IARIW 2021

IARIW 2021

Monday 23 – Friday 27 August

Statistisk sentralbyrå
Statistics Norway

Hedonic Imputation with Tree-based Decision Approaches

Shipei Zeng

(University of New South Wales and Shenzhen Research Institute of Big Data)

Paper prepared for the 36th IARIW Virtual General Conference
August 23-27, 2021
Session 2: The Potential and Challenges of Big Data and other Alternative Data in the
Production of Prices, National Accounts, and Measures of Economic Well-Being
Time: Tuesday, August 24, 2021 [14:00-16:00 CEST]

Hedonic Imputation with Tree-based Decision Approaches

Shipei Zeng

University of New South Wales and Shenzhen Research Institute of Big Data

August 11, 2021

Abstract

Linear hedonic regression is commonly utilised to estimate missing prices in cases where there is product entry and exit, or product "churn", but the linear assumption of prices in product characteristics is dubious. Actual consumer purchase patterns show that product characteristics are not perfectly substitutable so that the prediction capacity of linear models is challenged. I consider alternative estimations of hedonic prices by introducing tree-based machine learning models that are highly recommended for prediction accuracy. Particular attention is paid to the micro-economic explanation of tree-based models. A tree decision structure is compatible with consumer preferences when product characteristics are complements. Model performance metrics from (electronic-point-of-sale) scanner data confirm prediction accuracy gains from the appropriate model selection that follows consumer behaviour foundation. I find that random forests are the best fitted model with largest $\bar{R}^2$-type measures among a series of models. Price indexes with random forests display correct predictions that are robust in the single, double and full imputations. The variable importance estimated for product characteristics is consistent with the actual coefficients of hedonic functions in price simulation. It is advisable that tree-based decision approaches, especially random forests, can be effectively employed for unmatched products in hedonic imputation due to their prediction accuracy and compatibility with consumer utility types.

Keywords: hedonic imputation; machine learning; price indexes; unmatched products

1 Introduction

The compilation of price indexes is an essential instrument for our understanding of economy performance and will directly influence decision-making process of the government and industry organisations. For example, the calculation of consumer price indexes helps to measure the cost of consumer goods and services, which serves as an economic indicator for monetary policy and wage negotiations (Hill and Melser, 2008). The Reserve Bank conducts open-market operations to keep prices stable at a reasonable level, depending on inflation or deflation reflected by consumer price indexes. Employees seek wage escalations and employers determine wage adjustments when consumer price indexes increase. Price indexes play a key role in guiding government policies and labour market agreements. Common bilateral price indexes used by statistical offices include Laspeyres indexes, Paasche indexes, Fisher indexes and Törnqvist indexes. They capture the price movement of matched products, or balanced data where all products repeat in successive periods.

However, the main challenge for applying these indexes, is that new and disappearing products in a rapidly growing market result in unmatched items. Prices of new products in the base period are unavailable, and so are prices of disappearing products in the current period. To undertake unmatched products in practice, hedonic imputation has been utilised to estimate missing prices. Hedonic imputation is built upon on a hedonic model that assumes product prices to be decided by product characteristics. The linear hedonic model aligns a product with a number of characteristics and computes contributions of each characteristic to product prices. One unit increase in product characteristics leads to a corresponding increase in hedonic prices. Given specific product characteristics, missing prices of unmatched products can be estimated with hedonic regression.

The objective of this paper is to consider alternative accurate estimations of hedonic prices by introducing machine learning algorithms. I examine the prediction performance of linear regression models and tree-based machine learning approaches. Tree-based machine learning approaches, especially random forests, are recommended due to impressive prediction capacities in data analysis cases (Bajari et al., 2015). I use simulated scanner data to test whether machine learning achieves accuracy gains in price prediction. Price indexes are further computed with estimated prices for model comparison between linear regression and machine learning. Simulated scanner data are flexible in utility functions so that I can also inspect how prediction performance metrics respond to consumer preference types. Linear hedonic regression claims linear preferences on product characteristics while tree-based machine learning is compatible with Leontief preferences. A sensitivity analysis of model performance is necessary when consumer preferences vary in a wide range.

One possible factor by which the performance of hedonic imputation can be limited is heteroscedasticity. It has an adverse effect on linear regression. Estimators in an unweighted hedonic model will be inefficient under the heteroscedastic condition (Miller and Startz, 2019). A solution proposed by Diewert et al. (2009) is to use expenditure shares as weights and then apply weighted least squares (WLS). Expenditure shares interpret WLS regression from an economic perspective, but it can be argued whether expenditure shares are exactly actual weights within the context of WLS. Constant variance can be transformed from non-constant variance only if the unweighted model is divided by actual weights. Once expenditure shares are not actual WLS weights, the prediction of missing prices can be technically challenged.

Apart from heteroscedasticity, a more critical issue of hedonic imputation is the selection of function forms. Hedonic linear regression assumes that product prices are linear in product characteristics. Product characteristics are perfectly substitutable in the linear form. But consumers may find it hard to substitute one characteristic with another characteristic freely when making real purchases. The linear form may not represent preferences on product characteristics. It reduces the prediction accuracy of price imputation by assuming a linear relationship when the true function form is non-linear. Although linear hedonic regression can provide causal explanations of product characteristics, prediction accuracy takes priority over coefficient estimates for the purpose of price imputation.

A comparison between linear regression models and tree-based machine learning models is made in this paper to test the capability of estimating missing prices in unbalanced data sets where products enter or exit a market over periods. Linear models include ordinary least squares, feasible generalised least squares with value share weights or exponential weights, while tree-based models include regression trees, bagging trees and random forests. Random forests prove to be the best fitted model in in-sample and out-of-sample tests, with $\bar{R}^2$-type measures close to 1. Price indexes constructed by estimated prices also validate the prediction of random forests, indicating 60% correct values in the single, double and full imputations. In addition to price imputation, random forests provide variable importance that is analogous to coefficient estimates of linear models in terms of variable contributions. Variable importance quantifies the impact of independent variables on dependent variables. It helps to demystify the "black box" argument of many machine learning models.

An economic perspective is proposed to explain the remarkable prediction performance of tree-based machine learning compared with linear models. Linear models stand out when adopted to assess causal effects, but they are not recommended for price imputation. The linear set-up does not capture the relationship between product prices and product characteristics. In many cases, product characteristics, for example flavour and size, cannot be interchangeably treated. The consumer utility is incompatible with linear models that assume free substitutability among product characteristics. As opposed to linear models, tree-base machine learning matches the consumer behaviour foundation. To clarify this mechanism, the elasticity of substitution is allowed to vary so that simulations can cover prices in a range of utility types. For Leontief sub-utility where product characteristics are complements, tree-based models are suitable because the change of prices in a tree structure is jointly decided by a number of characteristics. Since prices will not necessarily increase or decrease due to a single characteristic, tree-based models follow the property of a Leontief function. For linear sub-utility where product characteristics are substitutes, linear models are generally better because the sub-utility is linear in product characteristics. Interestingly, random forests still have a remarkable fitting ability in the linear sub-utility. This algorithm uses random samples and variables that reconstitute the tree structure, so its prediction performance is stable regardless of substitutability between product characteristics.

This paper is a novel study to combine microeconomic explanations with tree-based machine learning models, including regression trees, bagging trees and random forests, to estimate prices on scanner data. Machine learning is widely adopted in predictive modelling cases, while its implementation in economics is still at an early stage. Concerns are expressed about the paucity of economic interpretations in machine learning. This paper explores the prediction mechanism of tree-based machine learning models, and explains why a tree structure is suitable for consumer preferences. Tree-based models are not only useful in predictive analytics, but also consistent with the microeconomic foundation of consumer behaviour. Using machine learning algorithms to predict missing prices for unbalanced products extends hedonic imputation options. These results are validated by real scanner data in the robustness check.

The remaining part of this paper is organised as follows. Section 2 lists available hedonic price approaches including linear hedonic regression and tree-based machine learning. Section 3 provides the generating process of scanner data based on consumer preferences. Section 4 presents the results by focusing on model selection criteria, in-sample and out-of-sample predictions, price indexes, variable importance and elasticity of substitution. Section 5 validates the results by applying models to scanner data from the real dataset. The final section concludes.

2 Price imputation approaches

2.1 Linear hedonic regression

Hedonic imputation is based on a hedonic model that relates prices to time effects and characteristics of certain commodities. Diewert (2003) worked on the originally sophisticated hedonic framework by Rosen (1974) and revised it in a more simplified way. I build upon the hedonic model modified by Diewert (2003) where a consumer theory approach confirms the economic interpretation of hedonic pricing. Suppose the consumer's utility of a hedonic product that has a vector of characteristics $\boldsymbol{z} = (z_1, \ldots, z_J)$ can be presented by $f(\boldsymbol{z})$. The utility $f(\boldsymbol{z})$ is defined as a separable sub-utility function. If a consumer acquires N units of these hedonic products, the total utility can be modelled equivalently as $f(N\boldsymbol{z})$. With the separable sub-utility function $Z = f(\boldsymbol{z})$ and an aggregate price ρ^t for one unit of Z, the hedonic aggregate price of product i in period t can be expressed as:

$$p_{it} = \rho_t f(\boldsymbol{z_{it}}) \tag{1}$$

The relationship between hedonic prices and the consumer utility is clear in this hedonic equation. For each unit of utility obtained from a hedonic product that has characteristics $\boldsymbol{z}$, a consumer needs to pay an aggregate price ρ_t. It separates the quality adjustment from the price change.

For hedonic regression, the separable sub-utility function is commonly assumed as a linear form. Available linear forms include log-log, semi-log and linear equations. To ensure that hedonic prices are always positive, the semi-log form of utility functions is preferred:

$$\ln f(\boldsymbol{z_{it}}) = \alpha + \sum_j \beta_j z_{ijt} \tag{2}$$

The semi-log form generates an unweighted time dummy hedonic regression model that is frequently employed in the literature (see for example de Haan (2010)):

$$\ln p_{it} = \alpha + \delta D_t + \sum_j \beta_j z_{ijt} + \epsilon_{it} \tag{3}$$

In the right hand of the hedonic regression model, $\ln \rho_t$ is replaced by δD_t, and ϵ_{it} is added. D_t refers to a time dummy variable and ϵ_{it} denotes an error term. α, β and δ are coefficients to be estimated.

An issue that needs to be addressed in the unweighted time dummy hedonic regression is heteroscedasticity, that is, the changing variance of error terms. Diewert et al. (2009) proposed a solution by using expenditure shares:

$$s_{it} = \frac{p_{it} q_{it}}{\sum_k p_{kt} q_{kt}} \tag{4}$$

p is the product price and q is the product quantity. Multiplying the unweighted hedonic model with the square root of expenditure shares leads to a weighted time dummy hedonic model:

$$s_{it}^{\frac{1}{2}} \ln p_{it} = s_{it}^{\frac{1}{2}} \alpha + s_{it}^{\frac{1}{2}} \delta D^t + s_{it}^{\frac{1}{2}} \sum_j \beta_n z_{ijt} + u_{it} \tag{5}$$

Thus the newly constructed error term u_{it} is equal to $s_{it}^{\frac{1}{2}} \epsilon_{it}$. For the case of heteroscedasticity where $Var(\epsilon_{it}) = \sigma_u^2 h_{it}$, the assumption of constant variance σ_u^2 for the error term u_{it} can be satisfied if the expenditure share s_{it} is exactly the actual weight $1/h_{it}$ within the context of WLS regression. Using expenditure shares as weights is favoured in hedonic regression for its economic explanation. Expenditure shares are monetary proportions of total purchases and they are literally close to the concept of weights. The limitation of expenditure share weighted regression, however, is that the expenditure share s_{it} is not necessarily the actual weight $1/h_{it}$ in WLS. This can reduce the prediction capacity when the regression is conducted for hedonic imputation.

If the economic explanation of weights is discarded, an optional approach can be considered for variance estimation. Assuming the conditional function h_{it} to be exponential, the conditional variance function can be estimated as what Wooldridge (2015) proposed:

$$\ln \hat{\epsilon}_{it}^2 = \theta + \lambda D_t + \sum_j \gamma_j z_{ijt} + v_{it} \tag{6}$$

The predicted value $\widehat{\ln \hat{\epsilon}_{it}^2}$ is then converted as the weight $1/\widehat{\hat{\epsilon}_{it}^2}$. The weights based on regression may be closer to actual weights than expenditure shares. But we need to be cautious that the WLS method does not aim at prediction enhancement. Even if we obtain actual weights in WLS and figure out unbiased estimators, accuracy gains of price prediction can still be limited.

