Ekonomi-tek Volume / Cilt: 7 No: 1 Ocak / January 2018, 1-16

Forecasting Inflation Using Summary Statistics of Survey Expectations: A Machine-Learning Approach

Bige Küçükefe[1]

Abstract

This paper aims to produce more accurate short-term inflation forecasts based on surveys of expectations by employing machine-learning algorithms. By treating inflation forecasting as an estimation problem consisting of a label (inflation) and features (summary statistics of surveys of expectations data), we train a suite of machine-learning models, namely, Linear Regression, Bayesian Ridge Regression, Kernel Ridge Regression, Random Forests Regression, and Support Vector Machines, to forecast the consumer-price inflation (CPI) in Turkey. We employ the Time Series Cross Validation Procedure to ensure that the training data exclude forecast horizon data. Our results indicate that these machine-learning algorithms outperform the official forecasts of the Central Bank of Turkey (CBT) and a univariate model.

JEL Codes: C82, E31

Keywords: Machine learning, forecast evaluation, inflation forecasting, surveys of expectations, summary statistics

[1] Namık Kemal University, bkucukefe@nku.edu.tr

1. Introduction

Applications of machine-learning algorithms in many fields have gained momentum in recent years. In this study, we explore the use of machine-learning methods to improve the accuracy of surveys of inflation expectations as a viable alternative to existing inflation-forecasting models.

Univariate models, expectation surveys, and Phillips curve models have been widely used to forecast future inflation. Comparison across different forecasting models is difficult due to differences in data, sample periods, and country-specific conditions. Debate over model performances, however, has attracted many researchers over the years. A comprehensive study by Öğünç et al. (2013) used a collection of econometric models that include univariate models, decomposition-based approaches, a Phillips curve motivated time-varying parameter model, a suite of VAR and Bayesian VAR models, and dynamic-factor models to forecast short-term inflation in Turkey. Their result revealed that a combination of these models leads to a reduction in forecast error.

In a similar approach, Kapetanios et al. (2008) argued that a single model is outperformed by combinations of various models. A milestone work by Atkeson and Ohanian (2001) compared the performance of a naïve moving-average model with a series of Phillips curve forecasting models and argued that the former performed better than the latter. However, a later work by Stock and Watson (2007) found that Phillips curve methods performed better for the period 1970-83, and the results of Atkeson and Ohanian were specific to the period 1984-99.

Surveys of inflation expectations offer an alternative approach to inflation forecasting. The superior forecasting performance of surveys has been highlighted by several researchers. In particular, Grothe and Meyler (2015) argued that short-term inflation expectations derived from survey and market data for the euro area and the United States were informative predictors of future inflation developments. Similarly, Ang et al. (2007) show that inflation expectations from survey data beat a wide variety of forecasting models that include time-series ARIMA models; regressions using real activity data motivated from the Phillips curve, and term structure models that include linear, non-linear, and arbitrage-free specifications.

A similar study by Gil-Alana et al. (2012) revealed that survey-based expectations outperform standard time-series models in US quarterly inflation out-of-sample predictions. Furthermore, Altuğ and Çakmaklı (2016) formulated a statistical model of inflation that combines data from survey expecta-

tions of inflation and argued that the model with survey expectations yields superior predictive performance to the model lacking them, as well as popular benchmarks, such as the backward-looking Phillips curves and the naive forecasting rule.

The appeal of machine learning stems from its ability to uncover complex structures hidden in large datasets without explicit programming. Classification and regression are two central applications of "supervised" machine learning. Both involve making estimations of an unknown target from a set of known features by applying various algorithms, such as support-vector machines, random forests, and deep neural networks.

We employ "supervised" machine-learning models for short-term inflation forecasting by using the Central Bank of Turkey (CBT)'s survey of inflation expectations. We treat summary statistics of survey data as features without time stamp for a supervised machine-learning problem for which the label to estimate is the future inflation. From the viewpoint of machine learning, this is a standard regression problem. We implemented the machine- learning models in Python software by using the Scikit-learn (Pedregosa et al., 2012) library, which is opensource software. The codes are available from the author upon request.

