Using Machine Learning to Construct Hedonic Price Indices*

Michael J. Cafarella† Gabriel Ehrlich‡ Tian Gao§

John Haltiwanger¶ Matthew D. Shapiro‖ Laura Yi Zhao**

May 2023

Abstract

This paper uses machine learning (ML) to estimate hedonic price indices at scale from item-level transaction and product characteristics. The procedure uses state-of-the-art approaches from hedonic econometrics and implements them with a neural network ML approach. Applying the methodology to Nielsen Retail Scanner data leads to a large hedonic adjustment to the Tornqvist index for food product groups: Cumulative food inflation over the period from 2007 through 2015 is reduced by half—from 5.9% to 2.8%—owing to quality adjustment. These results suggest that quality improvement via product turnover is important even in product groups that are not normally considered to feature rapid technological progress. The approach in the paper thus demonstrates the feasibility and importance of implementing hedonic adjustment at scale.

*We acknowledge financial support of the Alfred P. Sloan Foundation and the additional support of the Michigan Institute for Data Science and the Michigan Institute for Teaching and Research in Economics. Laura Zhao worked on this project as a graduate student and then subsequently as a post doc at the University of Maryland. Tian Gao worked on this project as a graduate student at the University of Michigan. The results here are in part based on researchers' own analyses calculated (or derived) based in part on data from The Nielsen Company (US), LLC and marketing databases provided through the Nielsen Datasets at the Kilts Center for Marketing Data Center at The University of Chicago Booth School of Business. The conclusions drawn from the Nielsen data are those of the researchers and do not reflect the views of Nielsen. Nielsen is not responsible for, had no role in, and was not involved in analyzing and preparing the results reported herein.

†M.I.T. and University of Michigan

‡University of Michigan

§Snowflake Inc.

¶University of Maryland and N.B.E.R.

‖University of Michigan and N.B.E.R.

**Bank of Canada and University of Maryland

1 Introduction

The availability of transactions data for retail sales has the promise to revolutionize the measurement of inflation, sales, and real consumption expenditures. Information systems of retailers capture the vast majority of retail purchases at the item level. These data should allow creation of economic statistics with little lag and reduced need for revisions. Economic statistics could also be constructed with much greater granularity—higher frequency, more detailed geography, and finer product detail. See Jarmin (2019), Ehrlich, Haltiwanger, Jarmin, Johnson, and Shapiro (2019), Ehrlich, Haltiwanger, Jarmin, Johnson, and Shapiro (2021), and Abraham, Jarmin, Moyer, and Shapiro (2022) for discussion of using transactions data at scale to re-engineer inflation measures.

The big data revolution also provides the possibility to improve measurement of quality change. While the system of official statistics features substantial effort to account for quality change, there is considerable evidence that quality change is mismeasured. This mismeasurement has implications for the measurement of inflation, productivity growth, and growth in real income and output. See Boskin et al. (1996) and Shapiro and Wilcox (1996) for evaluations for the Consumer Price Index (CPI) and Moulton (2018) for a summary of recent improvements.

This paper advances a machine learning approach to account for quality change. With item-level transactions data, one is immediately confronted by the rapid turnover of goods available in the market. This rapid turnover of goods at the item-level forces confronting accounting for quality change at scale. One must be able to distinguish between changes in new goods that are genuine changes in quality (i.e., ones that increase or decrease the value of goods) from those that are simply labeling, trivial changes in goods not valued by consumers, or perhaps merely the occasion to change price.

A large literature discusses the conceptual and practical issues related to hedonic price indices. Our machine learning approach builds on these hedonic models.[1] A key issue

[1] Court (1939) and Griliches (1961) pioneered the use of hedonic methods to construct price indices for automobiles.

highlighted by Pakes (2003) is that goods that enter or exit the marketplace are unlikely to be randomly selected in terms of their price trends. Such nonrandom selection is likely to bias "matched model" price indices constructed from products sold in both the base and subsequent periods. Silver and Heravi (2005) also demonstrate the empirical relevance of this selection problem.

Hedonic adjustment offers a potential solution to the selection bias caused by product turnover, but hedonic approaches can be difficult to implement at large scale. Traditional methods of hedonic estimation require well-coded data on product attributes data and for well-trained human analysts to specify sensible hedonic regression functions. Shapiro and Wilcox (1996) call this the "house-to-house combat" approach to hedonic estimation, evoking its labor-intensive and cumbersome nature. Another challenge to hedonic methods is that important product attributes may be unobservable; Erickson and Pakes (2011) propose an approach that can help to correct for time-varying valuations of unobservable product characteristics. Nonetheless, the practical and conceptual difficulties of hedonic methods remain important barriers to their wider application in official statistics. Currently, the Bureau of Labor Statistics (BLS) uses hedonic adjustment for only about 7.5 percent of goods and commodities in the CPI (our estimates based on Bureau of Labor Statistics, 2023).

Demand-based approaches offer an alternative approach to account for product turnover in price indices constructed from transactions-level data. Feenstra (1994) proposed a methodology to construct an exact price index that accounts for product turnover under the assumption of constant elasticity of substitution preferences, which Redding and Weinstein (2020) generalized to allow for time-varying product appeal. In a companion paper, Ehrlich, Haltiwanger, Jarmin, Johnson, Olivares, Pardue, Shapiro, and Zhao (2023) implement and compare these demand-based approaches with hedonic approaches in transactions-level data.[2]

[2]Ehrlich et al. (2023) focuses on a comparison of demand-based and hedonic models using general merchandise products from NDP. Unlike our data from Kilts, the NPD data contain rich sets of attributes, so it is possible to implement hedonic procedures using regression techniques. Ehrlich et al. (2023) briefly draws on results from this paper for comparison purposes. The ML hedonic approach and estimates are original to this paper. Ehrlich et al. (2023) is available at http://www-personal.umich.edu/~shapiro/papers/Price_Quantity_Scale.pdf. A key conclusion from that research is that

This paper presents an approach to hedonic estimation using machine learning methods that can be applied at scale and demonstrates its use with an application to the Nielsen Kilts Retail Scanner Data set. A key feature of the Nielsen data as distributed by Kilts is that they do not contain an extensive set of encoded product characteristics. Instead, they have unstructured text fields describing the products. These text fields—though they have discernable meaning—would be extremely difficult to encode into variables using standard techniques or hand-coding. For example, a product description for toilet paper is "DR W 1P 308S TT 6PK."[3] We discuss how we process such text in Section 4.1.

Two features of our methodology merit particular discussion. First, to convert text-based product descriptions into numerical characteristic representations, we use a hybrid feature encoding architecture that allows the system to incorporate "pre-trained" word embeddings (numerical representations) trained from an external corpus of text as well as "text-tailored" embeddings trained specifically on the product descriptions in the Nielsen Kilts Retail Scanner Data set. Second, our architecture does not predict prices or price changes directly, but rather predicts a set of probabilities that the price (or price change) lies in each of a set of bins that partition the observed range. Because of the noise in the estimated probabilities, it may not be optimal to form price predictions as the simple probability-weighted expected price. We use a receiver operating characteristic (ROC) curve procedure to determine the optimal number of bins to include in the price prediction.

Following Erickson and Pakes (2011), we use a two-step procedure to account for time-varying unobservables. In the first step, we use the ML architecture to predict the log level of prices. In the second step, we use the architecture to predict the percent change in price—using the residual from the first step as an additional input.

Our paper contributes to a small literature that explores the use of machine learning

the demand-based methods can be quite sensitive to the details of their specifications about consumer preferences and market structure. Hedonic methods are potentially more robust.

[3]The Nielsen Retail Scanner data are not unusual among scanner data in containing limited information on product attributes. As De Haan (2015) notes, "In many scanner data sets, only limited information on characteristics is available."

techniques to predict product prices and construct hedonic price indices. Bajari, Cen, Chernozhukov, Manukonda, Wang, Huerta, Li, Leng, Monokroussos, Vijajkuner, and Wan (2021) propose an ML architecture to form hedonic price predictions and construct hedonic price indices from Amazon's first-party apparel sales. They use both text and image embeddings in their hedonic prediction process. Zeng (2021) constructs hedonic price indices for juice products using random forest methods for price prediction but does not make use of text embeddings. Han, Schulman, Grauman, and Ramakrishnan (2021) use neural networks to produce image embeddings to study product differentiation in a marketplace for text fonts, and demonstrate that using machine learning approaches is a feasible way to quantify unstructured data on product attributes.

Our estimated hedonic price index for food product groups indicates 3.1 percentage points lower cumulative inflation from 2006q4 to 2015q4 than a traditional matched model index, a reduction of more than half. The estimated hedonic adjustment suggests that quality improvement from product turnover has been significant even in a sector (food) in which technological progress is less obvious than in other product categories. We estimate a smaller hedonic adjustment for nonfood product groups, which is likely to reflect consumers' changing patterns of purchases of nonfood items at grocery stores and the other types of retailers tracked in the Nielsen Scanner data, rather than economy-wide changes. Our results suggest that traditional price indices that do not account for product turnover systematically overstate the rate of inflation and understate the rate of real output growth in the Retail Trade sector.

2 Conceptual Framework: Accounting for Product Turnover with Hedonic Price Indices

This section defines and describes the traditional and hedonic price indices we consider in this paper. We defer a discussion of our approach to estimating the hedonic functions that

we use to construct the hedonic price indices until the following section.

Price indices aim to measure or approximate the change in the cost of living between two time periods–that is, to calculate how much more or less expensive it is to achieve the same standard of living as in some base period given the current set of products available for sale and their prices.

Traditional "matched-model" indices are constructed by comparing the prices of goods that were sold both in the base period and in the current period. Matched-model indices therefore cannot account for the possibility that the goods that enter the marketplace have more desirable features than the goods that exit the marketplace, potentially missing an important implication of product turnover. Hedonic price indices address that challenge by using predicted prices for goods in the periods prior to their entry and subsequent to their exit to measure the extent of quality upgrading via product turnover.

Our analysis in this paper focuses primarily on the traditional and hedonic versions of the Tornqvist index, and to a lesser extent the Laspeyres and Paasche indices.[4] Our companion paper, Ehrlich et al. (2023), considers additional price indices, including "exact price indices" that are meant to correspond exactly to the change in the consumer's cost of living under an assumed utility function.

2.1 Traditional "matched model" indices

The traditional geometric Laspeyres index measuring the change in prices from period $t-1$ to period t takes the form

$$\Phi_{t-1,t}^{L_{geo}} = \prod_{k \in \mathbb{C}_{t-1,t}} \left(\frac{p_{kt}}{p_{kt-1}} \right)^{s_{kt-1}}, \tag{1}$$

where p_{kt} is the price of good k in period t, p_{kt-1} is its price in period $t-1$, $\mathbb{C}_{t-1,t}$ is the set of "continuing goods" sold in both periods $t-1$ and t, and s_{kt-1} is the expenditure share of good k in period $t-1$ among the set of continuing goods. The ratio $\frac{p_{kt}}{p_{kt-1}}$ is known as the

[4] We will focus on the geometric versions of the Laspeyres and Paasche indices.

"price relative" for product k.

