Annual Food Price Inflation Forecasting: An Auto-Regressive Random Forest Approach

May 15, 2024

William N. McWilliams, Department Of Agricultural Applied Economics, College of Agriculture and Life Sciences, Virginia Tech University, Blacksburg, Virginia 24061, USA, wnm007@vt.edu

Olga Isengildina Massa, Department Of Agricultural Applied Economics, College of Agriculture and Life Sciences, Virginia Tech University, Blacksburg, Virginia 24061, USA, oimassa@vt.edu

Shamar L. Stewart, Department Of Agricultural Applied Economics, College of Agriculture and Life Sciences, Virginia Tech University, Blacksburg, Virginia 24061, USA, stewartls@vt.edu

Paper prepared for presentation at the 2024 Agricultural Applied Economics Association Annual Meeting, New Orleans, LA; July 28-30, 2024

1 Introduction

At the beginning of 2021, food prices began to rise at historical paces, re-prioritizing food price inflation as a concern to the US economy and policymakers. Given the significant volatility of food price inflation and the fact that food is a necessary good for all consumers, it is essential to have access to accurate information about food price inflation within an economy. The Economic Research Service (ERS) of the United States Department of Agriculture (USDA) fills this need by providing the public with a monthly "Food Price Outlook," which forecasts annual inflation rates across 22 different food-related inflation indices. As the primary source of information regarding US food price inflation, the Food Price Outlook (FPO) is a valuable source of information relied upon by food industry professionals, researchers, policymakers, and the media.

Despite their importance, FPO forecasts have not been extensively scrutinized. Joutz et al. (2000) provided some insights into the various forecast methodologies implemented by the USDA-ERS throughout the '80s and '90s, suggested alternative models, and offered analysis across model accuracy and reliability. Similarly, Kuhns et al. (2015) and Buck et al. (2023) offered a detailed explanation of the myriad of USDA-ERS forecast models revised and implemented from 2000 until 2015 and 2022 respectively, suggesting updates to models that reflect the current econometric methods. In response to the critiques of Nakamura (2008) and the papers mentioned above, Maclachlan et al. (2022) suggested a new optimized Seasonal ARIMA model for FPO forecasts, which was implemented in July of 2023. This "new time series approach" provides a standardized approach for defining a set of models, selecting a model based on information loss, and developing a prediction interval. This improved the previous approaches by considering uncertainty more rigorously, allowing for transparency and reproducibility within the FPO reports, and increasing model accuracy as measured by root mean squared error (RMSE).

However, the implementation of the optimized SARIMA model is limited by two main factors. First, the model uses only historical values of the CPI series to inform the model and does not include any additional exogenous variables despite evidence of several variables' ability to affect the prices of food items (Adjemian et al., 2023). Second, the choice of an ARIMA model assumes a linear relationship between the time series and its past observations and only allows for uni-variate time-series data. At the same time, the development and application of new random forest (RF) machine learning techniques to forecasting has been shown to provide performance increases in cases of high economic uncertainty. This is largely attributable to the model's ability to handle nonlinear relationships (Goulet Coulombe et al., 2022), making RF an ideal candidate for improving FPO forecasts.

The goal of this study is to implement the newly developed Auto-Regressive Random Forest (ARRF) model Coulombe (2020) to FPO forecasting. Our objective is to improve the forecast accuracy and performance of the FPO while maintaining the recent qualitative improvements gained by the SARIMA model, such as cross-category standardization and improved measures of uncertainty, but also allowing for additional exogenous variables. This ARRF model, we argue, will allow for the continuation of the improvements made by Maclachlan et al. (2022) while testing for and accommodating nonlinear forecasting dynamics otherwise absent in the current approach to the FPO.

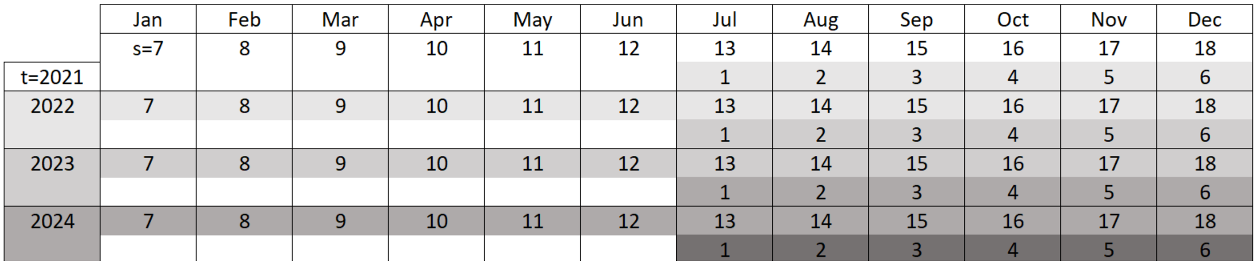
Another contribution of this study will be a comprehensive evaluation of the proposed as well as the optimized SARIMA forecasts that have yet to be thoroughly evaluated since their adoption. While keeping the forecast's data, rolling window size, and 18 horizons consistent with that currently used by the USDA-ERS, the MRF model will be implemented such that forecast intervals for the years 2003 to 2022 will be available for comparison to the historical data provided by the FPO. This will allow for a comparison of the models' performance across 20 years for 22 categories of food items. Following Isengildina-Massa et al. (2012), our forecast evaluation will incorporate measures for accuracy and bias, or the model's tendency to over- or under-estimate the actual annual inflation rates.

2 Data

Food price inflation and other measures of inflation are tracked monthly by The Bureau of Labor Statistics (BLS). The percentage changes in retail food prices are reflected in the Consumer Price Index(CPI). Percent changes in wholesale food prices are reported within the Producer Price Index (PPI). BLS publishes these measures with a 1-month delay, i.e., estimates for September 2023 were released by BLS on October 12, 2023.

Because it is important for food industry participants to anticipate food price inflation, not just observe it ex-post, USDA's Economic Research Service provides monthly forecasts of annual food price inflation in their Food Price Outlook (FPO) reports. Figure 1 illustrates the forecasting cycle for FPO reports as the new forecast estimates are produced, and observed index values are included when they become available. This figure demonstrates that the first forecast of annual inflation is released in July of the previous year, or 18 steps (months) before the end of the forecasted year. It is important to recognize that the cycles overlap with the 18-month-long forecasting cycle, and a forecast for the next year is started before the current year is finalized.

