# High-Frequency Inflation Forecasting: A Two-Step Machine Learning Methodology[*]

SSSSSS[†]

April 3, 2025

### Abstract

This study introduces a novel two-step machine learning methodology to generate high-frequency (daily and weekly) inflation forecasts in developing economies, where official statistics are typically available only at a monthly frequency and with delays. High-frequency forecasting in this context is understood as nowcasting or interpolation—real-time prediction ahead of official releases or within-period estimation using mixed-frequency indicators—while also functioning as a data-augmentation strategy. Applying the approach to Bolivia, the framework first constructs monthly-aligned features by aggregating high-frequency predictors, followed by training and validating machine learning models. Feature selection via L1 regularization and fine-tuned hyperparameters yield a final model that achieves superior out-of-sample accuracy. This model is deployed to produce high-frequency year-on-year CPI inflation nowcasts. Forecasts exhibit strong temporal alignment with observed monthly values, while distributional equivalence is confirmed via Kolmogorov–Smirnov tests. Compared to benchmark econometric models, the two-step machine learning approach delivers improved predictive performance, offering timely, granular insights to support forward-looking monetary policy in data-scarce environments.

**Keywords:** Inflation; Forecasting; Nowcasting; Machine Learning; High-frequency data; Mixed-frequency models; Data-Augmentation
**JEL Codes:** C53, E31, C22, C55

[*] The author's views and perspectives expressed herein are personal and may not reflect the positions of affiliated institutions. E-mail: `ssss.ssssss@gmail.com`

[†] Institution, City, Country.

# 1 Introduction

Inflation forecasting plays a central role in monetary policy formulation, economic planning, and private sector decision-making. Accurate and timely inflation estimates are essential for anchoring expectations, calibrating policy instruments, and mitigating adverse macroeconomic shocks (Bernanke and Woodford, 1997; Stock and Watson, 1999). However, a critical limitation persists: inflation statistics, such as the Consumer Price Index (CPI), are generally released with significant publication lags and typically at a monthly frequency. This delay leaves policymakers with an informational void during periods when rapid responses are most needed.

While developed countries benefit from abundant high-frequency data—such as real-time commodity prices, financial indicators, and digital consumption proxies—that support nowcasting and rapid updates to inflation projections (Ghysels and Marcellino, 2018; Varian, 2014), developing economies face a different reality. The scarcity of high-frequency data and lack of public access to granular economic indicators hinder timely monitoring. Bolivia serves as a representative case: despite weekly collection of price data by the national statistical office, only monthly CPI inflation reports are officially released, limiting the capacity of policymakers and economic agents to anticipate and respond to inflationary shocks in real time.

Traditional econometric techniques—such as ARIMA models, Vector Autoregressions (VAR), state-space frameworks, and Dynamic Stochastic General Equilibrium (DSGE) models—have long served as workhorses of inflation forecasting (Faust and Wright, 2013). These models, while theoretically grounded and interpretable, often fail to capture complex nonlinear dynamics, evolving structural relationships, or distributional shifts. Mixed-frequency approaches, including Bridge Equations and Mixed-Data Sampling (MIDAS) regressions, improve upon this by accommodating data at different temporal resolutions (Baffigi et al., 2004; Ghysels et al., 2007; Clements and Galvao, 2008). However, their reliance on linear and parametric assumptions constrains their flexibility and robustness in volatile environments.

Emerging machine learning (ML) methodologies offer a compelling alternative. Techniques such as Ridge, Lasso, Random Forest, and Gradient Boosting regressors can flexibly model nonlinearities and high-dimensional interactions with minimal structural assumptions. A growing body of evidence shows that ML-based models frequently outperform traditional approaches, particularly during turbulent periods marked by structural change or unusual economic conditions (Moshiri and Cameron, 2000; Medeiros et al., 2021; Goulet Coulombe et al., 2022; Masini et al., 2023).

Nevertheless, despite the empirical success of ML in macroeconomic forecasting, the literature has predominantly focused on monthly-frequency predictions in data-rich settings. Little attention has been paid to leveraging ML for high-frequency inflation nowcasting in developing countries, where low-frequency official statistics coexist with fragmented, high-frequency indicators. This gap poses both a challenge and an opportunity: designing an approach that reconciles disparate data sources, preserves temporal structure, and delivers timely, reliable forecasts in data-constrained environments.

This study responds to that need by proposing a novel *two-step machine learning methodology* tailored for high-frequency inflation forecasting. In this context, *"high-frequency forecasting"* refers to nowcasting—generating weekly or daily estimates of year-on-year inflation ahead of official monthly releases. Furthermore, the methodology can be understood as an interpolation or data-augmentation technique, as it leverages high-frequency inputs to generate plausible intra-month estimates, effectively enriching the observed monthly series with temporally disaggregated, model-consistent forecasts.

The methodological contribution is twofold. First, the proposed framework addresses the challenge of mixed-frequency data by aggregating daily and weekly indicators into monthly-compatible inputs and validating their distributional consistency with the target variable. Second, a machine learning model trained on monthly data is deployed to produce high-frequency forecasts by treating aggregated daily or weekly data as model inputs. This separation between aggregation and forecasting enhances robustness, reduces dimensionality, and ensures statistical alignment between training and prediction datasets.

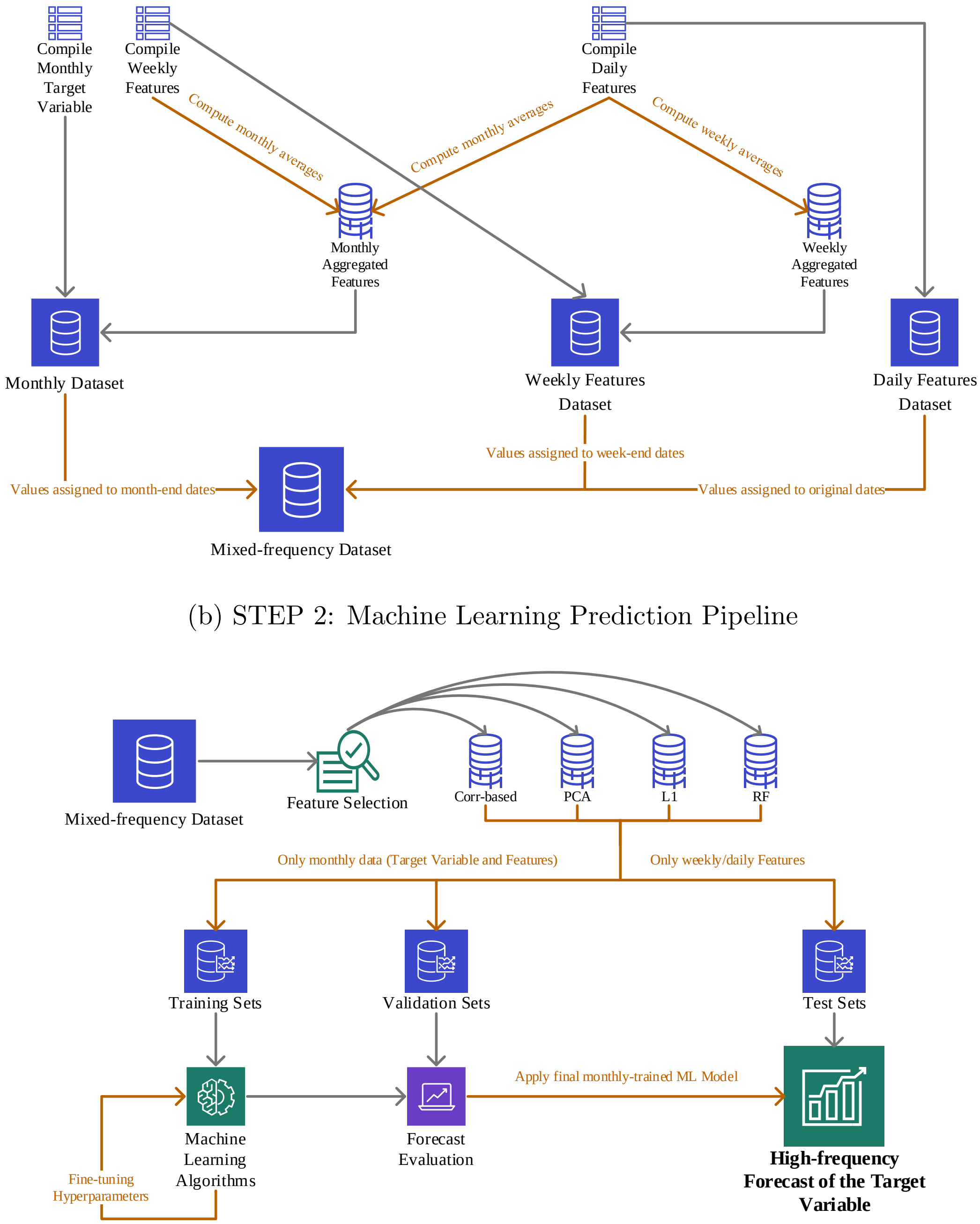
An empirical application to Bolivia illustrates the feasibility and effectiveness of this approach. Leveraging over 460 candidate predictors across wholesale prices, financial indicators, global commodities, and Google Trends data, the final model—Ridge regression using L1-regularized feature selection—demonstrates superior out-of-sample predictive accuracy compared to benchmark econometric approaches including Bridge Equations and MIDAS. Daily and weekly forecasts replicate long-term trends, uncover within-month fluctuations, and provide real-time insights suitable for policy intervention.

The remainder of the paper is structured as follows. Section 2 details the two-step machine learning framework. Section 3 presents the empirical application and comparative evaluation. Section 4 concludes with implications for forecasting practice and avenues for future research.

# 2 Methodology

The proposed approach combines mixed-frequency data aggregation with machine learning techniques in a structured two-step framework (visual representation in Figure 1).

Figure 1: Two-Step Machine Learning Methodology



(a) STEP 1: Mixed-Frequency Data Preparation

Compile Monthly Target Variable
Compile Weekly Features
Compile Daily Features
Compute monthly averages
Compute monthly averages
Compute weekly averages
Monthly Aggregated Features
Weekly Aggregated Features
Monthly Dataset
Weekly Features Dataset
Daily Features Dataset
Values assigned to week-end dates
Values assigned to month-end dates
Values assigned to original dates
Mixed-frequency Dataset

(b) STEP 2: Machine Learning Prediction Pipeline

Mixed-frequency Dataset
Feature Selection
Corr-based
PCA
L1
RF
Only monthly data (Target Variable and Features)
Only weekly/daily Features
Training Sets
Validation Sets
Test Sets
Machine Learning Algorithms
Fine-tuning Hyperparameters
Forecast Evaluation
Apply final monthly-trained ML Model
High-frequency Forecast of the Target Variable

The goal is to produce high-frequency inflation estimates (e.g., weekly and daily) using models trained on lower-frequency (monthly) data. Below, the two main steps are described, emphasizing the theoretical justification for mixing frequencies, the validation of distributional consistency, and the machine learning pipeline for feature selection and prediction.