2.2 Machine learning algorithms

Although linear hedonic regression can provide causal explanations of coefficients, the linear model is not the best option for hedonic price imputation. First, the fitting capacity of linear models is not stable in different scenarios. Linear models are suitable for consumer behaviour when the separable sub-utility function is linear, but they are not fitting non-linear hedonic prices. The prediction accuracy will decrease dramatically if the separable sub-utility function turns to be non-linear. Second, the linear assumption is not fully compatible with consumer preferences on product characteristics. The linear form of separable utility indicates that product characteristics are perfectly substitutable. It means that consumers will be at the same utility level if one product characteristic decreases and another product characteristic increases correspondingly. But in practice, product characteristics have their respective importance. Consumers may find it hard to substitute one characteristic with another characteristic freely. Third, linear models are not designed for price prediction. Linear regression features estimated coefficients that indicate the response of dependent variables to independent variables. Unbiased and consistent coefficients do not guarantee an excellent model prediction ability.

Compared with linear regression models, machine learning begins to attract economic researchers due to its striking prediction ability. Machine learning involves a great number of algorithms that can be categorised as supervised learning and unsupervised learning. Supervised learning applies to problems with output labels like classification and regression, while unsupervised learning applies to problems without output labels like clustering and dimensionality reduction. Bajari et al. (2015) evaluated traditional econometric models and machine learning algorithms for demand estimation, and found noticeable lower prediction errors of machine learning methods. Considering the importance of prediction in hedonic imputation cases, it is sensible to introduce machine learning to price estimation.

Random forests are highlighted in this paper for hedonic imputation. As one of the machine learning methods, random forests have demonstrated an outstanding prediction ability in model comparison organised by Bajari et al. (2015). Apart from prediction accuracy, random forests are preferred due to the tree structure implied by this algorithm. The tree structure is formed when the data set splits into two parts repeatedly at each node. A series of branches is produced according to these nodes and they jointly decide prediction values at terminating nodes. With the tree structure, prediction values will not be easily changed with the increase or decrease in a single independent variable. This structure is compatible with the Leontief separable sub-utility where utility remains unchanged unless multiple product characteristics change together. Therefore, tree-based algorithms are more appropriate for hedonic prices than linear models when product characteristics are not substitutes. In addition to random forests, I include regression trees and bagging trees for methodology clarification. Random forests are based on the method of bagging trees, and bagging trees are derived from regression trees.

To obtain the economic explainability of machine learning, I only include these tree-based models that can be potentially related with consumer preferences and utility functions. The complete machine learning framework, however, is more than what this paper has covered. A key component of the mechanism that enables machines to learn from data is an iterative training process. By evaluating the learning cost (the cost function or the loss function), machines are taught by the data to find appropriate parameters of the model so that the cost can be reduced in a desired direction. For example, neural networks, an algorithm designed to simulate how the human brain works, use backpropagation and gradient descent to find best weights and biases of the model. An initial attempt to fit the data produces the learning cost in neural networks. Then the algorithm evaluates the learning cost and updates the weights and biases, expecting a lower learning cost. This training process repeats until a satisfactory cost is produced. Considering the model interpretation, I choose not to include such training process and only focus on tree-based algorithms. But the iterative training process really reveals why machines can learn, and it may be adopted for future research if I seek to explain the learning capacity of these tree-based models.

2.2.1 Regression trees

Regression trees are a type of CART (classification and regression trees) algorithms. Breiman et al. (1984) initiated early attempts on CART that can be categorised as classification trees for discrete responses and regression trees for continuous responses. The tree structure facilitates a non-linear prediction model for hedonic prices. A more general form of hedonic regression can be expressed as:

$$\ln p_{it} = g(D_t, \boldsymbol{z_{it}}) + \epsilon_{it} \tag{7}$$

The unweighted time dummy hedonic regression adopts a linear form of $g(D_t, \boldsymbol{z_{it}})$ by assuming a linear separable sub-utility $f(\boldsymbol{z})$ on product characteristics. Regression trees discard the linear assumption and estimate hedonic prices in a non-parametric way. For the convenience of illustration, I use a general notation where $\ln p_{it}$ is denoted by y_i and $g(D_t, \boldsymbol{z_{it}})$ is denotes by $g(\boldsymbol{x_i})$:

$$y_i = g(\boldsymbol{x_i}) + \epsilon_i \tag{8}$$

$\boldsymbol{x}$ is the vector of independent variables $(x_1, \ldots, x_K)$. The objective of regression trees is to split observations into distinct and non-overlapping regions $\{R_1, \ldots, R_M\}$ such that least squares are satisfied within any region R_m. It is easy to see that the average value $\bar{y}_{R_m}$ within region R_m solves the function estimation:

$$\bar{y}_{R_m} = \underset{\hat{g}(\boldsymbol{x_i})}{\arg\min} \sum_{y_i \in R_m} (y_i - \hat{g}(\boldsymbol{x_i}))^2 \tag{9}$$

The key step is to figure out the set of regions that minimises the sum of squared errors:

$$\{R_1^*, \ldots, R_M^*\} = \underset{\{R_1, \ldots, R_M\}}{\arg\min} \sum_{R_m} \sum_{y_i \in R_m} (y_i - \bar{y}_{R_m})^2 \tag{10}$$

Ideally, I can enumerate all possible combinations of regions to decide the optimal set of regions, but it is computationally infeasible for a large number of observations. For example, to split 3000 observations into 10 regions, the enumeration approach forces me to examine $C(3000, 10) \approx 1.6 \times 10^{28}$ possibilities to find the optimal set of regions. Large data sets even require a huge amount of computational resource. To reduce excessive possibilities, regression trees utilise recursive binary splitting when growing the tree structure. For the independent variable x_k in $(x_1, \ldots, x_K)$, regression trees choose a partition point ψ_k to split y_i into two parts: the left part $\Psi_L = \{y_i \mid x_{ik} < \psi_k\}$ and the right part $\Psi_R = \{y_i \mid x_{ik} \geq \psi_k\}$. The objective of setting ψ_k is to minimise weighted mean squared errors (MSE) while least squares are satisfied in each part:

$$MSE^*(x_k) = \min_{\psi_k} \left(\sum_{y_i \in \Psi_L} P_L \frac{1}{N_L} (y_i - \hat{y}_{\Psi_L})^2 + \sum_{y_i \in \Psi_R} P_R \frac{1}{N_R} (y_i - \hat{y}_{\Psi_R})^2 \right) \tag{11}$$

Regression trees use the mean value of y_i in each part as the estimation, that is, $\hat{y}_{\Psi_L} = \bar{y}_{\Psi_L}$ and $\hat{y}_{\Psi_R} = \bar{y}_{\Psi_R}$. P_L is the proportion of observations that are split into the left part by ψ_k, N_L is the number of observations in the left part, and $\hat{y}_{\Psi_L}$ is the function estimation of y_i in the left part. P_R, N_R and $\hat{y}_{\Psi_R}$ also refer to these figures but in the right part. At the root node (the first level of partition), the computation of optimal weighted MSE loops for each independent variable. The best partition is determined by selecting the independent variable and the partition point that result in the least optimal weighted MSE. Then the partition repeats at internal nodes or split nodes (subsequent levels) until some terminating rule applies. A set of regions can be efficiently confirmed for regression trees. Note that efficiency is at the cost of accuracy. Recursive binary splitting is a greedy algorithm that carries out the optimal split at each single node. It neglects a better split that may appear when examining a few nodes ahead.

2.2.2 Bagging trees

To improve prediction accuracy, Breiman (1996) developed a bagging procedure, namely an acronym of bootstrap aggregating. Rather than obtaining a single value for each prediction, the bagging procedure takes an average value (or majority voting in classification issues) of predictions on multiple bootstrapped samples when these samples are drawn from original observations with replacement. Consider the original data set $\Omega = \{(y_i, \boldsymbol{x_i}) \mid i = 1, \ldots, N\}$ with N observations. The bagging procedure repeats drawing N random observations with replacement for B times to construct bootstrapped samples Ω_b, $b = 1, \ldots, B$. A number less than N for bootstrapped samples is also reasonable depending on computational efficiency. Each bootstrapped sample produces a trained model. Given input values, these trained models generate output values for prediction. The average value of predictions that are obtained from these trained models is the bagging prediction.

When the bagging procedure applies to regression trees, bagging trees are formed with regression trees growing on bootstrapped samples. The average value of predictions from regression trees is used as the prediction of bagging trees:

$$\hat{G}(\boldsymbol{x_i}) = \frac{1}{B} \sum_b \hat{g}_b(\boldsymbol{x_i}) \tag{12}$$

$\hat{G}(\boldsymbol{x_i})$ is the bagging prediction, and $\hat{g}_b(\boldsymbol{x_i})$ is the prediction on the b-th regression tree. Averagely speaking, the bagging prediction performs better than one single prediction that composes the bagging prediction. Given fixed input values, the squared error of bagging prediction is $(y_i - \hat{G}(\boldsymbol{x_i}))^2$ and the squared error of one regression tree prediction is $(y_i - \hat{g}_b(\boldsymbol{x_i}))^2$. Note that the average of squares is larger than or equal to the square of an average:

$$\begin{aligned}\sum_b \hat{g}_b^2 &= \sum_b \Big((\hat{g}_b - \hat{G})^2 + 2\hat{G}(\hat{g}_b - \hat{G}) + \hat{G}^2\Big) \\ &= \sum_b (\hat{g}_b - \hat{G})^2 + 2\hat{G}(\sum_b \hat{g}_b - B\hat{G}) + B\hat{G}^2 \\ &\geq B\hat{G}^2\end{aligned} \tag{13}$$

Then it can be demonstrated that the squared error of prediction averaged over all regression trees is larger than or equal to the squared error of bagging prediction:

$$\begin{aligned}\frac{1}{B}\sum_b (y_i - \hat{g}_b(\boldsymbol{x_i}))^2 = y_i^2 - 2y_i\frac{1}{B}\sum_b \hat{g}_b(\boldsymbol{x_i}) + \frac{1}{B}\sum_b \hat{g}_b^2(\boldsymbol{x_i}) \\ \geq (y_i - \hat{G}(\boldsymbol{x_i}))^2\end{aligned} \tag{14}$$

For each point estimation on $(y_i, \boldsymbol{x_i})$, the inequality indicates that bagging prediction is averagely better than a single prediction. If both sizes of the inequality are integrated over the data set Ω, the statement that bagging trees averagely outweigh a single regression tree will still hold true.

The benefits of bagging are more than reducing prediction errors. Since bootstrapping is a technique of random sampling with replacement, some observations may not be selected when one bootstrapped sample is produced. These observations are called out-of-bag (OOB) data because they are not situated in the bootstrapped bag. Models trained over bootstrapped samples can be tested on OOB data so that additional cross validation is not necessary in bagging trees. Besides model testing, it is convenient to use OOB data to measure variable importance. Variable importance helps to enhance the explainability of bagging trees. As a data-driven model, machine learning is often challenged because it is not fully explainable. Variable importance provides relative impacts of independent variables on the dependent variable, which specifies how independent variables relatively contribute to the prediction on the dependent variable.

2.2.3 Random forests

Although the bagging procedure tends to reduce prediction errors, it is primarily effective only when predictions are independent between bootstrapped samples. Accuracy gains of bagging trees will not be noticeable if predictions of trees are correlated with each other. To reduce the correlation of trees, Ho (1995) proposed a stochastic model where subspaces are selected over the whole feature space. The term "feature" in machine learning is analogous to the term "independent variable" in econometrics. A random subset of independent variables is selected for growing each tree. Since each tree grows on a different set of independent variables, the correlation of predictions can be reduced. This is the initial version of random forests. Breiman (2001) extended this algorithm by allowing for random selection of independent variables at each node, which further decouples these trees. The general procedure of random forests is equivalent to bagging trees according to Breiman (2001), except that independent variables are randomly sampled at each node. For a bootstrapped sample Ω_b in bagging trees, the feature space X is the set of all variables, that is, $X = \{x_1, \dots, x_K\}$. While for a bootstrapped sample in random forests, a subspace $X_s \subseteq X$ at each node is employed with independent variables randomly selected. The bagging procedure is once again conducted but with the subspace X_s. The random forest prediction is generated by taking an average value over predictions of trees. Both observations and variables are randomly sampled in random forests.

