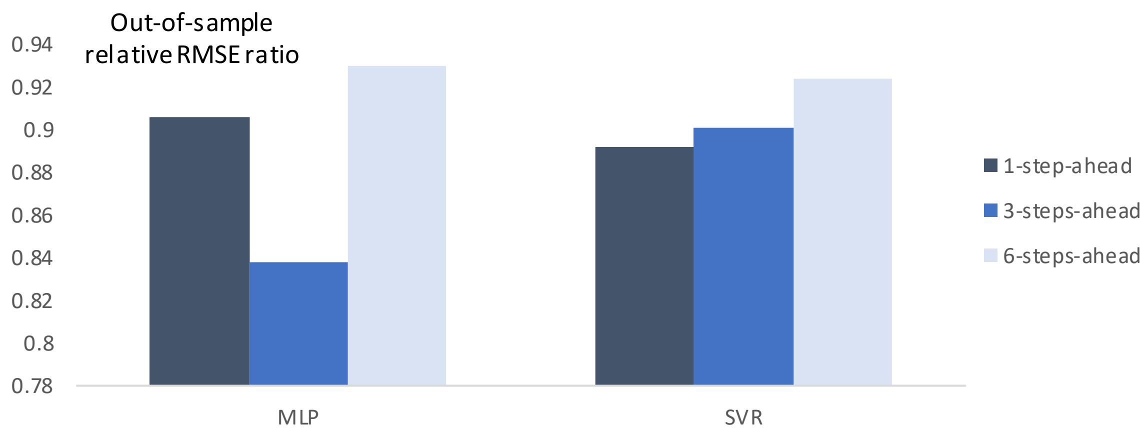
6. Sensitivity testing

In this section, we conduct sensitivity testing around the baseline results presented above by varying the parameters input into the models. We begin with varying the forecast horizon from one step ahead to three and six steps ahead. We then change the value of the penalty parameter of the elastic net, which implies changing the number of variables picked up by the elastic net that are used in the models.

6.1 Longer forecast horizon

We vary the number of forecast horizons from one to three and six to test for the robustness of the baseline results.[9] Consistent with the one-step-ahead forecast, the estimated out-of-sample relative RMSE ratios for predicting monthly national house price growth are mostly below 1 in the three- and six-steps-ahead forecasts. This implies that the out-of-sample RMSEs produced by the machine learning models are generally lower than the RMSEs produced by benchmark linear model across different forecast horizons when forecasting monthly house price growth (**Chart 4**).

Chart 4: The out-of-sample RMSEs of the ML models are generally below the linear model in predicting house price growth across forecast horizons

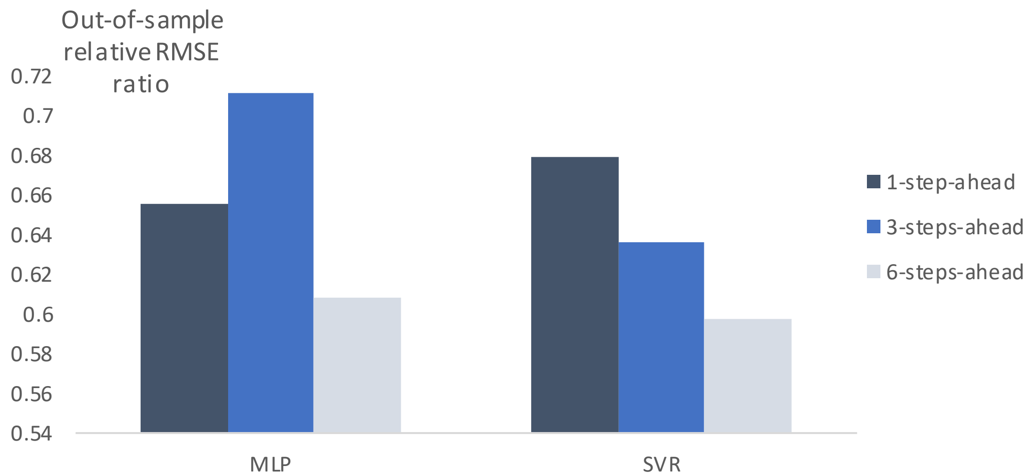


Note: The relative RMSE ratio is the root mean squared error (RMSE) of the machine learning model divided by the RMSE of the benchmark linear model. MLP is multi-layer perceptron, and SVR is support vector regression.
Sources: Bank of Canada calculations.