The remainder of this paper is organized as follows: Section 2 briefly explains the machine- learning methods we employed to improve the forecasting accuracy of inflation-expectation surveys. Section 3 presents the data and methodology by which we compared the forecasts. Section 4 discusses the forecast performance of the machine-learning algorithms and compares their accuracy with the survey data, the CBT official forecasts, a naive MA method, and a univariate model, and Section 5 contains the conclusion.

2. Machine-Learning Models

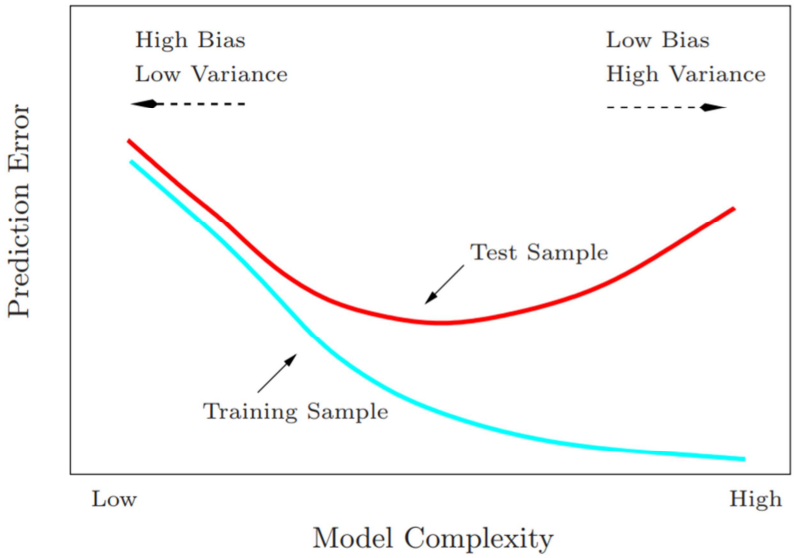
The success of machine learning primarily lies in its ability to discover unknown complex structures hidden in datasets. The common principle that underlies a supervised machine-learning model is to learn a target function (f) that maps input variables (X) to an output variable (Y). Without defining an explicit solution methodology that may not even exist, supervised machine-learning models "learn" from sample data and make estimations for out-of-sample data primarily for the purpose of binary classification, multiclass classification, and regression, among many other applications.

Dealing with under-fitting and over-fitting problems is important when selecting and tuning supervised machine models. The under-fitting problem occurs if the model doesn't represent the sample adequately. The over-fitting,

on the other hand, occurs if a model fits best on sample data and fails on out-of-sample data.

Therefore, bias-variance trade-off in machine learning is closely related to model complexity. As the model complexity increases, the variance tends to go up, and the bias tends to decrease—and vice versa [Mullainathan and Spiess, 2017]. Fig.1 depicts the prediction error as a function of model complexity with bias-variance combinations for a machine-learning model [Hastie et al., 2017, p. 38)].

Figure 1. Test and Training Error as a Function of Model Complexity



The performance evaluation of a supervised machine-learning model involves dividing a dataset into test and training sets. The training set is used to train the model, which is being evaluated against the test dataset. In our study, the summary statistics of inflation-expectation surveys (the features) and the actual inflation (label), without time information, are divided into two groups: the training and forecast sets.

We use five types of machine-learning algorithms: Linear Regression, Random Forests, Support-Vector Machines, Bayesian Ridge Regression, and Kernel Ridge Regression. Before introducing each of these models, we first present some fundamental aspects of supervised machine learning. See Hastie et al. (2009) and James et al. (2013) for details about the models.

2.1. Linear Regression

The underlying assumption of linear regression is that there is approximately a linear relation between response (Y) and variables (X) in a dataset with only quantitative values. We can write this relationship as

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon \quad (1)$$

Where $\beta_0,\ \beta_1, \ldots, \beta_p$ are the regression coefficients and ϵ is the error term. For p = 1, (1) transforms into a simple linear regression. The regression coefficients in (1) are estimated by using the least-squares approach in the formula:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta} x_2 + \cdots + \hat{\beta}_p x_p + \epsilon \quad (2)$$

such that the sum of squared residuals (RSS) is minimum. RSS is defined as

$$RSS = \sum_{i=1}^{n} \left(y_i - \left(\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + \cdots + \hat{\beta}_p x_{ip} \right) \right)^2 \quad (3)$$