I have demonstrated that the bagging prediction is averagely better than a single tree prediction. At each point estimation, the bagging prediction provides a lower squared error than squared errors averaged over all trees. I can then integrate squared errors over all data points to see how tree correlation affects prediction accuracy. The sum of squared errors of bagging trees is expressed as:

$$\begin{aligned}\sum_i (y_i - \hat{G}(\boldsymbol{x_i}))^2 &= \sum_i \left(y_i - \frac{1}{B}\sum_b \hat{g}_b(\boldsymbol{x_i}) \right) \left(y_i - \frac{1}{B}\sum_d \hat{g}_d(\boldsymbol{x_i}) \right) \\ &= \frac{1}{B^2}\sum_i \sum_b \sum_d \left(y_i - \hat{g}_b(\boldsymbol{x_i})\right)\left(y_i - \hat{g}_d(\boldsymbol{x_i})\right)\end{aligned} \tag{15}$$

$\hat{G}(\boldsymbol{x_i})$ is the bagging prediction, $\hat{g}_b(\boldsymbol{x_i})$ is the prediction on the b-th regression tree, and $\hat{g}_d(\boldsymbol{x_i})$ is the prediction on the d-th regression tree. $\hat{g}_b(\boldsymbol{x_i})$ and $\hat{g}_d(\boldsymbol{x_i})$ are rotationally symmetric. The difference of b and d is used to show that these predictions of regression trees are summed in different orders. Recall $\hat{g}_b(\boldsymbol{x_i})$ is the average value of y_i within a partition region. Therefore, $\left(y_i - \hat{g}_b(\boldsymbol{x_i})\right)\left(y_i - \hat{g}_d(\boldsymbol{x_i})\right)$ is a covariance-type term. The covariance-type term can be related to a sample correlation coefficient:

$$\rho_{bd} = \frac{\sum_i \left(y_i - \hat{g}_b(\boldsymbol{x_i})\right)\left(y_i - \hat{g}_d(\boldsymbol{x_i})\right)}{\sqrt{\sum_i (y_i - \hat{g}_b(\boldsymbol{x_i}))^2}\sqrt{\sum_i (y_i - \hat{g}_d(\boldsymbol{x_i}))^2}} \tag{16}$$

The coefficient ρ_{bd} captures the correlation between $\hat{g}_b(\boldsymbol{x_i})$ and $\hat{g}_d(\boldsymbol{x_i})$. Based on ρ_{bd}, a weighted correlation coefficient $\bar{\rho}$ is defined as:

$$\bar{\rho} = \frac{\sum_b \sum_d \rho_{bd}\sqrt{\sum_i (y_i - \hat{g}_b(\boldsymbol{x_i}))^2}\sqrt{\sum_i (y_i - \hat{g}_d(\boldsymbol{x_i}))^2}}{(\sum_b \sqrt{\sum_i (y_i - \hat{g}_b(\boldsymbol{x_i}))^2})^2} \tag{17}$$

Using the weighted correlation coefficient, the sum of squared errors of bagging trees can be written as:

$$\begin{aligned}\sum_i (y_i - \hat{G}(\boldsymbol{x_i}))^2 &= \bar{\rho}\frac{1}{B^2}\left(\sum_b \sqrt{\sum_i (y_i - \hat{g}_b(\boldsymbol{x_i}))^2} \right)^2 \\ &= \bar{\rho}\left(\frac{1}{B}\sum_b \sqrt{\sum_i (y_i - \hat{g}_b(\boldsymbol{x_i}))^2} \right)^2 \\ &\leq \bar{\rho}\frac{1}{B}\sum_b \sum_i \left(y_i - \hat{g}_b(\boldsymbol{x_i})\right)^2\end{aligned} \tag{18}$$

The inequality is produced because the square of an average is no greater than the average of squares (see the proof in bagging trees). It can be seen from the inequality that the sum of squared errors of bagging prediction is lower than or equal to $\bar{\rho}$ times the sum of squared errors averaged over all trees. This is the reason why random forests take random selection on variables. In a standard regression tree, a few variables may always be chosen at every node even though their corresponding mean squared errors are only slightly lower. These variables dominate the recursive binary splitting when a tree grows. Random forests allow independent variables to be randomly sampled so that every variable can be possibly selected. This ensures the variety of trees and reduces their correlation. With the correlation of predictions on trees dropping down, the prediction bias is diminished in random forests.

3 Scanner data simulation

The data set adopted to compare prediction performance in this paper is constructed by Monte Carlo simulation to approximate real scanner data. Scanner data include historical information on prices, quantities and characteristics of commodities that can be obtained by scanning corresponding bar codes. These bar codes specify various product items. I simulate 1000 different products in the scanner data set and they are labelled with Universal Product Codes (UPC). These fictional products are assumed to appear in consecutive 52 weeks, that is, a one-year period without seasonal adjustment. 70% of the complete observations are randomly selected as training data and the remaining 30% observations are selected as testing data. I run different models on training data and measure the performance of price prediction on testing data.

The most distinctive feature that makes scanner data different from online web-scraping data, is the quantity information. Raw scanner data in databases like

Dominick's should include two quantity-related variables: the number of items sold and the number of items in a bundle. Dividing the number of items sold by the number of items in a bundle generates the number of bundles sold, which is the product quantity q in this paper. To set a solid microeconomic foundation for data simulation, I follow the construction of artificial data from Diewert and Fox (2017) where a constant elasticity of substitution (CES) function underlies preferences of consumers. Consider the utility function with constant elasticity of substitution:

$$U = \left(\sum_i a_i^{\frac{1}{\sigma}} q_i^{\frac{\sigma-1}{\sigma}} \right)^{\frac{\sigma}{\sigma-1}} \tag{19}$$

where q_i is the product quantity, a_i is a series of positive parameters amounting to 1, and σ is the elasticity of substitution. The property of CES utility function depends on the value of σ. When σ approaches 0, it turns to be a Leontief utility function that indicates perfect complements among products. When σ approaches infinity, it turns to be a linear utility function that indicates perfect substitutes among products. I choose $\sigma = 1$ for scanner data simulation so that the utility function approaches a Cobb-Douglas form that is an intermediate state regarding product substitution (technically $\sigma = 1.0001$ for programming convenience). Given the price p_i on item i, it can be demonstrated that the corresponding expenditure share is:

$$s_i = \frac{a_i p_i^{1-\sigma}}{\sum_i a_i p_i^{1-\sigma}} \tag{20}$$

I assume that the expenditure e follows a uniform distribution in the range of $[0, 1]$ during different periods. Product quantities can be computed straightforwardly:

$$q_i = \frac{e s_i}{p_i} \tag{21}$$

This process repeats for 52 weeks and the issue of product quantity in simulated scanner data is resolved based on the CES utility function.

A common assumption of hedonic imputation is that the separable sub-utility function $f(\boldsymbol{z})$ adopts a semi-log form. The $\ln f(\boldsymbol{z})$ is formed by aggregating linear terms of product characteristics. This is a strong assumption because characteristics may not be substitutes. For example, the volume of soft drinks cannot easily replace the flavour of them. Consumer may not choose to lose their favourite flavours by receiving a larger size of drinks. To be consistent with the product quantity generating process, the separable sub-utility function in this paper is assumed to follow a CES function form rather than a linear function:

$$\ln f(\boldsymbol{z}) = \left(\sum_j \beta_j^{\frac{1}{\sigma}} z_j^{\frac{\sigma-1}{\sigma}} \right)^{\frac{\sigma}{\sigma-1}} \tag{22}$$

When σ is large enough, it approaches the semi-log form that is frequently used by traditional hedonic imputation. I allow σ to vary around 1 so that the model performance can be compared in various elasticity scenarios. All product characteristics follow a uniform distribution on $[0, 1]$. The time effect δ is 0 in the first week and follows a normal distribution in other periods. Heteroscedasticity is included by conditioning the error term on an exponential function. Specifically, product prices are constructed in the following way:

$$\ln p_{it} = \alpha + \sum_k \delta_k D_k + \left(\sum_j \beta_j^{\frac{1}{\sigma}} z_{ijt}^{\frac{\sigma-1}{\sigma}} \right)^{\frac{\sigma}{\sigma-1}} + \epsilon_{it} \tag{23}$$

$$\delta_k \sim \mathcal{N}(0, \sigma_\delta^2) \tag{24}$$

$$z_{ijt} \sim \mathcal{U}(0, 1) \tag{25}$$

$$\epsilon_{it} \sim \mathcal{N}(0, \sigma_\epsilon^2) \tag{26}$$

$$\sigma_\epsilon^2 = h(\boldsymbol{\delta}, \boldsymbol{z}_{it}) \sigma_u^2 \tag{27}$$

$$h(\boldsymbol{\delta}, \boldsymbol{z}_{it}) = exp(\alpha + \sum_k \delta_k D_k + \sum_j \beta_j z_{ijt}) \tag{28}$$

α refers to the constant term. δ_k denotes the time effect and D_k denotes the time dummy for $k = 2, \ldots, 52$. β_j is the coefficient of product characteristics and I set up $\beta_1 = 0.4, \beta_2 = 0.3, \beta_3 = \beta_4 = \beta_5 = 0.1$ for five product characteristics. The variance of error terms is based on a common variance σ_u^2 and a function $h(\boldsymbol{\delta}, \boldsymbol{z}_{it})$ conditioning on time effects and product characteristics. I choose an exponential form of $h(\boldsymbol{\delta}, \boldsymbol{z}_{it})$ to create severe heteroscedasticity. Table 1 lists parameters adopted for the simulation of product prices.

The summary statistics of simulated scanner data and assisting variables are presented in Table 2. The variable p refers to the retail price of a bundle of commodities and the variable q indicates the number of bundles sold. Note that the variable q is multiplied by 1000 for the purpose of being displayed clearly in the table. All product characteristics employed in the simulation are numerical variables, including z_1, z_2, z_3, z_4 and z_5. The variable ϵ is sampled from a normal distribution with different variances so that heteroscedasticity is simulated. Other variables such as the coefficient a in the CES utility, the expenditure share s and the total expenditure e in each period assist to compute product quantities. With these assisting variables, a microeconomic foundation is provided for scanner data simulation.

Table 1: Parameters for hedonic price simulation

Parameters	Definitions	Values
α	Constant term	1
β_1	Coefficient of product characteristic 1	0.4
β_2	Coefficient of product characteristic 2	0.3
β_3	Coefficient of product characteristic 3	0.1
β_4	Coefficient of product characteristic 4	0.1
β_5	Coefficient of product characteristic 5	0.1
σ	Elasticity of substitution	1.0001
σ_δ^2	Variance of time effects	0.2
σ_u^2	Common variance of error terms	0.01

Note: σ is 1.0001 rather than 1 for programming convenience. Following the literature about the CES utility function, I use σ to denote elasticity of substitution and it should be distinguished with the standard deviations σ_δ and σ_u.

Table 2: Summary statistics of scanner data

	Definitions	Mean	St. Dev.	Min	Max
Product features					
p	Retail prices	19.492	15.825	1.932	185.016
$1000q$	Number of bundles	0.050	0.061	0.000	0.616
z_1	Characteristic 1	0.499	0.289	0.000	1.000
z_2	Characteristic 2	0.501	0.288	0.000	1.000
z_3	Characteristic 3	0.502	0.288	0.000	1.000
z_4	Characteristic 4	0.501	0.287	0.000	1.000
z_5	Characteristic 5	0.501	0.290	0.000	1.000
Assisting set-up					
ϵ	Error terms	0.000	0.046	-0.266	0.266
a	Utility coefficients	0.001	0.001	0.000	0.002
s	Expenditure shares	0.001	0.001	0.000	0.002
e	Expenditures	0.590	0.286	0.009	0.996

Note: Zero values are caused by rounding numbers to 3 decimal places for display while original values are not necessarily zero. I multiply the number of product bundles by 1000 for the purpose of displaying it in detail. Heteroscedasticity is considered for error terms. Coefficients in the CES utility function, expenditures shares and expenditures are used to generate product quantities. The full data set includes 52,000 observations.

After applying these rules for 52 weeks and 1000 products, the data set has been scaled to include 52,000 observations. They are randomly divided into the training data set and the testing data set with a common ratio of 70% versus 30%. The training set is adopted to generate estimation models. These models are evaluated in the testing data set by measuring the performance of predicting hedonic prices. I draw on a series of information criteria so that the model performance can be evaluated from a variety of perspectives. The testing data set is also used as a benchmark when prices indexes are constructed. Prices in the testing data are treated as actual prices to replace predicted prices in single, double and full hedonic imputations. Price indexes are compared to determine which method has resulted in the measurement of price change closest to the testing data.