For each year, t, the annual inflation index is calculated as a sum of monthly price changes



	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
	s=7	8	9	10	11	12	13	14	15	16	17	18
t=2021							1	2	3	4	5	6
2022	7	8	9	10	11	12	13	14	15	16	17	18
							1	2	3	4	5	6
2023	7	8	9	10	11	12	13	14	15	16	17	18
							1	2	3	4	5	6
2024	7	8	9	10	11	12	13	14	15	16	17	18
							1	2	3	4	5	6

Figure 1: USDA FPO Forecast Cycles from Years 2021 to 2024

from the previous year (t-1) for each index j, as follows:

$$\Delta\hat{\Pi}^j_{t,m|s} = \begin{cases} 100 \cdot \dfrac{\sum\limits_{m=1}^{12} \widehat{\mathcal{I}}^j_{t,m} - \sum\limits_{m=1}^{12} \widehat{\mathcal{I}}^j_{t-1,m} + \sum\limits_{m=1}^{12} \mathcal{I}^j_{t-1,m}}{\sum\limits_{m=1}^{12} \widehat{\mathcal{I}}^j_{t-1,m} + \sum\limits_{m=1}^{12} \mathcal{I}^j_{t-1,m}} & \textit{for } s = \textit{1...6} \\ \\ 100 \cdot \dfrac{\sum\limits_{m=1}^{12} \widehat{\mathcal{I}}^j_{t,m} - \sum\limits_{m=1}^{12} \mathcal{I}^j_{t-1,m}}{\sum\limits_{m=1}^{12} \mathcal{I}^j_{t-1,m}} & \textit{for } s = \textit{7} \\ \\ 100 \cdot \dfrac{\sum\limits_{m=1}^{12} \widehat{\mathcal{I}}^j_{t,m} + \sum\limits_{m=1}^{12} \mathcal{I}^j_{t,m} - \sum\limits_{m=1}^{12} \mathcal{I}^j_{t-1,m}}{\sum\limits_{m=1}^{12} \mathcal{I}^j_{t-1,m}} & \textit{for } s = \textit{8...18} \end{cases} \tag{1}$$

Where $\Delta\hat{\Pi}^j_{t,m|s}$ is the percent change in prices from year $t-1$ to year t for a given inflation index j (All Food, Food Away From Home, poultry, etc), at step s of the forecast. Each of the 18 steps within the forecasting cycle estimates the same target year, t, consisting of steps 7-18, as shown in Figure 1. Observed monthly inflation index values $\mathcal{I}^j_{t,m}$ are released by the BLS with a one-month lag around the 13th of every month. Since the FPO reports are released around the 25th of the month, this information is included in the published estimate. The difference between each forecast within the same cycle lies in the amount of forecasted versus observed inflation data available at the step of the forecast. Therefore, as the forecast begins in step 1 and moves to step 18 three cases exist for calculating $\Delta\hat{\Pi}^j_{t,m|s}$ as shown in Equation 1.

For steps 1 through 6 of a forecast cycle ($s = 1...6$), case 1 of Equation 1 demonstrates the appropriate equation to employ. As shown in Figure 1, these steps always occur from July to December of year $t-1$; thus, at each of these steps, there are unobserved values in the calendar year $t-1$ ($\widehat{\mathcal{I}}^j_{t-1,m}$), and in the calendar year t ($\widehat{\mathcal{I}}^j_{t,m}$).

As the forecast cycle moves into January(s=7 in Equation 1), all monthly inflation index values for the year $t-1$ have been reported by the BLS. Therefore, all sum operators related to estimated index values($\widehat{\mathcal{I}}^j_{t-1,m}$) will be dropped from the numerator and denominator of the general equation, resulting in a form such as case 2 in Equation 1. Case two is a special case only applicable to step 7 since, at this point in time, there are no observed index values for the year t ,and therefore, a single sum operator is needed for the year t index estimates($\widehat{\mathcal{I}}^j_{t,m}$). This does not hold as the BLS begins to report index values for the year t as will be shown next.

As the forecast cycle moves into step 8, a new sum operator is needed to handle newly observed inflation index values($\mathcal{I}^j_{t-1,m}$). In February of year t (s=8), the BLS will report the first observed values for the year t, and as the forecast cycle progresses from step 8 to 18, additional observed values will be reported(see Figure 1. For these final 11 steps, case 3 of Equation 1 is employed.

2.1 Example FPO Report: October 25, 2023

FPO reports released from July through December include forecast estimates for both the current year and next year's change in food price inflation. For example, an FPO report released on October 25, 2023, included the estimate at the 16th step for the forecast targeting year 2023 ($\Delta\hat{\Pi}^j_{2023,m|16}$), and the estimate at the 4th step for the forecast targeting year 2024 ($\Delta\hat{\Pi}^j_{2024,m|4}$), as shown in Figure 1. Based on the discussion above, the 4th estimate is a pure forecast, and the 16th estimate is a combination of observed values for months 7-15 and forecasts for months 16-18, as follows:

$$\Delta\hat{\Pi}^j_{2023,m|s=16} = 100 \cdot \frac{\left(\sum\limits_{m=10}^{12} \widehat{\mathcal{I}}^j_{2023} + \sum\limits_{m=1}^{9} \mathcal{I}^j_{2023}\right) - \sum\limits_{m=1}^{12} \mathcal{I}^j_{2022}}{\sum\limits_{m=1}^{12} \mathcal{I}^j_{2022}} \tag{2}$$

$$\Delta\hat{\Pi}^j_{2024,m|s=4} = 100 \cdot \frac{\sum\limits_{m=1}^{12} \widehat{\mathcal{I}}^j_{2024,m} - \left(\sum\limits_{m=10}^{12} \widehat{\mathcal{I}}^j_{2023,m} + \sum\limits_{m=1}^{9} \mathcal{I}^j_{2023,m}\right)}{\left(\sum\limits_{m=10}^{12} \widehat{\mathcal{I}}^j_{2023,m} + \sum\limits_{m=1}^{9} \mathcal{I}^j_{2023,m}\right)} \tag{3}$$

where forecast inflation index estimates are included in both the years 2023 and 2024.