The traditional geometric Paasche index is defined similarly, but using the second period expenditure shares as weights:

$$\Phi_{t-1,t}^{P_{geo}} = \prod_{k \in \mathbb{C}_{t-1,t}} \left(\frac{p_{kt}}{p_{kt-1}} \right)^{s_{kt}}. \tag{2}$$

The traditional Tornqvist index is defined as the geometric average of the geometric Laspeyres and Paasche indices:

$$\Phi_{t-1,t}^{TQ} = \sqrt{\Phi_{t-1,t}^{L_{geo}} \Phi_{t-1,t}^{P_{geo}}} = \prod_{k \in \mathbb{C}_{t-1,t}} \left(\frac{p_{kt}}{p_{kt-1}} \right)^{\frac{s_{kt-1}+s_{kt}}{2}}. \tag{3}$$

The geometric price indices can be re-written as log inflation rates as follows:

$$\begin{aligned}
\ln \Phi_{t-1,t}^{L_{geo}} &= \sum_{k \in \mathbb{C}_{t-1,t}} s_{kt-1} \ln \frac{p_{kt}}{p_{kt-1}} \\
\ln \Phi_{t-1,t}^{P_{geo}} &= \sum_{k \in \mathbb{C}_{t-1,t}} s_{kt} \ln \frac{p_{kt}}{p_{kt-1}} \\
\ln \Phi_{t-1,t}^{TQ} &= \sum_{k \in \mathbb{C}_{t-1,t}} \left(\frac{s_{kt-1} + s_{kt}}{2} \right) \ln \frac{p_{kt}}{p_{kt-1}}.
\end{aligned}$$

The Laspeyres, Paasche, and Tornqvist indices are thus all weighted averages of product-level log price changes; they differ only in their weights. The geometric Laspeyres index uses base-period expenditure shares as weights, the geometric Paasche index uses end-period expenditure shares, and the Tornqvist uses average expenditure shares across the two periods.[5] In the case of arithmetic price indices and strictly normal goods, the Paasche index must lie below the Laspeyres index, and the two indices bound the exact change in the consumer's cost of living.[6] In the case of geometric price indices, that relationship need not hold.

[5] The weights are set equal to zero for goods that are not sold in a given period, i.e., $s_{kt-1} = 0$ for an entering good and $s_{kt} = 0$ for an exiting good.

[6] In that case, the Paasche index bounds the equivalent variation from below, while the Laspeyres index bounds the compensating variation from above.

Nonetheless, the Tornqvist index is a "superlative" price index, meaning that it provides a second-order approximation to the change in the true unit expenditure function (cost of living) for a wide class of utility functions (Diewert, 1978).

Calculating these price indices from UPC-level data avoids many of the challenges that confront traditional price indices, because product characteristics are unlikely to change over time without a corresponding change in UPC code (Redding and Weinstein, 2020).

Nonetheless, the matched model price indices are subject to bias from non-representativeness in the price trends of goods that enter and exit the marketplace. For example, if goods with the most rapidly falling prices are more likely to exit the marketplace, then any matched-model index will have an inflationary bias. Hedonic methods can address such biases by providing estimates of those goods' prices had they been sold prior to entry or subsequent to exit.

2.2 Hedonic indices

Hedonic methods provide a relatively simple procedure to correct for the "missing" prices from entering and exiting goods. There is a wide range of possible methods to implement a hedonic price index. In this paper, we focus on one approach to estimating hedonic price indices, which uses a machine learning architecture to apply the "time-varying unobservables" approach of Erickson and Pakes (2011) at scale.

We estimate hedonic models in both log-levels and log-differences. Our log-level hedonic models takes the form

$$\ln p_{kt} = h_t(z_{kt}) + u_{kt}, \tag{4}$$

where z_{kt} is a vector of observable characteristics for good k. The hedonic function $h_t()$ in equation (4) is estimated separately period-by-period, allowing it to vary over time along with changing consumer valuations and market structure. Traditionally, the function $h_t()$ is linear in parameters and the hedonic equation is estimated with ordinary or weighted least

squares regression (e.g., Pakes, 2003; Benkard and Bajari, 2005; Erickson and Pakes, 2011; Byrne et al., 2019). Our main innovation in this paper is to develop and demonstrate a machine learning procedure to estimate the function $h_t()$, which is potentially nonlinear in the characteristics z_{kt}. Our ML architecture additionally eliminates the researcher's need to specify the functional form of $h_t()$, alleviating the "house-to-house combat" problem discussed by Shapiro and Wilcox (1996).

We adapt an estimation scheme proposed by Erickson and Pakes (2011) for hedonic regressions that can account for unobservable product characteristics. They posit a hedonic data generating process of the form

$$\begin{aligned} \ln(p_{kt}) &= h_t(Z_k) + \eta_{kt} \\ \eta_{kt} &= \gamma_{kt} + u_{kt} \end{aligned} \tag{5}$$

where η_{kt}, the component of price unexplained by the hedonic function, is the sum of the market's valuation of product k's unobserved characteristics at time t, γ_{kt}, and a residual u_{kt}.

Erickson and Pakes (2011) suggest a two-step approach to estimation. First, estimate the model

$$\ln(p_{kt-1}) = h_{t-1}(Z_k) + \eta_{kt-1}. \tag{6}$$

Next, estimate a hedonic model for log price changes, including the lagged residual from the first-stage regression in equation 6 as an additional predictor,

$$\Delta \ln p_{kt} = \tilde{h}_t(Z_k, \hat{\eta}_{kt-1}) + v_{kt}, \tag{7}$$

where $\tilde{h}_t$ is the hedonic function for log price changes from period $t-1$ to period t rather than for log price levels in period t.

Including the first-stage residuals as predictors in equation (7) allows the hedonic function

account for heterogeneity in observationally identical goods that nevertheless feature different prices, to the extent that the differences in base period prices are correlated with price changes.[7]

We define our geometric hedonic price indices as follows:

$$\begin{aligned}
\ln \Phi_{t-1,t}^{LH_{geo}} &= \sum_{k \in \mathbb{CX}_{t-1,t}} s_{kt-1} \ln \widehat{\left(\frac{p_{kt}}{p_{kt-1}}\right)} \\
\ln \Phi_{t-1,t}^{PH_{geo}} &= \sum_{k \in \mathbb{CE}_{t-1,t}} s_{kt} \ln \widehat{\left(\frac{p_{kt}}{p_{kt-1}}\right)} \\
\ln \Phi_{t-1,t}^{TQH_{geo}} &= \sum_{k \in \mathbb{CEX}_{t-1,t}} \left(\frac{s_{kt-1} + s_{kt}}{2}\right) \ln \widehat{\left(\frac{p_{kt}}{p_{kt-1}}\right)}.
\end{aligned} \tag{8}$$

The set $\mathbb{CX}_{t-1,t}$ comprises continuing goods sold in both periods $t-1$ and t as well as exiting goods, sold in period $t-1$ but not in period t. The set $\mathbb{CE}_{t-1,t}$ comprises continuing goods and entering goods, which are sold in period t but not in period $t-1$. The set $\mathbb{CEX}_{t-1,t}$ comprises continuing, entering, and exiting goods. The expenditure shares s_{kt-1} and s_{kt} are defined relative to all goods sold in periods $t-1$ and t, respectively (i.e., the set $\mathbb{CX}_{t-1,t}$ for period $t-1$, and the set $\mathbb{CE}_{t-1,t}$ for period t).[8]

Note that the hedonic predictions that result from estimating equation (7) are for changes in log prices. In other words, we predict $\widehat{\Delta \ln p_{kt}} = \ln \widehat{\left(\frac{p_{kt}}{p_{kt-1}}\right)}$, rather than predicting prices individually in both periods and then calculating the change in predicted prices. We then enter those predicted log price changes into the price indices in equation (8) directly. That is, our procedure predicts the variables directly as they enter the index number formulas.

[7]Note that this procedure treats entering and exiting goods asymmetrically, because entering goods do not have a price or residual in period $t-1$. Erickson and Pakes (2011) only consider a Laspeyres-type index, so they do not confront this issue, but we encounter this asymmetry when estimating the Paasche and Tornqvist indices. To implement the procedure for those indices, we assume that $\eta_{kt-1} = 0$ for entering goods. In a robustness exercise reported in Ehrlich et al. (2023) using data from the NPD Group with traditional econometrics rather than machine learning, we we find very similar results if we replace the predicted price relatives for entering goods with those from a hedonic regression that uses current period rather than lagged residuals and is otherwise equivalent to equation (7).

[8]Note that the weights are calculated using observed expenditure shares—they do not make use of imputed prices. Doing so is feasible because the Laspeyres index uses the lagged-period weights and the Paasche uses the current-period weights, which are available even when goods exit or enter, respectively.

The definitions in equation (8) also show that we focus on "full-imputation" hedonic price indices in this paper consistent with Erickson and Pakes (2011) and Bajari et al. (2021). Full-imputation indices use exclusively predicted prices or price changes and are interpretable as characteristics indices. As shown in our companion paper (Ehrlich et al. (2023)), they are less subject to chain drift given that predicted prices and price changes are less volatile.

3 Data

We use the Nielsen Retail Scanner data (also referred to as RMS) from the Kilts Center at the University of Chicago Booth School of Business for the 2006 to 2015 period. The original data consists of over 2.6 million products identified by the finest level of aggregation—12-digit universal product codes (UPCs) that uniquely identify specific goods.[9] The Retail Scanner data are collected from over 40,000 individual stores from approximately 90 retail chains in over 370 MSAs in the United States. Total sales in Nielsen Retail Scanner are worth around $2 trillion and represent 53% of all sales in grocery stores, 55% in drug stores, 32% in mass merchandisers and 2% in convenience stores.

Nielsen organizes barcode-level goods into 10 departments, 119 product groups and over 1,000 product modules.[10] A typical department is, for example, dry grocery, which consists of 41 product groups such as baby food, coffee, and carbonated beverages. Within the carbonated beverage product group, there are product modules such as soft drinks and fountain beverages. The product groups are classified into food and nonfood sectors based on a correspondence developed by the Bureau of Labor Statistics.

The Retail Scanner data consists of more than 100 billion unique observations at the week-store-UPC level. We first aggregate the weekly data into monthly according to the National Retail Federation (NRF) calendar and then aggregate the monthly data to quarterly.[11] The

[9] Technically, the data include both a UPC code and a UPC version code. The unique product identifier used in the analysis is the combination of the UPC and UPC version code.

[10] We analyze data for the 110 product groups listed in Ehrlich et al. (2023).

[11] NRF calendars are collected from the NRF website: https://nrf.com/resources/4-5-4-calendar. Before

NRF calendar is a guide for retailers that ensures sales comparability between years by dividing a year into months based on a 4 weeks-5 weeks-4 weeks format. The layout of the calendar lines up holidays and ensures the same number of Saturdays and Sundays in comparable months. The NRF calendar thus ensures the comparability of the aggregated sales over time. The median number of individual products in a product group is $2,767$. Average product-level sales within the quarter range from $\$16,000$ at the 5th percentile to $\$290,000$ at the 95th percentile.

Products of different sizes and packaging feature different UPC codes. To ensure comparability between prices, we follow Hottman, Redding, and Weinstein (2016) and normalize UPC prices to the same units (e.g., ounces), utilizing the size and packaging information provided by Nielsen. We also follow those authors and Redding and Weinstein (2020) by winsorizing product-level price changes at the top and bottom 1% of price changes in each product group. The normalization of UPC prices to the same units carries over to the normalization of quantities to the same units.

4 Hedonic Prediction Using a Machine Learning Model

This section describes our machine learning approach to estimating the hedonic functions h_t and $\tilde{h}_t$ described in the previous section. In addition to describing the system architecture, the section also discusses how we test, train, and validate the model and how we convert the prediction model's primitive outputs into price indices. The next section presents our results.

aggregating to the quarterly frequency, we drop outliers, defined as the observations with price above 3 times median or below 1/3 of median for each UPC in a given month. We also drop observations with sales quantities greater than about 24 times the median sales quantity in a given month.

4.1 Overview of model structure

Our machine learning approach is designed to take advantage of the unstructured information available in the product descriptions from the Nielsen data provided by the Kilts Center for Marketing at the University of Chicago. The lack of pre-encoded variables and the large amount of data overall combine to make traditional econometric techniques poorly suited to applying hedonic methods to this data. In contrast, our machine learning approach is designed to address all of these challenges.