4 Model performance

4.1 Model selection criteria

Researchers use R squared (R^2) and adjusted R squared ($\bar{R}^2$) to evaluate the performance of estimation methods and determine the best fitted model. R^2 measures the fraction of variance explained in terms of the total variance, while additional variables in the model will spuriously increase R^2 even though the prediction performance is not really improved. $\bar{R}^2$ avoids this misleading information by posing penalty to the number of variables and so it serves as a more accurate indicator. Apart from being used as an indicator to identify which model outweighs, $\bar{R}^2$ is selected for characterising the quality of the model fit, that is, how $\bar{R}^2$ of a certain model is close to 1.

Due to the benefits of $\bar{R}^2$ on model comparison and model fit quality, Fox (2000) facilitated the transformation of some information criteria into the same pattern as

$\bar{R}^2$. The construction of $\bar{R}^2$-type expressions starts with the basic element R^2:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y}_i)^2} \tag{29}$$

where y_i denotes the observed value, $\hat{y}_i$ refers to the fitted value produced by model estimations, and $\bar{y}_i$ indicates the mean value of observed data. Drawing on R^2, Table 3 summarises options of available information criteria that convey the same dimension of information as $\bar{R}^2$. In Table 3, N denotes the number of observations, and J indicates the number of independent variables. Model selection criteria of AIC (Akaike's information criterion), SC (Schwartz's criterion, or Bayesian information criterion), and HQ (Hannan and Quinn criterion) include a penalised log likelihood form that is related to the sum of squared errors. The sum of squared errors bridges to the construction of $\bar{R}^2$. Fox (2000) defined corresponding $\bar{R}^2$-type expressions for AIC, SC and HQ by extracting a special item that consists of the sum of squared errors and a penalty coefficient, dividing it by the sum of squared total, and subtracting it from 1. Other model selection criteria such as Jp, Sp and GCV (generalised cross validation criterion) are directly enclosed with the sum of squared errors, making the transformation even straightforward. By utilising these modified information criteria, I can compare the performance of models as well as the quality of the model fit for the aforementioned OLS, FGLS and machine learning specifications.

Table 3: Information criteria and related $\bar{R}^2$-type expressions

Expressions	References
$\bar{R}^2 = 1 - (N/(N-J))(1-R^2)$	Theil (1961)
$\bar{R}^2(AIC) = 1 - (\exp(2J/N))(1-R^2)$	Akaike (1973)
$\bar{R}^2(SC) = 1 - N^{(J/N)}(1-R^2)$	Schwarz et al. (1978)
$\bar{R}^2(HQ) = 1 - (\ln N)^{(2J/N)}(1-R^2)$	Hannan and Quinn (1979)
$\bar{R}^2(Jp) = 1 - ((N+J)/(N-J))(1-R^2)$	Amemiya (1980)
$\bar{R}^2(Sp) = 1 - (N^2/((N-J)(N-J-1)))(1-R^2)$	Hocking (1976)
$\bar{R}^2(GCV) = 1 - (1-J/N)^{-2}(1-R^2)$	Craven and Wahba (1979)

Note: AIC (Akaike's information criterion), SC (Schwartz's criterion, or Bayesian information criterion) and HQ (Hannan and Quinn criterion) feature a penalised log likelihood form. Jp, Sp and GCV (generalised cross validation criterion) feature the sum of squared errors. These criteria are transformed into $\bar{R}^2$-type expressions according to Fox (2000).

These $\bar{R}^2$-type expressions should be computed with caution when FGLS methods are considered. Willett and Singer (1988) clarified the use of R^2 in the weighted least squares. Variables combined with weights in WLS are called transformed variables. The output of R^2 provided by statistical computing packages is based on these transformed variables. Denote y^* as the transformed variable $w^{\frac{1}{2}}y$, and the R squared of WLS is computed by:

$$R^2_{WLS} = 1 - \frac{\sum_i (y_i^* - \hat{y}_i^*)^2}{\sum_i (y_i^* - \bar{y}_i^*)^2} \tag{30}$$

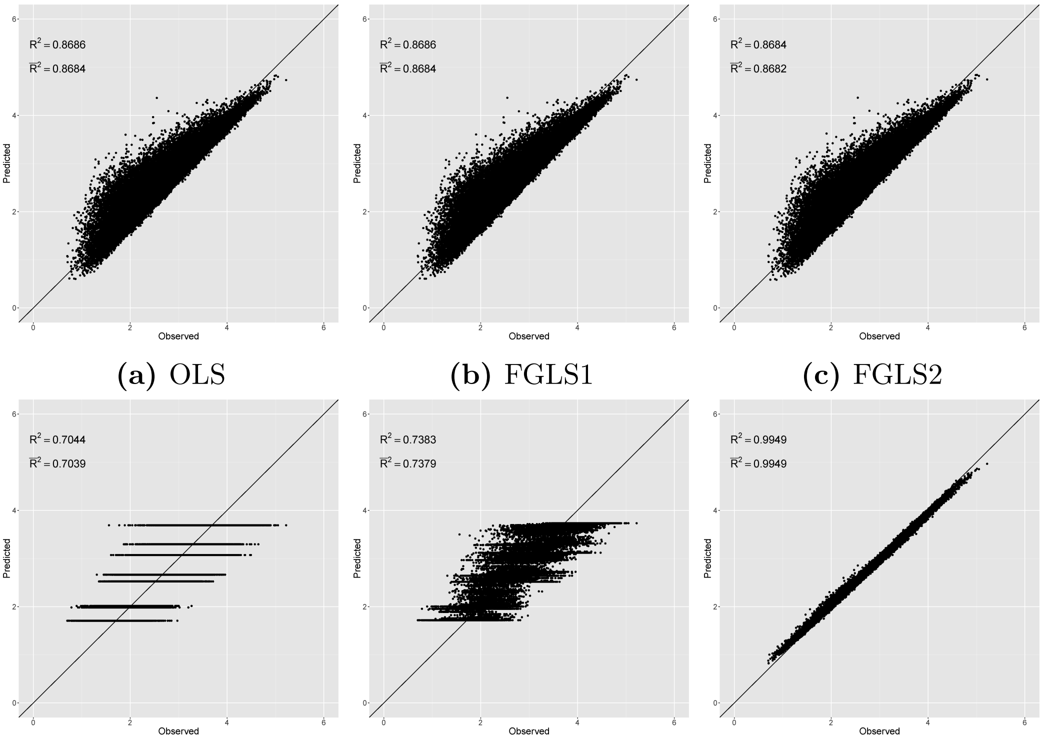
Since WLS minimises the sum of squared errors for the transformed variable, R^2_{WLS} will frequently become larger than R^2 given an appropriate weighting structure. This may lead to the interpretation that WLS regression has better performance in prediction, though it is not necessarily the truth. When R^2_{WLS} is significantly larger than R^2, estimated coefficients in OLS and WLS may be almost the same, leading to similar fitted values between OLS and WLS. Using R^2_{WLS} will mislead the performance comparison for these models. Additionally, R^2 needs to be computed manually in the FGLS regression rather than taking R^2_{WLS} reported by the statistical software because our focus is on the original variable, not the transformed variable.

The process of determining model performance is running models on the training data set prior to price prediction on the testing data. The transformed variable $w^{\frac{1}{2}}y$ cannot be predicted with the testing data set since the information of weights is unknown.

4.2 In-sample and out-of-sample prediction

I firstly conduct models in the training data set so that the in-sample estimation of prices can be carried out. These models are OLS, FGLS with value share weights from Equation 4 (FGLS1), FGLS with exponential weights from Equation 6 (FGLS2), regression trees, bagging trees and random forests. Figure 1 provides the comparison between observed prices and fitted prices (both in logarithmic forms) for them. Predicted prices by OLS tend to be higher than observed prices, indicating an over-estimating bias. Recall that in the scanner data simulation, a Cobb-Douglas function of product characteristics is included to construct product prices. But the OLS model estimates prices by assuming a linear form of product characteristics. This results in a systematic prediction bias, which is an upward bias in this case. The FGLS models are also built on linear regression so that they share the same upward bias as OLS. Compared with OLS, FGLS methods generate different estimated coefficients by introducing a weighting function. FGLS1 denotes weighted least squares using value shares as weights while FGLS2 denotes weighted least squares using an natural exponential weighting function. However, plots of observed prices and predicted prices for FGLS1 and FGLS2 are fairly close to that of OLS, showing limited prediction accuracy gains. Note the weighting function in FGLS2 exactly follows the heteroscedasticity pattern in scanner data simulation. It can be found that solving the heteroscedasticity problem does not remarkably enhance the prediction ability when comparing Figures 1a and 1c.

To explore models that exclude the linear assumption, the prediction results of machine learning models are plotted. Regression trees produce a set of discrete predicted prices in Figure 1d. This special prediction type is due to the tree-based algorithm that develops into terminating nodes (see Figure 2). A limited number of predicted values are generated and the prediction performance of regression trees is unsatisfactory. Bagging trees undertake this issue by taking an average value of all trees and allowing predicted values to be between discrete levels. The transition from Figure 1d to Figure 1e implies the bootstrapping and averaging process in bagging trees. Random forests move into even higher flexibility since variable selection is randomly decided. This relaxes the restriction that predicted values must be between multiple levels. Prediction performance is greatly improved and predicted prices by random forests are almost equal to observed prices in Figure 1f.



(a) OLS **(b)** FGLS1 **(c)** FGLS2

(d) Regression trees **(e)** Bagging trees **(f)** Random forests

Figure 1: In-sample observed prices and predicted prices

A closer inspection of R^2 and $\bar{R}^2$ reveals that the random forest model captures the

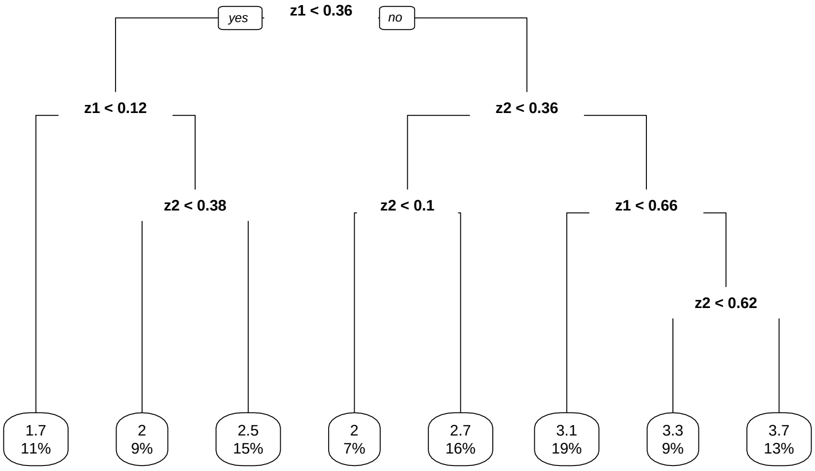


Figure 2: Nodes and branches in regression trees

most variability (0.99) of product prices around the average value. Linear models present a relatively weak performance. OLS, FGLS with value shares and FGLS with the natural exponential function have similar R^2 and $\bar{R}^2$ at approximately 0.87. However, tree-based models like regression trees and bagging trees do not match random forests in price prediction. Only 0.7 variability is obtained by regression trees and 0.74 variability is obtained by bagging trees. In addition to R^2 and $\bar{R}^2$, other alternative information criteria have been adopted to quantify how estimated prices fit actual prices. It can be seen from various $\bar{R}^2$-type expressions in Table 4 that random forests once again extract the most detailed information from the training data set. The performance metrics of OLS and FGLS methods are less impressive than those of random forests, but outweigh the results of regression trees and bagging trees. From the perspective of the strictest criterion SC, random forests score 0.99 for $\bar{R}^2(SC)$, demonstrating a high quality of the model fit capacity. What stands out in the last three rows is that $\bar{R}^2(Jp)$, $\bar{R}^2(Sp)$ and $\bar{R}^2(GVC)$ share the same values, which is due to the rounding of numbers to four decimal places.