As the forecast moves into a new step, the newly observed inflation index values for the previous month reported by the BLS will replace the forecasted value used for that month in the previous step. Using this new observed value, all forecasted values for months that remain unobserved will be re-estimated using a method that is covered in the section 3.1. This means that the value estimated for an unobserved inflation index value will change at each step, or written in an equation:

$$\widehat{\mathcal{I}}^{j}_{t,m|s} \neq \widehat{\mathcal{I}}^{j}_{t,m|s+1} \tag{4}$$

3 Methods

3.1 Seasonal ARIMA

By considering lags(auto-regression), changes between consecutive observations(differencing/integration), and moving average terms of the observed values at point s of a single inflation index (I^j_t), Maclachlan et al. proposed the current ARIMA model used by USDA-ERS for monthly FPO updates. An ARIMA(p,d,q) is defined for a forecast variable y_t, or in this case I_t, as follows:

$$\mathcal{I}^{j}_{t,m|s} = c + \phi_1 \mathcal{I}^{j}_{t-1,m} + \cdots + \phi_p \mathcal{I}^{j}_{t-p,m} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t \tag{5}$$

Box et al. (2015). The coefficients on the left side of the equation ϕ_p and θ_q relate to the auto-regressive and moving average independent variables, respectively.

In addition to this, the FPO forecast model also considers the presence of seasonality within each of the series, resulting in an additional three terms: seasonal auto-regressive terms (P), seasonal differencing (D), and seasonal moving averages (Q). The general equation for a SARIMA(p,d,q)(P,D,Q) can be written as:

$$\Phi\left(\mathcal{L}^m\right)\phi(\mathcal{L})\left(\mathcal{I}_\bullet - \mu\right) = \Theta\left(\mathcal{L}^m\right)\theta(\mathcal{L})\varepsilon_t \tag{6}$$

where

$$\text{AR: } \phi(\mathcal{L}) = 1 - \phi_1\mathcal{L} - \ldots - \phi_p\mathcal{L}^p$$
$$\text{MA: } \theta(\mathcal{L}) = 1 + \theta_1\mathcal{L} + \ldots + \theta_q\mathcal{L}^q$$
$$\text{Seasonal AR: } \Phi\left(\mathcal{L}^m\right) = 1 - \Phi_1\mathcal{L}^m - \ldots - \Phi_P\mathcal{L}^{Pm}$$
$$\text{Seasonal MA: } \Theta\left(\mathcal{L}^m\right) = 1 + \Theta_1\mathcal{L}^m + \ldots + \Theta_Q\mathcal{L}^{Qm}$$

This includes the three ARIMA terms: p auto-regressive terms(with coefficients $\phi ... \phi_p$) q moving average terms(with coefficients from $\theta_1 ... \theta_P$), and d degrees of differencing along with their seasonal terms: P seasonal auto-regressive terms (with the coefficients $\Phi_1 ... \Phi_P$), Q seasonal moving average terms(with coefficients $\Theta_1 ... \Theta_Q$), and D is the order of seasonal differencing. The notation $\mathcal{I}_\bullet$ will be used to represent $\mathcal{I}^{j}_{jt,m|s}$ from this point forward for notational ease.

To standardize the forecast approach across steps and cycles, Maclachlan et al. implemented the *auto.arima()* function from the popular *forecast* package in R that uses a variation of the Hyndman-Khandar algorithm (Hyndman and Khandakar, 2008), which employs Canova Hansen-Hansen tests for seasonality(Canova and Hansen, 1995), successive KPSS unit-root tests(Kwiatkowski et al., 1992) and maximum likelihood estimation(MLE) to automate the process of determining the value of each of the 6 model parameters. The ERS pre-specifies maximum values for each model parameter in the following Table 1 .

Table 1: SARIMA Paramter Values

Parameter	Maximum Value
Autoregressive (p)	$p \leq 12$
Differencing (d)	$d \leq 4$
Moving Average(q)	$q \leq 2$
Seasonal Autoregressive(P)	$P \leq 1$
Seasonal Differencing(D)	$D \leq 2$
Seasonal Moving Average(Q)	$Q \leq 1$

Based on the maximum values in Table 1, there is a set of 2,340 possible models to choose from for each forecast. To choose which model to use, the automated SARIMA model compares models by Bayesian Information Criteria(BIC). This method uses a likelihood function that helps balance the model fit and the model parsimony(Schwarz, 1978). It follows the general function below:

$$BIC = k \ln(n) - 2\ln(\hat{\mathcal{L}}) \tag{7}$$

where k is the number of parameters the model estimates, n is the number of data points observed, and $\hat{\mathcal{L}}$ is the maximum likelihood function that measures the model's fit.

Once the best model specifications have been determined for the current forecast step, a point forecast for the inflation index($\hat{I}_t^j$) is estimated for the current step, and each of the remaining steps in the forecast cycle(for a total of $(18 - s + 1)$ for a given s). Once all values for $\hat{I}^j$ have been estimated, these and the observed values I^j are plugged into the appropriate equation mentioned in Section 2 to determine the value reported by the FPO.

The FPO's current forecasting methodology does consider past observations of a single series of inflation indices but not for the inclusion of other relevant information. Maclachlan et al. tested the potential of expanding the set of exogenous variables of the forecast model to include futures prices of commodities relevant to the dependent variable, noting increases to the model's in-sample fit. The increases in model fit proved true for only specific inflation indices. This suggests that broadening the set of exogenous variables considered in the current model will require a trade-off between including variables that increase the model's performance and maintaining a standardized approach across all series at all points in time, one of the benefits mentioned by Maclachlan et al.. In addition to this, as the number of variables is increased in the model, concerns around over-fitting and the computational costs of optimizing need to be considered.

Linear regression is at the core of the SARIMA model, leading to limitations that must be considered when applying it in specific contexts. The auto-regressive(AR) and moving average(MA) terms allow for a nonlinear functional form, but the model is still linear in its parameters. This is, of course, an issue when forecasting something like US food price inflation that is affected by changes in the economic environment.

4 Tree-Based Methods

Among the many Machine Learning(ML) applications that have risen in popularity within the econometrics community, forecasting has been an area rich with experimental studies looking for increased prediction accuracy. Of the ML models that have come of interest, tree-based methods, such as the random forest (RF) model we use as a framework in this study, have become one of the favored methods for several reasons.

The strength of these models lies in their ability to manage data with many complicated and nonlinear relationships. Tree-based models do this by repeatedly dividing a sample of observations into smaller sub-samples, a method called recursive partitioning. Each time the observations are divided and subdivided, the interactions within the resulting smaller sub-samples become less complicated until a simple model can fit them. This approach's added benefits allow for many explanatory variables and complex relationships while circumnavigating the risk of over-fitting models.