The product descriptions in the data are generally not coded to be human-intelligible. For instance, two product descriptions for soft drinks are: `ZR DT LN/LM CF NBP CT` and `NATURAL R CL NB 12P`. A product description for toilet paper is: `DR W 1P 308S TT 6PK`. A human analyst could decipher portions of these descriptions: `DT` means "diet," `12P` means "twelve pack," `1P` means "one ply," `308S` means "308 sheets," etc. It would not be feasible for human analysts to encode such data at scale, and simple dictionaries would be fooled (e.g., the `P`-suffix means "pack" for soft drinks and "ply" for toilet paper).

To address these challenges, we have implemented deep neural networks to predict product prices and price changes from these product descriptions.As a supervised machine learning method, the system is trained to produce these predictions by being shown a large number of product description and price pairs. Deep neural networks require relatively little explicit input transformation work on the part of the developer. Instead, each input to the network is presented in a relatively "raw" form; the training process learns both how to transform the input into a useful format, and how to use the transformed input to make a price prediction. Unlike most explicit input transformations implemented by human engineers, the input transformations formulated by a neural network are generally not human-understandable. A network's transformation of an input, prior to its final classification, is sometimes referred to as an "embedding," because the input is embedded into a high-dimensional numerical space.

Today's best-performing textual systems often rely on "pretrained embedding" approaches. The core idea is first to train a neural network on a vast library of labeled examples, thereby

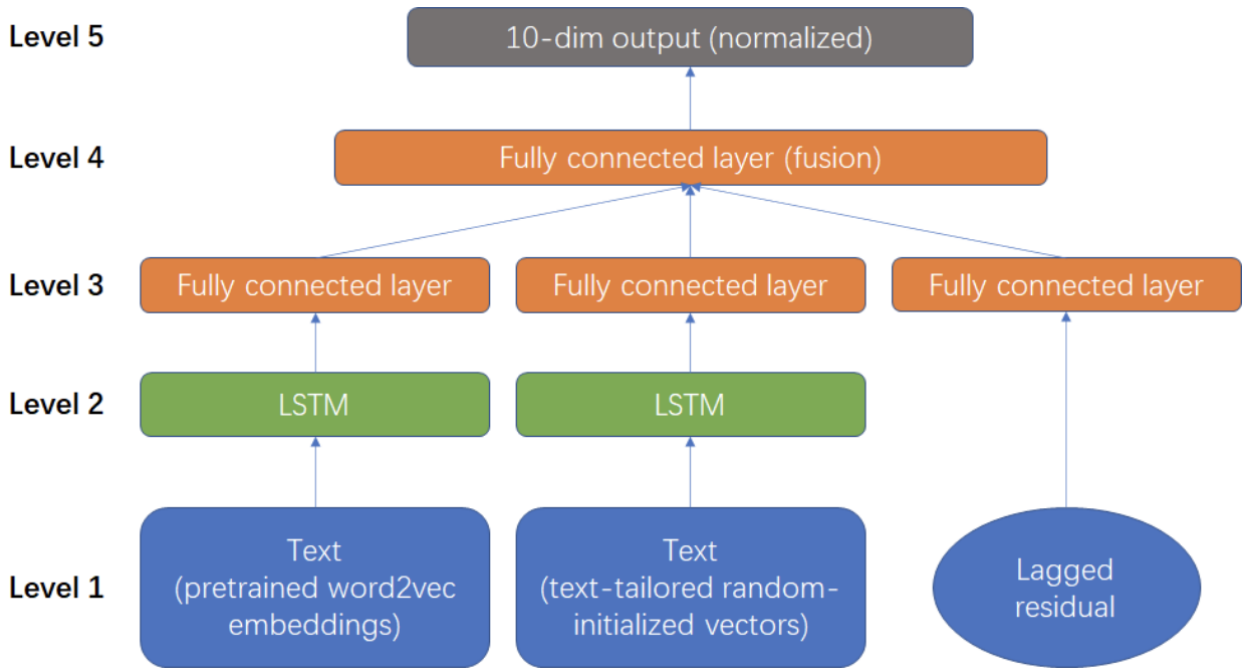
yielding a network that both transforms the input and makes an accurate prediction using that transformed representation. After the pretraining step, the researcher takes the portion of the network that performs the input transformation—that is, the learned embedding—and uses it in a novel network, which aims to accomplish a related but distinct prediction task. In this way, a practitioner can benefit from the vast data sets used in the pretraining set, even when the practitioner's own task may not have such data volumes available.

Pretrained embedding approaches now dominate the field, but it is useful to remember that this pretrain-and-transplant-the-embedding sequence is not strictly necessary. The standard neural network training procedure yields both an input transformation function and a predictor based on that transformed input, i.e., the machine learning procedure can produce a set of "custom" embeddings trained exclusively on the source data as an output. Therefore, the vast pretraining data set may be helpful but is not essential. That said, a recent review found that pretraining helps to make breakthroughs in language representation learning, and thus improves performance for many natural language processing tasks, such as reading comprehension tasks, even if the task features a limited data set (Lan, Chen, Goodman, Gimpel, Sharma, and Soricut, 2019).

Figure 1 displays a schematic diagram of our machine learning architecture. The bottom layer of the architecture illustrated in the figure encodes the product description for use by the network. We use the UPC text descriptions from the Nielsen product master file as the primary input feature. We represent each word in the text as two numeric vectors in a 300-dimensional space, which are the embeddings described above. We use two separate embeddings for each input. The first is a set of pretrained embeddings produced by the Word2Vec algorithm (Mikolov, Chen, Corrado, and Dean, 2013). The second is a set of embeddings that we train from scratch on the dictionary of Nielsen product description codes. As seen in the examples, the Nielsen product descriptions are a mix of standard English usage that one would find in other corpora and strange combinations of characters that are unique to the stock-keeping task. Our hybrid approach allows the network to

exploit external pretrained embeddings when possible, but it does not prevent the system from learning representations for odd Nielsen-specific text sequences. When we use the model to predict price changes, as opposed to price levels, we include the lagged residual from the price level estimation in equation 6 as an additional input into the network.

Figure 1: Machine Learning for Hedonic Prediction Model Architecture



In the second layer of the network architecture, we feed each distinct encoding into a "Long Short-Term Memory" component, or "LSTM." An LSTM component offers a way for the network to represent sequences of words, not just standalone words. The exact construction and functioning of LSTMs are beyond the scope of this paper, but practically, an LSTM takes as an input a sequence of word-by-word input representations and emits a single "full sequence" representation (Hochreiter and Schmidhuber, 1997). In other words, the arrows from layer 1 to layer 2 of Figure 1 describe single words, while the arrows from layer 2 to layer 3 and above describe full word sequences (in our case, an entire product description).

Layer 3 of the ML architecture permits combinations of features that are specific to

one input mode (i.e., pretrained or customized embeddings). These combinations are then combined and potentially transformed in fully-connected fusion layer in level 4. The fully-connected fusion layer allows the system to learn how to combine the competing evidence supplied by the alternative pretrained and custom embeddings. In Section 4.7 below, we document the performance of this hybrid encoding scheme relative to using either set of embeddings individually for a select number of product modules.[12]

Finally, in Layer 5, the system emits a prediction about the product's price or price change. The network does not emit a price directly, but rather predicts a "price bin" for the input. The networks emit a series of weights that can be interpreted as probabilities over these price bins; that is, they sum to 1. Continuous-quantity prediction tasks in neural networks present various technical issues, which are usually sidestepped by transforming them into discrete classification tasks (Le, Aldeneh, and Provost, 2017). We convert the probability-weighted bins into a traditional continuous price prediction using a procedure described in Section 4.4 below.

4.2 The machine learning classifier objective function

Our ML system classifies product prices and price changes into one of B price bins by optimizing a cross-entropy loss function. The product text descriptions are entered as inputs into the ML system described in Section 4.1, which translates them into 300-dimensional vectors or embeddings X_k. The system uses these embeddings to produce a bin classification Y^k, where Y^k is a B-dimensional vector with classifier scores for each of B equally-sized bins that partition the observed range of product prices or price changes:

$$\underbrace{Y^k}_{B\times 1} = f(\underbrace{X^k}_{300\times 1}). \tag{9}$$

We set $B = 10$ in this application.

[12] In early experiments, we also tried a third method that encodes text using pretrained character-level embeddings. This additional method offered no accuracy improvement over the two-item approach above.

The classifier scores Y_b^k, $b = 1 \cdots B$, are then translated into probabilities as:

$$\underbrace{P(Y = Y^k | X = X^k)}_{B \times 1} = \frac{e^{Y^k}}{\sum_{b=1}^{B} e^{Y_b^k}}. \tag{10}$$

Noting that we can observe true product prices, denote the bin in which the price for product k truly lies as c. We use these observed price bins to calculate the cross-entropy loss for product k as:

$$L^k = -\log P(Y = Y_c^k | X = X^k) = -\log \left(\frac{e^{Y_c^k}}{\sum_{b=1}^{B} e^{Y_b^k}} \right). \tag{11}$$

We define quantity weights w_k as each product k's share of unit sales within its product group:

$$w_k = \frac{N_k}{\sum_j N_j}, \tag{12}$$

where N_k is the unit sales quantity for product k.[13]

We then define the total system loss as the weighted sum of product-level cross-entropy losses:

$$L = \sum_k (w_k L^k). \tag{13}$$

In the training process, the system searches for the function f that minimizes the total loss L.

4.3 Training and validation

The machine learning system described above is not designed to be "human interpretable" in the same manner as classical regression techniques. The model implicitly includes nonlinear transformations and high-order interactions of the input features that a human analyst would be unlikely to include in a classical specification. Likewise, standard assessments of model fit and selection such as R^2, AIC, and BIC are not directly applicable to the deep neural

[13] In the first-difference specification, we define N_k to be the average unit sales quantity of product k in periods $t-1$ and t.

network. Nonetheless, it is possible to assess the model's performance and control for over-fitting with robust validation using out-of-sample tests.

We split the sample into models defined by product group-year-quarter combinations and perform model training and predictions in each of the models independently. We split the data into training, validation, and test sets using proportions of 70%, 20%, and 10%, respectively, which is common in the machine learning literature. The training data set is used to estimate the embeddings X^k, classifier function f, and bin probabilities $P(Y^k)$. We train separate models for each product group-quarter. We train each model on the training data set for a number of "epochs," periods in which each example in the training set (i.e., a product description and observed price bin in the product group-quarter) is presented to the classifier one time. We randomly initialize the system to begin the first epoch. We begin subsequent epochs with the system in the state that it concludes the previous epoch. The loss function in equation (13) is minimized within each epoch using the Adam gradient descent algorithm (Kingma and Ba, 2014).[14]

In practice, this training procedure is likely to over-fit the model to the training data in later epochs. To avoid this overfitting problem, we train the model for a significantly larger number of epochs than is likely to be optimal (in this application, 50 epochs). We then apply the models trained in each of the epochs to the validation data set to assess the models' out-of-sample accuracy. We select the model from the best-performing epoch using the model's unweighted "near accuracy" as our selection criterion. We define the model's near accuracy as the unweighted proportion of products in the validation data set for which the model assigns the highest probability to the correct price bin or an adjacent bin. Finally, we assess the out-of-sample performance of our model in the test data set.[15]

We prefer unweighted accuracy to weighted accuracy as our model (epoch) selection

[14]The Adam algorithm requires users to choose certain hyperparameters. We explored the hyperparameters in a limited set of product groups in unreported preliminary experiments and then held the hyperparameters fixed in the main analysis. A strength of the Adam algorithm is that "The hyper-parameters have intuitive interpretations and typically require little tuning" (Kingma and Ba, 2014).