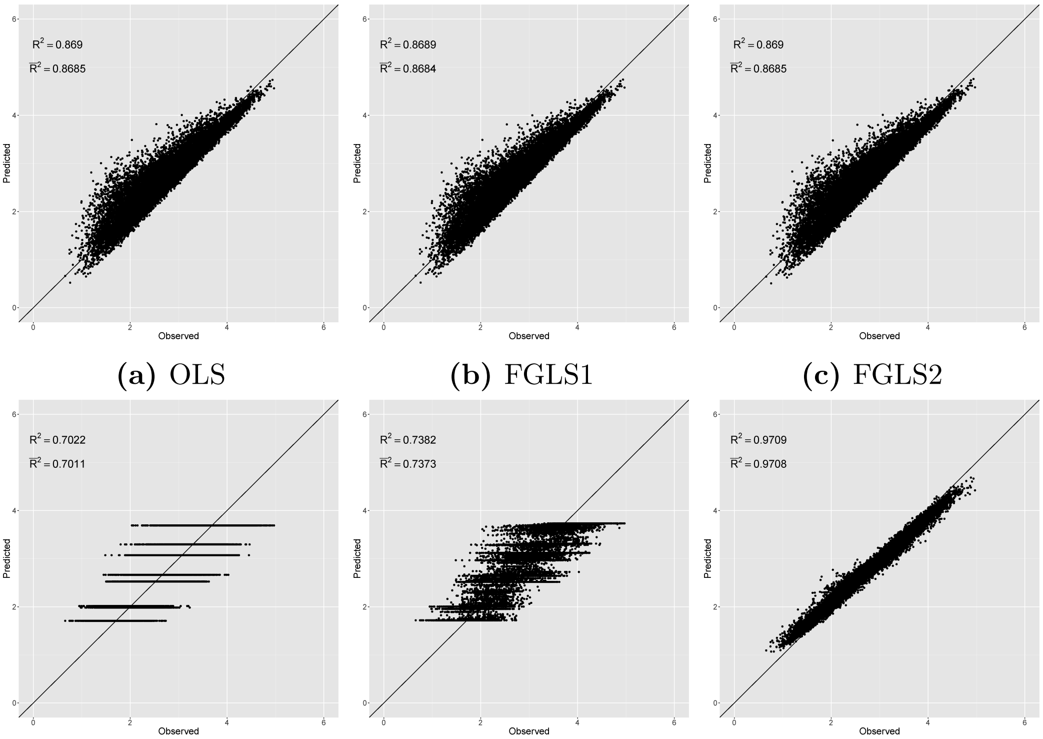
In-sample prediction results are not strongly convincing to determine model perfor-

Table 4: In-sample prediction performance in R^2 types

	OLS	FGLS		Machine learning		
		Value shares	Exponents	Regression trees	Bagging trees	Random forests
	(1)	(2)	(3)	(4)	(5)	(6)
R^2	0.8686	0.8686	0.8684	0.7044	0.7383	0.9949
$\bar{R}^2$	0.8684	0.8684	0.8682	0.7039	0.7379	0.9949
$\bar{R}^2(AIC)$	0.8682	0.8682	0.8680	0.7035	0.7375	0.9949
$\bar{R}^2(SC)$	0.8665	0.8664	0.8663	0.6996	0.7340	0.9948
$\bar{R}^2(HQ)$	0.8677	0.8676	0.8675	0.7022	0.7364	0.9949
$\bar{R}^2(Jp)$	0.8682	0.8682	0.8680	0.7035	0.7375	0.9949
$\bar{R}^2(Sp)$	0.8682	0.8681	0.8680	0.7035	0.7375	0.9949
$\bar{R}^2(GCV)$	0.8682	0.8682	0.8680	0.7035	0.7375	0.9949

Note: Same values occur due to rounding decimals.

mance due to the potential over-fitting problem. Models with outstanding in-sample performance may not guarantee the same prediction accuracy for out-of-sample data. The ability of random forests to fit external data may be suspected because the large R^2 of random forests can be produced by taking an overly complex approximation to the training data. To avoid the over-fitting problem, I conduct these parametric and non-parametric estimations again but on the testing data set. Prediction results of the testing data are evident in Figure 3 with observed prices on the horizontal axis and predicted prices on the vertical axis. Still, the random forest algorithm presents extraordinary prediction performance with R^2 and $\bar{R}^2$ at almost 0.97, followed by a series of linear models with R^2 and $\bar{R}^2$ at 0.87. Predictions with bagging trees report R^2 and $\bar{R}^2$ that are lower than those of random forests, but they outweighs the predictions with regression trees. Table 5 describes optional information criteria in $\bar{R}^2$-type expressions that have been employed apart from R^2 and $\bar{R}^2$. The most useful information can be obtained through the price estimation with random forests, which is followed by the methods of OLS and FGLS. Regression trees and bagging trees are relatively weak in price imputation. Specifically, price prediction with random forests scores 0.97 for $\bar{R}^2(SC)$ which is the most parsimonious indicator among these information criteria. Note that in the last three rows the differences between $\bar{R}^2(Jp)$, $\bar{R}^2(Sp)$ and $\bar{R}^2(GVC)$ are eliminated because the number rounding is associated with finite decimal places.



(a) OLS **(b)** FGLS1 **(c)** FGLS2

(d) Regression trees **(e)** Bagging trees **(f)** Random forests

Figure 3: Out-of-sample observed prices and predicted prices

4.3 Price indexes in product churn

Hedonic imputation covers more than pure price prediction. Price prediction is used to further figure out the price indexes in the context of new and disappearing products. Given any two periods of product churn, those products that appear in both periods are defined as matched items. Unmatched items are composed of new products and disappearing products. New products exist in the current period while disappearing products exist in the base period. Since common price indexes such as Fisher indexes and Törnqvist indexes are based on matched price data, missing

Table 5: Out-of-sample prediction performance in R^2 types

	OLS	FGLS		Machine learning		
		Value shares	Exponents	Regression trees	Bagging trees	Random forests
	(1)	(2)	(3)	(4)	(5)	(6)
R^2	0.8690	0.8689	0.8690	0.7022	0.7382	0.9709
$\bar{R}^2$	0.8685	0.8684	0.8685	0.7011	0.7373	0.9708
$\bar{R}^2(AIC)$	0.8681	0.8679	0.8680	0.7000	0.7363	0.9707
$\bar{R}^2(SC)$	0.8644	0.8643	0.8643	0.6917	0.7290	0.9699
$\bar{R}^2(HQ)$	0.8669	0.8667	0.8668	0.6973	0.7339	0.9705
$\bar{R}^2(Jp)$	0.8681	0.8679	0.8680	0.7000	0.7363	0.9707
$\bar{R}^2(Sp)$	0.8681	0.8679	0.8680	0.7000	0.7363	0.9707
$\bar{R}^2(GCV)$	0.8681	0.8679	0.8680	0.7000	0.7363	0.9707

Note: Same values occur due to rounding decimals.

values regarding these new and disappearing products need to be completed with appropriate estimation models. The product churn can be simulated by the training data set that consists of matched items and unmatched items. The testing data set assists to provide actual price information of unmatched items. In the out-of-sample test, I have compared the prediction accuracy of different methods on testing data. These out-of-sample price estimations as well as actual prices in the testing data set are applied to the construction of price indexes in product churn.

Taking the Törnqvist index as an example, de Haan (2010) considered the bilateral index of two periods and proposed the single imputation:

$$\hat{P}_{T,SI} = \prod_{i \in U_M} \left(\frac{p_i^1}{p_i^0} \right)^{0.5(s_i^0 + s_i^1)} \prod_{i \in U_D} \left(\frac{\hat{p}_i^1}{p_i^0} \right)^{0.5(s_i^0)} \prod_{i \in U_N} \left(\frac{p_i^1}{\hat{p}_i^0} \right)^{0.5(s_i^1)} \tag{31}$$

U_M is the set of matched products, U_D is the set of disappearing products and U_N is the set of new products. Actual prices in the base period and in the current period are denoted as p_i^0 and p_i^1 correspondingly, while missing values are estimated as $\hat{p}_i^0$ and $\hat{p}_i^1$ for new and disappearing items. s_i^0 refers to the value share of product i in the base period and s_i^1 refers to that in the current period. With price estimations $\hat{p}_i^0$

and $\hat{p}_i^1$, the $\hat{P}_{T,SI}$ price index can be computed. It features the "single" imputation because only missing values of unmatched products are estimated to compute price indexes. To apply the prediction method further, the double imputation can be defined by replacing not only missing values but also actual prices of disappearing and new products with estimated prices:

$$\hat{P}_{T,DI} = \prod_{i \in U_M} \left(\frac{p_i^1}{p_i^0} \right)^{0.5(s_i^0 + s_i^1)} \prod_{i \in U_D} \left(\frac{\hat{p}_i^1}{\hat{p}_i^0} \right)^{0.5(s_i^0)} \prod_{i \in U_N} \left(\frac{\hat{p}_i^1}{\hat{p}_i^0} \right)^{0.5(s_i^1)} \tag{32}$$

If price estimations are systematically biased, allowing numerators and denominators to be both estimated prices in the double imputation may cancel out the bias. Similarly, the full imputation is defined by replacing all prices of matched and unmatched products with estimated prices:

$$\begin{aligned} \hat{P}_{T,FI} &= \prod_{i \in U_M} \left(\frac{\hat{p}_i^1}{\hat{p}_i^0} \right)^{0.5(s_i^0 + s_i^1)} \prod_{i \in U_D} \left(\frac{\hat{p}_i^1}{\hat{p}_i^0} \right)^{0.5(s_i^0)} \prod_{i \in U_N} \left(\frac{\hat{p}_i^1}{\hat{p}_i^0} \right)^{0.5(s_i^1)} \\ &= \prod_{i \in U^0} \left(\frac{\hat{p}_i^1}{\hat{p}_i^0} \right)^{0.5(s_i^0)} \prod_{i \in U^1} \left(\frac{\hat{p}_i^1}{\hat{p}_i^0} \right)^{0.5(s_i^1)} \end{aligned} \tag{33}$$

The hedonic imputation of Törnqvist indexes can be extended to a variety of price indexes. As Diewert et al. (2017) highlighted, the essence of undertaking new and disappearing products for constructing price indexes is to assign zero quantities to these missing products. Value shares will not be affected by estimated prices because zero quantities multiplied by any prices are still zero values. For instance, the single imputation of Laspeyres indexes can be defined as:

$$\hat{P}_{L,SI} = \sum_{i \in U_M} s_i^0 \left(\frac{p_i^1}{p_i^0} \right) + \sum_{i \in U_D} s_i^0 \left(\frac{\hat{p}_i^1}{p_i^0} \right) + \sum_{i \in U_N} s_i^0 \left(\frac{p_i^1}{\hat{p}_i^0} \right) \tag{34}$$

Note that the term regarding new products in the right hand side is technically zero as value shares of new products in the base period (s_i^0 with $i \in U_N$) are zero. I keep this term to clarify the distinction between different price indexes. For Paasche indexes, the single imputation can be defined as:

$$\hat{P}_{P,SI} = \left[\sum_{i \in U_M} s_i^1 \left(\frac{p_i^1}{p_i^0} \right)^{-1} + \sum_{i \in U_D} s_i^1 \left(\frac{\hat{p}_i^1}{p_i^0} \right)^{-1} + \sum_{i \in U_N} s_i^1 \left(\frac{p_i^1}{\hat{p}_i^0} \right)^{-1} \right]^{-1} \tag{35}$$

The term regarding disappearing products in the right hand side is also zero, though it remains in the formula for clarification. Taking the geometric mean of Laspeyres indexes and Paasche indexes with the single imputation results in Fisher indexes $\hat{P}_{F,SI}$. In terms of the double imputation and the full imputation, Fisher indexes can be deduced in the same way as Törnqvist indexes by replacing missing prices or actual prices with estimated prices.

Table 6 shows the rates of correct predictions for Fisher indexes and Törnqvist indexes with the single, double and full imputation during 52 weeks. I compare hedonic imputation indexes with benchmark indexes and determine the percentage of correct predictions. Benchmark indexes are constructed with actual prices from the testing data set, which replaces the missing prices of new products and disappearing products. Predicted price indexes that are within ±0.5% of benchmark indexes are considered to be correct predictions. Indexes 1–6 are related to OLS, FGLS with value shares, FGLS with the natural exponential function, regression trees, bagging trees and random forests respectively. For Fisher indexes with the single imputation, it can be seen that the imputation with random forests has the highest percentage of correct predictions, that is, Index 6 with 58.82% precision. The accuracy of Index 6 drops to 50.98% when using the double imputation that includes more estimated prices. Note indexes with linear estimations have gained accuracy improvement when I switch from the single imputation to the double imputation. This is because the upward bias of linear price estimations can be relatively reduced in the double imputation. The effect of cancelling out systematic errors is also confirmed by the full imputation. Index 1 (where prices are overestimated) with the full imputation still obtains 41.18% correct predictions. Fisher indexes with hedonic imputation validate the impressive prediction performance of random forests for the single imputation, and demonstrate accuracy gains of cancelling out price estimation biases in double imputation and full imputation. For Törnqvist indexes, similar patterns are observed.