4.1 Data Partitioning With Decision Trees

Tree-based methods are based on a decision tree framework that takes a sample of observations and divides it into several sub-samples based on some splitting criteria. Figure 2 depicts the general form of a decision tree and some of the common terminology used to reference it. Nodes refer to places on the decision tree diagram that represent samples of data and are depicted as blue squares in Figure 2. The top node is where the tree begins, so it is fittingly referred to as the "root" node. The root node contains all available observations of $\mathcal{I}_{\bullet}$, the

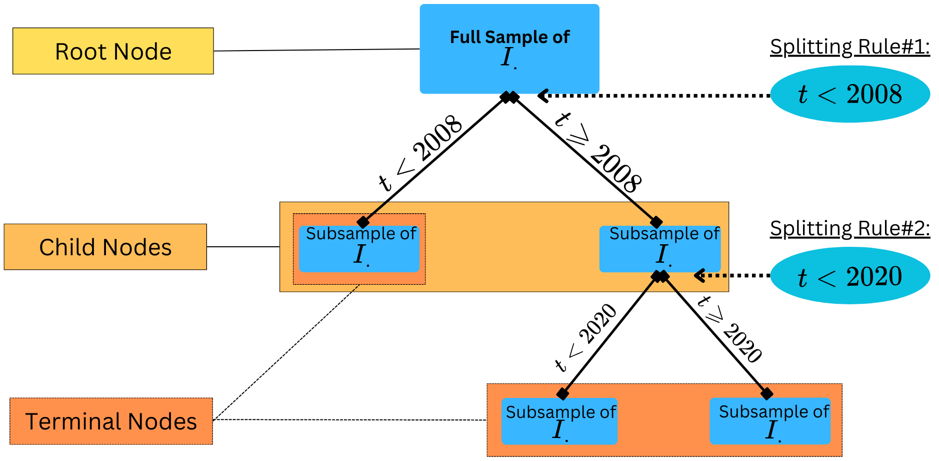


Figure 2: General Anatomy of a Decision Tree

full sample of all available inflation index values. AT this first node, the available observation of $\mathcal{I}_\bullet$ is divided into two sub-samples based on some criteria, often called a splitting rule. In the example provided in Figure 2, the first splitting rule, $t < 2008$, divides the data into two sub-samples, one with values for $\mathcal{I}_\bullet$ that occurred before 2008, and another with values that were observed after 2008. Further discussion about methods to determine splitting rules will come later in this section. Following the two lines in the decision tree down from the root node in Figure 2 makes it easy to see that the two sub-samples are stored in two new nodes due to the splitting rule. All observations with $t < 2008$, where t refers to years, are stored to the left, and all other values are stored in the node to the right. In Figure 2 a second splitting rule, $t < 2020$, is implemented on the sub-sample of $\mathcal{I}_\bullet$ that was stored in the right node with values of with $t \geq 2008$ following the first splitting rule. This resulted in two additional sub-samples and marked the end of this particular decision tree. This method of repeatedly dividing the data in to smaller sub-samples is called *recursive partitioning.*

As labeled in Figure 2 any node resulting from a splitting rule is considered to be a ”child” node of the node that was split to create it. Any nodes left undivided by the decision tree are called terminal nodes(or sometimes leaves. In the provided example, 3 terminal nodes are left once the decision tree is complete. Each of the terminal nodes contains a sub-sample of $\mathcal{I}_\bullet$. If the number of observations in each of the terminal nodes was added together, they should sum to equal the total number of $\mathcal{I}_\bullet$ in the original full sample found in the root node. The final form of the decision tree can then be used to make predictions for values of $\mathcal{I}_\bullet$ by using the sub-samples of data in the terminal nodes. This is accomplished by fitting a model to each of the sub-samples found in the terminal nodes.

4.2 Regression Trees

The term regression tree refers to using a decision tree framework to make predictions about a dependent variable of interest. Splitting rules can be strategically chosen such that the *recursive partitioning* of the data creates sub-samples of $\mathcal{I}_\bullet$ that have a common set of explanatory variables. This allows for simpler models to be fit to each of the sub-samples of $\mathcal{I}_\bullet$ instead of fitting a more complicated model to the full sample. In a traditional regression tree, a simple model using the average of the observations in a terminal node estimates the value of $\hat{\mathcal{I}}_\bullet$ with a given value of t.

Following the example in Figure 2, consider a situation that required the prediction of an index value $\hat{\mathcal{I}}_\bullet$ with a given value of $t = 2007$. Following the decision tree in Figure 2 can be estimated by asking a series of questions coinciding with the splitting criteria to determine the terminal node that the observation belongs in. For this case, the first splitting decision asks if the observation has a value of $t < 2008$. Since $t = 2007$ the answer is yes, which leads to the sub-sample in the terminal node farthest to the left containing only observations of $\mathcal{I}_\bullet < 2008$. As this is a terminal node, the mean model fit to this node is then used to estimate the value of the inflation index as $\hat{\mathcal{I}}_\bullet = \frac{1}{n}\sum_{i=1}^{n}(\mathcal{I}_{\bullet N_j})$, the sample mean of the terminal node.

The within-sample performance of the regression tree can be found by comparing the differences between estimates of $\hat{\mathcal{I}}_\bullet$ and actual values of $\mathcal{I}_\bullet$. Equation 8 shows the residual sum of squares (RSS) formula used to measure the models in sample fit across all nodes of the tree where $\mathcal{I}_\bullet$ is the observed value, and $\hat{\mathcal{I}}_{\bullet Nj}$ is the mean response for the training observations within the jth node. The smaller an RSS value is, the better the within-sample fit of the regression tree is.

$$RSS = \sum_{j=1}^{J}\sum_{i\in N_j}(\mathcal{I}_{\bullet i} - \hat{\mathcal{I}}_{\bullet N_j})^2 \tag{8}$$

4.2.1 How to Grow a Tree

The basic regression-tree growing algorithm is called *recursive binary splitting*. It relies on the measure of model fit, RSS, mentioned in Equation 8 to automate the selection of splitting rules. It starts with the full sample of observations at the root node, searches over all the available independent variables $X_1 ... X_p$, and considers each of the possible splitting values b for each of these. The algorithm uses these values to survey all of the possible splitting rules available and the resulting RSS for each option. Traditionally, only binary splits for a single node are considered at a given point in time. When the optimal splitting rule has been found, the node is split accordingly to create two new child nodes. As the algorithm continues, the two new child nodes undergo the same process to determine the optimal splitting rules for each node. Of the two, the one resulting in the lowest RSS will be chosen for the next split. The process will then repeat itself until some predefined stopping criterion is met. This criterion is most commonly based on minimum decreases in the *RSS*, minimum sample sizes for terminal nodes, or a maximum length from the root node to the terminal nodes. Trade-offs between the model's fit and the model's predictive accuracy must be weighed when imposing a stopping criterion. Larger trees tend to lead to better within-sample fit(lower RSS), but are prone to being over-fit. Too short of a maximum length from the root node, or too large of a minimum sample size will stop the algorithm too early resulting in poor performance