## 2.1 Two-Step Machine Learning Methodology

**Step 1: Mixed-Frequency Data Preparation.** The first step constructs a dataset that links high-frequency indicators with the low-frequency target variable. Daily and weekly variables are aggregated into monthly equivalents so that they can be compared and combined with the monthly target variable. Let $x^d_{i,t}$ denote the value of daily feature $i$ on day $t$, and $x^w_{j,w}$ the value of weekly feature $j$ in week $w$. Monthly aggregates for each such feature are computed via a simple averaging protocol:

$$x^{\text{daily}}_{i,m} = \frac{1}{N_m} \sum_{t \in T_m} x^d_{i,t}, \qquad T_m = \text{days in month } m,\ N_m = |T_m|, \tag{1}$$

$$x^{\text{weekly}}_{j,m} = \frac{1}{K_m} \sum_{w \in W_m} x^w_{j,w}, \qquad W_m = \text{weeks in month } m,\ K_m = |W_m| \in \{4, 5\}. \tag{2}$$

Equation (1) collapses daily observations into a monthly average (e.g., averaging daily oil prices in month $m$), while equation (2) does the same for data available at a weekly frequency. This yields for each month $m$ a vector of aggregated daily-sourced features $X^{\text{daily}}_m$ and weekly-sourced features $X^{\text{weekly}}_m$. These are then combined to form a unified monthly feature matrix:

$$X_m = \left[ X^{\text{daily}}_m \,\|\, X^{\text{weekly}}_m \right],$$

concatenating the $P$ daily-derived features and $Q$ weekly-derived features into a single $1 \times (P+Q)$ vector for month $m$. By construction, $X_m$ preserves the informational content of higher-frequency predictors but expresses them at a common monthly frequency, thus resolving the temporal misalignment between predictors and target.[1]

A key theoretical justification for this aggregation approach is that monthly data (used as the training set in Step 2) inherently constitute aggregations of underlying higher-

[1] A similar procedure is followed when the aim is to aggregate and forecast on a weekly basis. Specifically, daily observations are aggregated into weekly averages using an analogous approach to equation (1), and the resulting weekly feature matrix $X_w$ is constructed by concatenating these daily-derived aggregates with the original weekly-frequency data.

frequency (daily or weekly) data. Consequently, daily and weekly data (employed as the test set in Step 2) naturally follow a similar underlying data-generating process as that of the monthly observations. Formally, this implies for each feature:

$$F_{X_d}(x) \approx F_{X_w}(x) \approx F_{X_m}(x),$$

where $F(\cdot)$ denotes the cumulative distribution function (CDF) corresponding to each frequency level. Since monthly data are derived by aggregating the underlying higher-frequency processes, it naturally follows that daily or weekly data or their aggregated counterparts should replicate the distributional properties of monthly observations. This similarity in distributional properties, termed *distributional equivalence*, ensures that a model trained on monthly data can validly generalize to higher-frequency inputs (daily or weekly) without introducing systematic biases.

To empirically verify this critical assumption, the methodology includes a formal statistical validation step using two-sample Kolmogorov–Smirnov (K–S) tests. The K–S test assesses whether two samples originate from the same underlying continuous distribution by comparing their empirical cumulative distribution functions (CDFs). Specifically, for each feature $i$, the following hypotheses are tested:

- For daily frequency:

$$H_0^{(1)} : F_{x_{i,d}}(x) = F_{x_{i,m}}(x), \quad \text{versus} \quad H_a^{(1)} : F_{x_{i,d}}(x) \neq F_{x_{i,m}}(x),$$

- For weekly frequency:

$$H_0^{(2)} : F_{x_{i,w}}(x) = F_{x_{i,m}}(x), \quad \text{versus} \quad H_a^{(2)} : F_{x_{i,w}}(x) \neq F_{x_{i,m}}(x),$$

where $F_{x_{i,d}}$, $F_{x_{i,w}}$, and $F_{x_{i,m}}$ represent the empirical distributions of the daily, weekly, and monthly aggregated values of feature $i$, respectively.

The K–S test is particularly suitable for validating distributional equivalence because it assesses differences across the entire distribution rather than solely at specific points or statistical moments. Failure to reject the null hypotheses at conventional significance levels (e.g., $p > 0.05$) indicates that daily or weekly observations are statistically indistinguishable from monthly observations. This empirical verification supports the assumption that machine learning models trained on monthly data can be appropriately applied to generate unbiased daily or weekly inflation nowcasts, as detailed in Step 2, thereby reinforcing the theoretical and methodological rigor of the framework.

Following aggregation and validation, the *mixed-frequency dataset* is constructed. Observations are timestamped such that monthly variables align with the final day of each month, weekly variables with the final day of each week (e.g., Sunday), and daily variables retain their exact observation dates. These variables are then concatenated row-wise into a single, coherent time-indexed database $D_{\text{mixed}}$. Within this dataset, the target variable—originally observed at monthly frequency—contains values aligned exclusively with month-end dates; intermediate missing values at higher frequencies are subsequently estimated using projections generated through the approach outlined in Step 2.

**Step 2: Machine Learning Prediction Pipeline.** In the second step, the constructed *mixed-frequency dataset* is used to train machine learning models and subsequently generate high-frequency forecasts of the target variable.

To assess out-of-sample validity, the monthly data is randomly partitioned into training and validation subsets. Specifically, 80% of observations are randomly selected as the training set, and the remaining 20% form the validation set. This methodological choice departs from the traditional temporal split, justified by the limited (monthly) sample size and the primary objective of this study: generating high-frequency inflation forecasts that effectively serve as interpolations within monthly intervals and real-time nowcasts. A random partitioning strategy, spanning the entire available sample period, ensures that the training set adequately captures diverse economic conditions and variability throughout the entire time series. Consequently, models trained in this manner exhibit enhanced generalization capability, essential for reliably estimating inflation during interim periods between official monthly releases.[2]

Before fitting any models, *feature selection* is performed to reduce dimensionality and enhance forecast interpretability. High-frequency datasets can include dozens or hundreds of candidate predictors, and incorporating all of them could risk overfitting or obscuring important relationships. Four selection techniques are considered to build corresponding predictor sets $FS^{(1)}$–$FS^{(4)}$:

1. *Correlation filtering*: Retaining predictors whose positive correlation with the monthly target variable exceeds a specified threshold (0.5 in this study). This selects the vari-

[2] Given the interpolation nature of the forecasting exercise, temporal dependencies across contiguous months are less critical compared to a scenario of long-horizon forecasting. Hence, the advantage gained from randomly sampling training observations—achieving broader representativeness and robustness against overfitting—outweighs potential concerns of violating strict temporal ordering, ultimately strengthening the model's ability to deliver unbiased, precise, and consistent intra-month nowcasts.

ables most linearly related to the target.

2. *Principal Component Analysis (PCA)*: Transforming features into a smaller set of orthogonal principal components that capture at least 95% of the total variance. The selected features $FS^{(2)}$ consist of the top $k$ principal component scores, which serve as aggregate factors.
3. *L1 regularization*: Fitting a penalized linear regression of the monthly target variable on all candidate predictors with an $L_1$ penalty and retaining those predictors with non-zero estimated coefficients. The L1 (Lasso) method shrinks less relevant coefficients to zero, effectively performing variable selection.
4. *Random forest importance*: Training a random forest on the training set and computing each variable's importance. Predictors with importance above the percentile 90th are selected, identifying variables that contribute most significantly to prediction in a nonlinear ensemble context.

Each of these methods yields a reduced feature set capturing potentially important signals. By comparing these methods, the approach integrates multiple criteria for predictor relevance (correlation, variance explained, predictive power in linear and nonlinear contexts). This multi-pronged filtering aligns with best practices for enhancing forecast parsimony and robustness, mitigating overfitting, and providing economic interpretability by highlighting key drivers.

For each selected feature set $FS^{(c)}$ (for $c = 1, 2, 3, 4$), a suite of machine learning models is trained on the training portion of the monthly data. Various algorithms commonly used in empirical forecasting are considered to ensure both linear and nonlinear patterns are captured:

1. *Penalized linear regressions*: Ridge, and Lasso, imposing $L_2$, $L_1$ penalties, respectively.
2. *Boosting algorithms*: AdaBoost and Gradient Boosting.
3. *Tree-based ensemble methods*: Random Forest and Extra Trees.
4. *Ensemble model combinations*: Simple or weighted averages of predictions, inspired by the forecast combination approach of Bolivar (2024).

Each model's hyperparameters are tuned using $k$-fold cross-validation on the training set, optimizing for the lowest mean squared error across validation folds. Final model selection is conducted via out-of-sample evaluation on the validation set, with the model (or ensemble) yielding the highest predictive accuracy (lowest MSE or highest $R^2$) selected for generating high-frequency forecasts.

With the *final monthly-trained forecasting model* ($f_{\theta^*}$, representing the best-performing algorithm), high-frequency predictions are produced. Specifically, the forecasting model $f_{\theta^*}$ is applied to high-frequency feature data, with each week's aggregated features serving as input observations. Let $X_w(s)$ denote the feature vector for week $s$ (constructed as described in Step 1). Weekly nowcasts are computed as:

$$\hat{Y}_w(s) = f_{\theta^*}\big(X_w(s)\big),$$

resulting in an estimate of the target variable for week $s$. For daily predictions, the feature vector $X_d(r)$ for day $r$ is similarly constructed, and forecasts are computed as $\hat{Y}_d(r) = f_{\theta^*}(X_d(r))$. The resulting series ($\hat{Y}_w$ or $\hat{Y}_d$) constitute the high-frequency forecasts. In essence, the forecasting model interprets each week (or day) analogously to a monthly observation, providing corresponding predictions. Since Step 1 ensures that daily/weekly inputs are statistically aligned with monthly data, daily/weekly forecasts share the same scale and interpretation as the monthly target variable. Under the maintained assumptions, this ensures that $E[\hat{Y}_w] \approx E[Y_m]$ and $E[\hat{Y}_d] \approx E[Y_m]$, anchoring high-frequency forecasts consistently to monthly values, thereby preventing systematic bias and ensuring forecast reliability over time.

## 2.2 Target Variable

This study implements a two-step methodology to generate high-frequency inflation forecasts, specifically addressing challenges faced by developing economies with limited access to high-frequency data. Bolivia is selected as a representative case study, exemplifying a scenario where official inflation statistics are published exclusively at a monthly frequency.

In Bolivia, inflation is officially measured as the year-on-year (y-o-y) percentage change in the Consumer Price Index (CPI). This index is compiled by the National Institute of Statistics of Bolivia (INE), based on comprehensive weekly price surveys conducted across major demographic and economic centers nationwide (INE, 2016). While INE computes weekly CPI values internally to establish monthly averages, weekly CPI data are not publicly available. Accordingly, the y-o-y CPI inflation rate is chosen as the target variable ($Y_m$).

The observed monthly y-o-y inflation data is utilized exclusively in Step 1 of the methodology to train and validate the monthly-frequency machine learning models. The dataset covers the period from January 2010 to October 2024.

Daily inflation nowcasts are subsequently generated for the period spanning January 1, 2010, to October 30, 2024. Weekly predictions, in contrast, are provided from December 30, 2018, to October 27, 2024, reflecting the availability of Google Trends data, which begins on December 30, 2018 (detailed explanation provided in Subsections 2.3 and 3.5).

## 2.3 Features

A comprehensive set of potential high-frequency inflation predictors (459 features) has been collected and organized into five distinct groups: wholesale prices, Google Trends data, financial variables, commodity prices, and lagged variables. This classification facilitates an integrated analysis, combining global economic insights, local market conditions, and the complex dynamics underlying inflation. A detailed description of these variables is provided in Appendix A.