2.2. Random Forests Regression

Random Forests (Breiman(2001)) are a collection of simple decision-tree predictors. Each decision tree in a Random Forest can produce a response for a set of input values. An algorithm determines the split points, splitting variables, and topology of a decision tree. The tree grows after solving for each split until a tuning parameter (tree size), which controls the model's complexity, is reached. Random Forests aim to improve the predictive performance of decision trees by aggregating many of them. Furthermore, Random Forests overcome the problem of the strong predictor estimation in the bagged-trees approach by allowing for a smaller number of randomly selected predictors for each split. The average of the predictions from all the trees is the ensemble estimation of the Random Forest model. Growing a regression tree requires an algorithm that automatically decides on the splitting variables and split points. The response of a model consisting of M regions $R_1, R_2, \ldots, R_M$ is determined by

$$f(x) = \sum_{i=m}^{M} c_i I(x \in R_m) \quad (4)$$

for each region. Starting with all data, the algorithm first considers the pair of half planes in terms of the splitting variable $j$and the split point s as

$$R_1(j,s) = \{X | X_j \leq s\} \text{ and } R_2(j,s) = \{X | X_j > s\} \quad (5)$$

and seeks the splitting variable j and split point s that minimize

$$\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_1(j,s)} (y_i - c_2)^2 \quad (6)$$

For any value of j and s, the inner minimization is solved by

$$\widehat{c_1} = average\big(y_i \big| x_i \in R_1(j,s)\big),$$

$$\widehat{c_2} = average\big(y_i \big| x_i \in R_2(j,s)\big), \quad (7)$$

By scanning through all the inputs, we determine the best pair of (j, s) is determined. Then the same process is repeated on resulting regions of data until the tree grows an optimal size according to a tuning parameter that determines the model's complexity.

One common strategy is to grow a large tree with some minimum node size, such as 5. Then this large tree is pruned using a *cost-complexity pruning* procedure. First, define a sub-tree $T \subset T_0$ of any tree obtained by pruning T_0, that is, collapsing any number from its non-terminal nodes. Letting

$$N_m = \#\{x_i \in R_M\},$$

$$\widehat{c_m} = \frac{1}{N_m}\sum_{x_i \in R_m} y_i, \quad (8)$$

$$Q_m(T) = \frac{1}{N_m}\sum_{x_i \in R_m} (y_i - \widehat{c_m})^2$$

where m is the index of terminal nodes in region R_M, we define the cost-complexity criterion

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T| \quad (9)$$

where $|T|$ denotes the number of terminal nodes in T, and α is the tuning parameter ($\alpha \geq 0$), which governs the bias-variance tradeoff in the model. The idea is to find the sub-tree $T_\alpha \subseteq T_0$ to minimize $C_\alpha(T)$ in (9) for each α. Small values of α yield larger trees Tα and vice versa. The full tree T_0 is returned with α = 0.

For each α there is a unique smallest sub-treeT_α that minimizes (9). To find T_α, the algorithm collapses the internal mode that produces the smallest per-node increase in $\sum_m N_m Q_m(T)$ until it produces the single-node (root) tree. This method is called *weakest link pruning* and gives a finite sequence of sub-trees that contain T_α. The tuning parameter α s estimated by choosing the value $\hat{\alpha}$ to minimize the cross-validated sum of squares, from which $T_{\hat{\alpha}}$ is the final tree.

Decision trees yield high-variance, low-bias output. One way to reduce high variance is to use the bagging technique, which simply fits the same regression tree many times to bootstrap-sampled versions of the training dataset

and compute the average. Since each tree generated in the bagging of B trees is identically distributed, the variance of the average is given by

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2 \tag{10}$$

The second term disappears in (10) as B increases (more trees), and, hence, the benefits of averaging get weaker due to the size of the correlation of pairs in the bagged trees. The Random Forests aim to solve this problem by randomly selecting the input variables in the tree-growing process. Before each split, m variables are selected randomly from the input variables (p) as candidates for splitting. After growing B trees with this procedure, the Random Forest regression predictor is defined as

$$\hat{f}_{rf}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T(x; \Theta_b) \tag{11}$$

where Θ_b characterizes the bth Random Forest tree in terms of split variables, cutpoints at each node, and terminal-node values.