The results for forecasting national monthly resales growth are similar to those for house price growth, that relative RMSE ratios are below 1 across different forecast horizons (**Chart 5**).

[9] The number of variables picked up by the elastic net is the same as in the one-step-ahead forecast (149 variables) to allow for adequate comparison between the different forecast horizons.

Chart 5: **The out-of-sample forecast performance of ML models also outperform the linear model in predicting resales growth across forecast horizons**

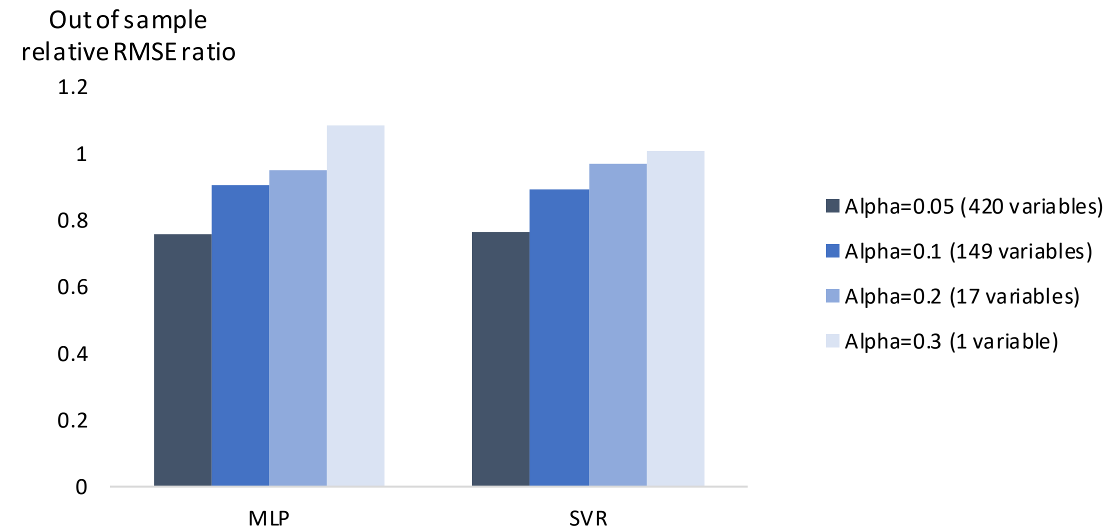


Note: The relative RMSE ratio is the root mean squared error (RMSE) of the machine learning models divided by the RMSE of the benchmark linear model. MLP is multi-layer perceptron, and SVR is support vector regression.
Source: Bank of Canada calculations

