Identifying Consumer Preferences from User-Generated Content on Amazon.com by Leveraging Machine Learning

Jikhan Jeong[1]

Job Market Paper

Version: December 15, 2020 (the latest version)

[1] Econ Ph.D. student at Washington State University. E-mail: Jikhan.jeong@wsu.edu. The preliminary results were presented at the 2020 Rocky Mountain Advanced Computing Consortium and the 2020 National Association for Business Economics (NABE) Tech conference. This paper also will be presented at and the 2021 American Economic Association (AEA). In addition, I thank Alan Love, Jinhui Bai, Bidisha Mandal, all professors, staff, and students at WSU SES, Melanie Lynette Thongs, Christopher Steven Duke, Roy Obenchain, Peter Mills at WSU. I also thank Julian McAuley (UCSD) to share the dataset in this study. All errors in this paper are mine alone.

Abstract

Inexperienced consumers may have high uncertainty about experience goods that require technical knowledge and skills to operate effectively; therefore, experienced consumers' prior reviews can be useful for inexperienced one. However, one-sided review systems (e.g., Amazon.com) only provide the opportunity for consumers to write a review as a buyer and contain no feedback from the seller's side, so the information displayed about individual buyers is limited. Therefore, this study analyzes consumers' digital footprints (DFs) for programmable thermostats (home energy management devices) to identify and predict unobserved consumer preferences, using a dataset of 141 million Amazon reviews including consumer reviews and product-specific information. In addition, this study identifies consumers' sentiment toward product content dimensions (PCDs) extracted from review text by applying topic modeling and domain expert annotations, while excluding questionable reviews (posted by "suspicious one-time reviewers" and "always-the-same rating reviewers").

This paper obtains three main results: first, I find that the factors that affect consumer ratings are: (a) user' DFs (e.g., average rating across all categories), (b) reviewers' attitudes toward eight product content dimensions (smart connectivity, easiness, energy saving, functionality, support, price value, privacy, and the Amazon's service quality effect), and (c) other prior reviewers DFs (e.g., length of the review summary). Second, extreme gradient boosting (XGBoost) is found to obtain the highest performance for predicting the ratings of potential consumers before they make a purchase or write a review. Third, a convolutional neural network (CNN) on top of Bidirectional Encoder Representations from Transformers (BERT) embedding shows the highest performance for classifying consumers' sentiment toward a specific PCD.

The two main contributions of this study are that it shows: (1) how to use user-generated-content about different products in different categories and at different times to identify unobserved consumer preferences; and (2) a detailed, step-by-step way to combine structured and unstructured data to predict individual heterogeneous consumer preferences and classify their sentiments. Overall, this approach developed in this paper is applicable, scalable, and interpretable for distinguishing important drivers of consumer reviews for different goods in a specific industry and can be used by industry to design customer-oriented marketing strategies.

JEL classification: D80; M21; M31; C45; Q40

Keywords: Consumer preference, Online Reviews, User-Generated Content, Discrete Choice Model, Machine Learning, Deep Learning, Natural Language Processing

1. INTRODUCTION

In recent years, big data analysis has experienced remarkable growth. This growth has been fostered by innovations in computation performance and remarkable successes with artificial intelligence (AI) algorithms. Additionally, these advances have benefitted from increasing volume, diversity, and value of the data.

There are two types of big data: structured data (which have a well-defined data type) and unstructured data (which lack a well-defined data type, such as image, voice, video, and text). Online product reviews generated by consumers contain both structured and unstructured data. For example, while consumers' product ratings fall into the category of structured data, their written reviews are unstructured data. User-generated online product review data can provide useful information for inexperienced consumers because they contain feedback from actual consumers who reveal their preferences for products; such data are quite different from the feedback provided by user focus groups or experts. By leveraging the information from prior review data, inexperienced consumers can reduce their search cost and uncertainty about product quality.

Some reviewers in this study mention the usefulness of previous reviews written by the crowd (other prior reviewers), such as the following: "We bought this model because of the exceptional Consumer Products review/ratings" and "After reading some of the negative reviews, I was hesitant to purchase these units." Firms can also employ user-generated review content to estimate individual consumer preferences, needs, satisfaction, and complaints and to design, develop, and promote new products. For example, Timoshenko and Hauser (2019) demonstrated how to identify consumer needs from user-generated review text on Amazon.com.