2.3. Support-Vector Regression

The idea behind Support-Vector Machines (SVM) (Vapnik (1995)) is to find hyperplanes that separate different classes in a training dataset. SVM Regression (SVR) is a form of SVM with a numerical dependent variable instead of a categorical one. SVR relies on kernel functions (linear, polynomial, radial basis, etc.) to construct optimal hyperplanes.

Kernel function transforms the training data from nonlinear space to linear space. This transformation allows SVR to find an optimum hyper plane. Mapping back to the original space completes the algorithm. For a linear-regression model, $f(x) = x^T\beta + \beta_0$, and estimation of β and β_0 are possible through minimization of

$$H(\beta, \beta_0) = \sum_{i=1}^{N} V\big(y_i - f(x_i)\big) + \frac{\lambda}{2}||\beta||^2 \tag{12}$$

where

$$V_t = \begin{cases} 0, & if|r| < \epsilon, \\ |r| - \epsilon, & otherwise \end{cases}$$

The solution functions for $\hat{\beta}$ and $\hat{\beta}_0$ that minimize H have the form

$$\hat{\beta} = \sum_{i=1}^{N}(\hat{\alpha}_i^* - \hat{\alpha}_i)\, x_i \tag{13}$$

$$\hat{f}(x) = \sum_{i=1}^{N}(\hat{\alpha}_i^* - \hat{\alpha}_i)\, \langle x, x_i\rangle + \beta_0 \tag{14}$$

Where $\widehat{\alpha}_i$ and $\widehat{\alpha}_i^*$ are called the support vectors, which solve the quadratic optimization problem

$$\min_{\alpha_i,\alpha_i^*} \epsilon \sum_{i=1}^{N}(\alpha_i^* + \alpha_{i,}) - \sum_{i=1}^{N} y_i(\alpha_i^* - \alpha_i) + \frac{1}{2}\sum_{i,i'=1}^{N}(\alpha_i^* - \alpha_i)$$

$$(\alpha_{i'}^* - \alpha_{i'})\langle x, x_i'\rangle \tag{15}$$

subject to

$0 \leq \alpha_i, 0 \leq \alpha_i^* \leq 1/\lambda,$

$$\sum_{i=1}^{N}(\alpha_i^* - \alpha_i) = 0,$$

$$\alpha_i\alpha_i^* = 0$$

2.4. Kernel Ridge Regression

Kernel Ridge Regression (KRR) (Hsiang (1975) is a form of linear regression. KRR imposes a penalty on the size of the regression coefficients, which are estimated by obtaining $\hat{\beta}^{ridge}$ values that minimize

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta^2 = RSS + \sum_{j=1}^{p}\beta_j^2 \tag{16}$$

where $\lambda \geq 0$ is a tuning parameter to be calculated separately. Limiting the size of the regression coefficients alleviates the high variance problem caused by the large coefficients of the correlated variables in a model. Writing the criterion (16) in matrix form,

$$RSS(\lambda) = (y - \boldsymbol{X}\beta)^T(y - \mathbf{X}\beta) + \lambda\beta^T\beta \tag{17}$$

we estimate the regression parameters as

$$\hat{\beta}^{ridge} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\mathbf{I})^{-1}\boldsymbol{X}^{\boldsymbol{T}}y \tag{18}$$

where I is the identity matrix with size $p \times p$. The choice of quadratic penalty in (17) makes the KRR solution a linear function of y. If the inputs are orthonormal, the KRR estimates are a scaled version of the least-squares estimates ($\hat{\beta}^{ridge} = \hat{\beta}/(1 + \lambda)$).

2.5. Bayesian Ridge Regression

Bayesian regression introduces uninformative priors over the hyper parameters of the model. The output y is assumed to be Gaussian distributed around X, w such that:

$$p(y|X, w, \alpha) = \mathcal{N}(w|X, w, \alpha) \tag{19}$$

Where α is treated as a random variable and estimated from data. In a probabilistic model estimated by Bayesian Ridge Regression, the prior for the parameter w is given by a spherical Gaussian:

$$p(w|\lambda) = \mathcal{N}(w|0, \lambda^{-1}\mathbf{I}_{\boldsymbol{p}}) \tag{20}$$

The priors over α and λ are gamma distributions (the conjugate prior for the precision of the Gaussian normal distribution). The parameters w, α, and λ are estimated jointly in the model.