The *recursive binary splitting* algorithm tends to result in overly complex trees that perform poorly on out-of-sample predictions. In order to address this, a minimum reduction in RSS may be required for proposed splitting rules. This will help limit the complexity of the tree but it is known for being too short-sighted of an approach and has hence been labeled a *greedy* algorithm. For example, a particular splitting rule may not lead to significant decreases in *RSS* but can result in subsequent splits, potentially leading to sub-optimal model performance.

4.2.2 Tree Pruning

An alternative method for growing decision trees looks to manage the negative tendencies of an unbounded recursive binary splitting algorithm while avoiding the greedy nature of imposed minimum reductions in RSS. Instead of limiting the tree size with a criterion, the trees are allowed to grow very large with many terminal nodes, which is likely to over-fit and perform poorly out-of-sample. This overgrown tree can be referred to as T_0. For every T_0, there exists a set of trees $T_0 ... T_r$ where T_r corresponds to the single root node, and all trees between it and T_0 are possible subtrees.

Many methods exist for *pruning* back large trees, most of which try to minimize a test error rate. *Reduced error pruning(REP)* is the simplest of these pruning techniques. With *REP* T_0 is *pruned* to smaller sub-trees by removing specific terminal nodes, starting with the largest tree (T_0) and working up to the root node (T_r). The predictive accuracy for each sub tree is validated using a hold-out sample, and if it has not gotten worse, the pruned node will be left out. Iterating over this process is intended to remove sections of the tree that have little or no importance when predicting the dependent variable, subsequently reducing the complexity of the tree and improving the predictive accuracy. This requires cross-validation must be employed across all potential sub-trees, which is inefficient as it can require validation across many sub-trees.

Cost complexity pruning is an alternative method used to limit the number of sub-trees considered during the pruning process. During this process, a parameter ($\alpha \geq 0$) is chosen and used to define the cost complexity measure in Equation 9:

$$\sum_{m=1}^{T} \sum_{x_i \in N_j} (\mathcal{I}_{\bullet i} - \hat{\mathcal{I}}_{\bullet N_m})^2 + \alpha |T| \tag{9}$$

Where $|T|$ is the number of terminal nodes on tree T, N_m is the sub-sample corresponding with the mth terminal node, and $\hat{\mathcal{I}_{\bullet \mathcal{N}_m}}$ is the predicted response associated with the mth node. The complexity measure controls the trade-off between a tree's fit and complexity by changing the value of α. As α increases, a more significant penalty is placed on the tree's number of terminal nodes.

Using *recursive binary splitting*, a large regression tree T_0 needs to be grown. Within T_0, a set of sub-trees exists that can be compared via the cost complexity measure in Equation 9, and for a given value of α there is a sub-tree T^* that will result in a minimum value. If $\alpha = 0$, then T will equal T_0 since Equation 9 will just measure the within-sample error. As α increases, a higher penalty will be placed on trees with more terminal nodes, so T^* will tend to be a smaller sub-tree.

The value for α will affect the final tree chosen to make predictions of the dependent variable, and thus, determining the value for these parameters is important. The most common way of determining a value for α begins by dividing the data set into 10 smaller data sets equal in size. One of these data sets will be held for cross-validation, while the other nine will each be subject to recursive binary splitting, resulting in nine large trees (T_0). For each of the nine trees, an optimal tree (T^*) will then be determined using a cost complexity function with the same value of α. The mean squared prediction error of each of the nine T^* on the data in the 10th data set that was withheld for validation is then evaluated. The average of all nine mean squared predictions is then found. This process is repeated for several α values, and the average value of the resulting mean squared prediction errors are then compared. Whichever value of α has the lowest average mean squared prediction errors is then used in the cost complexity function applied to the full data set.

4.3 Bagging

Decision trees are known to have a high variance issue. This means that if two decision trees were grown using random samples of the same training data, their results could be very different, negatively affecting test accuracy. One commonly used way of lowering the variance of a decision tree procedure involves employing bootstrap aggregation, often called *bagging* in the context of decision trees. *Bagging* involves taking many training sets from the population, growing a separate regression tree from each training set, and averaging the resulting predictions, resulting in a final tree with lower relative variance.

Consider a set of n trees $T_1 ... T_n$ each one grown on one of n different random samples taken from the same training set. The predictions of each of these trees can then be averaged to obtain:

$$T_{avg}(\hat{\mathcal{I}}_{\bullet}) = \frac{1}{n}\sum_{i=1}^{n} T_i(\hat{\mathcal{I}}_{\bullet}) \tag{10}$$

This typical *bagging* method can be applied across hundreds or even thousands of trees to reduce the variance within regression tree methods. Furthermore, this method may be applied across trees grown in several ways, such as cost complexity pruning, reduced error pruning, etc. In fact, no pruning at all is most commonly used on the trees as the averaging across the trees' predicted values lowers the high variance associated with large trees mentioned in Subsection 4.2.1 and allows employing several predictive models, such as mean, median, regression, etc.

Combining all these individually simple models will result in the final, potentially more effective, $T_{avg}(\mathcal{I}_{\bullet})$. The idea of aggregating model predictions such as this is called an *ensemble* method. While the ensemble model $T_{avg}(\mathcal{I}_{\bullet})$ typically results in improved predictive accuracy, it often increases the difficulties of interpreting the model. In many cases, practitioners are concerned about the predictive accuracy of the models and which of the predictor variables were the most significant to the model. Variable importance addresses this by summing the total amount that the RSS was decreased by splits made over a given predictor variable and averaging these values across all n of the bagged trees. Larger values of variable importance measure indicate the predictor variable is more important to the model.