[15]We split the data into training, validation, and test data sets once per product group-quarter; that is, we do not use k-fold cross-validation.

criterion for two reasons. First, the validation data sets can be based on relatively small samples, as they contain 20% of the product group-year-quarter's observations. Therefore, a model's weighted near accuracy can depend in practice on a small number of high-sales products, leading to instability. We have found that using unweighted near accuracy produces more stable results. Second, there is not an obvious reason *a priori* that higher-sales volume products should be more representative of the relationship between product descriptions and prices than lower-sales volume products. Therefore, the unweighted near accuracy arguably corresponds more closely to the primitive goal of the machine learning procedure, which is to match product descriptions to price bins. Section 4.6 discusses small sample issues in the context of machine learning. It presents data indicating that the model's performance is more reliable when using unweighted accuracy as the selection criterion than when using unweighted accuracy.

4.4 Converting classifier output to price predictions

The machine learning classifier estimates the probabilities of each item falling into quantiles of product-module-specific price and price change distributions.[16] In order to construct hedonic price indices, we must convert those probabilities into continuous point estimates. In principle, we could calculate point estimates by taking the inner product of the estimated bin probabilities with the mean values of prices or price changes in the bins. A problem that arises with that approach is that products will often have very low estimated probabilities of falling into some bins. Including very low-probability bins in the calculation of predicted prices may add more noise than signal to the estimate.

We use a statistical procedure closely related to the concept of a Receiver Operating Characteristic (ROC) curve to determine the optimal set of bins to include in the calculation of predicted prices or price changes. Our procedure chooses a cutoff estimated probability value that a product lies in a particular bin, which trades off between the classifier's specificity

[16] Those quantiles are calculated separately for each product module and each quarter.

and its sensitivity. The classifier's specificity is defined as one minus its false positive rate (FPR), or the rate at which it incorrectly classifies a product as belonging to a particular bin. The classifier's sensitivity is its true positive rate (TPR), the rate at which it correctly classifies a product as belonging to a particular bin. A "predicted positive" classification of a product into a particular bin occurs when the predicted probability is greater than the threshold, so both the FPR and the TPR will depend on the cutoff.

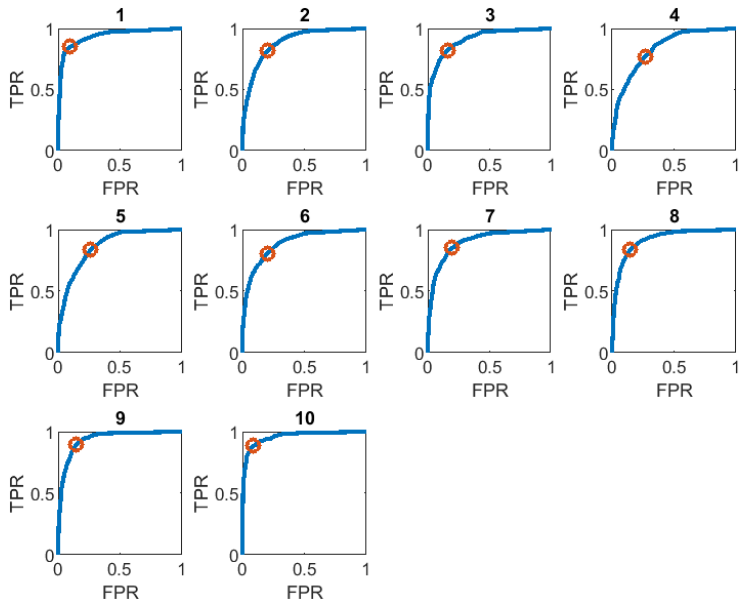
We calculate a set of modified bin probabilities to use for price prediction as follows. If the estimated probability for a bin is lower than the cutoff, we set the modified probability to zero. We then re-normalize the non-zero modified probabilities to sum to one. We calculate the inner product of the re-normalized bin probabilities and the mean prices to form the predicted price. We use an analogous procedure to predict price changes.[17] Appendix A describes the procedure in more detail.

Figures 2 and 3 below illustrate our ROC curve procedure using the example of soft drinks in 2014Q1. Figure 2 plots the classifier's FPR on the horizontal axis and the classifier's TPR on the vertical axis for each of 10 price bins. The blue curves in each plot show the FPRs and TPRs that result from a series of candidate cutoff probabilities ranging from 0 to 0.99. The blue ROC curves thus illustrate the tradeoff between classifier specificity and sensitivity posed by different potential values of the cutoff probability. As we formalize in Appendix A, for each bin, we choose the cutoff probability that minimizes the distance from the perfectly fitting model, represented by the $(0, 1)$ point on the ROC plots. The red circles in each plot represent the optimal choices. Figure 3 illustrates how the TPRs and FPRs vary with the candidate cutoff probability for each bin.

In addition to the ROC curve procedure described here, we have also experimented with a simpler *ad hoc* procedure in which we use only the bins with the two highest estimated probabilities and renormalize those two probabilities to sum to one. Aside from being *ad hoc*, the simple procedure fails to account flexibly for the variation in the model's fit across

[17] To be precise, we predict log prices and log price changes, rather than price levels and price changes *per se*.

Figure 2: ROC Curves for Soft Drinks, 2014q1

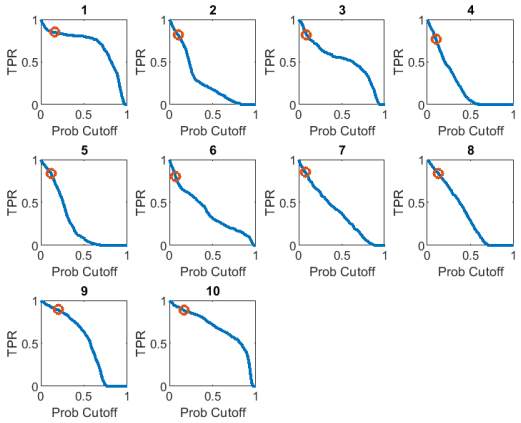


The figure displays ROC curves for each of the 10 price change bins for soft drinks in 2014q1. The false positive rate is on the horizontal axis of each plot, and the true positive rate is on the vertical axis. The optimal tradeoff between sensitivity and specificity for each bin is indicated by the red circle, which represents the point on the ROC curve that has the shortest Euclidian distance to the (0,1) point corresponding to perfect classification.

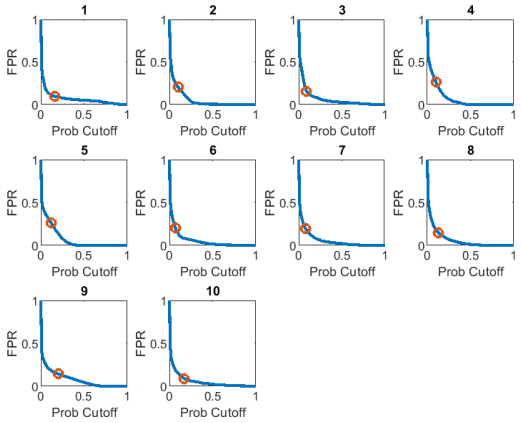
product groups and time. Some product groups exhibit concentrated probabilities across bins, while others have flatter distributions. The ROC procedure provides a principled and flexible way to select which price and price change bins to include in our price predictions.

Figure 3: True and False Positive Rates for Soft Drinks, 2014q1

(A) True Positive Rates



(B) False Positive Rates



The figure shows how the true and false positive rates for each bin move as the cutoff probability increases from 0 to 0.99. The red circles indicate the calculated optimal cutoff probabilities for each bin.

4.5 Model performance

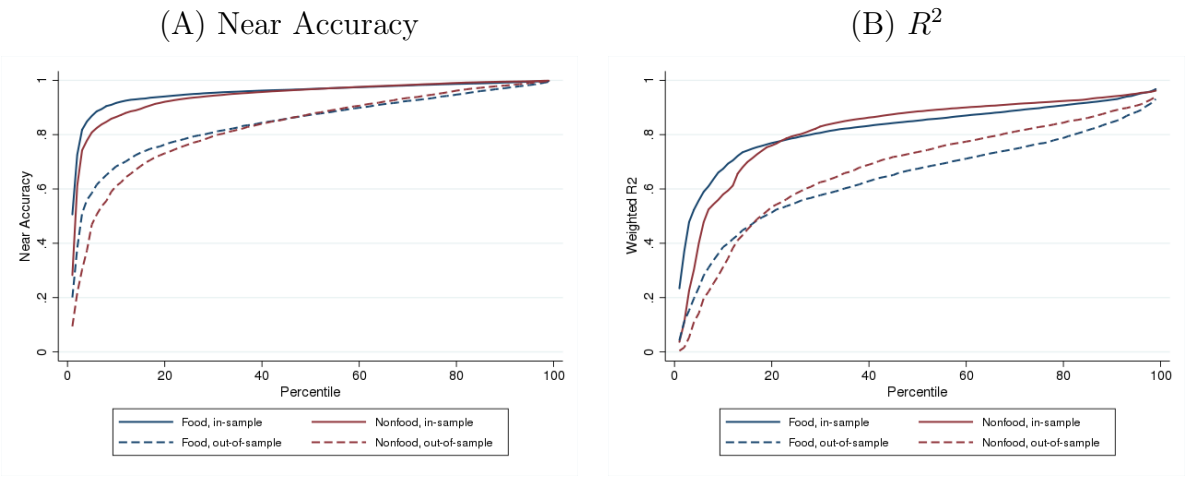
Figure 4 illustrates the procedure's performance in predicting log product prices, while Figure 5 illustrates the performance in predicting log price changes. In both figures, Panel (A) measures performance by the prediction near accuracy, and Panel (B) measures performance by the prediction R^2.[18] Both panels display percentiles of results across prediction models trained for each product group-quarter. Each panel displays four lines, for food and nonfood product groups, and both in-sample and out-of-sample predictions, for each product group-quarter.

Figure 4 shows that the model achieves high predictive accuracy for price levels in most product groups. Panel (A) shows that the median in-sample near accuracy for both food and nonfood product groups is above 90%. As expected, the model's out-of-sample accuracy is lower than the in-sample accuracy, but at the median it is still well above 80% for both food and nonfood product groups. Panel (B) shows the distribution across product group-quarters of R^2s from simple regressions of observed log prices on predicted log prices. The median

[18] As noted in Section 4.3, we define the near accuracy as the probability that the classifier correctly identifies the exact or an adjacent price bin as being the most likely.

in-sample R^2 is roughly 85% for the food product groups and nearly 90% for the nonfood product groups. Again, the model is less accurate out of sample, with median out-of-sample R^2s of roughly 70% for the food product groups and 75% for the nonfood product groups. The model's predictive performance is comparable to that of Bajari et al. (2021), who report out-of-sample R^2s of 80–90% in their best-performing specifications using the rich product text and image information in their data set.

Figure 4: Measures of Fit for Deep Neural Network, Log Prices



Panel (A) displays the classifier's "near accuracy," defined as the probability that it correctly identifies the exact or an adjacent log price bin as being the most likely. The percentiles displayed are across product group-quarters. Panel (B) displays the R^2 of predicted log prices across product group-quarters. The estimated hedonic price indices use price predictions for both in-sample and out-of-sample items.

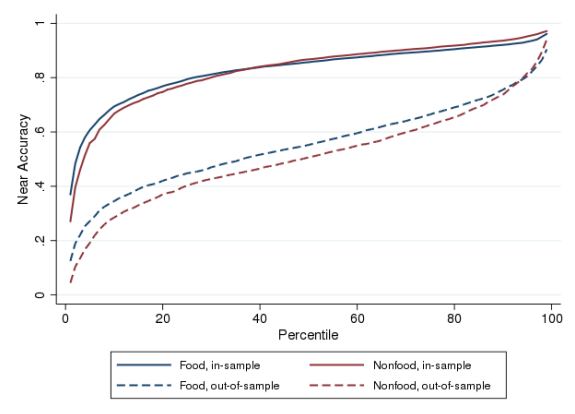
Figure 5 shows our model's predictive performance for log price changes, which are inherently harder to predict than price levels. Panel (A) shows that the median in-sample near accuracy for both food and nonfood price change bins is nonetheless above 80%. The out-of-sample near accuracy for the median product group-quarter is above 50% for the nonfood product groups and nearly 60% for the food product groups. Panel (B) shows the distribution of predictive R^2s for log price changes. The median in-sample R^2 is above 50% for the food product groups and above 40% for the nonfood product groups. The out-of-sample R^2s deteriorate to nearly 20% and below 10% for the food and nonfood product groups, respectively. It is important to recall that those R^2s are for quarterly log price changes, and

that both the in-sample and out-of-sample price changes enter into the hedonic price indices (with the in-sample predictions being substantially more numerous).