Liu, Lee, and Srinivasan (2019) suggested that review data are more likely to be influential for consumers when the product group has more competition, a shorter product history, and weaker brand power. Accordingly, inexperienced consumers may have high uncertainty about experience goods that require technical knowledge and skills when innovative entry firms enter the market. In this case, the information contained in reviews written by experienced consumers may be useful to inexperienced consumers.

This study analyzes the digital footprints (DFs) of reviewers who participate in Amazon's review system, which is a one-sided review system that contains no feedback from the seller's side (Tadelis 2016) and on which buyers can write a review without any fee (Cui, Lui, and Guo 2012.) The raw review data from He and McAuley (2016), gathered between May 1996 and July 2014, are used to generate DFs. This data set contains 142.8 million reviews, including consumer reviews and product-specific information.

In contrast to previous research, which has used Amazon's online reviews for general experience goods (e.g., books, DVDs, and music), this study investigates Amazon's online reviews for a specific experience good (programmable thermostats) requiring enough technical knowledge and skills to install, set up, program, and use it. Consumers who buy a thermostat need to know how different models will increase the energy efficiency in their house and reduce their energy bills. However, in the early stages of thermostat usage, people usually do not know their real-time energy consumption, the cost, and the amount of energy saving that a new thermostat can provide. This means that programmable thermostat consumers typically face high uncertainty.

In particular, consumer uncertainty may be higher than normal when disruptive innovation happens because innovative new firms (e.g., the Nest) enter the market, introduce innovative products (e.g., Wi-Fi thermostats that can provide remote access and control), and compete with the incumbent firms (e.g., Honeywell). Therefore, reviews of programmable thermostats by experienced consumers will be useful to potential consumers who need information on their product quality and benefits.

This study analyzes different subsets of reviews in all categories suitable for each data pre-processing step. For example, analyzing all the reviews in all the categories over the entire sample period to detect suspicious one-time reviewers or always-the same-rating reviewers is advantageous for companies and researchers to improve the credibility of the one-sided online review system and reduce potential biases in the review data.

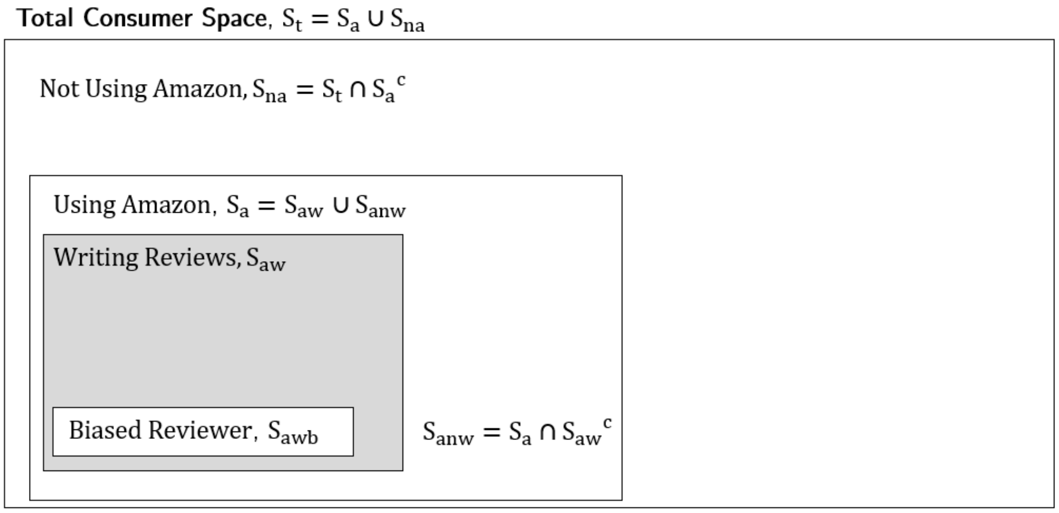
The purpose of this study is to identify unobserved consumers' characteristics and preferences by analyzing DFs; therefore, the sample group disregards reviewers and programmable thermostats containing no prior DFs. In addition, DFs from earlier reviewers (crowd) may have the greatest effect on subsequent reviewers when the reviewer posts his or her first review. This study therefore focuses on the target reviewers' first review of a programmable thermostat. After only selecting the first review of each reviewer for the thermostat group, the total number of reviewers and their first-time reviews is 5,307, and the total number of reviews written by these reviewers in all categories over the entire sample period is 169,809.