In addition to Fisher indexes and Törnqvist indexes, I compute GEKS indexes to ensure the requirement of circularity. The GEKS method was proposed by Gini (1931), Eltetö and Köves (1964) and Szulc (1964). It takes the geometric mean of price index ratios. Suppose P_{it} is the bilateral price index between periods i and t, and P_{jt} is the bilateral price index between periods j and t. The GESK index between periods i and j is defined with the base t taking all periods:

$$GEKS_{ij} = \prod_{t=1}^{T} (P_{it}/P_{jt})^{\frac{1}{T}} \tag{36}$$

GEKS indexes based on Fisher and Törnqvist indexes can be seen in Table 6. Index 6 which is associated with random forests has displayed approximately 60% correct predictions in the single imputation for the GEKS (Fisher) and GEKS (Törnqvist) methods. The OLS method in the single imputation has a better performance than random forests for GESK (Fisher) indexes, but its prediction accuracy dramatically decreases for GEKS (Törnqvist) indexes. It is reasonable to consider random forests when conducting the single imputation for price indexes. The double and full imputations serve to remove the price prediction bias of linear estimations when I make column-wise comparison. But their percentages of accuracy are much lower than the single imputation. The ability of cancelling out price estimation biases for multilateral indexes are less effective than for bilateral indexes.

Table 6: Accuracy of price indexes with hedonic imputation

		OLS	FGLS		Machine learning		
	Types	Index 1	Index 2	Index 3	Index 4	Index 5	Index 6
Fisher	Single	54.90%	47.06%	52.94%	19.61%	19.61%	58.82%
	Double	56.86%	52.94%	56.86%	7.84%	9.80%	50.98%
	Full	41.18%	23.53%	35.29%	7.84%	5.88%	27.45%
Törnqvist	Single	37.25%	31.37%	37.25%	21.57%	25.49%	62.75%
	Double	47.06%	39.22%	49.02%	9.80%	13.73%	49.02%
	Full	45.10%	11.76%	45.10%	5.88%	5.88%	27.45%
GEKS	Single	68.63%	52.94%	68.63%	15.69%	17.65%	58.82%
(*Fisher*)	Double	31.37%	33.33%	33.33%	3.92%	3.92%	17.65%
	Full	21.57%	13.73%	21.57%	3.92%	3.92%	13.73%
GEKS	Single	37.25%	25.49%	35.29%	27.45%	23.53%	62.75%
(*Törnqvist*)	Double	9.80%	11.76%	9.80%	7.84%	7.84%	17.65%
	Full	11.76%	5.88%	11.76%	1.96%	3.92%	17.65%

Note: Estimations within a range of ±0.5% to actual values are considered as accurate predictions. Bilateral indexes are calculated using the fixed base. Multilateral indexes are calculated using the GEKS method. The series of Index 1 is constructed using OLS to estimate the prices of new and disappearing products. Indexes 2–6 are related to FGLS with value shares, FGLS with exponents, regression trees, bagging trees and random forests.

4.4 Variable importance

In-sample and out-of-sample tests have manifested the predictive ability of random forests on scanner data. Indexes in product churn verify the reliability of random forests in hedonic imputation. With the quality of being precise in price prediction and index computation, random forests are recommended for hedonic imputation that facilitates the price index of unbalanced data. However, researchers may adhere to OLS and FGLS despite of the predictive ability of random forests. They prefer the economic interpretation of linear models, that is, estimated coefficients that indicate the response of prices to one variable while other variables remain unchanged (see

Table 7). The economic interpretation is one of the most frequently stated problems with machine learning. The machine learning approach is often challenged as a "black box" compared with specifically estimated coefficients in OLS and FGLS (Prasad et al., 2006).

Table 7: Regression results of logarithmic product prices

	OLS	FGLS	
		Value shares	Exponents
z_1	1.705***	1.700***	1.718***
	(0.005)	(0.005)	(0.005)
z_2	1.332***	1.335***	1.318***
	(0.005)	(0.005)	(0.005)
z_3	0.483***	0.481***	0.503***
	(0.005)	(0.005)	(0.005)
z_4	0.480***	0.482***	0.494***
	(0.005)	(0.005)	(0.005)
z_5	0.486***	0.488***	0.502***
	(0.005)	(0.005)	(0.005)
constant	0.484***	0.490***	0.461***
	(0.012)	(0.011)	(0.011)
week	Yes	Yes	Yes
Observations	36,437	36,437	36,437
F Statistic	4,295.004***	4,262.311***	4,400.363***

Note: $^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

For the purpose of model interpretation, I introduce variable importance measures of random forests. Although the response of prices to one unit change of product characteristics cannot be quantified in random forests, variable importance specifies the importance of independent variables and carries out the relative impact of product characteristics to prices.

Two types of variable importance measures can be adopted: the increase in mean squared errors or the increase in node purity (Breiman, 2001, 2002). The first practice specifies the change of MSE and is basically used for numerical responses. The second practice specifies the sum of reduced node impurities, that is, the variety of nodes when one variable is split. I focus on the first measure in this paper as it provides accuracy comparison with respects to numerical variables. Recall that bootstrapped samples are drawn from the scanner data with replacement and result in unselected data out of bag. I denote these OOB data as Ω_b^c. It presents the complement of bootstrapped data Ω_b given that Ω is the universe that contains all observations. The OOB scanner data set is initially used to measure the MSE, denoted as ζ_b for the b-th tree that grows on the b-th bootstrapped data bag:

$$\zeta_b = \sum_{(y,\boldsymbol{x})\in\Omega_b^c} \frac{1}{N(\Omega_b^c)} \left(y - \hat{y}(x_j, \boldsymbol{x_{-j}})\right)^2 \tag{37}$$

$N(\Omega_b^c)$ refers to the number of observations in Ω_b^c, x_j denotes the j-th variable that needs to be tested, and $\boldsymbol{x_{-j}}$ denotes variables except x_j. y is the actual value in the OOB data and $\hat{y}$ is the predicted value by running estimation models on the OOB data. The j-th variable x_j of the OOB data is then permuted as $\tilde{x}_j$. Estimation models are conducted on $\tilde{x}_j$ and $\boldsymbol{x_{-j}}$ to generate the predicted value $\hat{y}$, measuring a new MSE for the b-th tree:

$$\zeta_b^j = \sum_{(y,\boldsymbol{x})\in\Omega_b^c} \frac{1}{N(\Omega_b^c)} \left(y - \hat{y}(\tilde{x}_j, \boldsymbol{x_{-j}})\right)^2 \tag{38}$$

ζ_b measures the MSE given original variables while ζ_b^j measures the MSE given permuted variables. Differences between ζ_b^j and ζ_b are averaged on all bootstrapped samples to compute the change of MSE that measures the impact of variable permutation:

$$\mathit{\Delta MSE} = \frac{1}{B}\sum_{b=1}^{B}(\zeta_b^j - \zeta_b) \tag{39}$$

If the independent variable x_j is highly related to the dependent variable y, a prominent change in MSE will be produced after the permutation of x_j in the OOB data. $\varDelta MSE$ supports the economic interpretation of random forests because the relative contribution of time effects and product characteristics to prices can be detected.

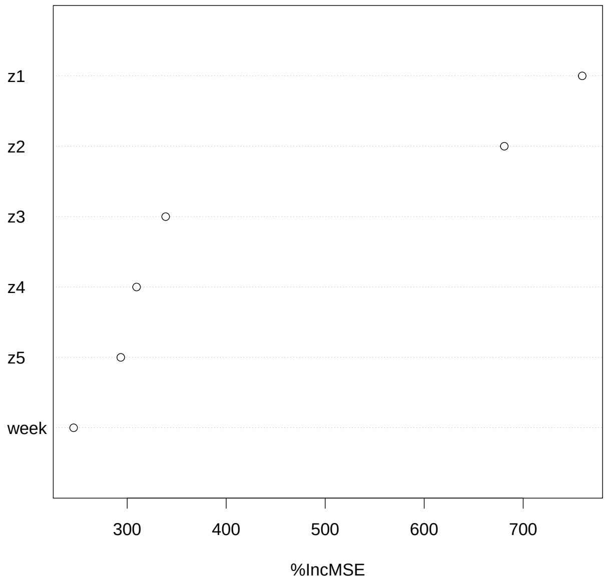


Figure 4: Variable importance in random forests

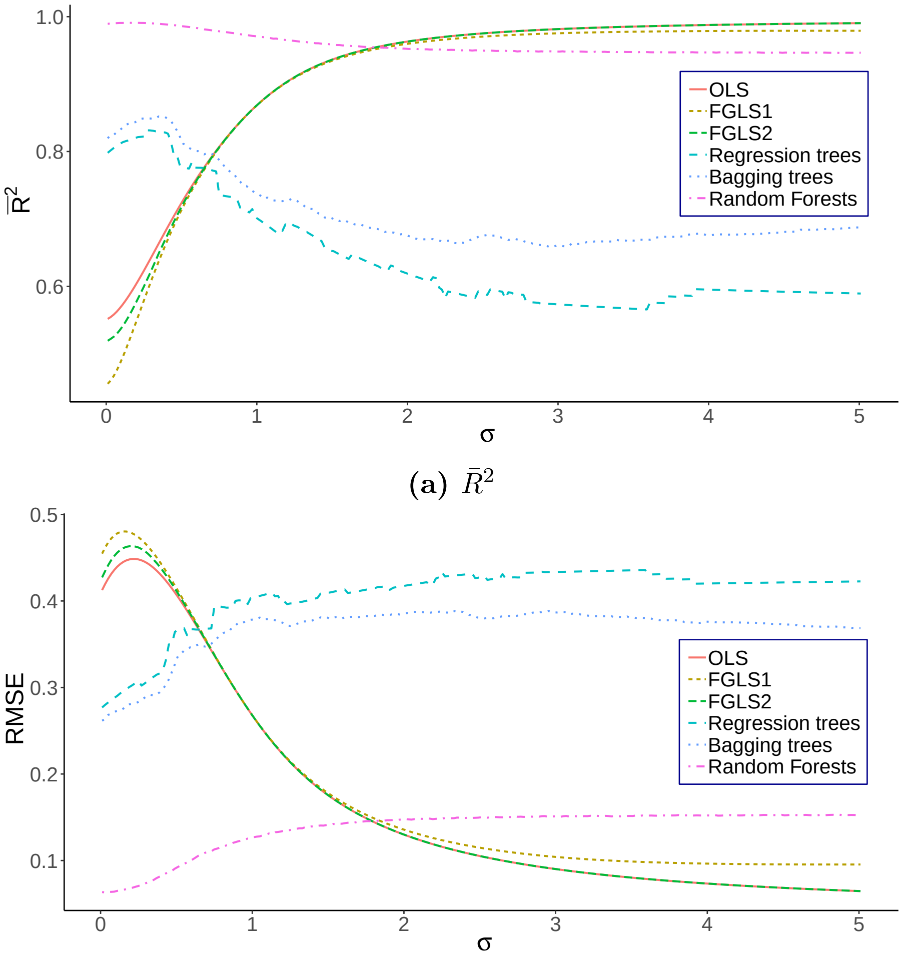
In Figure 4, there is a clear trend of decreasing importance from the product characteristic z_1 to the product characteristic z_5. The variable z_1 has strongly supported the prediction of product prices, followed by the variable z_2. Variables z_3, z_4 and z_5 share almost equivalent importance in price prediction. The variable importance comparison indicates that z_1 has the highest impact on product prices. The impact of z_2 is slightly less than z_1. Almost identical contributions of z_3, z_4 and z_5 to product prices are spotted. Recall $\beta_1 = 0.4, \beta_2 = 0.3, \beta_3 = \beta_4 = \beta_5 = 0.1$ in the CES separable sub-utility function. The relative size of impacts estimated for product characteristics is consistent with these coefficients in the set-up of hedonic prices.

4.5 Elasticity of substitution

I have assumed that hedonic prices follow a Cobb-Douglas form of product characteristics by choosing unit elasticity of substitution in the CES function. This is an intermediate status between the Leontief type where product characteristics cannot be substituted internally, and the linear type where product characteristics can be substituted equivalently. The price prediction capacity of random forests in the scenario with unit elastic utility has been confirmed. To provide a sensitivity analysis of random forests as well as other models, I allow the elasticity of substitution to vary between 0.01 and 5.01 so that the hedonic price prediction approximately extends to a Leontief sub-utility function and a linear sub-utility function. Out-of-sample metrics are recorded when σ increases with a step size at 0.02.