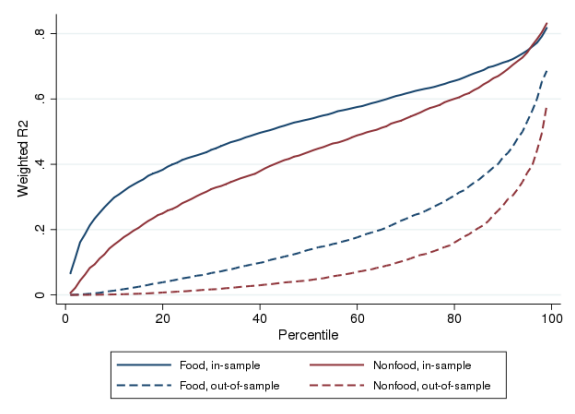
Overall, we consider our ML prediction procedure to be very successful in light of the limited attribute information available in the data set. We believe that commercially available data sets with richer product attribute information will allow for even more accurate prediction than what we have achieved here. In Section 4.6, we present data related to the stability of the ML model's results. We show that the model results are less stable for product group-quarters with smaller numbers of products. We also present evidence supporting our use of unweighted accuracy as our model selection criterion.

Figure 5: Measures of Fit for Deep Neural Network, Log Price Changes

(A) Near Accuracy



(B) R^2



Panel (A) displays the classifier's "near accuracy," defined as the probability that it correctly identifies the exact or an adjacent log price change bin as being the most likely. The percentiles displayed are across product group-quarters. Panel (B) displays the R^2 of predicted log price changes across product group-quarters. The estimated hedonic price indices use price predictions for both in-sample and out-of-sample items.

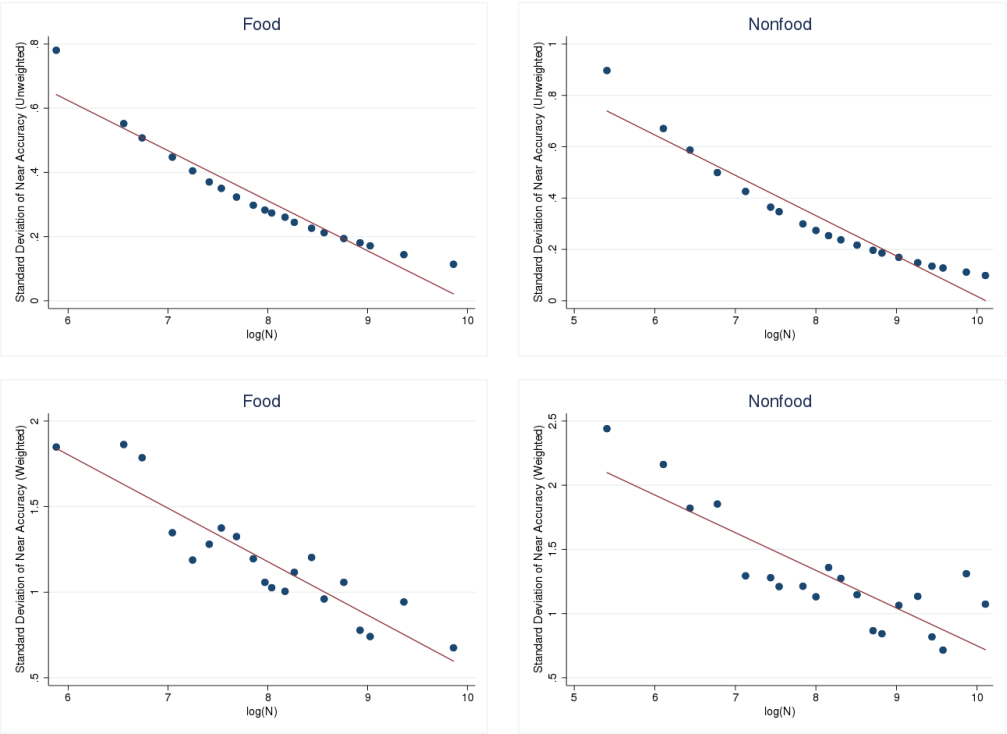
4.6 Assessing the variability of the ML procedure

In this section, we evaluate the stability of the ML procedure's results by bootstrapping the test sample for each of the more than 4,000 models (product group-quarters) that we estimate. As with any econometric exercise, our ML procedure may be sensitive to small

samples. The potential small sample problem is particularly acute in the validation stage of our model because the validation sample is only 20% of the full sample. We conducted a bootstrapping exercise to explore our procedure's sensitivity to small samples. We resampled 50% of the 10% holdout test sample from each product group-quarter with replacement 100 times. We then calculated the standard deviation of the models' near accuracy.

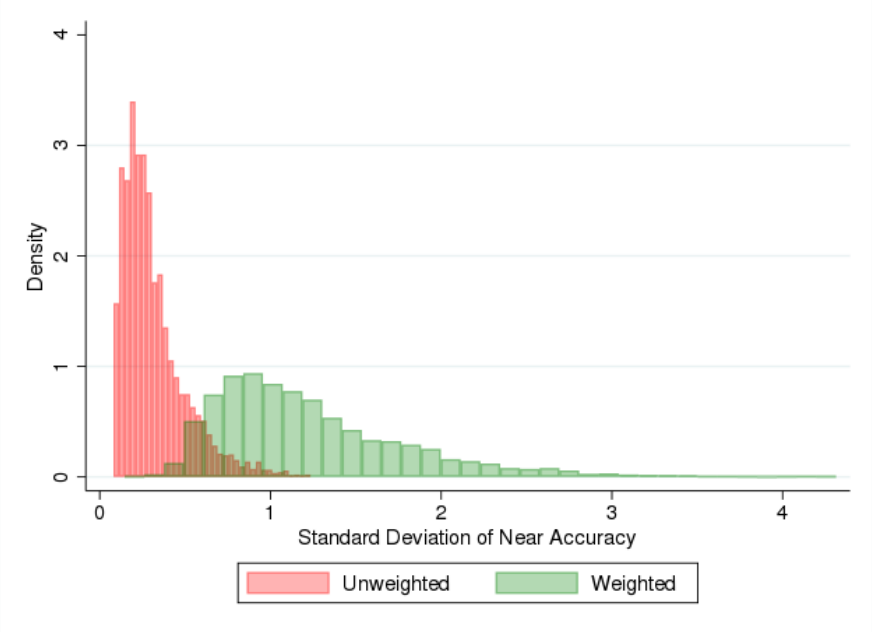
Figure 6 shows that the standard deviation of the model near accuracy decreases with the log sample size; i.e., model accuracy is more stable when sample sizes are larger. The top two panels display bootstrapped results using unweighted near accuracy, while the bottom two panels display bootstrapped results using weighted near accuracy. Unreported results indicate that high levels of product market sales concentration are also associated with more variable near accuracy levels.

Figure 6: Bootstrapping Standard Deviation vs Sample Size



Each panel contains a scatter plot of twenty bin averages of bootstrapped classifier near accuracy relative to log sample size, labeled $log(N)$, along with the estimated line of best fit. The top two panels display results using unweighted near accuracy, while the bottom two panels display weighted near accuracy. The left two panels display results for food product groups, while the right two panels display results for nonfood product groups.

Figure 7: Standard Deviation of Bootstrapping Near Accuracy



The figure displays histograms of the standard deviation of bootstrapped classifier near accuracy across product groups. The pink bars show the histogram for unweighted near accuracy, and the green bars show the histogram for weighted near accuracy.

Figure 7 shows the distribution of the standard deviation of both weighted and unweighted near accuracy across the 4,000-plus models in the bootstrap analysis. The unweighted near accuracy displays lower standard deviations on average, as well as a narrower range of near accuracies. We therefore conclude that using unweighted near accuracy as our model selection criterion in the validation step is likely to produce more stable results than using weighted near accuracy.[19]

[19]In unreported results, we have explored the sensitivity of the ML procedure's results to the potential small sample problem by re-calculating the aggregate Tornqvist indices after dropping the product groups in the bottom quartile of product group sample size. Fortunately, the product groups with small sample sizes also tend to have lower expenditure shares, and they therefore have smaller weights in the aggregate Tornqvist indices. The price indices dropping the product groups with small samples are similar to the full-sample indices, both in their traditional and their hedonic versions. We conclude that our results using our preferred procedure are not particularly sensitive to the potential problem of small samples.

4.7 Comparing pretrained, customized, and combined embeddings

In this section, we briefly compare the model's performance using pretrained, customized, and combined (hybrid) embeddings. We conducted a set of experiments in which we trained the models for a set of 20 product groups (10 in Food, 10 in Nonfood) using pretrained embeddings only, customized embeddings only, or the full combined model described in Section 4.1. Table 1 displays the results for the log price level model, and Table 2 displays results for the log price change model.

The near accuracy rates for the price level predictions in Table 1 range from approximately 77%–94% for Food products and 24%–96% for Nonfood products. The highest rates of near accuracy are for the Baby Food, Household Supplies, and Paper Products product groups. The lowest rate of near accuracy by a substantial margin is for Ice, a product group for which we would not have expected our system to have high predictive power. Overall, the customized, pretrained, and combined embeddings produce similar performances in terms of near accuracy in predicting log price levels, with no clear tendency for one set of embeddings to outperform the others.

The patterns of the price change predictions in Table 2 are similar to the results for the price level predictions. The predictive near accuracy is lower on average across the 20 product groups than for the price level predictions, but not uniformly so. Again, there is no clear tendency for one set of embeddings to outperform the others.

We interpret these results as suggesting that the machine learning system is able to do a surprisingly good job of interpreting the product descriptions in the Nielsen Kilts Center data using the external corpus, despite descriptions that appear difficult to parse to human eyes. We take the combined embedding architecture as our preferred specification and use that architecture for prediction over all of the product groups. We use the predictions from the combined embedding architecture to construct the hedonic price indices.

Table 1: Near accuracy for pretrained, customized and combined embeddings in the log level model

Selected Food Product Groups			
	Combined	**Pretrained**	**Customized**
BABY FOOD	93.5%	91.1%	91.9%
FRESH PRODUCE	91.8%	93.8%	93.3%
COFFEE	90.4%	89.3%	88.6%
CARBONATED BEVERAGES	85.9%	86.9%	86.5%
BREAD AND BAKED GOODS	82.4%	83.4%	83.3%
PREPARED FOODS-FROZEN	80.3%	80.1%	80.8%
MILK	79.3%	79.1%	77.7%
CANDY	77.2%	78.1%	77.8%
SNACKS	77.2%	77.4%	79.2%
CEREAL	77.2%	78.9%	78.1%
Selected Nonfood Product Groups			
	Combined	**Pretrained**	**Customized**
HOUSEHOLD SUPPLIES	96.2%	96.4%	96.0%
PAPER PRODUCTS	95.3%	95.2%	95.6%
SKIN CARE PREPARATIONS	89.5%	89.4%	90.2%
DISPOSABLE DIAPERS	85.5%	88.0%	83.7%
ELECTRONICS, RECORDS, TAPES	84.3%	84.2%	84.2%
HOUSEHOLD CLEANERS	83.9%	82.8%	83.8%
HOUSEWARES, APPLIANCES	75.6%	77.3%	74.0%
LIQUOR	73.9%	74.2%	74.2%
HARDWARE, TOOLS	65.0%	64.6%	64.6%
ICE	24.4%	27.4%	25.5%

This table shows the classifier's near accuracy for log prices in select product groups for various types of embeddings. The column "Pretrained" shows results using embeddings trained on an external corpus of text, while the column "Customized" shows results using embeddings trained on the product descriptions in the Nielsen Kilts Center Retail Scanner Panel. The column "Combined" shows results using a hybrid feature encoding architecture that allows the system to incorporate both pre-trained and customized embeddings.