The data compilation process ensures the construction of consistent time-series datasets at daily, weekly, and monthly intervals, designed explicitly to serve as features within the forecasting methodology.[3]

- *Wholesale Prices*: The dataset incorporates *daily* wholesale price information, obtained from Bolivia's Agro-Environmental and Productive Observatory (OAP), which serves as a key component for early detection of market supply and demand dynamics, thus providing preliminary indicators of inflationary pressures (Cushing and McGarvey, 1990). Detailed product-level price data, including meats, tubers, fruits, vegetables, and agro-industrial goods, are incorporated into the *mixed-frequency dataset*, represented as daily observations ($X_d^{\mathrm{WS}}$) alongside derived weekly ($X_w^{\mathrm{WS}}$) and monthly ($X_m^{\mathrm{WS}}$) averages.
- *Google Trends Data*: The inclusion of Google Trends data tends to enhance the forecasting framework by capturing real-time shifts in consumer sentiment regarding economic indicators and specific commodities (Eugster and Uhl, 2024; Simionescu, 2022; Bulut, 2018). Since Google Trends data is available at a maximum frequency of *weekly*, the dataset incorporates weekly values ($X_w^{\mathrm{GT}}$) and their corresponding monthly averages ($X_m^{\mathrm{GT}}$).
- *Financial Variables*: Given Bolivia's developing economic context and the absence of an extensive domestic stock market, the scope for identifying suitable high-frequency

[3] Superscripts "WS", "GT", "FIN", and "COM" denote subsets of features corresponding respectively to wholesale prices, Google Trends data, financial variables, and commodity prices.

financial indicators related to aggregate price dynamics is limited. Following an exhaustive review, two domestic financial indicators have been identified as relevant predictors: the Housing Development Unit (Unidad de Fomento a la Vivienda, UFV) and the USD/BOB exchange rate (obtained from Google Finance). The UFV, calculated and published *daily* by the Central Bank of Bolivia, tracks inflation explicitly and serves as a reference for domestic financial transactions, contracts, and legal obligations, ensuring value preservation relative to inflation. Concurrently, the *daily* USD/BOB exchange rate influences imported goods and services costs, thereby affecting domestic inflation dynamics.[4]

  Additionally, the *daily* LIBOR rate, an international financial benchmark, is included due to its significant and positive correlation with Bolivia's CPI. Monthly ($X_m^{\mathrm{FIN}}$) and weekly ($X_w^{\mathrm{FIN}}$) averages derived from daily observations ($X_d^{\mathrm{FIN}}$) provide an expanded perspective on the financial factors influencing domestic inflation (Huybens and Smith, 1999) and are added to the *mixed-frequency dataset*.

- *Commodity Prices*: Global economic influences on domestic inflation are captured through *daily* international commodity prices, encompassing oil, metals, and various agricultural products (Altansukh et al., 2017). Derived weekly ($X_w^{\mathrm{COM}}$) and monthly ($X_m^{\mathrm{COM}}$) series generated from daily data ($X_d^{\mathrm{COM}}$) are included in the dataset, providing insights into external price shocks affecting the domestic inflationary environment.
- *Lagged Variables*: To account explicitly for seasonality and persistence inherent in inflation dynamics, lagged values of the target variable—specifically at intervals of 1, 2, 3, 6, 9, and 12 months—are incorporated as potential predictors (Bolivar, 2024). When forecasting at weekly and daily frequencies, lagged values correspond to the y-o-y inflation observed on the last day of the respective lag month. This inclusion strengthens the predictive accuracy by embedding historical inflation trends directly into the forecasting models.

The proposed two-step methodology incorporates monthly ($X_m$), weekly ($X_w$), and daily ($X_d$) feature sets. The monthly feature set spans from January 2010 to October 2024 for Step 1. For Step 2, the weekly dataset covers December 30, 2018, to October 27, 2024, reflecting the availability of Google Trends data (originally weekly), which was obtained only for this period. In contrast, the daily dataset, excluding Google Trends

[4] Although Bolivia maintains a crawling-peg exchange rate regime with a fixed official exchange rate since November 2011, the publicly available USD/BOB rate from Google Finance, reflecting mid-market, spot, or interbank rates, offers greater variability suitable for forecasting purposes.

data, covers January 1, 2010, through October 30, 2024.

## 2.4 Mixed-frequency Dataset and Feature Selection

The mixed-frequency dataset $D_{\text{mixed}}$ consists of the y-o-y CPI inflation target variable (recorded exclusively at month-end) and 459 candidate features previously described in subsection 2.3. The dataset contains a total of 6,277 observations, including 190 monthly observations (month-end from January 2010 to October 2024), 305 weekly observations (week-end from December 30, 2018, to October 27, 2024), and 5,782 daily observations (January 1, 2010, to October 30, 2024). Daily observations exclude Google Trends data.

From this comprehensive dataset $D_{\text{mixed}}$, four distinct subsets of features were derived using alternative feature selection methods:[5]

- *Subset* $FS^{(1)}$: Comprising 41 features exhibiting a positive correlation greater than 0.5 with the target variable, inclusive of all y-o-y inflation lagged features.
- *Subset* $FS^{(2)}$: Containing 85 features, consisting of 79 features obtained from the top principal component scores capturing at least 95% of the total variance, augmented with six y-o-y inflation lagged variables.
- *Subset* $FS^{(3)}$: Comprising 36 features, including 30 features selected through Lasso regularization (L1 penalty), retaining only those predictors with non-zero coefficients, supplemented by six y-o-y inflation lagged variables.
- *Subset* $FS^{(4)}$: Consisting of 54 features, including 48 features selected based on their importance scores surpassing the 90th percentile in a random forest model, along with six y-o-y inflation lagged variables.

Further details regarding the features selected through correlation filtering, principal component analysis (PCA), Lasso regularization, and random forest importance strategies are provided in Appendix B.

## 2.5 Machine Learning Prediction Pipeline

This subsection describes the machine learning prediction pipeline employed to fulfill the research objectives. The selected machine learning algorithms capture both linear and non-linear relationships, and their predictive power is further enhanced through ensem-

[5] The Lasso and random forest models employed for feature selection were trained using the default hyperparameters from the `scikit-learn` package, that is, without fine-tuning.

ble methods to improve CPI inflation forecasting robustness. Hyperparameter tuning, critical for optimizing model performance and generalization capability, is conducted using a 5-fold cross-validation (5F-CV) procedure. Additionally, all features within subsets $FS^{(1)}$–$FS^{(4)}$ (daily, weekly, and monthly frequencies) are standardized using z-score normalization before training, validation and high-frequency forecasting, ensuring comparability across variables and reducing potential biases.

The implemented algorithms and their respective hyperparameter tuning procedures are detailed as follows:

***Penalized Linear Regressions*:**

1. *Ridge Regression* applies an L2 penalty to mitigate overfitting by shrinking coefficient magnitudes. Hyperparameters include the regularization parameter $\lambda$ (ranging from 0.1 to 10 in increments of 0.05) and binary parameters *intercept* and *positive* (evaluated as True or False), optimized through 5F-CV.
2. *Lasso Regression* employs an L1 penalty to promote coefficient sparsity. Its hyperparameters ($\lambda$, *intercept*, and *positive*) share identical ranges and tuning methods as Ridge regression.

***Boosting Algorithms*:**

3. *AdaBoost Regressor* aggregates weak learners iteratively for optimized performance. Hyperparameters comprise learning rate $\alpha$ (0.95 to 1.2 in 0.01 increments), *loss* function (linear, square, exponential), number of estimators $T$ (50 to 200), and tree depth $d$ (1 to 10), optimized via 5F-CV.
4. *Gradient Boosting Regressor* sequentially corrects residuals using weak learners. Hyperparameters include learning rate $\gamma$ (0.01 to 1.0 in 0.01 increments), estimators $T$ (100 to 300), and tree depth $d$ (3 to 10), tuned via 5F-CV.

***Tree-based Ensemble Methods*:**

5. *Random Forest Regressor* combines multiple decision trees to effectively model complex interactions. Hyperparameters (estimators $T$: 100 to 300; tree depth $d$: 3 to 10; *criterion*: squared error, absolute error, Friedman MSE) are optimized to balance accuracy and overfitting risk.
6. *Extra Trees Regressor* utilizes randomized decision trees for enhanced prediction stability. Hyperparameters include estimators $T$ (100 to 300), tree depth $d$ (3 to 10), *criterion* (squared error, absolute error, Friedman MSE), and bootstrapping (True/False), tuned via 5F-CV.

***Ensemble Model Combinations*:** Following Bolivar (2024), two forecast combinations are constructed:

7. *Weighted arithmetic mean (WAM)* of all individual forecasts, weighted inversely by mean squared errors (MSE).
8. *Weighted arithmetic mean of the three best-performing forecasts (WAM-Best)*, similarly weighted by inverse MSE.

All forecasting algorithms (1–8) are applied individually to each of the four feature subsets ($FS^{(1)}$–$FS^{(4)}$) derived through feature selection, conducting training and validation across all combinations.

High-frequency forecasts (weekly and daily) are generated exclusively using the *final monthly-trained forecasting model*, identified as the best-performing model among the 32 evaluated combinations (8 algorithms × 4 feature subsets).

# 3 Empirical Application to Bolivia

## 3.1 Features Distributional Equivalence

A fundamental premise underlying the proposed two-step machine learning methodology is the *distributional equivalence* between aggregated high-frequency (daily and weekly) and monthly feature datasets. This condition ensures that a forecasting model trained using monthly data can be reliably applied to higher-frequency observations without introducing systematic biases or distortions. Formally establishing this equivalence validates that daily and weekly data mirror the underlying data-generating processes of monthly aggregated values, which is critical for extending predictive models from lower-frequency training sets to high-frequency nowcasting.

To rigorously verify this assumption, two-sample Kolmogorov–Smirnov (K–S) tests are conducted for each high-frequency feature, assessing the null hypothesis that the empirical distributions of daily or weekly features coincide with their monthly counterparts. The comprehensive results of these K–S tests are reported in Appendix C. In all examined cases, the null hypothesis of distributional equivalence is not rejected at conventional significance levels (e.g., $p > 0.05$), confirming that high-frequency features are statistically indistinguishable from their monthly aggregates.

These findings substantiate the methodological foundation of utilizing monthly-trained models for accurate and unbiased daily or weekly inflation nowcasting, thereby enhancing the predictive robustness and practical applicability of the proposed forecasting framework.

## 3.2 Final Monthly-Trained Forecasting Model

Hyperparameter fine-tuning significantly enhanced forecasting accuracy across all machine learning algorithms, as documented in Appendix D. Furthermore, feature selection strategies demonstrated notable improvements in predictive performance compared to utilizing the full candidate feature set, as detailed by comparative mean squared error (MSE) analyses presented in Appendix E.

Table 1 summarizes the evaluation metrics (MSE and $R^2$) computed on the validation dataset for each forecasting algorithm across the four distinct feature-selected subsets ($FS^{(1)}$–$FS^{(4)}$). Ridge regression consistently emerged among the top-performing algorithms, reflecting its strength in modeling stable linear relationships inherent within CPI inflation dynamics. Specifically, the Ridge regression model trained on subset $FS^{(3)}$, obtained through L1 regularization, achieved the lowest MSE (0.144) and highest $R^2$ (0.984), thereby representing the most accurate single forecasting model identified.

Table 1: Forecast Evaluation on Validation Set (Out-of-sample)

| Forecast model | $FS^{(1)}$ | | $FS^{(2)}$ | | $FS^{(3)}$ | | $FS^{(4)}$ | |
|---|---|---|---|---|---|---|---|---|
| | MSE | $R^2$ | MSE | $R^2$ | MSE | $R^2$ | MSE | $R^2$ |
| WAM-Best[†] | 0.270 | 0.971 | 0.177 | 0.981 | 0.156 | 0.983 | 0.213 | 0.977 |
| WAM | 0.288 | 0.969 | 0.221 | 0.976 | 0.193 | 0.979 | 0.256 | 0.972 |
| Ridge | 0.300 | 0.967 | 0.180 | 0.980 | 0.144 | 0.984 | 0.198 | 0.978 |
| Lasso | 0.321 | 0.965 | 0.358 | 0.961 | 0.300 | 0.967 | 0.649 | 0.929 |
| ADA | 0.475 | 0.948 | 0.386 | 0.958 | 0.482 | 0.947 | 0.540 | 0.941 |
| GBR | 0.358 | 0.961 | 0.309 | 0.966 | 0.309 | 0.966 | 0.368 | 0.960 |
| RF | 0.404 | 0.956 | 0.466 | 0.949 | 0.404 | 0.956 | 0.527 | 0.943 |
| ET | 0.334 | 0.964 | 0.269 | 0.971 | 0.216 | 0.976 | 0.380 | 0.959 |

Note: WAM and WAM-Best denote Weighted Arithmetic Mean of all individual forecasts, and Weighted Arithmetic Mean of the Three Best-performing individual forecasts, respectively. 1st-best-performing ; 2nd-best-performing ; and 3rd-best-performing .