6.2 Varying elastic net penalty parameter

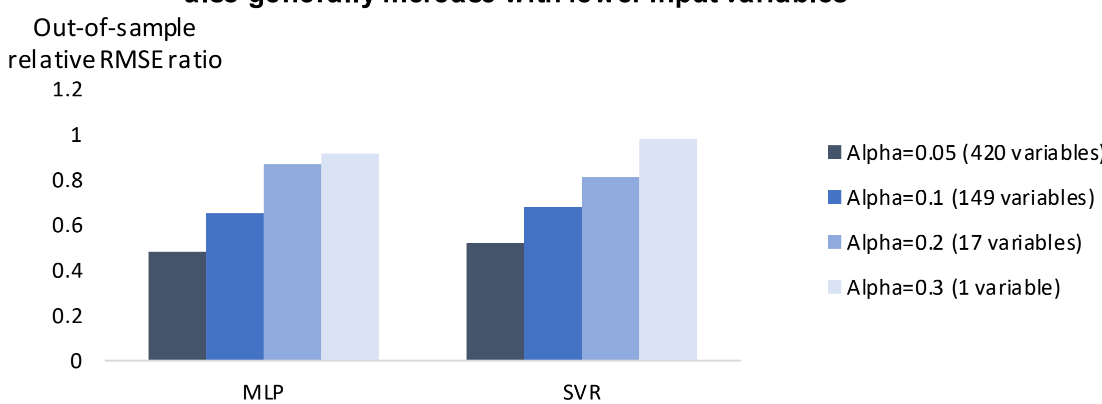
We also vary the elastic net penalty parameter to observe changes in the results when adjusting the number of variables input into the models. The baseline elastic net penalty parameter is 0.1. This implies including about 149 variables in the models. We decrease the penalty parameter to 0.05, which implies the number of variables input into the models increases to 420 (including lags of variables). We increase the penalty parameter to 0.2, which implies the number of variables input into the models decreases to 17. And increasing the penalty parameter to 0.3 would mean only 1 variable remains, namely the previous lags of the resales or price growth itself. As **Chart 6** and **Chart 7** show, increasing the value of the penalty parameter or decreasing the number of variables would generally increase the out-of-sample relative RMSE ratios of the machine learning models. This is because increasing the number of input variables generally punishes the forecast performance of the linear model as the RMSE of the linear model increases with the number of input variables. In contrast, there is no clear evidence that increasing the number of input variables punishes the forecast performance of the machine learning models (**Chart 8** and **Chart 9**). However, in the case with largest number of input variables (420 variables), the K-S statistics are significant, signifying that the RMSEs of the machine learning models are significantly different than those of the benchmark linear model.

Chart 6: **The out-of-sample relative RMSE ratios for predicting house price growth generally increase with fewer variables used in the models**



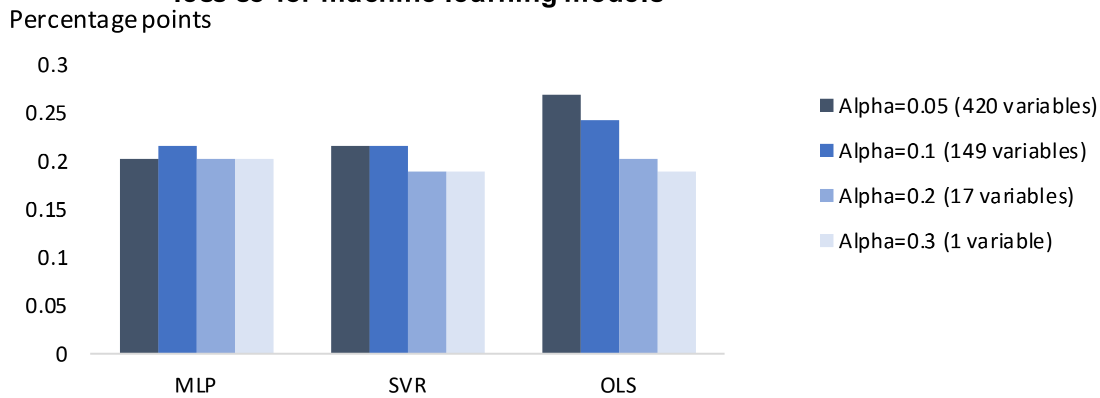
Note: The relative RMSE ratio is the root mean squared error (RMSE) of the machine learning models divided by the RMSE of the benchmark linear model. MLP is multi-layer perceptron, and SVR is support vector regression.
Source: Bank of Canada calculations

Chart 7: **The out-of-sample relative RMSE ratios for predicting resales growth also generally increase with fewer input variables**