3. Data and Methodology

The Central Bank of Turkey (CBT) conducts a survey each month to monitor the expectations of experts from the financial and real sectors. The questionnaire includes short-term inflation forecasts (current month, next month, and two months ahead) in addition to many other expectations of economic variables, such as exchange and interest rates. The survey reports also contain summary statistics that consist of mode, median, minimum, arithmetic mean, and maximum and minimum values. We obtained the survey of expectations data from the CBT and the Consumer-Price Inflation (CPI) data from TURKSTAT. The data cover the period from August 2001 to December 2017, for which we produced monthly inflation forecasts for three horizons: current month (h=1), next month (h=2), and two months ahead (h=3).

The Time-Series Cross-validation (TSCV) procedure uses the past data only for the training of the machine-learning models without any information about the forecast horizon or beyond, thereby producing out-of-sample forecasts. We used an expanding-window rather than the rolling-window estimation procedure to permit the use of more data for the learning process. Expanding the window-procedure estimates model on a sample running from $1, 2, \ldots, t$ and we produced forecasts of variables at date $t + h : h > 0$.

The performance metric we employed to evaluate the forecasts by the machine-learning models is the Relative Root Mean Square Error (R-RMSE), which is calculated by dividing the model forecast RMSE by the survey forecast RMSE:

$$R - RMSE = \frac{\sqrt{\sum_{i=1}^{N}(\pi_i - f_i^m)^2}}{\sqrt{\sum_{i=1}^{N}(\pi_i - f_i^s)^2}} \quad (21)$$

where π is the actual inflation, f^m is the model forecast, and f_s is the arithmetic mean of survey expectations. A relative RMSE value less than unity indicates that the model improved on the surveys' forecasts and vice versa.

Furthermore, we computed the Empirical Cumulative Distribution Function (ECDF) for Bootstrap RMSEs of the forecasts. ECDF is defined as:

$$\text{ECDF}\ (RMSE) = \frac{\text{number of elements in the sample} \leq RMSE}{N} \quad (22)$$

where N is the total number of elements in the sample. ECDF used the data from the Bootstrap replicates of the forecasts and their RMSEs to draw statistical inferences. We used 10,000 bootstrap replicates with replacements.

We compare the forecast performance of the machine learning models with a suite of forecasts that consist of survey data, the CBT forecasts, a time-series forecasting model (TBATS), and a naive moving-average model. TBATS is a state space-modeling framework (de Livera et al. (2011)) for univariate time series forecasting with complex seasonal patterns. This trigonometric framework has both linear and nonlinear time-series modeling capacity with single seasonality, multiple seasonality, high period, and non-integer seasonality. TBATS incorporates Box-Cox transformations, Fourier representations with time-varying coefficients, and ARMA error correction. We used the R implementation of the TBATS model to compute the point forecasts for the next three months, starting from the first month of the respective quarter, from 2016-Q4 to 2017-Q4.

The CBT doesn't publish its monthly inflation forecasts. Hence, we use the quarterly inflation forecasts for comparison. Although quarterly inflation forecasts are not provided separately in the CBT's inflation reports, one can calculate them by solving the following equation analytically for f_Q.

$$(1 + \pi_{9m})(1 + f_Q) = 1 + f_{12m} \quad (23)$$

where π_{9m} is the actual past nine months' inflation and f_{12m} is the CBT's yearly inflation forecast at the end of the quarter that is available in the CBT's inflation report.

Calculating the quarterly inflation forecasts makes use of the next month and two months ahead forecasts from the models, such that the next quarter inflation forecast ($\hat{\pi}_{Qm}$), starting with the current month m , is given by

$$\hat{\pi}_{Qm} = \left(1 + f_{m-1}^{h(2)}\right)\left(1 + f_{m}^{h(2)}\right)\left(1 + f_{m}^{h(3)}\right) - 1 \quad (24)$$

where $f_m^{h(n)}$ is the model forecast in month m for horizon n.