Figure 5a depicts the out-of-sample $\bar{R}^2$ of linear estimations and machine learning estimations. When product characteristics are complements ($\sigma = 0.01$), tree-based models show higher $\bar{R}^2$ in price prediction than linear models. This is because tree-based models are compatible with the data structure of hedonic prices produced by a Leontief sub-utility function. Since the Leontief function rules out substitutability between product characteristics, hedonic prices are essentially determined by the minimal value among product characteristics. The increase in prices is conditional on a joint contribution from product characteristics, rather than the increase in one characteristic. Recall that predicted values of regression trees will not necessarily fluctuate with one increasing variable. The tree structure ensures that the change of prices mainly depends on multiple independent variables. Therefore, tree-based models are more suitable for hedonic prices when substitutability is absent between product characteristics.

With the elasticity of substitution increasing, the performance of linear models turns to improve. The $\bar{R}^2$ of OLS surpasses that of regression trees with $\sigma = 0.65$, and



(a) $\bar{R}^2$

(b) RMSE

Figure 5: Sensitivity analysis of hedonic price prediction

surpasses that of bagging trees with $\sigma = 0.73$. It slightly outweighs the $\bar{R}^2$ of the random forests for σ >1.81. When product characteristics are substitutes ($\sigma = 5.01$), linear models demonstrate extraordinary prediction accuracy while regression trees and bagging trees only explain about two-thirds of price variability. This is reasonable because hedonic prices are approaching a linear relationship with product characteristics when elasticity of substitution is large enough in the separable sub-utility function. The tree structure is not well matched to the linear relationship. However, the $\bar{R}^2$ of random forests is still remarkable at around 0.95 as this algorithm adopts bootstrapped samples and random selection of variables to modify the tree structure. The performance of random forests is reliable regardless of the increasing σ. For the situation that the utility type of consuming behaviour is uncertain, random forests can always serve as a satisfactory method in price prediction.

Other $\bar{R}^2$-type expressions of in-sample and out-of-sample tests have exhibited almost indistinguishable values so it is unnecessary to repeat the sensitivity analysis with these expressions. As an additional indicator, the root of mean squared errors (RMSE) is used. The trend of out-of-sample RMSE for linear estimations and machine learning estimations is displayed in Figure 5b. Models with lower RMSE are considered to have a better prediction capacity. It can bee seen from the plot that linear estimations become accurate with increasing elasticity of substitution. Tree-based models are disadvantaged when hedonic prices are linear in product characteristics, but the prediction performance of random forests is still acceptable. The evidence from RMSE supports the remarks on price prediction using $\bar{R}^2$.

5 Evidence from the Dominick's dataset

To validate the results from simulated scanner data, I conduct a robustness check and draw real scanner data from the Dominick's database that is published by James

M. Kilts Center, University of Chicago Booth School of Business. The Dominick's database covers historical information on prices and product characteristics obtained by scanning bar codes. Both category-specific files and general files of scanner data are contained in the database, providing a variety of commodities for hedonic imputation. I select frozen juices in category-specific files for the robustness testing.

Summary statistics for numeric variables and categorical variables are presented in Table A1 and Table A2. The variable *price* refers to the retail price of a bundle of products and the variable *quantity* indicates the number of bundles sold. Note that the variable *quantity* is only utilised to generate value shares and is not adopted as one of product characteristics. Product characteristics from the Dominick's database are categorical variables, including *week*, $fruit$ and *size* in Table A2. The variable *week* is originally numerical, and it is converted to be categorical to present fixed time effects. Product flavours from $fruit$ are captured from product names. The variable *size* denoting product sizes is taken as a categorical variable considering its non-linear impact on hedonic prices. Other variables, like *move*, *qty*, *sale* and *ok* are not listed but they are essential to generate variables as required. The variable *move* (the number of items sold) is divided by the variable *qty* (the number of items in a bundle) to generate *quantity* in this paper. Suspected observations with *ok* equal to 0 or *sale* equal to "G" are removed. With these filter criteria, the data set of frozen juices is trimmed to include 150,437 observations that are randomly divided into the training data set and the testing data set with a common ratio of 70% versus 30%. Once again, the training set is adopted to generate estimation models and subsequently these models can be evaluated in the testing data set.

Figure A1 compares observed prices and fitted prices (both in logarithmic forms) for OLS, FGLS with value share weights (FGLS1), FGLS with exponential weights (FGLS2), regression trees, bagging trees and random forests in the training data set. R^2 and $\bar{R}^2$ metrics of FGLS1 and FGLS2 are lower than those of OLS. It means that dealing with the heteroscedasticity problem does not improve prediction abilities. The special prediction pattern of regression trees is plotted in Figure A1d. A few levels of predicted values are generated due to the tree-based algorithm. To allow predicted values to be between discrete levels, bagging trees use bootstrapped samples and average prediction values over all trees. This leads to prediction accuracy gains in bagging trees. Prediction performance is further improved when random forests choose randomly selected variables for price prediction. In addition to in-sample plots, prediction results of the testing data are evident in Figure A2. Still, the random forest algorithm provides the best prediction, followed by bagging trees and regression trees. The prediction with OLS reports lower R^2 and $\bar{R}^2$ than those of the prediction with tree-based models, but it outweighs the prediction with FGLS1 and FGLS2. A closer inspection of $\bar{R}^2$-type expressions of in-sample and out-of-sample tests can be seen in Table A3, which displays similar information to what is conveyed by these plots.

Estimated coefficients that indicate the response of prices to one variable while other variables remain unchanged are listed in Table A4. Product sizes are statistically significant in determining hedonic prices. Detailed coefficients of week indexes and fruit flavours are not displayed because of excessive dummies, though most of these dummies are significant in regressions. Variable importance that specifies the relative impact of product characteristics to prices is plotted in Figure A3. The variable $size$ has the largest impact on price prediction, followed by variables $fruit$ and $week$. Random forests take categorical variables as factor variables in computation so that the importance measures of $size$, $fruit$ and $week$ are incomparable with the coefficients of dummies in linear models.

The replication of models on the Dominick's dataset confirms the robustness of the results from simulated scanner data. Random forests are the best fitted tree-based model in hedonic price prediction. The tree structure helps to explain consumer preferences where product characteristics are not substitutable. It accounts for large $\bar{R}^2$-type expressions of tree-based models. Linear models fall behind about the price prediction of frozen juice data since a linear relationship between hedonic prices and product characteristics does not truly capture the behaviour of consumers. However, the accuracy of price indexes on the real scanner data cannot be examined. Prices of new products in base periods and prices of disappearing products in current periods are not available. It is not reasonable to decide the best model in computing price indexes when the benchmark is missing.

6 Conclusion

The motivation of this paper is to improve the prediction accuracy of hedonic imputation for unmatched products that enter or exit in multiple periods. A number of models are compared, including OLS, FGLS with value share weights, FGLS with exponential weights, regression trees, bagging trees and random forests. Linear regression models are adopted by following common approaches of hedonic imputation while tree-based machine learning models are adopted because tree structures are compatible with the behaviour foundation where product characteristics are not substitutes for consumers. I simulate scanner data with product prices and characteristics to test these models. The data generating process is based on constant elasticity of substitution and the elasticity is allowed to vary in a wide range. Either elastic utility or inelastic utility can be approached with appropriate elasticity coefficients.

Using unit elastic utility as the benchmark, the model comparison between linear hedonic regression and tree-based machine learning confirms the extraordinary prediction capability of random forests. Random forests are the best fitted model due to large in-sample and out-of-sample $\bar{R}^2$-type measures (close to 1). The prediction performance of random forests is also validated by the price index construction where bilateral indexes and multilateral indexes are computed with predicted prices of unmatched products. Price indexes with random forests display approximately 60% correct prediction values that are stable in the single, double and full imputations. Although causal effects cannot be quantified, random forests produce variable importance measures that specify relative effects of product characteristics and time periods on prices. The variable importance estimated for product characteristics is consistent with coefficients in the hedonic set-up.

In addition to the benchmark of unit elastic utility, I allow the elasticity of substitution to vary so that the separable sub-utility function underlying hedonic prices approximately extends to a Leontief function or a linear function. When product characteristics are complements, tree-based models show better price prediction than linear models. Since a Leontief sub-utility function rules out substitutability between product characteristics, the increase in prices is conditional on a joint contribution from product characteristics. The tree structure ensures that the change of prices mainly depends on multiple product characteristics and is therefore more suitable for explaining hedonic prices when substitutability is absent. When product characteristics are substitutes, linear models demonstrate extraordinary prediction accuracy because the separable sub-utility function is linear in product characteristics. The standard tree structure is no longer matched to the linear relationship. However, the performance of random forests is still remarkable as this algorithm employs random samples and variables to revise the standard tree structure. The price prediction of random forests is stable whether the substitutability between product characteristics is available or not.

The model performance on simulated scanner data and real scanner data in this paper suggests that machine learning approaches, especially random forests, can be effectively employed for hedonic imputation when new products and disappearing products exist as unmatched items. Although the paucity of causal explanations in machine learning remains to be concerned, using these tree-based models is suitable for hedonic prices from the perspective of prediction accuracy. Since product characteristics are not perfectly substitutes in practice, the pricing mechanism of products is likely to approach a Leontief function that shares some common features with a tree-based pricing type. This allows tree-based machine learning to fit scanner data and to produce accurate price prediction. Within the context of hedonic imputation, prediction accuracy is more essential than causal explanations. The focus of hedonic imputation is to estimate missing prices so the response of prices to the change of product characteristics or time dummies is not a priority. Standard econometric approaches in hedonic regression like FGLS attempt to improve prediction accuracy by excluding heteroscedasticity, but coefficients are just slightly revised and the accuracy enhancement is insignificant. It is advisable to introduce machine learning methods to hedonic models so that missing prices can be better estimated and price indexes can be better computed for unmatched products.

References

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principal. In *2nd International Symposium on Information Theory, 1973*. Akademiai Kiado.

Amemiya, T. (1980). Selection of regressors. *International Economic Review*, pages 331–354.

Bajari, P., Nekipelov, D., Ryan, S. P., and Yang, M. (2015). Machine learning methods for demand estimation. *The American Economic Review*, 105(5):481–485.

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

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

Breiman, L. (2002). Manual on setting up, using, and understanding random forests v3. 1. Technical report, Statistics Department University of California Berkeley.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Craven, P. and Wahba, G. (1979). Smoothing noisy data with spline functions. *Numer. Math*, 31:403.

de Haan, J. (2010). Hedonic price indexes: a comparison of imputation, time dummy and 're-pricing' methods. *Jahrbücher für Nationalökonomie und Statistik*, pages 772–791.

Diewert, E., Fox, K. J. F., and Schreyer, P. (2017). The digital economy, new products and consumer welfare. Technical report, Vancouver School of Economics.

Diewert, W. E. (2003). Hedonic regressions: a consumer theory approach. In *Scanner Data and Price Indexes*, NBER Chapters, pages 317–348. National Bureau of Economic Research, Inc.

Diewert, W. E. and Fox, K. J. (2017). Substitution bias in multilateral methods for CPI construction using scanner data. Microeconomics working papers, Vancouver School of Economics.

Diewert, W. E., Heravi, S., and Silver, M. (2009). Hedonic imputation versus time dummy hedonic indexes. In *Price index concepts and measurement*, pages 161–196. University of Chicago Press.

Eltetö, O. and Köves, P. (1964). On a problem of index number computation relating to international comparison. *Statisztikai Szemle*, 42:507–518.

Fox, K. J. (2000). Information-rich expressions for model selection criteria. *Applied Economics Letters*, 7(1):59–62.

Gini, C. (1931). On the circular test of index numbers. *Metron*, 9(9):3–24.

Hannan, E. J. and Quinn, B. G. (1979). The determination of the order of an autoregression. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 190–195.

Hill, R. J. and Melser, D. (2008). Hedonic imputation and the price index problem: an application to housing. *Economic Inquiry*, 46(4):593–609.

Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282.

Hocking, R. R. (1976). A biometrics invited paper: the analysis and selection of variables in linear regression. *Biometrics*, 32(1):1–49.

Miller, S. and Startz, R. (2019). Feasible generalized least squares using support vector regression. *Economics Letters*, 175:28 – 31.

Prasad, A. M., Iverson, L. R., and Liaw, A. (2006). Newer classification and regression tree techniques: bagging and random forests for ecological prediction. *Ecosystems*, 9(2):181–199.