(†) The three best-performing *individual* algorithms per feature subset ($F^{(c)}$) are: $F^{(1)}$ Ridge, Lasso, and ET; $F^{(2)}$ Ridge, ET, and GBR; $F^{(3)}$ Ridge, ET, and Lasso; $F^{(4)}$ Ridge, GBR, and ET.

The weighted arithmetic mean of the three best-performing individual algorithms

(WAM-Best) also consistently demonstrated high predictive accuracy across subsets, notably achieving an MSE of 0.156 and $R^2$ of 0.983 within subset $F^{(3)}$. Nevertheless, Ridge regression alone provided superior predictive performance, underscoring the efficacy of a carefully regularized linear model in capturing the core inflationary signals embedded within the selected features.

The following insights emerge from this evaluation:

- Ridge regression consistently exhibits robust predictive performance across different feature subsets, underscoring its ability to generalize effectively and model stable linear relationships present within Bolivia's CPI inflation data.
- Feature selection markedly reduces forecasting errors compared to using the full feature set, highlighting its effectiveness in isolating economically meaningful predictors and minimizing noise-driven variability.
- Among feature selection methods, the $F^{(3)}$ subset derived via L1 regularization consistently achieves the highest forecast accuracy. This highlights the value of enforcing sparsity and explicitly penalizing irrelevant or redundant features, ultimately resulting in improved inflation predictions.

In light of these evaluations, the *final monthly-trained forecasting model* selected for subsequent weekly and daily inflation predictions is Ridge regression utilizing the $F^{(3)}$ feature subset. This choice reflects an optimal balance of predictive accuracy, interpretability, and statistical robustness, thus enabling reliable, unbiased, and consistent high-frequency inflation forecasts suitable for policy analysis and economic decision-making.

## 3.3 Two-Step Machine Learning Methodology vs. Econometric Approaches

This subsection presents a rigorous comparative analysis between the forecasting accuracy of the proposed two-step machine learning methodology and traditional econometric techniques widely used in mixed-frequency data nowcasting. Specifically, the predictive performance of the final monthly-trained forecasting model (Ridge algorithm with the $FS^{(3)}$ subset) is benchmarked against the Bridge Equation and Mixed-Data Sampling (MIDAS) regression approaches.

The *Bridge Equation* method is a two-stage procedure: first, daily and weekly high-frequency indicators ($X^{HF}$) are aggregated into monthly averages ($\bar{X}_m^{HF}$) to align temporally with the monthly-frequency target variable ($Y_m$). In the second step, a standard linear regression relates these aggregated indicators to the monthly target:

$$Y_m = \alpha + \beta' \bar{X}_m^{HF} + \epsilon_m, \tag{3}$$

where $\alpha$ denotes the intercept, $\beta$ is the vector of regression coefficients, and $\epsilon_m$ represents the residual error term. Parameters estimated using historical monthly data enable subsequent generation of weekly and daily nowcasts by inserting respective aggregated predictor data into equation (3).

Alternatively, the *MIDAS regression* explicitly models data at their original sampling frequencies without requiring temporal aggregation. The MIDAS specification is defined as:

$$Y_m = \alpha + \sum_{k=0}^{K} \beta_k(\theta) X_{t-k}^{HF} + u_m, \tag{4}$$

where $Y_m$ is the monthly-frequency target, $X_t^{HF}$ indicates daily or weekly predictors, and $\beta_k(\theta)$ is a lag-weighting polynomial commonly parametrized through exponential Almon polynomials:

$$\beta_k(\theta) = \frac{\exp(\theta_1 k + \theta_2 k^2)}{\sum_{j=0}^{K} \exp(\theta_1 j + \theta_2 j^2)}, \quad k = 0, 1, \ldots, K. \tag{5}$$

The MIDAS parameters $\theta = (\theta_1, \theta_2)$, capturing decay and curvature of lag impacts, are estimated via nonlinear least squares (NLS).

To ensure methodological consistency, both econometric approaches employ the identical feature subset ($FS^{(3)}$), previously identified by L1 regularization as optimally informative, and utilize the same training dataset employed by the final monthly-trained forecasting model. Table 2 presents the Relative Mean Squared Error (Rel-MSE) computed as:

$$\text{Rel-MSE} = \frac{MSE_i^{FS^{(3)}}}{MSE_{\text{Ridge}}^{FS^{(3)}}}, \quad i \in \{\text{Bridge, MIDAS}\},$$

allowing a direct comparative evaluation of predictive performance on the identical out-of-sample validation set used for the machine learning model.

Table 2: Relative-MSE on Validation Set (Out-of-Sample): Econometric Approaches vs. Two-Step ML Methodology

| Forecasting Model | Rel-MSE |
|---|---|
| Two-Step ML Final Model | 1.00 |
| Bridge Equation | 1.21 |
| MIDAS Regression | 1.19 |

*Note*: The Two-Step ML Final Model refers to Ridge regression trained on the L1-regularization feature-selected subset ($FS^{(3)}$).

Table 2 demonstrates a forecasting advantage of the proposed Two-Step Machine Learning methodology. Specifically, the final monthly-trained ML model achieves a lower MSE relative to both econometric benchmarks, outperforming the Bridge Equation and MIDAS models by approximately 21% and 19%, respectively. These findings highlight the efficacy of systematically integrating feature selection and hyperparameter optimization inherent in the machine learning pipeline, thus underscoring the practical utility and superior predictive capability of the proposed methodological framework in high-frequency inflation forecasting contexts

## 3.4 Daily Forecast

Following the two-step forecasting methodology, the Ridge regression trained on the $F^{(3)}$ subset (selected via L1 regularization) serves as the final monthly-trained model. This subsection details the application of this selected model for generating daily y-o-y inflation forecasts.

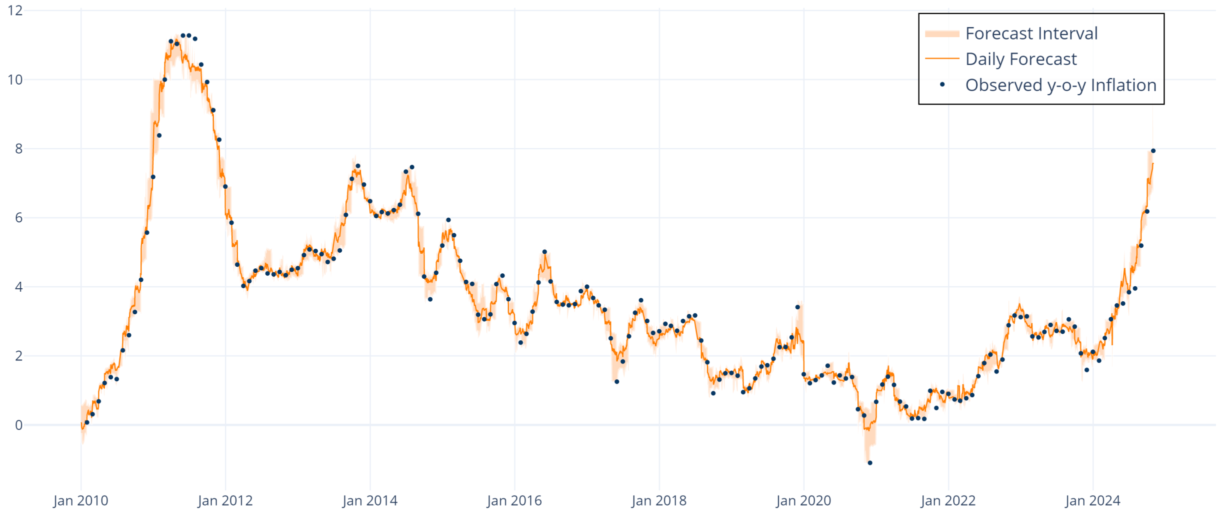
The daily forecast covers the period from January 1, 2010, to October 30, 2024.[6] The principal aim is to demonstrate the model's capacity to support real-time inflation nowcasting, perform within-period interpolation, and act as a data-augmentation strategy. By producing daily estimates that are statistically consistent with the monthly CPI inflation series, the model effectively bridges temporal gaps in official reporting, offering enhanced granularity and timeliness for economic analysis and policy response.

Daily forecasts, depicted in Figure 2, consistently replicate the overall trends and cyclical patterns exhibited by the official monthly inflation reports. Notably, the fore-

[6] Daily forecasts are generated excluding Google Trends data originally included in $F^{(3)}$, while maintaining the fine-tuned hyperparameters from the final monthly-trained forecasting model. This approach enables coverage of the period from January 1, 2010, to October 30, 2024.

casts capture meaningful within-month fluctuations, which serve as early indicators of potential inflationary changes. Such daily granularity is instrumental for policymakers and economic analysts aiming to respond swiftly and appropriately to emerging inflationary pressures.

Figure 2: Daily y-o-y Inflation Forecast



The forecast interval (shaded area) is determined by the maximum and minimum daily predictions generated by the three best-performing algorithms within the $F^{(3)}$ feature subset. This interval serves as a reference for the range of variability across individual model predictions.

Given the methodological design, this model facilitates daily real-time updates, enabling inflation nowcasts up to one month ahead or until the subsequent official monthly inflation data release. This forecasting capacity is especially valuable for detecting early signs of inflationary disturbances, thus allowing policymakers to consider proactive interventions. For example, an unexpected daily uptick in forecasted inflation could prompt immediate policy evaluations or adjustments, mitigating inflationary pressures before monthly data confirmation.

The advantages of employing daily inflation forecasts, relative to traditional monthly approaches, include:

- *Timeliness*: Daily data provides immediate detection of inflation dynamics, essential for rapid policy responses.
- *Granularity*: Within-month forecasts expose transient shocks or short-term trends potentially obscured by monthly aggregates.
- *Enhanced Precision*: Increased forecasting frequency allows a deeper and more pre-

cise understanding of inflationary processes, thereby refining policy formulation.

Overall, transitioning to daily inflation forecasts leverages the methodological strengths presented in this study, enabling detailed and proactive insights into Bolivia's inflation dynamics. This approach facilitates informed, timely, and effective policy interventions, and is similarly applicable to other economies facing comparable data constraints.

For visualization of daily CPI year-on-year inflation nowcasts obtained using the best-performing forecasting models from feature subsets $F^{(1)}$, $F^{(2)}$, and $F^{(4)}$, see Appendix F.

Lastly, the two-sample Kolmogorov–Smirnov (K–S) test is conducted to statistically validate the distributional equivalence between the daily inflation nowcasts and the observed monthly CPI y-o-y inflation. This validation step is essential since the methodological validity of applying monthly-trained algorithms for high-frequency forecasting relies on this assumption. The K–S test results yield a p-value of 0.76, which is sufficiently large to preclude rejection of the null hypothesis stating that both distributions are statistically indistinguishable. Additionally, as thoroughly documented in Appendix C, predictor features similarly exhibit statistical equivalence across daily, weekly, and monthly frequencies, further reinforcing the methodological soundness of frequency aggregation and interpolation. These findings robustly support the appropriateness of utilizing monthly-trained forecasting models for generating reliable daily inflation predictions. Appendix G provides graphical evidence illustrating the substantial overlap and similarity between the empirical distributions of observed monthly and daily predicted y-o-y CPI inflation.