As a benchmark, we employ a naive Moving Average (MA) model, in which the next quarter's inflation is equal to the arithmetic mean of the past four quarters' inflation rates (Atkeson and Ohanian, 2001).

$$\hat{\pi}_{t+Q} = \frac{1}{4}\left(\pi_{t-Q_1} + \pi_{t-Q_2} + \pi_{t-Q_3} + \pi_{t-Q_4}\right) \quad (25)$$

Even though it is well established that a naive MA model does not perform well for short time horizons for economies lacking stable inflation dynamics, the model nonetheless serves as a benchmark.

The results we obtained with the default parameters of the machine-learning models in the Scikit-learn library are satisfactory to the extent that they demonstrate the effectiveness of the proposed approach.

4. Forecasting Performance Evaluation

The forecast period covers five quarters, from 2016-Q4 to 2017-Q4. We evaluate the forecast performance of the machine-learning models, the surveys, the CBT, the TBATS model, and a naive MA model serving as a benchmark. The ECDF of Bootstrap mean of RMSEs offers useful information for comparing forecasting performances.

Table 1 shows the probability of fractional improvement by the machine-learning models at each time horizon, based on the Bootstrap mean of fractional improvements. At horizon 1, only the LR provides a fractional improvement, with a 68% probability, whereas the other models remain below the 50% critical level, which indicates no improvement.

On the other hand, RFR performs better than the survey, with an 87% probability at horizon 2, followed by the KRR and BRR, with 74% and 60%. Both the LR and SVR remain below 50% at horizon 2, meaning that they fail to improve the accuracy of survey forecasts.

Significant fractional improvement is achieved by the models at horizon 3, with 99% probability by RFR and 98% by KRR, BRR, and SVR. The accuracy gain by LR occurs with lower probability than the other models, but with 68%, it can still improve the survey forecast at horizon 3.

Table 1. Probability of Fractional Improvement by Machine-Learning Models

Forecast horizon	RFR	KRR	LR	BRR	SVR
h=1	0.33	0.36	**0.68**	0.32	0.39
h=2	**0.87**	0.74	0.49	0.60	0.30
h=3	**0.99**	0.98	0.68	0.98	0.98

Note: Forecast horizons are in months. Bold text indicates the most accurate model for the forecast horizon.
Abbreviations: RFR: Random Forests Regression, KRR: Kernel Ridge Regression, LR: Linear Regression, BRR: Bayesian Ridge Regression, SVR: Support Vector Regression

Table 2 gives RMSEs and relative RMSEs for the forecasts. Although the survey forecasts performed worse than the CBT and TBATS, all the models scored smaller RMSEs, in the range of 0.76 to 0.83. With 0.76 RMSE, LR outperformed the other machine-learning models, but the difference is negligible, except for KRR, which has an RMSE of 0.83. TBATS produced better forecasts than the CBT.

MA yielded the worst forecast performance due to the rapidly changing inflation dynamics during the forecast period. The CBT scored only 7% less than the RMSE. By comparison, the TBATS scored 18% better than the survey forecasts. The machine-learning models improved on the survey forecasts by up to 38%.

Table 2. Quarterly CPI Forecast Performances

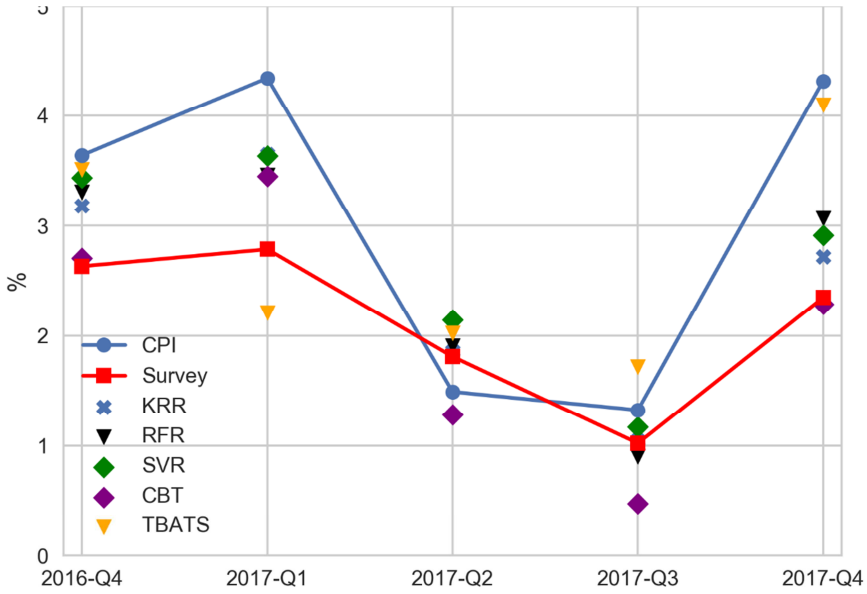
Forecast by	RMSE	Relative RMSE (MA)	Relative RMSE (Survey)
MA	1.70	1.00	1.39
CBT	1.14	0.67	0.93
TBATS	1.01	0.59	0.82
Survey	1.23	0.72	1.00
LR	0.76	0.45	0.62
BRR	0.78	0.46	0.64
RFR	0.77	0.45	0.63
KRR	0.83	0.49	0.68
SVR	0.77	0.45	0.63