Rosen, S. (1974). Hedonic prices and implicit markets: product differentiation in pure competition. *Journal of Political Economy*, 82(1):34–55.

Schwarz, G. et al. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464.

Szulc, B. (1964). Indices for multiregional comparisons. *Przeglad statystyczny*, 3:239–254.

Theil, H. (1961). *Economic forecasts and policy.* North-Holland, Amsterdam.

Willett, J. B. and Singer, J. D. (1988). Another cautionary note about R^2: its use in weighted least-squares regression analysis. *The American Statistician*, 42(3):236–238.

Wooldridge, J. M. (2015). *Introductory econometrics: a modern approach.* Nelson Education.

Appendices

Table A1: Summary statistics of numerical variables

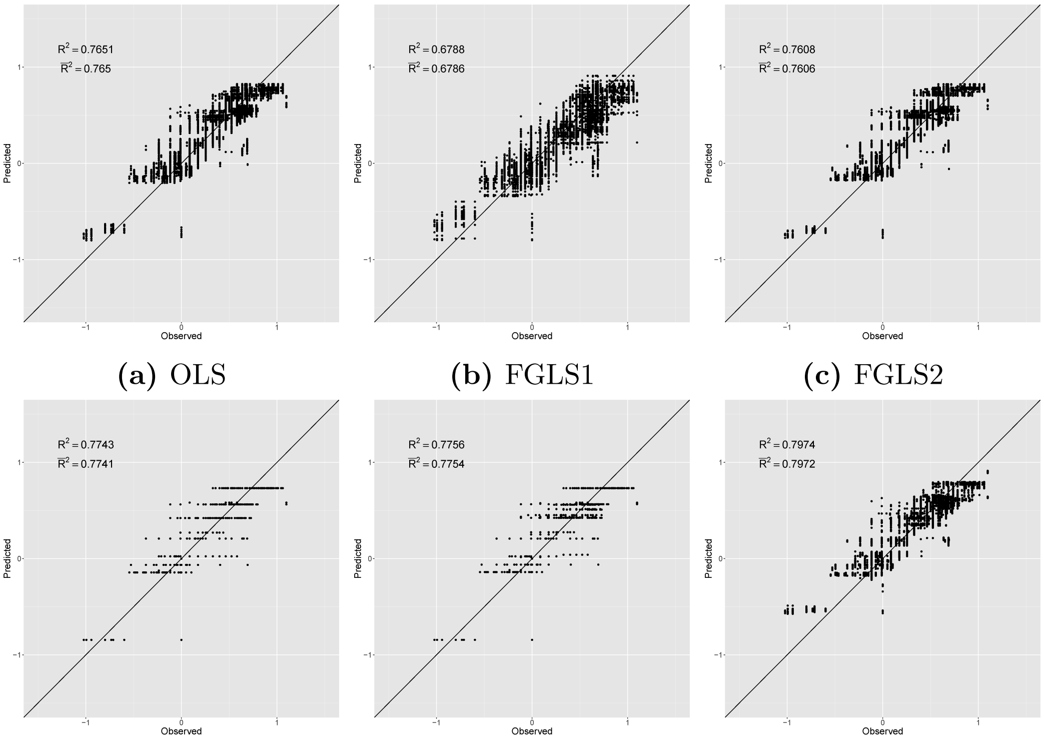
Variables	Definitions	Datasets	Mean	Std. Dev	Min	Max
price	Retail prices ($)	Full	1.407	0.490	0.360	3.000
		Training	1.409	0.491	0.360	3.000
		Testing	1.403	0.486	0.360	3.000
quantity	Bundles	Full	43.682	112.411	0.333	4,838
		Training	43.714	112.114	0.333	4,838
		Testing	43.607	113.104	0.500	4,121

Note: The full frozen juice data set with 150,437 observations is randomly divided into training data with 105,329 observations and testing data with 45,108 observations. The ratio follows 70% versus 30% as a common proportion of data partitions.

Table A2: Summary statistics of categorical variables

Variables	Definitions	Datasets	Levels	Top Three Frequent Items
week	Week indexes	Full	52	43, 50, 51
		Training	52	43, 42, 50
		Testing	52	51, 45, 50
fruit	Product flavours	Full	14	ORG, LEMO, GRAPE
		Training	14	ORG, LEMO, GRAPE
		Testing	14	ORG, GRAPE, LEMO
size	Product sizes	Full	6	12 OZ, 6 OZ, 16 OZ
		Training	6	12 OZ, 6 OZ, 16 OZ
		Testing	6	12 OZ, 6 OZ, 16 OZ

Note: The full frozen juice data set with 150,437 observations is randomly divided into training data with 105,329 observations and testing data with 45,108 observations. The ratio follows 70% versus 30% as a common proportion of data partitions. Levels denote the categorical level of each variable. Flavours are recognisable from brand names. "ORG" is short for oranges and "LEMO" is short for Lemonade. Product sizes are taken as a categorical variable due to their non-linear impact on hedonic prices.



(a) OLS

(b) FGLS1

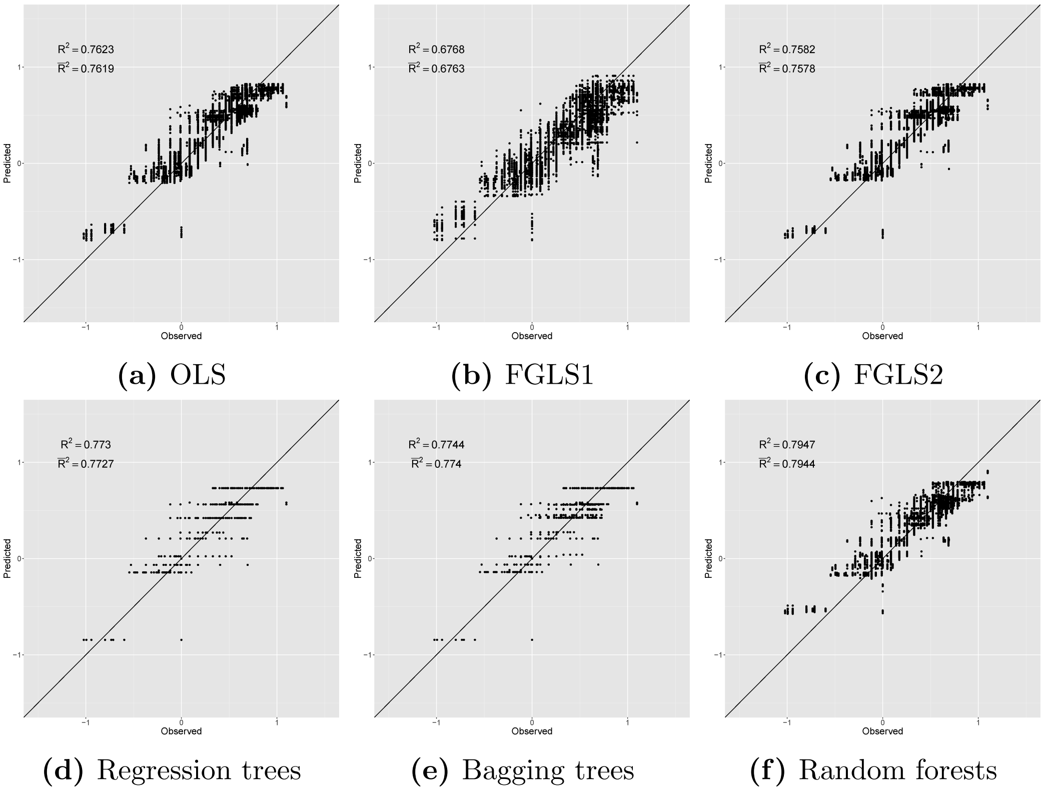
(c) FGLS2

(d) Regression trees

(e) Bagging trees

(f) Random forests

Figure A1: In-sample observed and predicted frozen juice prices



(a) OLS **(b)** FGLS1 **(c)** FGLS2

(d) Regression trees **(e)** Bagging trees **(f)** Random forests

Figure A2: Out-of-sample observed and predicted frozen juice prices

Table A3: Prediction performance on frozen juice data in R^2 types

	OLS	FGLS		Machine learning		
		Value shares	Exponents	Regression trees	Bagging trees	Random forests
	(1)	(2)	(3)	(4)	(5)	(6)
In-sample prediction						
R^2	0.7651	0.6788	0.7608	0.7743	0.7756	0.7974
$\bar{R}^2$	0.7650	0.6786	0.7606	0.7741	0.7754	0.7972
$\bar{R}^2(AIC)$	0.7648	0.6784	0.7605	0.7740	0.7753	0.7971
$\bar{R}^2(SC)$	0.7633	0.6764	0.7590	0.7726	0.7739	0.7958
$\bar{R}^2(HQ)$	0.7644	0.6778	0.7600	0.7736	0.7748	0.7967
$\bar{R}^2(Jp)$	0.7648	0.6784	0.7605	0.7740	0.7753	0.7971
$\bar{R}^2(Sp)$	0.7648	0.6784	0.7605	0.7740	0.7753	0.7971
$\bar{R}^2(GCV)$	0.7648	0.6784	0.7605	0.7740	0.7753	0.7971
Out-of-sample prediction						
R^2	0.7623	0.6768	0.7582	0.7730	0.7744	0.7947
$\bar{R}^2$	0.7619	0.6763	0.7578	0.7727	0.7740	0.7944
$\bar{R}^2(AIC)$	0.7616	0.6758	0.7574	0.7723	0.7737	0.7941
$\bar{R}^2(SC)$	0.7584	0.6715	0.7542	0.7693	0.7706	0.7914
$\bar{R}^2(HQ)$	0.7606	0.6745	0.7564	0.7714	0.7727	0.7932
$\bar{R}^2(Jp)$	0.7616	0.6758	0.7574	0.7723	0.7737	0.7941
$\bar{R}^2(Sp)$	0.7616	0.6758	0.7574	0.7723	0.7737	0.7941
$\bar{R}^2(GCV)$	0.7616	0.6758	0.7574	0.7723	0.7737	0.7941

Note: Same values occur due to rounding decimals.

Table A4: Regression results of frozen juice prices

	OLS	FGLS	
		Value shares	Exponents
size (base=10 OZ)			
11.5 OZ	0.146^{***}	0.044^{***}	0.166^{***}
	(0.004)	(0.007)	(0.004)
12 OZ	0.031^{***}	-0.083^{***}	0.084^{***}
	(0.003)	(0.004)	(0.004)
16 OZ	0.252^{***}	0.227^{***}	0.317^{***}
	(0.003)	(0.005)	(0.004)
6 OZ	-0.605^{***}	-0.623^{***}	-0.563^{***}
	(0.003)	(0.005)	(0.003)
7.5 OZ	0.398^{***}	0.275^{***}	0.447^{***}
	(0.005)	(0.008)	(0.004)
constant	0.112^{***}	0.125^{***}	0.042^{***}
	(0.005)	(0.006)	(0.005)
week	Yes	Yes	Yes
fruit	Yes	Yes	Yes
Observations	105,329	105,329	105,329
F Statistic	$4{,}968.887^{***}$	$2{,}968.019^{***}$	$8{,}989.181^{***}$

Note: $^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$. *week* and *fruit* are not displayed in detail in this table due to excessive dummies.

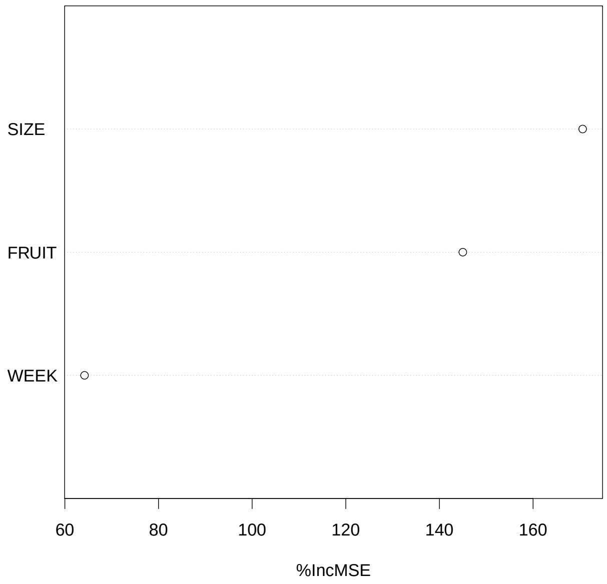


Figure A3: Variable importance in random forests on frozen juice data