Although the method of averaging across all of the $\hat{\mathcal{I}}_{\bullet}$ values predicted by each tree helps lower the model's variance, it is limited to the correlation of the trees across which it averages. If the trees are highly correlated, then this method will not result in a substantial decrease in variance, which in turn will affect the predictive accuracy of the model out of the sample. Despite growing trees on random training data samples, the *recursive binary splitting* method used to grow the bagged trees will often result in highly correlated trees. This is particularly true when some of the predictor variables used during the splitting rules are relatively more important than others which will lead to splits on these more important variables to occur earlier on in the tree's growth process, resulting in correlated trees. This often results in non-substantial decreases in the models' variance.

4.4 Random Forests

Random Forests are very similar to the bagging algorithm in that they take n random samples of the training data, grow n regression trees on each sample, and average across the predicted variable. The traditional *bagging* method may result in correlated trees when some of the predictor variables are found to be more relatively important than others. Random forests address this weakness of the traditional *bagging* method by adding an additional step that helps to decorrelate the trees during the splitting process.

Before each splitting rule is determined, a random sample(without replacement) of m candidate variables is taken from the full set of p predictor variables. This reoccurs at each split, allowing for $m = \sqrt{p}$ candidate variables each time. This means that for some trees, the more important predictor variables will not be considered for earlier splitting points which may result in less correlated trees, thereby making the $T_{avg}(\hat{\mathcal{I}}_{\bullet})$ less variable.

4.5 Auto-Regressive Random Forest

In their 2021 paper Coulombe proposed the Auto-Regressive Random Forest as an important adaptation of the basic RF model that is noted as being particularly well suited to predict measures of inflation among many other macroeconomic indicators. Coulombe notes the key difference between ARRF and Rf is the inclusion of a linear part within each tree leaf instead of just an intercept. An example of a single regression tree is shown for reference in Figure 2 where time t^* is the date where a structural shift occurred, and g_{t-1} is a lagged economic indicator such as the unemployment rates.

This is only one of the many regression trees that can be used to explain the change in inflation Π_t. The structure, variables used for dividing the independent variables, and the values of the splitting points must be determined. Iterative local updates are made to the trees known as *greedy* algorithms Breiman et al. (2017) that help optimize these factors while remaining computationally feasible. The results of this *greedy* algorithm can be highly variable Hastie et al. (2009).

High variance among the decision tree results is frequently addressed with a bootstrap aggregation technique often called bagging Breiman (1996). A Block Bayesian Bootstrap (BBB) Hans Kunsch (1989) randomly selects many training sets from the population, builds a separate prediction model using each training set, and averages the resulting predictions.

In Breiman's 2001 paper an additional step to decorrelate the trees was suggested. As trees are being built on the BBB-constructed training sets, a random sample of m predictor variables is chosen to be candidates when splitting the data. This additional step is, in fact, why the forest is said to be random. Without this precaution, the locally focused *greedy* algorithm will always follow the same series of splits. The m parameter is a tuneable metric within the model that defaults to $m = p/3$ where p is the total number of predictor variables.

The ARRF model we use for this paper is based off of the Tiny ARRF model mentioned in Coulombe (2020). A ten-year window of the historical values of $\mathcal{I}_\bullet$ are considered at each of the eighteen steps of the forecast. Eight lags of the dependent variable $\mathcal{I}_\bullet$ are used as the splitting variables used to fit 1000 different regression trees. At each of the 1000 regression tree's terminal nodes $\mathcal{I}_\bullet$ is regressed on the first and second lags of itself. These parameters are then used to produce forecasts of the monthly $\hat{\mathcal{I}}_\bullet$ which are ultimately aggregated using Equation 1 to produce an estimate of $\Delta\Pi^j_{t|s}$. At this time, no additional variables will be considered as we look to motivate the use of the ARRF within this context. Future iterations of this model hope to discuss the addition of appropriate explanatory models and the additional benefits they may offer to the forecast's performance.

5 Model Evaluation

Since the models are intended to be used to forecast annual percent changes in food price inflation during the monthly FPO report, we evaluate each of the models based on its out-of-sample performance. We chose to limit our current results to the All food series for the current study and intend to explore cross categorical performances at a later date. We first assess each model's accuracy at each of the 18 steps that the FPO reports for targeted years. This allows us to observe the performance dynamics at differing forecast horizons. Next, we determine whether or not the difference in predictive accuracy and performance between the models are significantly different from each other, or essentially the same. Finally, we explore whether or not either of the model exhibits tendencies to over or under forecast the realized values.

5.1 Forecast Accuracy

We evaluate the predictive accuracy of the models discussed in this study based on their out-of-sample performance at each step for the target years 2003-2022. Forecast errors are calculated by comparing the predicted year over year percent change in inflation index value ($\Delta\hat{\Pi}^j_{t,m|s}$) to the actual year-over-year percent change in index value ($\Delta\Pi^j_{t,m|s}$) at each step s for a given target year t and index j using Equation 11:

$$e^j_{s,t} = \Delta\Pi^j_{t,m|s} - \Delta\hat{\Pi}^j_{t,m|s} \qquad t = 2003, ..., 2022 \qquad s = 1, ..., 18 \tag{11}$$

With 18 observations for each target year our results contain a total of 360 error terms for the entire period and 20 per individual step . Since inflation index is measured in year over year percent changes, there is no need to calculate percent errors to compare alternative forecasts or categories.

We determine the forecast's accuracy by assessing the size of the forecast errors in the two most common ways: mean absolute error (MAE) and root mean squared error (RMSE), defined as:

$$RMSE^j = \sqrt{\frac{1}{n}\sum_{s=1}^{n} e^j_{s,t}} \qquad t = 2003, ..., 2022 \qquad s = 1, ..., 18 \tag{12}$$

$$MAE^j = \frac{1}{n}\sum_{s=1}^{n} e^j_{s,t} \qquad t = 2003, ..., 2022 \qquad s = 1, ..., 18 \tag{13}$$

5.2 Tests for Differences Between Forecasts

When suggesting an alternative model for adoption it is important to also consider the costs incurred when switching. If an alternative model is essentially the same as the current model the costs of switching would logically not be rewarded. Cases may occur when two models have accuracy measures relatively close to each other, so it is helpful to determine whether or not the models are producing estimates that are statistically different from each other. To formally test whether or not two models perform significantly different from each other we will implement a modified Mariano Diebold(DM) test following the methods proposed by Harvey et al. as seen in Equation 14.

$$\begin{gathered}
\mathrm{DM} = \frac{\bar{d}}{\sqrt{\frac{1}{n}\left(\delta_0 + 2\sum_{q=1}^{t-1}\delta_q\right)}} \cdot \sqrt{\frac{n+1-2h+\frac{h(h-1)}{n}}{n}} \\
\delta_0 = \frac{1}{n}\sum_{t=1}^{n}\left(d_t - \bar{d}\right)^2; \quad \delta_q = \frac{1}{n}\sum_{t=q+1}^{n}\left(d_t - \bar{d}\right)\left(d_{t-q} - \bar{d}\right) \quad t = 2003, \ldots, 2022
\end{gathered} \tag{14}$$

Where h is the forecast horizon, d_t is the difference between the errors of the two competing models at a given step, $\bar{d}$ is the average value of d_t across the entire test period, δ_0 is the variance of d_t, and δ_q is the q-th order auto-covariance term. The null hypothesis is that the two methods have the same forecast accuracy (Hyndman and Khandakar, 2008).