Table 2: Near accuracy for pretrained, customized and combined embeddings in the log first-difference model

Selected Food Product Groups			
	Combined	**Pretrained**	**Customized**
BABY FOOD	68.9%	68.8%	68.9%
FRESH PRODUCE	55.7%	55.8%	55.4%
COFFEE	55.0%	57.0%	53.4%
CARBONATED BEVERAGES	68.1%	68.7%	66.2%
BREAD AND BAKED GOODS	50.8%	50.9%	51.5%
PREPARED FOODS-FROZEN	57.4%	58.0%	59.8%
MILK	54.4%	56.2%	54.4%
CANDY	50.5%	50.8%	50.5%
SNACKS	53.6%	54.7%	55.2%
CEREAL	46.5%	46.7%	47.9%
Selected Nonfood Product Groups			
	Combined	**Pretrained**	**Customized**
HOUSEHOLD SUPPLIES	45.8%	46.5%	45.1%
PAPER PRODUCTS	48.6%	49.1%	47.7%
SKIN CARE PREPARATIONS	50.9%	51.8%	51.0%
DISPOSABLE DIAPERS	66.5%	66.1%	68.2%
ELECTRONICS, RECORDS, TAPES	43.7%	45.8%	43.0%
HOUSEHOLD CLEANERS	42.5%	39.7%	40.1%
HOUSEWARES, APPLIANCES	50.7%	51.7%	50.3%
LIQUOR	53.2%	53.7%	55.4%
HARDWARE, TOOLS	51.1%	51.2%	50.0%
ICE	38.8%	34.3%	38.5%

This table shows the classifier's near accuracy for log price changes in select product groups for various types of embeddings. The column "Pretrained" shows results using embeddings trained on an external corpus of text, while the column "Customized" shows results using embeddings trained on the product descriptions in the Nielsen Kilts Center Retail Scanner Panel. The column "Combined" shows results using a hybrid feature encoding architecture that allows the system to incorporate both pre-trained and customized embeddings.

5 Estimated Price Indices

We calculate aggregate price indices over the period 2006q4–2015q4 separately for food and nonfood product groups. The aggregate indices for each category are calculated as Tornqvist indices using Divisia weights.[20] Figure 8 shows traditional and hedonic Tornqvist indices for the food and nonfood categories, respectively. The indices are normalized to a level of one at the start of the sample period and have been chained quarter-over-quarter.

The two product categories exhibit different price trends and hedonic adjustments over our sample period. There has been modest inflation in the food product groups, with the traditional Tornqvist index rising 5.9% cumulatively over the sample period. The hedonic Tornqvist index shows cumulative inflation of 2.8% percent over the period, implying a total hedonic adjustment of negative 3.1 percentage points for food. In contrast, there has been steady deflation in the nonfood product groups. The traditional Tornqvist index indicates cumulative deflation of 14.4 percent over the sample period, while the hedonic Tornqvist index indicates cumulative deflation of 12.9 percent. Thus, the total hedonic adjustment was positive 1.5 percentage points for nonfood.

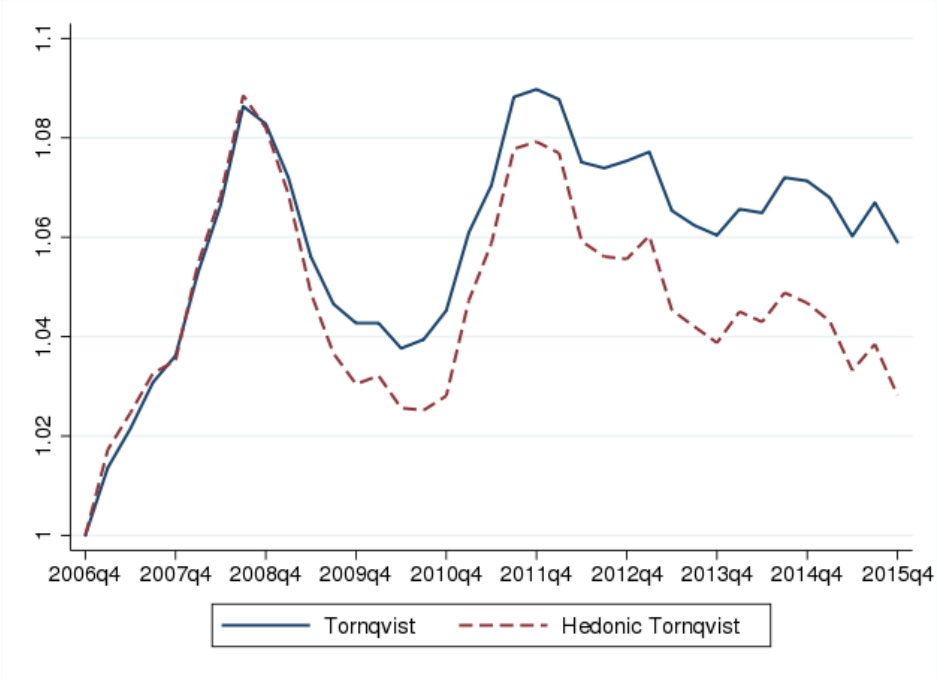
Table 3 and Figure 9, which display inflation as measured by several price indices over our sample period, shed some light on why the hedonic adjustments for food and nonfood product groups take different directions. The negative 3.1 percentage point cumulative hedonic adjustment to the Tornqvist index for the food product groups stems from a negative 2.6 percentage point adjustment to the Laspeyres index and a negative 3.5 percentage point adjustment to the Paasche index.[21] Equations (1) and (8) show that the hedonic adjustment to the Laspeyres index reflects two factors: first, the use of imputed price changes rather than observed price changes for continuing items, and second, the inclusion of exiting products in the index. The negative hedonic adjustment to the Laspeyres index suggests that exiting

[20] In other words, the weights for each product group are based on the product group-level average expenditure shares in quarters $t-1$ and t and are updated each quarter.

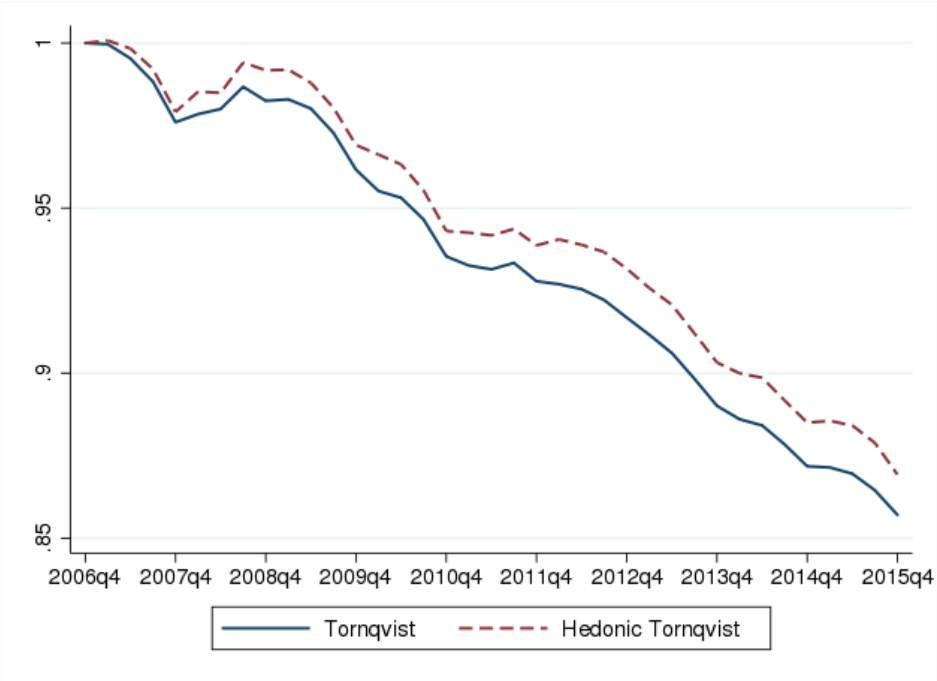
[21] The Tornqvist index is the geometric mean of the geometric Paasche and Laspeyres indices, so the hedonic adjustment to the Tornqvist index equals the simple average of the hedonic adjustments to the Paasche and Laspeyres indices.

Figure 8: Traditional and Hedonic Price Indices for Food and Nonfood Product Groups

A. Food Product Groups



B. Nonfood Product Groups



This figure shows Tornqvist price indices aggregated across product groups for food in the top panel and nonfood in the bottom panel. The indices are chained quarter-over-quarter and are aggregated across product groups using Divisia expenditure weights. The blue lines show traditional Tornqvist indices and the dashed red lines show hedonic Tornqvist indices calculated using the combined product embeddings.

Table 3: Cumulative Inflation Rate, 2006q4 to 2015q4

	Food		Nonfood	
	Cumulative	Quarterly	Cumulative	Quarterly
Laspeyres	15.2%	0.39%	-15.7%	-0.47%
Hedonic Laspeyres	12.6%	0.33%	-10.5%	-0.31%
Paasche	-2.6%	-0.07%	-13.1%	-0.39%
Hedonic Paasche	-6.1%	-0.17%	-15.2%	-0.46%
Tornqvist	5.9%	0.16%	-14.4%	-0.44%
Hedonic Tornqvist	2.8%	0.08%	-12.9%	-0.38%

The data come from the Nielsen Scanner Panel. The quarterly inflation rates are compound (geometric) averages.

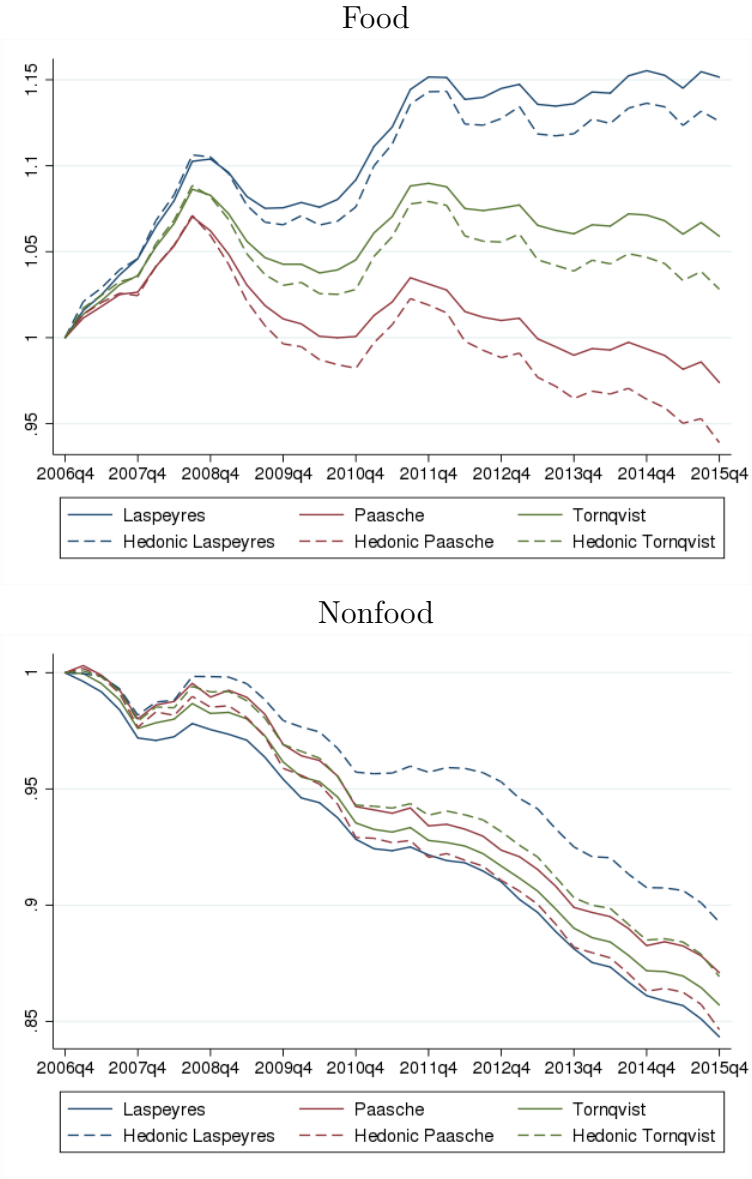
items were negatively selected in terms of consumer desirability (Pakes, 2003). Equations (2) and (8) show that the hedonic adjustment to the Paasche index reflects the use of imputed price changes and the inclusion of entering items. The negative hedonic adjustment to the Paasche index suggests that entering items were positively selected in terms of consumer desirability. An important implication of the hedonic adjustment to the price index for food product groups is that this sector has seen meaningful quality improvement via product turnover, despite not normally being considered to feature rapid technological progress.[22]