Note: Forecast period from 2016-Q4 to 2017-Q5.
Abbreviations: CPI: Consumer-Price Inflation, MA: Naïve Moving Average forecast, CBT: Central Bank of Turkey forecasts, TBATS: A univariate model by de Livera et al. (2011). See note to Table 1 for other abbreviations.

The quarterly inflation forecasts of the machine-learning models, surveys, CBT and TBATS are shown in Figure 2. Due to the limited number of forecasting points and changing performances of the models for different periods, it is difficult to judge the forecast accuracies from the figure. On the other hand, Figure 3 shows the ECDF of the Bootstrap means of forecast absolute errors, which reveals the forecasting performances.

It is apparent from the figure that the machine-learning models outperform the CBT forecasts, which are slightly more accurate than the surveys. Even though the TBATS produced the best forecasts, its performance was not consistent, and large forecasting errors moved the TBATS to a ranking between the CBT and the machine-learning models.

Figure 2. Quarterly CPI Forecasts

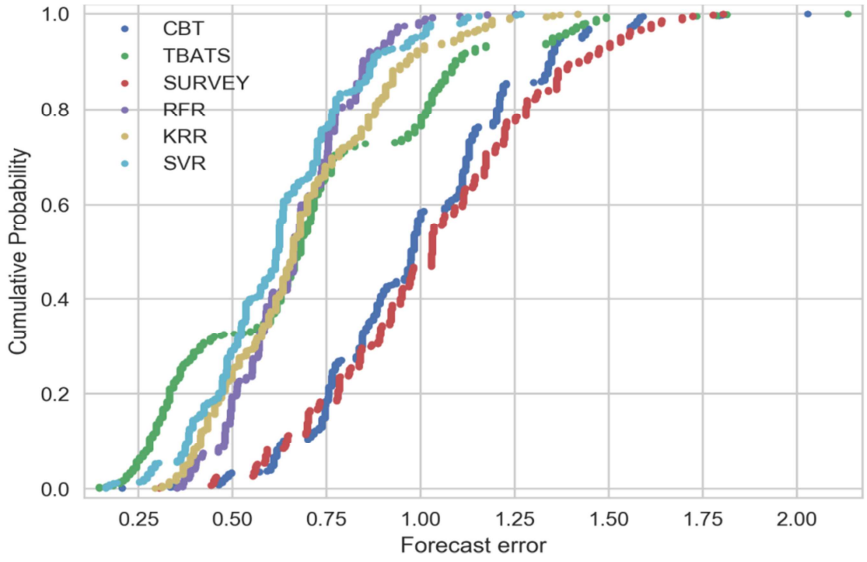


Note: See the note to Table 2 for abbreviations.

To summarize, even though the surveys performed worse than the CBT and the TBATS forecasts, the machine-learning models relying only on the summary statistics of the survey data produced significant levels of accuracy, which led to better forecasts than those of the CBT and the TBATS, as measured by the RMSE. In particular, the RFR yielded the best performance among the selected machine-learning models.

Further accuracy improvement of the models is possible through parameter optimization and an increase in the size of the training data. In addition, extending the forecast period with new data as they become available will contribute to better evaluation of the machine-learning models and improve the forecasting capability of inflation-expectations surveys.

Figure 3. Bootstrap Means of Forecast Errors.



Note: See the note to Table 2 for abbreviations.