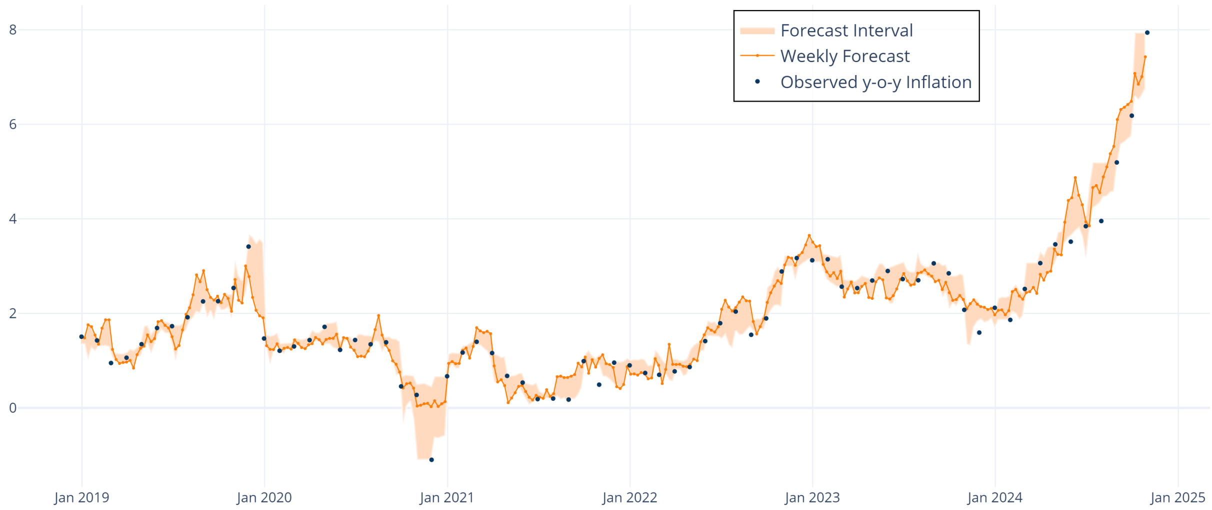
## 3.5 Weekly Forecast

Complementing the daily inflation nowcasts, this subsection presents weekly forecasts covering the period from December 30, 2018, to October 27, 2024. In contrast to daily forecasts, which provide within-month interpolations across the entire original monthly-frequency sample (January 2010–October 2024), weekly predictions illustrate a scenario where certain predictors are only available at a weekly frequency. In this study, this limitation specifically arises due to the incorporation of Google Trends data, which are only available at weekly intervals and having a limited forecasting sample.

Concretely, the weekly inflation nowcasts utilize the same final monthly-trained Ridge regression model derived from the $F^{(3)}$ subset; the only distinction is the inclusion of Google Trends features at their original weekly frequency. Therefore, the weekly nowcasts not only add the predictive signals of Google Trends data but also reflect a practical forecasting scenario pertinent to contexts where daily high-frequency data are either unavailable or incomplete.

Figure 3 presents the weekly CPI y-o-y inflation forecasts, demonstrating robust alignment with observed monthly inflation trends, while simultaneously capturing more granular fluctuations within each month. The weekly forecasts thus serve as a pragmatic benchmark, enabling early detection of short-term inflation dynamics and offering policymakers valuable insights for timely economic interventions.

Figure 3: Weekly y-o-y Inflation Forecast



Note: Forecast intervals are defined by the maximum and minimum weekly predictions generated by the three best-performing algorithms using the $F^{(3)}$ subset, inclusive of weekly Google Trends data. This interval serves as a reference for the range of variability across individual model predictions.

In summary, weekly forecasts enrich the analytical perspective offered by this study's methodology, highlighting its flexibility in adapting to varying data frequency conditions prevalent in developing economies.

Weekly CPI y-o-y inflation forecasts obtained from the best-performing models corresponding to feature subsets $F^{(1)}$, $F^{(2)}$, and $F^{(4)}$ are presented in Appendix F.

Furthermore, the K–S test yields a p-value of 0.82 when comparing the observed monthly and weekly predicted y-o-y CPI inflation distributions. This high p-value indicates no statistically significant difference between the two distributions, thereby reinforcing the assumption of distributional equivalence. Appendix G presents visualization of the substantial overlap and close alignment between the observed monthly and weekly predicted inflation distributions.

# 4 Conclusions

This paper introduces a novel two-step machine learning (ML) framework for generating high-frequency (daily and weekly) forecasts of year-on-year CPI inflation in data-scarce environments. The proposed methodology is tailored to developing economies, where official inflation statistics are typically published at monthly intervals and with reporting lags. In this context, high-frequency forecasting is interpreted as both nowcasting—real-time estimation ahead of official data release—and interpolation, whereby intra-month inflation estimates are derived from available high-frequency signals. The method can also be viewed as a data augmentation strategy, enriching the informational granularity of monthly datasets.

The framework is operationalized through a sequential structure. First, high-frequency indicators are aggregated into monthly equivalents and validated for statistical alignment with monthly inflation data. Then, supervised ML models are trained on the aggregated monthly dataset and subsequently applied to generate high-frequency forecasts. Applied to Bolivia as a case study, the approach demonstrates strong predictive performance. The final model—a Ridge regression trained on a Lasso-selected subset of features ($FS^{(3)}$)—achieves the lowest mean squared error on the validation set. The model's daily and weekly predictions not only align closely with official inflation trends but also capture within-month dynamics, offering granular early-warning signals to policymakers.

Methodological robustness is reinforced through formal statistical testing. Distributional equivalence between observed monthly inflation values and the predicted high-frequency series is confirmed by Kolmogorov–Smirnov (K–S) tests, validating the assumption that monthly-trained models can be reliably used to infer higher-frequency behavior. Additional K–S tests show that features across daily, weekly, and monthly frequencies follow statistically indistinguishable distributions once properly aggregated. Moreover, the proposed Two-Step ML methodology outperforms benchmark econometric models—namely, Bridge Equations and MIDAS regressions—under the same validation protocol, underscoring its comparative advantage in handling mixed-frequency inflation data.

The findings offer several practical implications. For policymakers and analysts in developing economies, the framework provides a scalable, interpretable, and data-efficient solution to monitor inflation in near real-time. This improved temporal resolution can support faster policy responses to inflationary shocks and enhance macroeconomic surveillance systems.

Nevertheless, the study is subject to some limitations. First, the methodology assumes a stable data-generating process across time and frequencies—an assumption that may be violated under structural breaks or abrupt policy shifts. Second, although the two-step framework addresses frequency alignment and reduces model complexity, it does not explicitly model nonlinear or lagged interactions across frequencies. Third, while feature selection enhances parsimony and forecasting accuracy, it may limit economic interpretability, particularly when data-driven algorithms prioritize statistical relevance over theoretical clarity.

Future research may expand the utility of the proposed framework along several dimensions. Applying the methodology to nowcast other macroeconomic indicators—such as GDP, industrial production, or exchange rates—would offer a robust test of its generalizability. Beyond real-time forecasting, the framework also lends itself to interpolation and imputation tasks, serving as a data-augmentation strategy for reconstructing high-frequency series from lower-frequency aggregates or filling in missing observations. This is particularly relevant in settings where data gaps or limited temporal granularity constrain empirical analysis and hinder timely policy responses. Further enhancements could include incorporating structural break detection, dynamic model updating, and the integration of richer data sources such as financial transactions, mobility patterns, or sentiment indicators to improve accuracy and adaptability. Finally, cross-country applications in similarly data-constrained environments would facilitate external validation and offer comparative insights into the broader applicability of the proposed methodology.

### Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author used ChatGPT in order to improve language and readability. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the content of the publication.

## References

Altansukh, G., Becker, R., Bratsiotis, G., and Osborn, D. R. (2017). What is the globalisation of inflation? *Journal of Economic Dynamics and Control*, 74:1–27.

Baffigi, A., Golinelli, R., and Parigi, G. (2004). Bridge models to forecast the euro area gdp. *International Journal of forecasting*, 20(3):447–460.

Bernanke, B. S. and Woodford, M. (1997). Inflation forecasts and monetary policy. *Journal of Money, Credit and Banking*, 29(4):653.

Bolivar, O. (2024). Gdp nowcasting: A machine learning and remote sensing data-based approach for bolivia. *Latin American Journal of Central Banking*, 5(3):100126.

Bulut, L. (2018). Google trends and the forecasting performance of exchange rate models. *Journal of Forecasting*, 37(3):303–315.

Clements, M. P. and Galvao, A. B. (2008). Macroeconomic forecasting with mixed-frequency data: Forecasting output growth in the united states. *Journal of Business & Economic Statistics*, 26(4):546–554.

Cushing, M. J. and McGarvey, M. G. (1990). Feedback between wholesale and consumer price inflation: A reexamination of the evidence. *Southern Economic Journal*, pages 1059–1072.

Eugster, P. and Uhl, M. W. (2024). Forecasting inflation using sentiment. *Economics Letters*, page 111575.

Faust, J. and Wright, J. H. (2013). Forecasting inflation. *Handbook of Economic Forecasting*, 2:2–56.

Ghysels, E. and Marcellino, M. (2018). *Applied economic forecasting using time series methods*. Oxford University Press.

Ghysels, E., Sinko, A., and Valkanov, R. (2007). Midas regressions: Further results and new directions. *Econometric Reviews*, 26(1):53–90.

Goulet Coulombe, P., Leroux, M., Stevanovic, D., and Surprenant, S. (2022). How is machine learning useful for macroeconomic forecasting? *Journal of Applied Econometrics*, 37(5):920–964.

Huybens, E. and Smith, B. D. (1999). Inflation, financial markets and long-run real activity. *Journal of monetary economics*, 43(2):283–315.

INE (2016). Índice de precios al consumidor: Documento metodológico. Technical report, Instituto Nacional de Estadística.

Masini, R. P., Medeiros, M. C., and Mendes, E. F. (2023). Machine learning advances for time series forecasting. *Journal of economic surveys*, 37(1):76–111.

Medeiros, M. C., Vasconcelos, G. F., Veiga, Á., and Zilberman, E. (2021). Forecasting inflation in a data-rich environment: the benefits of machine learning methods. *Journal of Business & Economic Statistics*, 39(1):98–119.

Moshiri, S. and Cameron, N. (2000). Neural network versus econometric models in forecasting inflation. *Journal of forecasting*, 19(3):201–217.

Simionescu, M. (2022). Econometrics of sentiments-sentometrics and machine learning: the improvement of inflation predictions in romania using sentiment analysis. *Technological Forecasting and Social Change*, 182:121867.

Stock, J. H. and Watson, M. W. (1999). Forecasting inflation. *Journal of Monetary Economics*, 44(2):293–335.

Varian, H. R. (2014). Big data: New tricks for econometrics. *Journal of economic perspectives*, 28(2):3–28.

# A Potential Features for High-frequency Inflation Forecasting

| Group | Variables | Source |
|---|---|---|
| Wholesale prices (422 features) | **Products:** apple, banana (2 types), bean, beef, carrot, cassava, chicken, chili, corn (2 types), flour (2 types), grapefruit, greenbean, lard, lemon, milk (2 types), noodle, oil (2 types), onion (2 types), orange (2 types), potato (2 types), papaya, paprika, peas, pineapple, quinoa, redpepper, rice (4 types), sorghum, soy, squash, sugar, tomato, watermelon, and wheat. They are collected by cities: La Paz (LP), Santa Cruz (SC), Cochabamba (CB), Sucre (SU), Oruro (OR), Potosí (PO), Tarija (TJ), Trinidad (TR), Cobija (CO); and aggregated at a national level (BO). | Agro-Environmental and Productive Observatory |
| Google Trends[†] (17 features) | **Bolivia's searches related to inflation and prices:** Inflación; demanda; desempleo; dinero; economia; ine; ine bolivia; inflacion; inflacion en bolivia; inflación bolivia; ipc; la inflacion; la inflación; pib; pib bolivia; que es inflacion; que es pib; | Google Trends |
| Financial | Housing Development Unit.<br>USD/BOB exchange rate.<br>London Interbank Offered Rate (LIBOR). | Central Bank of Bolivia<br>Google Finance<br>Bloomberg |
| Commodity prices (11 features) | WTI oil, natural gas, gold, silver, zinc, tin, soybean, soybean meal, soybean oil, lead, and copper. | Bloomberg, FRED, Trading Economics |

(†) Google Trends' variables are named as their original Spanish search.