After the data preprocessing is discussed, the following three questions are investigated:

1. Can consumers' preferences be identified through the analysis of DFs?

The conceptual consumer space shows the segmentation of consumers (Figure 1). The purpose of this consumer space concept is to derive the group of consumers who become reviewers on Amazon.com. The total consumer group is denoted as $\mathrm{S_t}$. This total group is divided into two groups, those who are users of Amazon, $\mathrm{S_a}$, and those who are not, $\mathrm{S_{na}}$. This study assumes that members of the non-Amazon user group $\mathrm{S_{na}}$ do not write and read reviews on Amazon. The Amazon user group $\mathrm{S_a}$ is split into two subgroups, those who write reviews, $\mathrm{S_{aw}}$, and those who do not, $\mathrm{S_{anw}}$. It should be noted that even though consumers in $\mathrm{S_{aw}}$ write reviews, it is possible that their review data contains bias. Accordingly, this study assumes these biased reviews reduce the credibility of the information found in the reviews. Above all, if a researcher analyzes the review data written by the consumer group $\mathrm{S_{aw}}$ is analyzed and used to estimate and predict the individual consumer preferences of the entire Amazon user group $\mathrm{S_a}$, it will cause sample bias because there is no information about $\mathrm{S_{anw}}$. Therefore, this study aims to estimate and consumer preferences for the group of Amazon users who write a review, i.e. $\mathrm{S_{aw}}$, by using the review data written by this group (S_{aw}) while excluding biased reviews (from the subgroup $\mathrm{S}_{\mathrm{awb}}$). Consequently, this paper implements specific pre-processes to remove the reviews written by $\mathrm{S}_{\mathrm{awb}}$.

Figure 1. Total Consumer Space


Total Consumer Space, $S_t = S_a \cup S_{na}$
Not Using Amazon, $S_{na} = S_t \cap {S_a}^c$
Using Amazon, $S_a = S_{aw} \cup S_{anw}$
Writing Reviews, S_{aw}
Biased Reviewer, S_{awb}
$S_{anw} = S_a \cap {S_{aw}}^c$

Both revealed- and stated-preference methods have been widely applied to estimate consumer preferences. Revealed-preference methods reflect the actual consumer choices in a real-life situation while stated-preference methods reflect respondents' hypothetical choices in a well-designed survey or field experiment (Train 2009.)

The one-sided review system (e.g., Amazon.com) only provides buyers with the opportunity to write a review (Tadelis 2016); therefore, the information displayed about individual buyers is limited. Consequently, the conventional revealed- and stated-preference methods cannot directly identify unobserved consumer characteristics and preferences from online product review data.

This study identifies unobserved consumer characteristics and preferences by extracting (1) users' and prior other reviewers' digital footprints (DFs) from user-generated content (UGC) and (2) consumers' sentiment toward product content dimensions (PCDs) from review text data. This study defines this approach as the user-generated-preference (UGP) method.

In contrast to previous studies using aggregated review summary statistics at the product level (Table 1), this study extracts individual reviewers' DFs for a specific product group from a dataset of 141 million Amazon reviews by making use of Python coding along with high-performance computing (HPC.) The DFs are divided into two groups.

1.User DFs: reviewer i's DFs before writing a review of thermostat p on day $\mathrm{t_i}$.

$$\sum_{\mathrm{t_i^a}}^{\mathrm{t_i^b}} \mathrm{df}_{\mathrm{ipt_i}}(\cdot), \text{where } \mathrm{t_i^a} \ = \underset{\mathrm{t_i^a}}{\mathrm{argmax}} \,|\, \mathrm{t_i} - \mathrm{t_i^a}| \text{ and } \mathrm{t_i} > \mathrm{t_i^a}$$

$$\mathrm{t_i^b} = \underset{\mathrm{t_i^b}}{\mathrm{argmin}} \,|\mathrm{t_i} - \mathrm{t_i^b}| \ \text{ and } \mathrm{t_i} > \mathrm{t_i^b} \geq \mathrm{t_i^a}$$

$\mathrm{df}_{\mathrm{ipt_i}}(\cdot)$ is a DF function for reviewer i who writes a review of p before $\mathrm{t_i}$.