The pattern of hedonic adjustments in the nonfood product groups is more complex. The small positive hedonic adjustment to the Tornqvist index stems from a negative 2.1 percentage point hedonic adjustment to the Paasche index, which is slightly more than offset by a positive 5.2 percentage point hedonic adjustment to the Laspeyres index. The Paasche index reflects the influence of entering products, so its negative hedonic adjustment is consistent with the intuition that entering items are positively selected in terms of hedonic desirability relative to their prices. The Laspeyres index reflects the influence of exiting products, so its

[22]In our companion paper (Ehrlich et al., 2023), we consider the implications of chain drift finding that the Tornqvist index is more subject to chain drift than the hedonic Tornqvist. This inference is based on comparisons of GEKS adjustments to the Tornqvist and hedonic Tornqvist indices using NPD data. Exploring chain drift in the current machine learning setting is more complex because implementation of GEKS adjustments requires estimation of hedonic models over different horizons, which would be computationally challenging. Nonetheless, the inference from the companion paper is likely to hold more generally. Hence, the gap we find between the traditional Tornqvist and hedonic Tornqvist is likely to be even larger taking into account chain drift.

positive adjustment suggests that exiting products, rather than being negatively selected, as seen in the food product groups and argued by Pakes (2003), are positively selected. This result is counter-intuitive from the perspective of the observed economywide changes in the items sold in the nonfood product groups.

Figure 9: Alternative Traditional and Hedonic Price Indices



This figure shows geometric Laspeyres, Paasche, and Tornqvist price indices aggregated across product groups for food in the top panel and nonfood in the bottom panel. The indices are chained quarter-over-quarter and are aggregated across product groups using Divisia expenditure weights. The hedonic indices are calculated using the combined product embeddings.

Figure 10 provides evidence that the coverage and composition of nonfood items sold in the stores in the Nielsen Retail Scanner Panel changed substantially over our study period. We compare the growth of nominal expenditures in the Nielsen Retail Scanner Panel and the Bureau of Economic Analysis's (BEA's) Personal Consumption Expenditures (PCE) data for select product groups.[23] The figure contains two panels with bar charts that display the ratios of nominal expenditures in the Nielsen Retail Scanner Panel to the corresponding PCE measure calculated in 2015q4, divided by the parallel ratio calculated in 2008q1. A value of one for a product group indicates that expenditures in the Nielsen Retail Scanner Panel grew at the same rate as the PCE data over that period. Values below one indicate that expenditures in the Nielsen Retail Scanner Panel grew more slowly than the BEA's economywide PCE measure, suggesting that the Nielsen Retail Scanner Panel's coverage of sales may have deteriorated. Conversely, values above one indicate that expenditures in the Nielsen Retail Scanner Panel grew more quickly than the BEA's economywide PCE measure.

The top panel displays this measure for a selection of food product groups; the bars generally take values near one, suggesting that the Nielsen Retail Scanner Panel's coverage of economywide food sales was roughly steady over our study period. The bottom panel displays this measure for a selection of nonfood product groups. The values of the bars are much more variable across product groups, with many groups showing values well below one. Ehrlich et al. (2023) presents further evidence that, while the Nielsen Retail Scanner Panel's coverage of economywide spending on food product groups was roughly constant over our sample period, the data's coverage of economywide spending on nonfood product groups deteriorated considerably over that time. We conclude that the Nielsen Retail Scanner Panel's coverage of sales in nonfood product groups tracked the PCE measure less reliably during our study period than its coverage of sales in food product groups.

Figure 11 displays bar graphs showing the relative cumulative changes in prices from

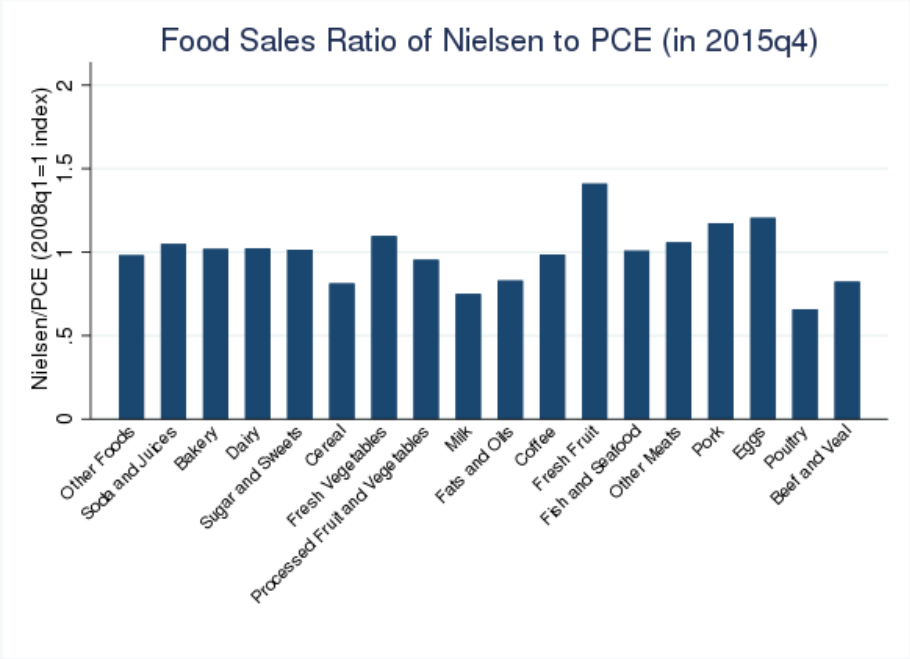
[23]Figures 10 and 11 use a concordance provided to us by BLS between PCE categories and the 1000 or so Nielsen product modules. We have found that at the aggregate food and nonfood levels, using a concordance the 100 plus product groups in the Nielsen data with the PCE does not make an important difference relative to the product module concordance.

2007q1 to 2015q4 between the Nielsen Retail Scanner Panel and the BEA PCE price index in the same product groups shown in Figure 10. Each bar shows the ratio of the cumulative chained growth of the traditional Tornqvist index for a given product group over that period divided by the corresponding cumulative growth of the PCE price index for that product group. A value of one indicates that prices rose at the same rate in the Nielsen data as in the PCE; values above one indicate that prices rose more quickly in the Nielsen data, while values below one indicate that prices rose more slowly in the Nielsen data. The top panel shows that prices in the Nielsen scanner data's food product groups generally tracked the PCE closely during our study period, with the exceptions of sugar and sweets and fresh fruit. The bottom panel shows that, in contrast, many of the nonfood product groups exhibit much lower price growth in the Nielsen scanner data than in the PCE. Notable product groups exhibiting this behavior include dishes, photo equipment, cellphone, personal computer, and clocks.

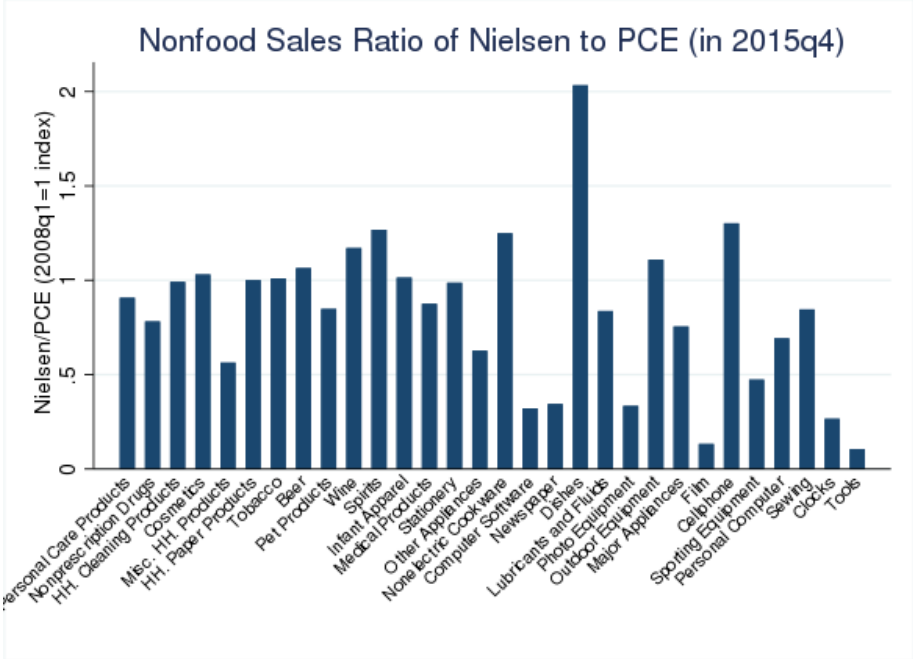
We conclude that our potentially surprising finding of a positive hedonic adjustment to the Laspeyres index for nonfood product groups may thus reflect changes in the Nielsen Retail Scanner Panel's coverage of items sold, rather than a true economywide phenomenon related to the desirability of exiting items. For example, nonfood items covered in the Nielsen Retail Scanner data may increasingly reflect lower quality items relative to those sold in other outlets not covered by Nielsen. In any case, many nonfood items are mainly sold at outlets other than grocery stores and pharmacies, so it is not surprising that the Nielsen Scanner Data do not capture economy-wide trends in the nonfood sector.