# B Feature Selection Variables

| Subset | Features |
|---|---|
| Correlation-based ($FS^{(1)}$) | lag 1, lag 2, lag 3, sugar TJ, sugar BO, chicken TJ, sugar PO, sugar CB, sugar LP, sugar OR, oil2 BO, sugar SC, lag 6, lard CO, oil2 TJ, rice4 SU, squash CO, watermelon PO, oil2 OR, peas CO, onion2 TJ, oil2 SU, oil SU, sugar SU, apple TR, chicken SC, lard OR, oil2 PO, oil BO, oil PO, oil2 SC, redpepper CO, silver, rice4 LP, rice4 TJ, banana2 CO, oil SC, chicken CO, pineapple CO, corn2 TJ |
| PCA ($FS^{(2)}$) | lag 1, lag 2, lag 3, lag 6, lag 9, lag 12, PC 1, PC 2, PC 3, PC 4, PC 5, PC 6, PC 7, PC 8, PC 9, PC 10, PC 11, PC 12, PC 13, PC 14, PC 15, PC 16, PC 17, PC 18, PC 19, PC 20, PC 21, PC 22, PC 23, PC 24, PC 25, PC 26, PC 27, PC 28, PC 29, PC 30, PC 31, PC 32, PC 33, PC 34, PC 35, PC 36, PC 37, PC 38, PC 39, PC 40, PC 41, PC 42, PC 43, PC 44, PC 45, PC 46, PC 47, PC 48, PC 49, PC 50, PC 51, PC 52, PC 53, PC 54, PC 55, PC 56, PC 57, PC 58, PC 59, PC 60, PC 61, PC 62, PC 63, PC 64, PC 65, PC 66, PC 67, PC 68, PC 69, PC 70, PC 71, PC 72, PC 73, PC 74, PC 75, PC 76, PC 77, PC 78, PC 79 |
| L1 regularization ($FS^{(3)}$) | lag 1, lag 2, lag 3, lag 6, lag 9, lag 12, oil CB, sugar SC, milk2 CB, watermelon PO, flour LP, chicken TJ, lard CO, onion2 TJ, corn2 TJ, banana2 CO, oil SU, chicken CO, rice CB, *inflacion*, tomato TJ, chicken TR, papa2 TJ, silver, sugar TJ, chicken CB, squash CO, peas TJ, rice3 PO, peas SU, papaya SU, carrot TJ, bean CB, yuca LP |
| Random Forest ($FS^{(4)}$) | lag 1, lag 2, lag 3, lag 6, lag 9, lag 12, oil CB, sugar BO, chicken TJ, oil2 TJ, oil2 BO, sugar TJ, rice4 TJ, sugar CB, rice3 PO, sugar OR, sugar SC, rice4 SU, corn2 TJ, chicken BO, papa2 TJ, oil2 PO, sugar LP, papaya TR, oil SC, banana CO, rice BO, milk SC, chicken TR, pineapple TR, rice2 LP, rice3 SU, ufv, paprika OR, squash CO, potato1 PO, rice2 PO, flour LP, chicken PO, silver, chicken LP, corn2 SC, chicken SC, tomato CO, libor, banana2 CO, sugar PO, carrot CO, tomato TJ, oil TR, lard LP, rice4 OR |

Note: WS products with a number at the end indicate a different type of such products.

# C Kolmogorov–Smirnov Test on Features

| Grp. | Feature | pval | Grp. | Feature | pval | Grp. | Feature | pval | Grp. | Feature | pval |
|---|---|---|---|---|---|---|---|---|---|---|---|
| WS | beef LP | 0.962 | WS | lemon LP | 0.499 | WS | apple LP | 0.532 | WS | onion LP | 0.661 |
| WS | beef CB | 0.858 | WS | lemon CB | 0.461 | WS | apple CB | 0.642 | WS | onion CB | 0.850 |
| WS | beef SC | 0.981 | WS | lemon SC | 0.272 | WS | apple SC | 0.999 | WS | onion SC | 0.456 |
| WS | beef OR | 0.927 | WS | lemon OR | 0.403 | WS | apple OR | 0.517 | WS | onion OR | 0.892 |
| WS | beef PO | 0.998 | WS | lemon PO | 0.480 | WS | apple PO | 0.101 | WS | onion PO | 0.403 |
| WS | beef SU | 0.818 | WS | lemon SU | 0.909 | WS | apple SU | 0.972 | WS | onion SU | 0.798 |
| WS | beef TJ | 1.000 | WS | lemon TJ | 0.564 | WS | apple TJ | 0.784 | WS | onion TJ | 0.749 |
| WS | beef TR | 1.000 | WS | lemon TR | 0.699 | WS | apple TR | 0.999 | WS | onion TR | 0.432 |
| WS | beef CO | 1.000 | WS | lemon CO | 0.444 | WS | apple CO | 0.992 | WS | onion CO | 0.979 |
| WS | beef BO | 0.883 | WS | lemon BO | 0.504 | WS | apple BO | 0.149 | WS | onion BO | 0.929 |
| WS | chicken LP | 0.722 | WS | orange LP | 0.127 | WS | grapefruit LP | 0.519 | WS | onion2 LP | 0.865 |
| WS | chicken CB | 0.387 | WS | orange CB | 0.895 | WS | grapefruit CB | 0.814 | WS | onion2 CB | 0.014 |
| WS | chicken SC | 0.285 | WS | orange SC | 0.924 | WS | grapefruit SC | 0.335 | WS | onion2 SC | 0.920 |
| WS | chicken OR | 0.878 | WS | orange OR | 0.409 | WS | grapefruit OR | 0.767 | WS | onion2 OR | 1.000 |
| WS | chicken PO | 0.836 | WS | orange PO | 0.194 | WS | grapefruit PO | 0.983 | WS | onion2 PO | 1.000 |
| WS | chicken SU | 0.770 | WS | orange SU | 0.272 | WS | grapefruit SU | 0.801 | WS | onion2 SU | 1.000 |
| WS | chicken TJ | 0.155 | WS | orange TJ | 0.521 | WS | grapefruit TJ | 0.845 | WS | onion2 TJ | 1.000 |
| WS | chicken TR | 0.515 | WS | orange TR | 0.044 | WS | grapefruit TR | 0.474 | WS | onion2 TR | 1.000 |
| WS | chicken CO | 0.987 | WS | orange CO | 0.440 | WS | grapefruit CO | 0.694 | WS | onion2 CO | 0.437 |
| WS | chicken BO | 0.851 | WS | orange BO | 0.178 | WS | grapefruit BO | 0.210 | WS | onion2 BO | 0.287 |
| WS | papa1 LP | 0.996 | WS | papaya LP | 0.664 | WS | orange2 LP | 0.003 | WS | bean LP | 0.718 |
| WS | papa1 CB | 0.734 | WS | papaya CB | 0.409 | WS | orange2 CB | 0.953 | WS | bean CB | 0.307 |
| WS | papa1 SC | 0.769 | WS | papaya SC | 0.005 | WS | orange2 SC | 0.356 | WS | bean SC | 0.005 |
| WS | papa1 OR | 0.477 | WS | papaya OR | 0.591 | WS | orange2 OR | 0.001 | WS | bean OR | 0.092 |
| WS | papa1 PO | 0.833 | WS | papaya PO | 0.412 | WS | orange2 PO | 0.035 | WS | bean PO | 0.212 |
| WS | papa1 SU | 0.922 | WS | papaya SU | 0.507 | WS | orange2 SU | 0.613 | WS | bean SU | 0.126 |
| WS | papa1 TJ | 0.827 | WS | papaya TJ | 0.247 | WS | orange2 TJ | 0.964 | WS | bean TJ | 0.550 |
| WS | papa1 TR | 0.785 | WS | papaya TR | 1.000 | WS | orange2 TR | 0.997 | WS | bean TR | 0.050 |
| WS | papa1 CO | 1.000 | WS | papaya CO | 0.449 | WS | orange2 CO | 0.707 | WS | bean CO | 0.904 |
| WS | papa1 BO | 0.823 | WS | papaya BO | 0.062 | WS | orange2 BO | 0.096 | WS | bean BO | 0.085 |
| WS | papa2 LP | 0.946 | WS | pineapple LP | 0.126 | WS | tomato LP | 0.346 | WS | chili LP | 0.142 |
| WS | papa2 CB | 0.922 | WS | pineapple CB | 0.272 | WS | tomato CB | 0.128 | WS | chili CB | 0.230 |
| WS | papa2 SC | 0.660 | WS | pineapple SC | 0.097 | WS | tomato SC | 0.008 | WS | chili SC | 0.008 |
| WS | papa2 OR | 0.759 | WS | pineapple OR | 0.754 | WS | tomato OR | 0.120 | WS | chili OR | 0.137 |
| WS | papa2 PO | 0.397 | WS | pineapple PO | 0.440 | WS | tomato PO | 0.156 | WS | chili PO | 0.021 |
| WS | papa2 SU | 0.373 | WS | pineapple SU | 0.433 | WS | tomato SU | 0.101 | WS | chili SU | 0.614 |
| WS | papa2 TJ | 0.113 | WS | pineapple TJ | 0.257 | WS | tomato TJ | 0.024 | WS | chili TJ | 0.213 |
| WS | papa2 TR | 0.438 | WS | pineapple TR | 0.995 | WS | tomato TR | 0.832 | WS | chili TR | 0.311 |
| WS | papa2 CO | 0.671 | WS | pineapple CO | 1.000 | WS | tomato CO | 0.999 | WS | chili CO | 0.997 |
| WS | papa2 BO | 0.883 | WS | pineapple BO | 0.300 | WS | tomato BO | 0.187 | WS | chili BO | 0.011 |
| WS | yuca LP | 0.474 | WS | platano LP | 0.252 | WS | redpepper LP | 0.995 | WS | corn LP | 0.411 |
| WS | yuca CB | 0.175 | WS | platano CB | 0.272 | WS | redpepper CB | 0.998 | WS | corn CB | 0.427 |
| WS | yuca SC | 0.476 | WS | platano SC | 0.331 | WS | redpepper SC | 1.000 | WS | corn SC | 0.490 |
| WS | yuca OR | 0.099 | WS | platano OR | 0.084 | WS | redpepper OR | 0.977 | WS | corn OR | 0.289 |
| WS | yuca PO | 1.000 | WS | platano PO | 0.295 | WS | redpepper PO | 1.000 | WS | corn PO | 0.740 |
| WS | yuca SU | 0.919 | WS | platano SU | 0.611 | WS | redpepper SU | 0.934 | WS | corn SU | 0.738 |
| WS | yuca TJ | 1.000 | WS | platano TJ | 0.671 | WS | redpepper TJ | 1.000 | WS | corn TJ | 0.967 |
| WS | yuca TR | 0.916 | WS | platano TR | 0.633 | WS | redpepper TR | 1.000 | WS | corn TR | 0.989 |
| WS | yuca CO | 0.239 | WS | platano CO | 0.078 | WS | redpepper CO | 1.000 | WS | corn CO | 0.423 |
| WS | yuca BO | 0.034 | WS | platano BO | 0.042 | WS | redpepper BO | 0.802 | WS | corn BO | 0.324 |
| WS | banana LP | 0.063 | WS | watermelon LP | 0.243 | WS | peas LP | 0.507 | WS | paprika LP | 0.248 |
| WS | banana CB | 0.153 | WS | watermelon CB | 0.791 | WS | peas CB | 0.026 | WS | paprika CB | 0.102 |
| WS | banana SC | 0.025 | WS | watermelon SC | 0.870 | WS | peas SC | 0.004 | WS | paprika SC | 0.125 |
| WS | banana OR | 0.003 | WS | watermelon OR | 0.189 | WS | peas OR | 0.162 | WS | paprika OR | 0.354 |
| WS | banana PO | 0.107 | WS | watermelon PO | 0.677 | WS | peas PO | 0.106 | WS | paprika PO | 0.133 |
| WS | banana SU | 0.353 | WS | watermelon SU | 0.838 | WS | peas SU | 0.110 | WS | paprika SU | 0.297 |
| WS | banana TJ | 0.698 | WS | watermelon TJ | 0.898 | WS | peas TJ | 0.175 | WS | paprika TJ | 0.448 |
| WS | banana TR | 0.935 | WS | watermelon TR | 1.000 | WS | peas TR | 0.112 | WS | paprika TR | 0.997 |
| WS | banana CO | 0.994 | WS | watermelon CO | 0.989 | WS | peas CO | 0.987 | WS | paprika CO | 0.849 |
| WS | banana BO | 0.147 | WS | watermelon BO | 0.112 | WS | peas BO | 0.036 | WS | paprika BO | 0.059 |