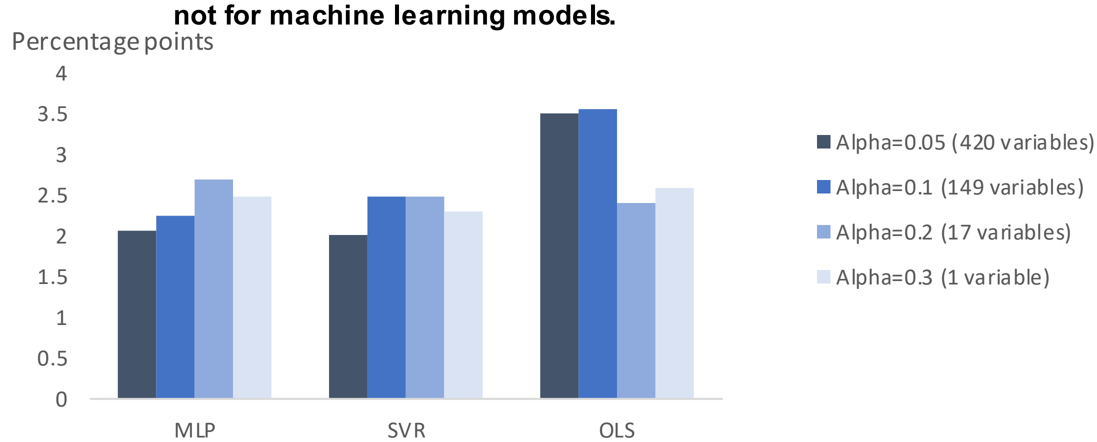
Note: The relative RMSE ratio is the root mean squared error (RMSE) of the machine learning models divided by the RMSE of the benchmark linear model. MLP is multi-layer perceptron, and SVR is support vector regression.
Source: Bank of Canada calculations

Chart 8: The out-of-sample RMSE for predicting house price growth decreases with fewer input variables used in the linear model, but less so for machine learning models



Source: Bank of Canada calculations

Chart 9: The out-of-sample RMSE for predicting resales growth generally decreases with fewer input variables used in the linear model, but not for machine learning models.



Source: Bank of Canada calculations

7. Conclusion

Our baseline results indicate that although the two machine learning algorithms (SVR and MLP) can perform slightly better than the linear model in forecasting one-step-ahead national monthly growth in house prices and resales, we cannot conclude that they perform significantly different. Similar to the national findings, the RMSEs produced by the machine learning models are slightly lower than those produced by the linear benchmark model when predicting monthly house price and resales growth in Toronto. However, the results are not statistically significant. Results are also similar when the forecast horizon is changed to three and six steps ahead. In contrast, increasing the number of input variables generally punishes the forecast performance of the linear benchmark model, but there is no clear evidence that it does the same for the machine learning models. Therefore, changing the number of variables supports the baseline finding that machine learning models produce slightly lower RMSEs than the linear model. This is in line with the conclusion reached by Coulombe et al. (2022) that more data and nonlinearities are very useful for machine learning techniques in predicting real activity series.[10] However, this marginally improved forecasting performance comes at a price: machine learning models are significantly more complex, and the economic interpretation of the results is less clear (interpreting machine learning forecasts is an active field of research in its own right [see, for instance, Hall and Gill 2018]).

However, our current approach, which is based on traditional time-series data, has limitations. It's possible that machine learning methods can significantly outperform linear regression in forecasts using non-traditional data sets (i.e., unstructured, high-frequency or sentiment data).

[10] Truncating our dataset to end in 2009 lowers the relative RMSEs to between 0.28 and 0.73, supporting the view that machine learning models perform significantly better in times of rapid change.

References

Coulombe, P. G., M. Leroux, D. Stevanovic and S. Surprenant. 2022. "How is Machine Learning Useful for Macroeconomic Forecasting?" *Journal of Applied Econometrics* 37 (5): 920–946.

Fortin-Gagnon, O., M. Leroux, D. Stevanovic and S. Surprenant. 2019. "A Large Canadian Database for Macroeconomic Analysis."

Hoerl, A. E. and R. W. Kennard. 1970. "Ridge Regression: Biased Estimation for Nonorthogonal Problems." *Technometrics* 12 (1): 55–67.

Khan, M. and T. Webley. 2019. "Disentangling the Factors Driving Housing Resales." Bank of Canada Staff Analytical Note No. 2019-12.

Milunovich, G. 2020. "Forecasting Australia's Real House Price Index: A Comparison of Time Series and Machine Learning Methods." *Journal of Forecasting*. 39 (7): 1098–1118.

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

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

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