Figure 10: Assessing the Change in Nielsen's Coverage of Food and Nonfood Products, 2008–2015

(A) Relative Change in Nominal Expenditures: Food



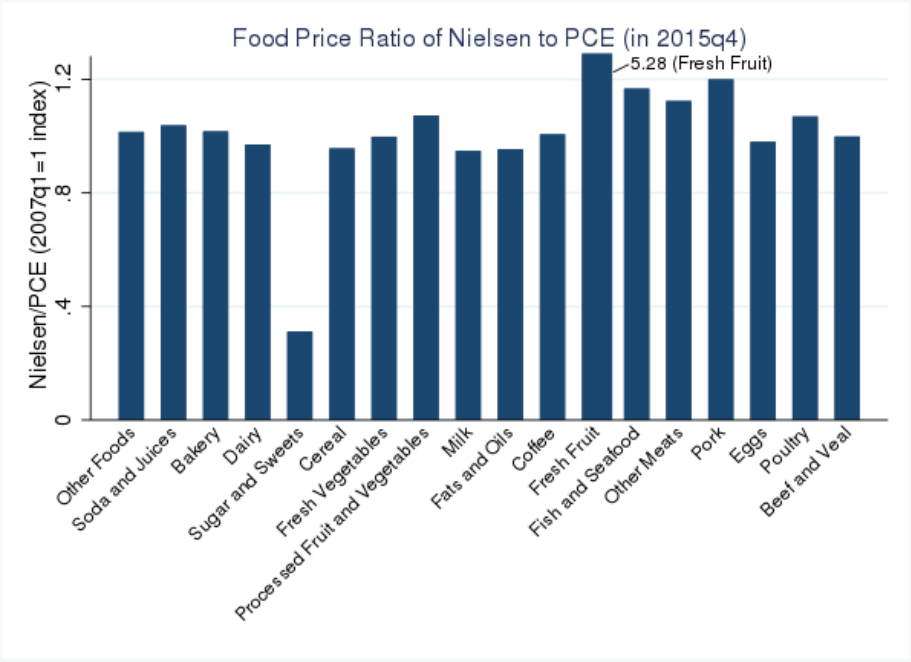
(B) Relative Change in Nominal Expenditures: Nonfood



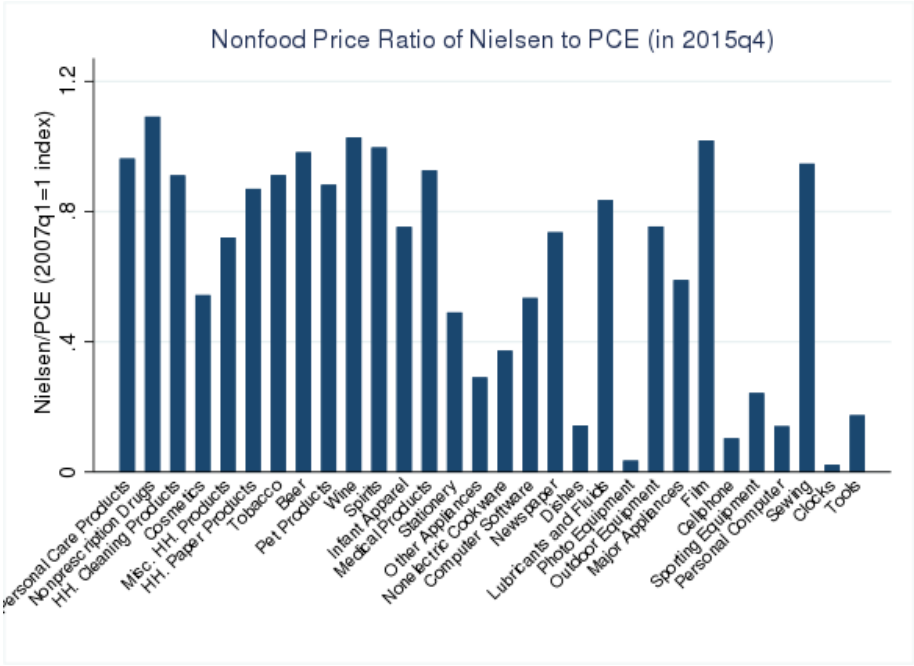
Each bar displays the ratio of nominal expenditures for a product group in the Nielsen Retail Scanner Panel to the corresponding PCE measure calculated in 2015q4, divided by the parallel ratio calculated in 2008q1. A value of one for a product group indicates that expenditures in the Nielsen data grew at the same rate as the PCE data over that period; values below one indicate slower growth in the Nielsen data.

Figure 11: Comparing Measured Price Changes in Nielsen's Food and Nonfood Product Groups to the PCE, 2008–2015

(A) Relative Change in Nominal Prices: Food



(B) Relative Change in Nominal Prices: Nonfood



The bars display the relative cumulative changes in various product groups' prices from 2007q1 to 2015q4 between the Nielsen Retail Scanner Panel and the BEA PCE price index. Each bar shows the ratio of the cumulative chained growth of the traditional Tornqvist index for a given product group over that period divided by the corresponding cumulative growth of the PCE price index for that product group. A value of one indicates that prices rose at the same rate in the Nielsen data as in the PCE; values above one indicate that prices rose more quickly in the Nielsen data.

6 Conclusion

Hedonic price indices hold the potential to estimate the change in consumers' cost of living more accurately than traditional indices, but estimating them at scale from item-level data entails significant challenges. We propose a machine learning procedure to estimate hedonic price indices that allows them to be implemented at scale with little human involvement. Our procedure demonstrates how statistical agencies can make enhanced use of item-level sales and attribute data to produce official statistics. We estimate a large hedonic adjustment to the Tornqvist index for food product groups, which reduces cumulative inflation from 2006q4 to 2015q4 by more than half for those groups. These results suggest that traditional price indices systematically overstate the rate of inflation and understate the rate of real output growth in the Retail Trade sector, even in product groups that do not obviously feature fast technological progress.

References

Abraham, K. G., R. S. Jarmin, B. C. Moyer, and M. D. Shapiro (2022). *Big Data for Twenty-First-Century Economic Statistics*, Volume 79. University of Chicago Press.

Bajari, P., Z. Cen, V. Chernozhukov, M. Manukonda, J. Wang, R. Huerta, J. Li, L. Leng, G. Monokroussos, S. Vijajkuner, and S. Wan (2021). Hedonic prices and quality adjusted price indices powered by AI. Technical report, Institute for Fiscal Studies Cenmap Working Paper CWP 04/21.

Benkard, C. L. and P. Bajari (2005, jan). Hedonic price indexes with unobserved product characteristics, and application to personal computers. *Journal of Business and Economic Statistics 23*(1), 61–75.

Boskin, M. J., E. R. Dulberger, R. J. Gordon, Z. Griliches, and D. Jorgenson (1996). Toward a more accurate measure of the cost of living: Final report to the senate finance committee from the advisory commission to study the consumer price index. Committee print 104-72 (December 1996), Committee on Finance, U.S. Senate.

Bureau of Labor Statistics (2023). Quality adjustment in the cpi. Accessed February 9, 2023.

Byrne, D. M., D. E. Sichel, and A. Aizcorbe (2019). Getting Smart About Phones: New Price Indexes and the Allocation of Spending Between Devices and Services Plans in Personal Consumption Expenditures. *Finance and Economics Discussion Series 2019*(012).

Court, A. (1939). Hedonic price indexes with automotive examples. In *The Dynamics of Automobile Demand*, pp. 98–119. General Motors.

De Haan, J. (2015). A framework for large scale use of scanner data in the dutch cpi. In *Report from Ottawa Group 14th meeting, International Working Group on Price Indices, Tokyo (Japan), May, 2015*. Ottawa Group.

Diewert, W. E. (1978). Superlative index numbers and consistency in aggregation. *Econometrica 46*(4).

Ehrlich, G., J. Haltiwanger, R. Jarmin, D. Johnson, E. Olivares, L. Pardue, M. D. Shapiro, and L. Y. Zhao (2023). Quality adjustment at scale: Hedonic vs. exact demand-based price indices. Unpublished working paper, Census, Maryland and Michigan.

Ehrlich, G., J. Haltiwanger, R. Jarmin, D. Johnson, and M. D. Shapiro (2019). Minding your Ps and Qs: Going from micro to macro in measuring prices and quantities. *AEA Papers and Proceedings 109*, 438–443.

Ehrlich, G., J. Haltiwanger, R. Jarmin, D. Johnson, and M. D. Shapiro (2021). Re-engineering key national economic indicators. *Big Data for Twenty-First Century Economic Statistics*.

Erickson, T. and A. Pakes (2011). An experimental component index for the CPI: From annual computer data to monthly data on other goods. *American Economic Review 101*(5), 1707–1738.

Feenstra, R. C. (1994). New Product Varieties and the Measurement of International Prices. *The American Economic Review 84*(1), 157–177.

Griliches, Z. (1961). Hedonic prices for automobiles: an econometric analysis of quality change. In *The Price Statistics of the Federal Government, General Series No. 73*, pp. 137–196. Columbia University Press for the National Bureau of Economic Research.

Han, S., E. H. Schulman, K. Grauman, and S. Ramakrishnan (2021). Shapes as product differentiation: Neural network embedding in the analysis of markets for fonts. *arXiv preprint arXiv:2107.02739*.

Hochreiter, S. and J. Schmidhuber (1997, nov). Long short-term memory. *Neural Comput. 9*(8), 1735–1780.

Hottman, C. J., S. J. Redding, and D. E. Weinstein (2016). Quantifying the sources of firm heterogeneity. *The Quarterly Journal of Economics 131*(3), 1291–1364.

Jarmin, R. S. (2019). Evolving measurement for an evolving economy: thoughts on 21st century us economic statistics. *Journal of Economic Perspectives 33*(1), 165–84.

Kingma, D. P. and J. Ba (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lan, Z., M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut (2019). A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Le, D., Z. Aldeneh, and E. M. Provost (2017). Discretized continuous speech emotion recognition with multi-task deep recurrent neural network. In *Interspeech*, pp. 1108–1112.

Mikolov, T., K. Chen, G. Corrado, and J. Dean (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Moulton, B. R. (2018). The measurement of output, prices, and productivity. *Report, Productivity Measurement Initiative under The Hutchins Center on Fiscal and Monetary Policy, Brookings Institution, Washington, DC*.

Pakes, A. (2003). A Reconsideration of Hedonic Price Indexes with an Application to PC's. *American Economic Review 93*(5), 1578–1596.

Redding, S. J. and D. E. Weinstein (2020). Measuring Aggregate Price Indices with Taste Shocks: Theory and Evidence for CES Preferences. *The Quarterly Journal of Economics 135*(1), 503–560.

Shapiro, M. D. and D. W. Wilcox (1996). Mismeasurement in the consumer price index: An evaluation. *NBER Macroeconomics Annual 11*, 93–142.

Silver, M. and S. Heravi (2005). A failure in the measurement of inflation: Results from a hedonic and matched experiment using scanner data. *Journal of Business & Economic Statistics 23*(3), 269–281.

Zeng, S. (2021). Hedonic imputation with tree-based decision approaches. Paper prepared for the 36th IARIW Virtual General Conference.

Appendix

A The ROC Procedure

This appendix formalizes our ROC (receiver operating characteristic) curve procedure for selecting the cutoff probability described in Section 4.4.

We calculate the classifier's true positive rate (TPR) for bin b using cutoff probability $\overline{P}$ as:

$$TPR_b(\overline{P}) = \frac{\sum_{k \in \Omega_b} \widehat{P(Y_b^k | X^k)} > \overline{P}}{N_b}, \tag{14}$$

where N_b is the number of products that truly belong to bin b, Ω_b is the set of products that fall into bin b, and $\widehat{P(Y_b^k | X^k)}$ is the estimated probability that product k falls into bin b. In other words, the true positive rate is the sum of the estimated number of products falling into bin b divided by the total number of products that truly fall into the bin. Different values for the cutoff probability $\overline{P}$ will produce different true positive rates. The classifier's true positive rate is also called its *sensitivity*.

We calculate the classifier's false positive rate (FPR) as:

$$FPR_b(\overline{P}) = \frac{\sum_{k \notin \Omega_b} \widehat{P(Y_b^k | X^k)} > \overline{P}}{N - N_b}, \tag{15}$$

where N is the total number of products sold in the product group-quarter under consideration, so the denominator $N - N_b$ is the total number of products that do not fall into bin b. The false positive rate is thus the ratio of the estimated number of products incorrectly classified as falling into bin b, divided by the total number of products that do not truly fall into the bin. The classifier's *specificity* is defined as one minus its false positive rate.

Ideally, the classifier would have a true positive rate of one and a false positive rate of zero. We choose the cutoff probability $\overline{P}^\star$ to minimize the sum of the classifier's squared

false negative rate (one minus its true positive rate) and false positive rate:

$$\overline{P}^{\star} = \arg\min_{\overline{P}} \left(1 - TPR_b(\overline{P})\right)^2 + \left(FPR_b(\overline{P})\right)^2. \tag{16}$$

It is straightforward to extend this procedure to allow for quantity or other weights. A strength of our ROC curve procedure to select the cutoff probabilities is that it allows for arbitrary patterns of estimated probabilities across bins. For instance, the procedure can accommodate "U-shaped" probability profiles across low- and high-priced bins without difficulty.