| Grp. | Feature | pval | Grp. | Feature | pval | Grp. | Feature | pval | Grp. | Feature | pval |
|---|---|---|---|---|---|---|---|---|---|---|---|
| WS | greenbean LP | 0.233 | WS | rice2 BO | 0.990 | WS | sugar CO | 1.000 | WS | milk2 TR | 1.000 |
| WS | greenbean CB | 0.050 | WS | rice3 LP | 0.987 | WS | sugar BO | 0.931 | WS | milk2 CO | 0.640 |
| WS | greenbean SC | 0.000 | WS | rice3 CB | 0.995 | WS | noodle LP | 0.968 | WS | milk2 BO | 0.149 |
| WS | greenbean OR | 0.004 | WS | rice3 SC | 0.997 | WS | noodle CB | 1.000 | WS | lard LP | 0.925 |
| WS | greenbean PO | 0.027 | WS | rice3 OR | 0.454 | WS | noodle SC | 1.000 | WS | lard CB | 1.000 |
| WS | greenbean SU | 0.049 | WS | rice3 PO | 0.933 | WS | noodle OR | 0.389 | WS | lard SC | 0.967 |
| WS | greenbean TJ | 0.765 | WS | rice3 SU | 1.000 | WS | noodle PO | 0.969 | WS | lard OR | 0.997 |
| WS | greenbean TR | 0.009 | WS | rice3 TJ | 0.992 | WS | noodle SU | 0.803 | WS | lard PO | 0.945 |
| WS | greenbean CO | 0.797 | WS | rice3 TR | 1.000 | WS | noodle TJ | 0.690 | WS | lard SU | 1.000 |
| WS | greenbean BO | 0.036 | WS | rice3 CO | 1.000 | WS | noodle TR | 1.000 | WS | lard TJ | 0.659 |
| WS | squash LP | 0.497 | WS | rice3 BO | 0.686 | WS | noodle CO | 0.097 | WS | lard TR | 1.000 |
| WS | squash CB | 0.172 | WS | rice4 LP | 0.995 | WS | noodle BO | 0.960 | WS | lard CO | 0.993 |
| WS | squash SC | 0.160 | WS | rice4 CB | 1.000 | WS | flour LP | 0.998 | WS | lard BO | 0.893 |
| WS | squash OR | 0.086 | WS | rice4 SC | 0.963 | WS | flour CB | 1.000 | WS | veglard LP | 0.892 |
| WS | squash PO | 0.021 | WS | rice4 OR | 1.000 | WS | flour SC | 1.000 | WS | veglard CB | 1.000 |
| WS | squash SU | 0.078 | WS | rice4 PO | 1.000 | WS | flour OR | 0.977 | WS | veglard SC | 0.997 |
| WS | squash TJ | 0.155 | WS | rice4 SU | 1.000 | WS | flour PO | 1.000 | WS | veglard OR | 0.086 |
| WS | squash TR | 1.000 | WS | rice4 TJ | 1.000 | WS | flour SU | 1.000 | WS | veglard PO | 0.929 |
| WS | squash CO | 0.997 | WS | rice4 TR | 1.000 | WS | flour TJ | 1.000 | WS | veglard SU | 0.887 |
| WS | squash BO | 0.058 | WS | rice4 CO | 0.996 | WS | flour TR | 0.998 | WS | veglard TJ | 0.961 |
| WS | carrot LP | 0.345 | WS | rice4 BO | 0.395 | WS | flour CO | 1.000 | WS | veglard TR | 1.000 |
| WS | carrot CB | 0.399 | WS | oil LP | 0.999 | WS | flour BO | 0.988 | WS | veglard CO | 1.000 |
| WS | carrot SC | 0.274 | WS | oil CB | 1.000 | WS | flour2 LP | 0.697 | WS | veglard BO | 0.951 |
| WS | carrot OR | 0.570 | WS | oil SC | 1.000 | WS | flour2 CB | 0.987 | WS | ycorn LP | 0.987 |
| WS | carrot PO | 0.210 | WS | oil OR | 1.000 | WS | flour2 SC | 0.899 | WS | ycorn CB | 0.949 |
| WS | carrot SU | 0.177 | WS | oil PO | 1.000 | WS | flour2 OR | 0.824 | WS | ycorn SC | 0.918 |
| WS | carrot TJ | 0.300 | WS | oil SU | 0.999 | WS | flour2 PO | 0.787 | WS | ycorn OR | 0.843 |
| WS | carrot TR | 0.035 | WS | oil TJ | 1.000 | WS | flour2 SU | 0.999 | WS | ycorn PO | 0.987 |
| WS | carrot CO | 0.670 | WS | oil TR | 1.000 | WS | flour2 TJ | 0.815 | WS | ycorn SU | 0.980 |
| WS | carrot BO | 0.213 | WS | oil CO | 1.000 | WS | flour2 TR | 0.987 | WS | ycorn TJ | 0.764 |
| WS | rice LP | 0.706 | WS | oil BO | 1.000 | WS | flour2 CO | 0.946 | WS | ycorn TR | 0.916 |
| WS | rice CB | 0.874 | WS | oil2 LP | 1.000 | WS | flour2 BO | 0.948 | WS | ycorn CO | 0.999 |
| WS | rice SC | 0.844 | WS | oil2 CB | 1.000 | WS | milk LP | 0.976 | WS | ycorn BO | 0.690 |
| WS | rice OR | 0.498 | WS | oil2 SC | 0.906 | WS | milk CB | 1.000 | WS | quinoa LP | 1.000 |
| WS | rice PO | 0.737 | WS | oil2 OR | 0.937 | WS | milk SC | 1.000 | WS | quinoa CB | 1.000 |
| WS | rice SU | 1.000 | WS | oil2 PO | 0.979 | WS | milk OR | 0.989 | FIN | ufv | 1.000 |
| WS | rice TJ | 0.995 | WS | oil2 SU | 0.858 | WS | milk PO | 1.000 | FIN | exchange | 0.144 |
| WS | rice TR | 1.000 | WS | oil2 TJ | 0.996 | WS | milk SU | 1.000 | COM | wti | 1.000 |
| WS | rice CO | 1.000 | WS | oil2 TR | 1.000 | WS | milk TJ | 0.989 | COM | gas ny | 0.993 |
| WS | rice BO | 0.300 | WS | oil2 CO | 0.698 | WS | milk TR | 1.000 | COM | gold | 1.000 |
| WS | rice2 LP | 0.331 | WS | oil2 BO | 0.986 | WS | milk CO | 1.000 | COM | silver | 0.996 |
| WS | rice2 CB | 1.000 | WS | sugar LP | 0.986 | WS | milk BO | 0.965 | COM | zinc | 0.999 |
| WS | rice2 SC | 0.954 | WS | sugar CB | 0.999 | WS | milk2 LP | 1.000 | COM | tin | 1.000 |
| WS | rice2 OR | 0.979 | WS | sugar SC | 0.949 | WS | milk2 CB | 1.000 | COM | soybean | 0.999 |
| WS | rice2 PO | 0.988 | WS | sugar OR | 0.893 | WS | milk2 SC | 1.000 | COM | soy flour | 0.977 |
| WS | rice2 SU | 1.000 | WS | sugar PO | 0.999 | WS | milk2 OR | 1.000 | COM | soy oil | 1.000 |
| WS | rice2 TJ | 0.830 | WS | sugar SU | 1.000 | WS | milk2 PO | 0.018 | COM | lead | 0.958 |
| WS | rice2 TR | 1.000 | WS | sugar TJ | 0.999 | WS | milk2 SU | 1.000 | WS | copper | 1.000 |
| WS | rice2 CO | 0.999 | WS | sugar TR | 1.000 | WS | milk2 TJ | 1.000 | FIN | libor | 1.000 |

Findings comparing weekly and monthly distributions similarly support distributional equivalence; however, these results are omitted here to conserve space.

# D Fine-Tuned Hyperparameters

Table D.1: Fine-tuned Hyperparameters

| Algorithm | Hyperparameters |
|---|---|
| | $FS^{(1)}$ |
| Ridge | $\lambda = 1.1$; *intercept* =True; *positive* =True |
| Lasso | $\lambda = 0.1$; *intercept* =True; *positive* =True |
| ADA | $\alpha = 1.1$; *loss* = square; $T = 55$ |
| GBR | $\gamma = 0.5$ |
| RF | *criterion* = absolute error; $d = 5$ |
| ET | $d = 8$; *bootstrap* = True; ; *oob_score* = True |
| | $FS^{(2)}$ |
| Ridge | $\lambda = 6.5$; *intercept* =True; *positive* =False |
| Lasso | $\lambda = 0.1$; *intercept* =True; *positive* =False |
| ADA | $\alpha = 1.1$; $T = 55$ |
| GBR | $\gamma = 0.5$; $T = 130$ |
| RF | $T = 105$; $d = 4$ |
| ET | All default values |
| | $FS^{(3)}$ |
| Ridge | $\lambda = 5.3$; *intercept* =True; *positive* =False |
| Lasso | $\lambda = 0.1$; *intercept* =True; *positive* =False |
| ADA | $\alpha = 0.99$; $d = 7$ |
| GBR | $\gamma = 0.3$; $T = 120$ |
| RF | *criterion* = absolute error; $d = 5$; $T = 130$ |
| ET | $d = 10$ |
| | $FS^{(4)}$ |
| Ridge | $\lambda = 3.5$; *intercept* =True; *positive* =False |
| Lasso | $\lambda = 0.37$; *intercept* =True; *positive* =False |
| ADA | $\alpha = 0.98$; *loss* = square; $d = 5$ |
| GBR | $\gamma = 0.5$; $T = 110$ |
| RF | *criterion* = absolute error; $d = 6$ |
| ET | $d = 9$; *criterion* = absolute error; *bootstrap* = True; ; *oob_score* = True |

Note: Throughout the fine-tuning process, it was determined that the default values in the scikit-learn functions were the most suitable, hence, these default values have been omitted from the table.