5. Conclusion

We have employed a suite of machine-learning models to improve the accuracy of surveys of inflation expectations, conducted by the Central Bank of Turkey (CBT). A training set consisting of only the summary statistics of survey data and actual inflation in Turkey was used. The comparison of forecast performances vis-a-vis the forecasts by the CBT and a univariate model (TBATS) shows that the proposed method not only improves the accuracy of the surveyed forecasts but also outperforms the CBT and TBATS, which are themselves more accurate than the surveys.

We treat the inflation forecasting as an estimation problem in machine learning. The summary statistics of survey data form the features set and the actual inflation is used as labels. The time-series cross validation procedure ensures that the forecast horizon data are not included in the training set for the machine-learning model. Among the models, the RFR yielded the best fractional improvement.

References

Altuğ, S., & C. Çakmaklı, (2016), "Forecasting Inflation Using Survey Expectations and Target Inflation: Evidence for Brazil and Turkey," *International Journal of Forecasting*, *32* (1), pp. 138-153. https://doi.org/10.1016/j.ijforecast.2015.03.010

Ang, A., Bekaert, G., & M. Wei, (2007), "Do Macro-variables, Asset Markets, or Surveys Forecast Inflation Better?" *Journal of Monetary Economics*, *54* (4), pp. 1163-1212. https://doi.org/10.1016/j.jmoneco.2006.04.006

Atkeson, A., & L. E. Ohanian, (2001), "Are Phillips Curves Useful for Forecasting Inflation?" *Federal Reserve Bank of Minneapolis Quarterly Review*, *25* (1), pp. 2-11.

Breiman, L., (2001), "Random Forests," *Machine Learning, 45* (1), pp. 5-32. https://doi.org/10.1023/A:1010933404324

Gil-Alana, L., A. Moreno, & F. de Gracia, (2012), "Exploring Survey-Based Inflation Forecasts," *Journal of Forecasting*, 31 (6), pp. 524-539. https://doi.org/10.1002/for.1235

Grothe, M., & A. Meyler, (2015), "Inflation Forecasts: Are Market-based and Survey-based Measures Informative?," *European Central Bank, Working Paper No. 1865*.

Hastie, T., R. Tibshirani, & J. Friedman, (2017), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer.

Hsiang, T. C., (1975), "A Bayesian View on Ridge Regression," *Journal of the Royal Statistical Society, Series D (The Statistician)*, 24 (4), pp. 267-268.

James, G., D. Witten, T. Hastie, & R. Tibshirani, (2013), *An Introduction to Statistical Learning*, Vol. 103, ISBN 1461471389, doi:10.1007/978-1-4614-7138-7

Kapetanios, G., V. Labhard, & S. Price, (2008), "Forecast Combination and the Bank of England's Suite of Statistical Forecasting Models," *Economic Modeling*, *25* (4), pp. 772-792. https://doi.org/10.1016/j.econmod.2007.11.004

Livera, A. M., R. J. Hyndman, & R. D. Snyder, (2011), "Forecasting Time Series with Complex Seasonal Patterns Using Exponential Smoothing," *Journal of the American Statistical Association*, 106 (496), pp. 1513-1527. https://doi.org/10.1198/jasa.2011.tm09771

Mullainathan, S., & J. Spiess, (2017), "Machine Learning: An Applied Econometric Approach," *Journal of Economic Perspectives*, 31 (2), pp. 87-106. https://doi.org/10.1257/jep.31.2.87

Öğünç, F., K. Akdoğan, S. Başer, M. G. Chadwick, D. Ertuğ, T. Hülagü, & N. Tekatlı, (2013), "Short-term Inflation Forecasting Models for Turkey and A Forecast Combination Analysis," *Economic Modeling*, *33*, pp. 312-325.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, & E. Duchesnay, (2012), "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, 12, pp. 2825-2830. https://doi.org/10.1007/s13398-014-0173-7.2

Stock, J. H., & M. W. Watson, (2007), "Why Has US Inflation Become Harder to Forecast?" *Journal of Money, Credit and Banking*, 39, pp. 3-33. https://doi.org/10.1111/j.1538-4616.2007.00014.x

Vapnik, V. N., (1995), *The Nature of Statistical Learning: Theory*, Springer. https://doi.org/10.1109/TNN.1997.641482