2. Crowd DFs: the crowd's (other prior reviewers') DFs for thermostat p before i writes a review of thermostat p on day $\mathrm{t_i}$.

$$\sum_{\mathrm{j \neq i}}^{\mathrm{J}} \sum_{\mathrm{t_j^a}}^{\mathrm{t_j^b}} \mathrm{df}_{\mathrm{jpt_j}}(\cdot)\,, \text{where } \{\forall \mathrm{J} \ \in \ \mathrm{R} \text{ and } 1 \leq \mathrm{j} \leq \mathrm{J} < \infty |\ \mathrm{i}, \mathrm{t_i}, p\}$$

$$\mathrm{t_j^a} = \underset{\mathrm{t_j^a}}{\mathrm{argmax}} \,|\, \mathrm{t_i} - \mathrm{t_j^a}| \text{ and } \mathrm{t_i} > \mathrm{t_j^a}$$

$$\mathrm{t}_\mathrm{j}^\mathrm{b} = \underset{\mathrm{t}_\mathrm{j}^\mathrm{b}}{\operatorname{argmin}} |\mathrm{t}_\mathrm{i} - \mathrm{t}_\mathrm{j}^\mathrm{b}| \text{ and } \mathrm{t}_\mathrm{i} > \mathrm{t}_\mathrm{j}^\mathrm{b} \geq \mathrm{t}_\mathrm{j}^\mathrm{a}$$

In detail, the user DFs are: (1) five-star rating, brand, product, review summary and body length, product title and description length, and price (at time of web scraping) on day t_i, (2) other reviewers' helpfulness vote for the target reviewers' prior reviews before day t_i, (3) the patterns of star-ratings for all reviewed products in all categories before day t_i, (4) the pattern of the review summary and body length in all categories before day t_i, (5) the volume of prior reviews in all categories before day t_i, (6) the volume of prior reviews in each category before day t_i, (7) the category diversity of all reviews before day t_i, (8) the patterns of reviewed products' price in all categories before day t_i, (9) a time dummy for t_i, a holiday dummy, and a binary dummy indicating whether t_i is before the day that the Nest thermostat was available on Amazon.com, (10) product dummies for each programmable thermostat, and (11) the time interval between day $\mathrm{t}_\mathrm{j}^\mathrm{a}$ and day t_i.

The crowd's DFs for each programmable thermostat are : (1) the rating patterns of the crowd before day t_i, (2) the length of the review summary (headline) and body written by the crowd before day t_i, (3) the star rating, the length of the review summary, and the body written by other reviewers on the most recent day $\mathrm{t}_\mathrm{j}^\mathrm{b}$.

The review text often contains information that is useful for identifying the latent product content dimensions (PCDs; Liu, Lee, and Srinivasan 2019), each reviewer's sentiment, and the direct or indirect reasons for the star rating given, such as “4 stars, minus a star for the lack of an adequate update to fix my WiFi issues.” and “I did not give 5 stars because the installation directions are poor.” Therefore, the author determines five PCDs in the review text by applying unsupervised machine learning (i.e., topic modeling) and extends the five dimensions to nine based on domain knowledge and the purpose of the research design. The nine dimensions are: (1) smart-connectivity, (2) easiness, (3) energy saving, (4) functionality, (5) support, (6) perceived price value, (7) privacy, (8) the Amazon effect, and (9) environmental friendliness. The domain expert annotates each reviewer’s sentiment toward each PCD to transfer domain knowledge from the expert to the empirical models.

After the DF mining and expert review sentiment annotation, the study applies a heteroskedastic ordered probit (HETOP) analysis to identify latent consumer characteristics and preferences regarding programmable thermostats (PTs.) All the HETOP models show the existence of heteroskedasticity in the likelihood ratio (LR) test. All the models that contain DFs and sentiment variables show a much better model-fit than the base model without DFs and sentiment variables.

The results show that a reviewer is less likely to give a five-star rating for the reviewed PT who (1) writes a longer review summary and body for a PT, (2) has lower variance of the review summary length in prior reviews, a larger volume of prior reviews in all categories, and a higher average rating in the prior reviews in all categories, (3) writes a review for the PT that has a higher average length of review summary and/or lower variance of the review summary length in prior reviews, and (4) writes a larger volume of prior reviews in the specific product categories (“Amazon instant video”, “apps for Android”, “cell phones”, “clothes, shoes, and jewelry”, “grocery gourmet food”, “health and personal care”, “magazine subscriptions”, and “software”) and writes a smaller volume of reviews in the “appliance” category.