Table D.2: MSE on Validation Set (out-of-sample): With and Without Fine-tuning

| Forecast | $F^{(1)}$ | | $F^{(2)}$ | | $F^{(3)}$ | | $F^{(4)}$ | |
|---|---|---|---|---|---|---|---|---|
| | With | Without | With | Without | With | Without | With | Without |
| Ridge | 0.300 | 0.330 | 0.180 | 0.211 | 0.144 | 0.158 | 0.198 | 1.246 |
| Lasso | 0.321 | 2.280 | 0.358 | 2.280 | 0.300 | 2.280 | 0.649 | 0.796 |
| ADA | 0.475 | 0.528 | 0.386 | 0.425 | 0.482 | 0.522 | 0.540 | 1.398 |
| GBR | 0.358 | 0.461 | 0.309 | 0.245 | 0.309 | 0.366 | 0.368 | 0.557 |
| RF | 0.404 | 0.471 | 0.466 | 0.484 | 0.404 | 0.406 | 0.527 | 0.446 |
| ET | 0.334 | 0.356 | 0.269 | 0.269 | 0.216 | 0.239 | 0.380 | 0.995 |

# E Forecast Evaluation With and Without Feature Selection

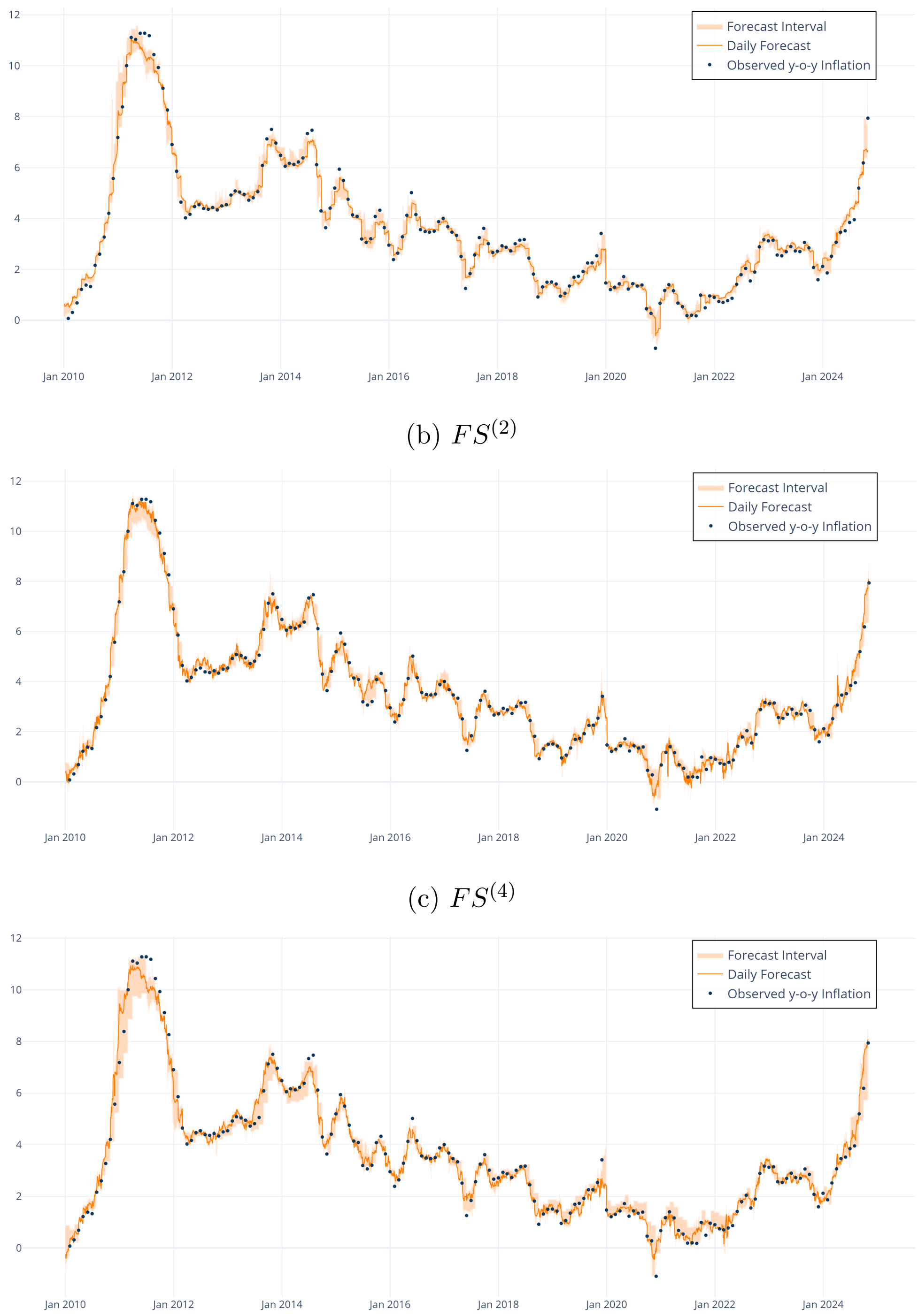
Table E.1: Relative-MSE on Validation Set (out-of-sample): With and Without Feature Selection

| Forecast | $No\ FS$ | $FS^{(1)}$ | $FS^{(2)}$ | $FS^{(3)}$ | $FS^{(4)}$ |
|---|---|---|---|---|---|
| Ridge | 2.9 | 2.1 | 1.3 | 1.0 | 3.1 |
| Lasso | 3.2 | 2.2 | 2.5 | 2.1 | 2.7 |
| ADA | 4.0 | 3.3 | 2.7 | 3.3 | 2.0 |
| GBR | 3.6 | 2.5 | 2.1 | 2.1 | 2.1 |
| RF | 4.3 | 2.8 | 3.2 | 2.8 | 2.0 |
| ET | 3.3 | 2.3 | 1.9 | 1.5 | 3.8 |

Note: "$No\ FS$" stands for No Feature Selection. Relative-MSE computed as: Rel-MSE = $\frac{MSE_i^{FS^{(j)}}}{MSE_{Ridge}^{FS^{(3)}}}$

# F Other High-frequency Forecasts

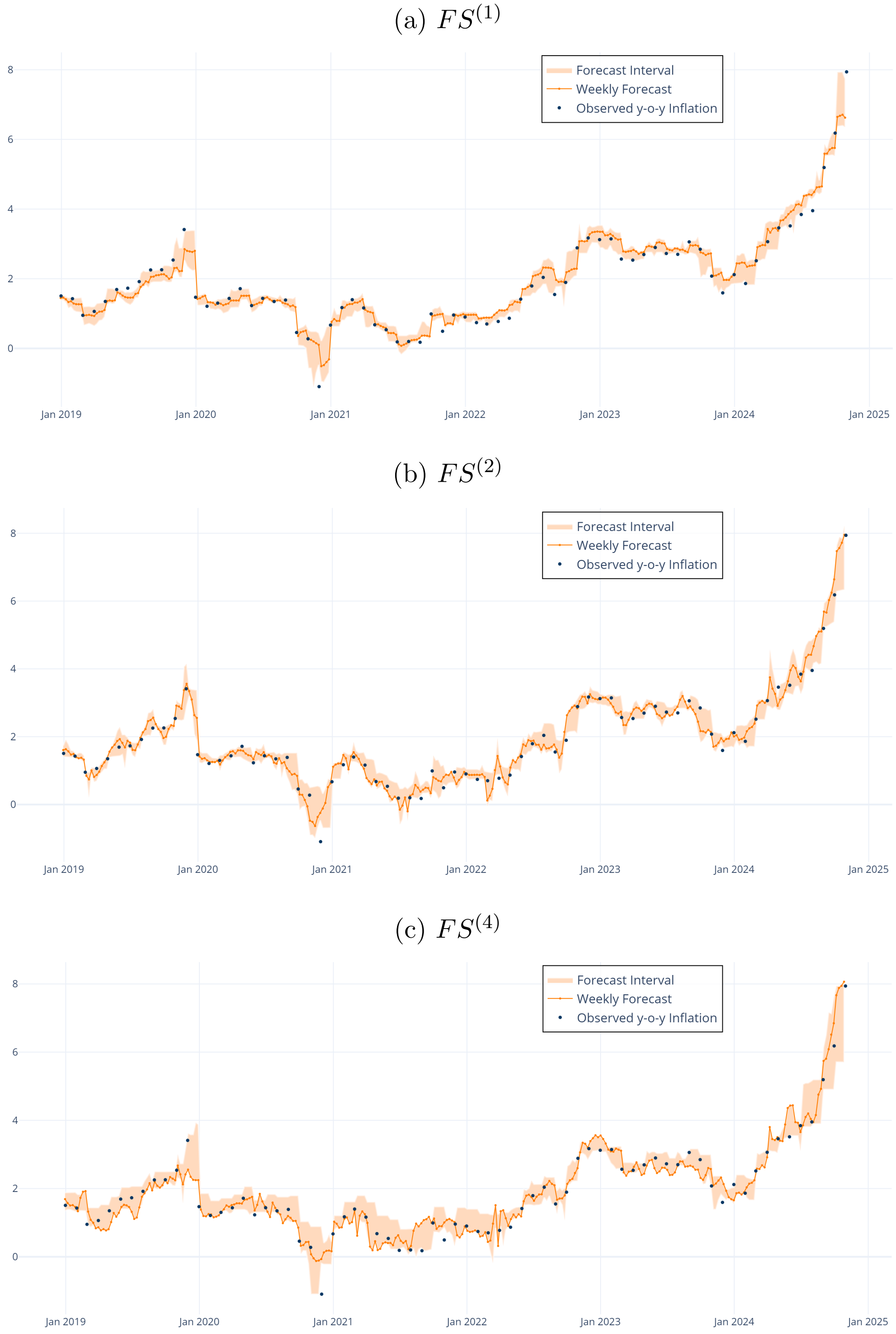
Figure F.1: Daily y-o-y Inflation Forecast


(a) $FS^{(1)}$

(b) $FS^{(2)}$

(c) $FS^{(4)}$

Figure F.2: Weekly y-o-y Inflation Forecast

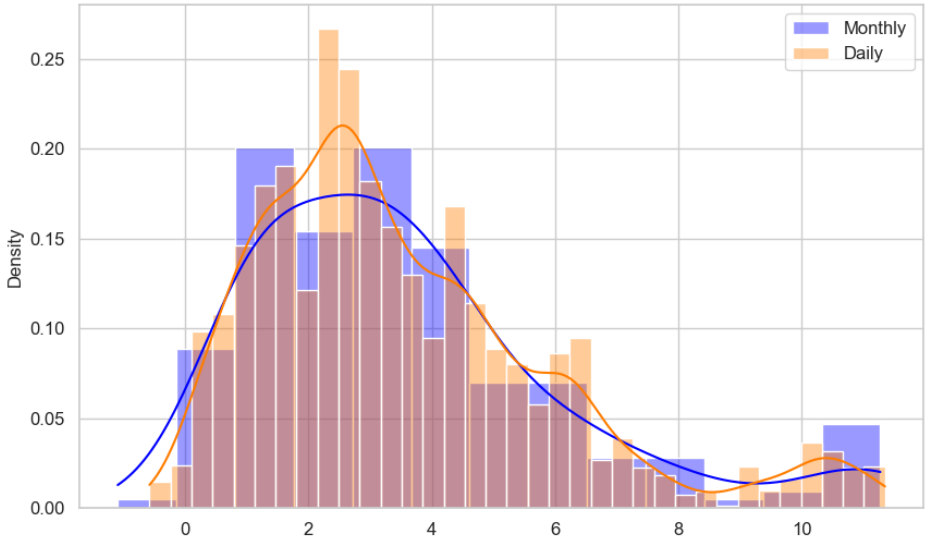


(a) $FS^{(1)}$

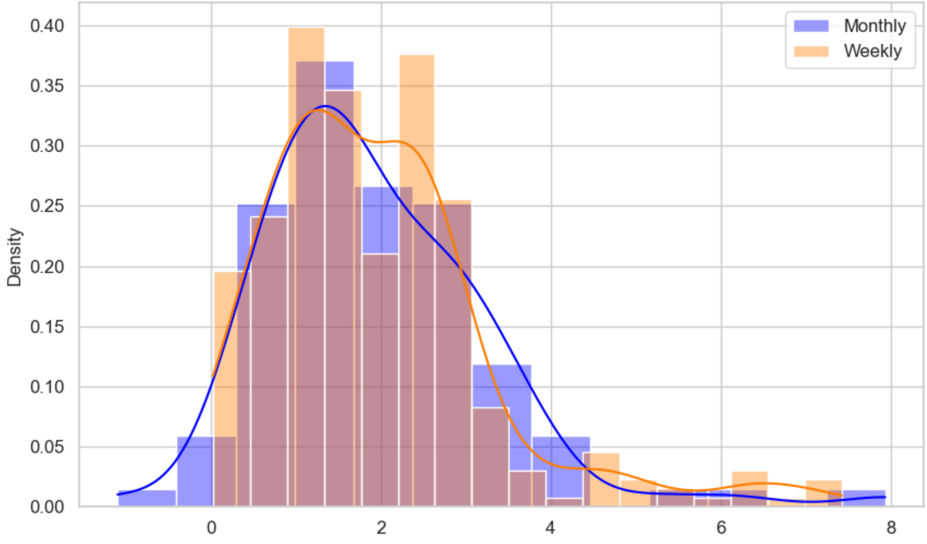
(b) $FS^{(2)}$

(c) $FS^{(4)}$

# G Distribution Comparison Between Observed and High-frequency Predicted y-o-y Inflation



(a) Monthly vs. Daily



(b) Monthly vs. Weekly