5.3 Forecast Bias

While the MAE and RMSE inform our evaluation criteria with respect to the magnitude of the forecast errors, these measures do not allow us to observe whether there is a systematic bias to consider. Forecast bias generally refers to the model's tendency to over or under-forecast its target. To assess the directional bias of the candidate models, we use the mean errors (ME) defined as:

$$ME^j = \frac{1}{n}\sum_{s=1}^{n} e^j_{s,t} \qquad t = 2003, ..., 2022 \qquad s = 1, ..., 18 \tag{15}$$

A negative ME value suggests that the forecast over predicts the annual percent change, while a positive value indicates the opposite. A two-tailed t-test is performed on the sample of observations for each forecast step to determine whether the ME significantly differs from zero. Mean Errors found to be significantly different from zero indicate the presence of forecast bias. We perform the two-tailed t-test for each step in order to observe the dynamics of the forecast bias as their horizons slowly decrease from steps 1 to 18.

6 Results

To evaluate the out-of-sample performance of the current model used during FPO publications and the alternative model proposed here, we begin by estimating equations 11, 12, 13, and 15 using the point estimates at each of the 18 steps for the target years 2003 to 2022. The point estimates used during this process for the SARIMA model along with the observed values needed for these calculations were taken from the FPO historical values provided by the USDA ERS.[1] Section 2 describes how these values were calculated. Using the same target years, we estimated the ARRF model across 18 steps for each year using the same methodology as the SARIMA model to enable a clear comparison between the two. A description of the ARRF model is provided in section 4.5.

6.1 Forecast Accuracy

Performance evaluation results for both forecast models are presented in Table 2 where the out-of-sample accuracy of the models can be compared across forecast steps. As both forecast methods move from step 1 to step 18 the new information provided to the models results in the general decrease across all measures of accuracy indicating that short term

[1] The historical FPO estimates and realized values used in this study encompass the years 2003 to 2022 and can be found at https://www.ers.usda.gov/data-products/food-price-outlook/

forecast errors tend to be smaller than longer term forecast errors. ARRF generally out performs the SARIMA model across all steps during the test period with a few exceptions at step 18 for the MAE measure and step 12 for RMSE. The differences between the models performance decreases with the forecast horizon indicated that the ARRF model outperforms the SARIMA specifically when the horizons are larger.

Table 2: **Accuracy Measures**:Forecast for All Food CPI series(2003-2022)

	MAE		RMSE		Diebold-Mariano	
step	SARIMA	ARRF	SARIMA	ARRF	t-stat	p-val
1	1.38	0.56	2.13	0.74	-1.50	0.15
2	1.53	0.56	2.22	0.76	-1.62	0.12
3	1.51	0.48	2.21	0.58	-1.42	0.17
4	1.45	0.42	2.05	0.49	-1.74	0.10
5	1.28	0.35	1.78	0.45	-2.41	0.03
6	1.30	0.40	1.78	0.48	-1.91	0.07
7	1.09	0.40	1.42	0.54	-2.40	0.03
8	0.96	0.31	1.14	0.46	-2.93	0.01
9	0.67	0.26	0.87	0.33	-2.27	0.03
10	0.46	0.25	0.68	0.31	-1.72	0.10
11	0.47	0.26	0.68	0.47	-1.55	0.14
12	0.33	0.24	0.44	0.45	0.17	0.87
13	0.26	0.15	0.38	0.19	-0.75	0.46
14	0.14	0.12	0.21	0.16	-0.58	0.57
15	0.10	0.08	0.15	0.11	-0.52	0.61
16	0.07	0.06	0.11	0.08	-0.51	0.62
17	0.04	0.04	0.07	0.06	-0.29	0.78
18	0.01	0.04	0.04	0.04	0.17	0.87

Note: Mean Absolute Error (MAE) and Root Mean Squared Errors (RMSE) were calculated using errors for each step across the target years 2003 to 2022 using the Equations 13 and 12 respectively.

The accuracy measures in Table 2 suggests that there are differences between the SARIMA and ARRF model's performances. We test that hypothesis with a Mariano-Diebold test at each step following equation 14. The results of test can be found in the final columns of Table 2 and show that during steps 4 to 10 the ARRF and SARIMA models are significantly different from each other[2]. As the ARRF outperforms the SARIMA during these periods, by both measures of predictive accuracy, we conclude that it outperformed the SARIMA model at 7 of the 18 horizons reported for FPO target years and performed equally as well in the remaining horizons.

[2] Statistical significance was determined using the 90 percent confidence interval

6.2 Forecast Bias

Table 3 summarises the tests of forecast bias performed on each of the models for each individual step of the forecast. The Mean Error (ME) measure calculated using Equation 15 at each step acts as an indicator as to whether or not the the model over or under predicted the realized value at each step. Negative and positive values indicate and over and under prediction respectively. We test whether or not the ME values are significantly different from zero with two-tailed tests at each step that results in columns containing a test-statistic and its corresponding p-value for both models.

The SARIMA model was found to have error terms significantly different from zero[3] at 5 of the 18 horizons(steps: 1, 2, 3, 6, and 17). Similarly, ARRF model was found to have error terms significantly different from zero at 5 of the 18 horizons(steps: 3, 4, 6, 7, and 17) indicating the potential of forecast bias. The majority of the steps were during the first 6 months of the forecast at which point the known information is at its lowest point and the horizon is at its largest. During these periods both models suffered from under prediction. Both models had the opposite issue at step 17 where they were found to over predict the target value by a similar margin of error.