All the sentiment variables (excluding the environmental friendliness dimension) are statistically significant and positively influence the probability of a five-star rating. To the best of the author’s knowledge, this is the first study about the effect of the online retail market platform’s service quality on ratings and it is found that without considering the online platform’s service quality effect, empirical results can be biased.

Overall, this paper shows (1) how to extract digital footprints (DFs) generated from target consumers writing their first review for a target product group in a specific industry and members of the crowd (other prior reviewers) writing reviews for the target products and (2) how to identify unobserved consumer characteristics and preferences. The approach also shows (1) how to analyze unstructured review text data to extract unobserved product content dimensions for a specific product group and (2) how to identify the target consumers’ sentiment toward the product content dimensions.

Firms could apply this approach (1) to understand consumers’ prior digital trajectory, review behaviors, and preferences regarding target products, and (2) to identify product content dimensions and consumers’ sentiment toward the dimensions from product review text data when marking consumer-oriented business decisions.

2. **Can potential consumers’ preferences be predicted?**

Firms often want to know potential individual consumers’ preferences concerning target product groups in a specific industry (e.g., programmable thermostats) instead of a general product category level (e.g., book). Better short-term predictions of potential consumers' preferences for industry-specific product groups may also help firms to improve their business decisions. To predict potential consumers’ preferences for the target product groups (programmable thermostats), six popular supervised machine learning models are applied, including two base models (kernel support vector machine and decision tree), tree ensemble models (random forest and extreme gradient boosting), and deep learning (artificial neural net and long- short- term memory). Extreme gradient boosting (XGB) shows the best prediction performance in all cases.

This study defines two different counterfactual scenarios as “full ex ante” and “partial ex ante” predictions. The designation “ex ante” indicates a firm’s prediction of potential consumers’ preferences before they make a purchase (full ex ante) or write a review of the purchased product (partial ex ante.) In the full ex ante prediction, firms do not know the potential consumers’ purchased product, star-rating, and review. In the partial ex ante prediction, firms know consumers’ purchased product type; however, they do not have access to consumers’ reviews since they have not written them yet.

Each machine learning model predicts potential consumers' star ratings with six different ex ante variable sets to identify the effect of adding digital footprint variables, the volume of prior reviews in each category, product dummies, and potentially biased price variables. Without DF variables, the prediction performances of machine learning models are low and similar to each other. However, adding DF variables and the volume of prior reviews in each category increases prediction machines' performance. Surprisingly, adding product dummies and potential biased price variables (at the time of web scraping) does not increase the prediction performance of machines much.

The star ratings (label) are skewed to five-star ratings (majority classes); therefore, the Amazon review dataset is an imbalanced dataset. The prediction of an imbalanced dataset is still a big challenge in current machine learning since the predicted labels in an imbalanced dataset are often skewed to the majority classes while the prediction performance for minority classes is low. The potential for an even greater bias may exist in the case of multi-classification.

To identify the effect of label ranges on the prediction performance, this study applies three different ranges of classes: the original five-star rating, three-class, and binary class classifications. Each machine also runs its model in the five-, three, and binary classification cases. The original five-star rating classification shows the lowest prediction performance while the prediction performances of the three-class and the binary classifications are better than that of the five-star rating classification. However, prediction for the minority class (the three-star rating) is almost impossible. This point sheds light on the potential bias in the minority class in the prediction task.

To mitigate the biased prediction problem in imbalanced data, the class weighting method is applied to the loss function (i.e., the cross-entropy loss function) of each machine learning model during the training steps. Extreme gradient boosting sometimes shows better prediction performance with class weights; however, most machine models produce unstable prediction results with the class weighting method.

This approach in this section could help firms to determine how to design prediction tasks to predict potential consumers' star-rating with imbalanced data (with ex ante variables only) and how to improve the prediction performance as well as reducing the potential bias toward the majority classes. Therefore, this approach will be useful for firms' short-term target advertisement, recommendation, pricing, and promotion to potential consumers. However, the low prediction performance of the minority class in the imbalanced review dataset has the potential to cause social inequality.

3. Can individual consumers' sentiments be classified using NLP?

Social science researchers have been studying online reviews to draw meaningful interpretations and insights from structured and unstructured text data. Many social science studies have applied lexicon methods (which create dummies or variables for each unique word over all documents); however, measuring the similarity, ambiguity, and contextual meaning of the language is difficult with lexicon methods. Recently, natural language processing (NLP) has shown success by applying deep learning models to diverse language tasks (e.g., word embedding and text classification.)