7 Conclusion

This study sought to explore using the ARRF model for FPO forecasting. Our analysis of the out-of-sample forecast performance from 2003 to 2022 demonstrates that the proposed ARRF model outperforms the currently used SARIMA model regarding predictive accuracy. After comparing the two forecasting models by the two most common accuracy measures, RMSE and MAE, the ARRF resulted in lower values across almost all of the 18 separate steps reported by the FPO.

Both methods performed better with shorter forecast horizons, as noted by a general decrease in the accuracy measures as the forecast moved from step 1 to 18. The ARRF outperforms the SARIMA model by the largest margins in the earlier steps of the forecast when the forecast horizons are their longest. As the forecast horizon shortened, the differences between the two forecasts decreased, and the estimates of both models began to converge with the realized value for the target year.

Upon careful examination of the DM test results, we found that the ARRF model's performance was on par with the SARIMA model at 11 of the 18 horizons, and superior at the remaining 7. Notably, the ARRF forecast excelled during steps 4 through 10 of the

[3] Statistical significance was determined using the 90 percent confidence interval

Table 3: **Tests for Bias**: Forecast for All Food CPI series(2003-2022)

	SARIMA			ARRF		
	ME	t-stat	p-val	ME	t-stat	p-val
1	0.87	1.95	0.07	0.23	1.45	0.16
2	1.01	2.24	0.04	0.24	1.43	0.17
3	0.92	1.98	0.06	0.23	1.90	0.07
4	0.74	1.69	0.11	0.21	2.02	0.06
5	0.36	0.90	0.38	0.08	0.77	0.45
6	0.78	2.10	0.05	0.23	2.36	0.03
7	0.46	1.51	0.15	0.28	2.69	0.01
8	-0.06	-0.23	0.82	-0.01	-0.10	0.92
9	-0.03	-0.15	0.88	-0.08	-1.05	0.31
10	0.09	0.55	0.59	0.06	0.88	0.39
11	-0.08	-0.52	0.61	-0.05	-0.47	0.64
12	-0.02	-0.15	0.88	-0.08	-0.78	0.45
13	0.00	0.06	0.96	-0.01	-0.17	0.86
14	0.01	0.31	0.76	-0.00	-0.09	0.93
15	0.02	0.43	0.67	-0.01	-0.21	0.83
16	-0.01	-0.40	0.69	-0.01	-0.56	0.58
17	-0.03	-2.33	0.03	-0.03	-1.91	0.07
18	0.00	0.57	0.58	0.00	0.38	0.71

Note: Mean Errors(ME) was calculated using errors for each step across the target years 2003 to 2022 using Equation 15.

forecast, where the horizon ranged from 15 to 9 months, further demonstrating its reliability and accuracy.

Both methods were found to under and overestimate the realized values depending on the forecast horizon. During earlier steps, both models were found to underestimate the annual food price inflation. At step 17, both models had issues with overestimation. While both models were found to have bias, the magnitude of forecast bias was larger for the SARIMA forecast.

Our findings strongly suggest that the ARRF model has the potential to significantly enhance the accuracy and reduce the bias of the annual food price inflation estimates reported during monthly FPO publications. The ARRF model offers these improvements in predictive accuracy over 7 of the 18 steps, without compromising the accuracy at the other steps currently provided by the SARIMA model, thereby highlighting its value and potential impact.

Much like the SARIMA model proposed by Maclachlan et al. (2022) , the ARRF model offers a standardized approach that can be easily applied to all of the food-related series covered by the FPO while offering the USDA-ERS the even greater flexibility to include exogenous variables into the model. Unlike the SARIMA, the ARRF method is capable of managing many complex and non-linear relationships. For example, tree-based models can easily include explanatory variables while avoiding the risk of over fitting that would eventually befall a similar approach with a SARIMA model. In fact, Coulombe notes the potential for even further increases to the predictive accuracy of tree-based methods with the inclusion of relevant macroeconomic variables. We intend to explore this topic further in the future.

References

Adjemian, M. K., Arita, S., Meyer, S., and Salin, D. (2023). Factors affecting recent food price inflation in the United States. *Applied Economic Perspectives and Policy.*

Box, G., G.M. Jenkins, G.C. Reinsel, and G.M. Ljung (2015). *Time Series Analysis: Forecasting and Control.* John Wiley & Sons, 5th edition.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (2017). *Classification And Regression Trees.* Routledge.

Buck, E., Hinz, M., Jiang, Y., Wen, X., Kuethe, T. H., Buck, E., Hinz, M., Jiang, Y., Wen, X., and Kuethe, T. H. (2023). The Rationality of USDA's Retail Food Price Inflation Forecasts. Technical report.

Canova, F. and Hansen, B. E. (1995). Are Seasonal Patterns Constant over Time? A Test for Seasonal Stability. *Journal of Business & Economic Statistics*, 13(3):237.

Coulombe, P. G. (2020). The Macroeconomy as a Random Forest.

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.

Hans Kunsch (1989). The Jackknife and The Bootstrap for General Stationary Observations. *The Annal of Statistics*, 17(3):1217–1241.

Harvey, D., Leybourne, S., and Newbold, P. (1997). Testing the equality of prediction mean squared errors. *International Journal of Forecasting*, 13(2):281–291.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning.* Springer New York, New York, NY.

Hyndman, R. J. and Khandakar, Y. (2008). Automatic Time Series Forecasting: The ¡b¿forecast¡/b¿ Package for ¡i¿R¡/i¿. *Journal of Statistical Software*, 27(3).

Isengildina-Massa, O., MacDonald, S., and Xie, R. (2012). A Comprehensive Evaluation of USDA Cotton Forecasts. *Journal of Agricultural and Resource Economics*, 37(1):98–113.

Joutz, F., Trost, R., Hallahan, C., Clauson, A., and Denbaly, M. (2000). Retail Food Price Forecasting at ERS the process methodology and performance from 1984 to 1997.

Kuhns, A., Volpe, R., Leibtag, E., and Roeger, E. (2015). United States Department of Agriculture How USDA Forecasts Retail Food Price Inflation. Technical report.

Kwiatkowski, D., Phillips, P. C., Schmidt, P., and Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root. *Journal of Econometrics*, 54(1-3):159–178.

Maclachlan, M. J., Chelius, C. A., and Short, G. (2022). Time-Series Methods for Forecasting and Modeling Uncertainty in the Food Price Outlook. Technical report.

Nakamura, E. (2008). Pass-through in retail and wholesale. In *American Economic Review*, volume 98, pages 430–437.

Schwarz, G. (1978). Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2):461–464.