As a digital experiment, target reviewers' sentiment toward the "functionality" dimension in the review text data is classified by deep learning models. The classification models are divided into "partial" (using only text data) and "full" (using numerical data and text) models. The machine learning models with both numerical and text data shows better prediction performances than the text-only case.

Word embedding is an NLP method to map unstructured review text to numerical vectors. This method is a way to use text as input data for machine learning models. Three word embedding methods are applied in this study: (1) term frequency inverse document frequency (TF-IDF), (2) Word2vec (W2V), and (3) Bidirectional Encoder Representations from Transformers (BERT.)

Three different W2V embedding models are applied in this study as follows: (1) the W2V model trained on review text written by the target reviewers in all categories, (2) the W2V model trained on all reviews in the "tool and home improvement" category, and (3) Google's pre-trained embedding model trained on the large corpus from Google News. The second W2V model trained on all the reviews in the category shows a better representation of words in the review text data.

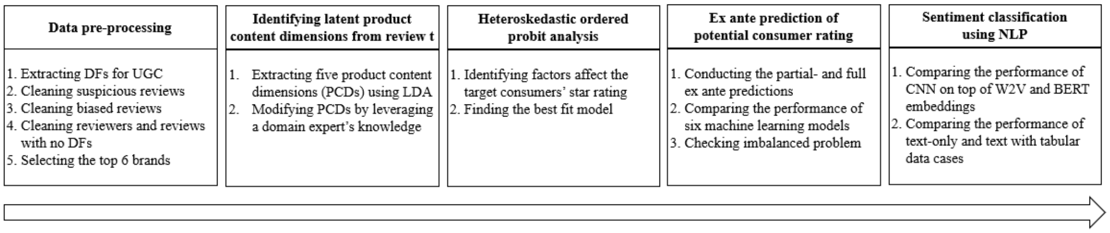
The BERT embeddings are divided into fine-tuned BERT embedding and further pre-trained BERT embedding trained on reviews written by target reviewers in all the categories or all the reviews in the category. The fine-tuned BERT embedding is the best word embedding among the three types. This suggests that firms can improve their sentiment classification performance simply by leveraging pre-trained contextual word embedding models pre-trained on big text data.

The author compared the classification performance of the RF and XGB with TF-IDF embedding and the convolutional neural network (CNN) on top of the BERT and W2V embeddings. The CNN on top of the fine-tuned BERT embedding shows the best performance for the reviewers' sentiment toward a product content dimension.

These findings will be helpful for firms to identify consumer preferences, predict potential consumer preferences, extract product content dimensions for a specific product group from review text, and classify consumers' sentiments.

Figure 2 shows the overall research steps. Section 1 describes the previous literature. Section 2 presents the data-preprocessing for cleaning noisy reviews and extracting target reviewers' sentiment toward the product content dimensions. Section 3 describes the empirical analysis of HETOP. Section 4 demonstrates the ex ante prediction of potential consumers' ratings. Section 5 shows the sentiment classification of specific product content in the reviews. Finally, section 6 offers conclusions and implications.

Figure 2. A summary of the overall research steps



Data pre-processing

1. Extracting DFs for UGC
2. Cleaning suspicious reviews
3. Cleaning biased reviews
4. Cleaning reviewers and reviews with no DFs
5. Selecting the top 6 brands

Identifying latent product content dimensions from review t

1. Extracting five product content dimensions (PCDs) using LDA
2. Modifying PCDs by leveraging a domain expert's knowledge

Heteroskedastic ordered probit analysis

1. Identifying factors affect the target consumers' star rating
2. Finding the best fit model

Ex ante prediction of potential consumer rating

1. Conducting the partial- and full ex ante predictions
2. Comparing the performance of six machine learning models
3. Checking imbalanced problem

Sentiment classification using NLP

1. Comparing the performance of CNN on top of W2V and BERT embeddings
2. Comparing the performance of text-only and text with tabular data cases

2. Prior Literature

When a consumer purchases a product through the online retail market, there is uncertainty about the quality of product because the consumer is not in physical contact with it. There are two different types of goods based on consumers' degree of information about product quality (McCluskey 2000). First, search goods are types of products for which consumers know the quality before they make a purchase with perfect information. Second, experience goods are a type of product for which consumers know the quality only after making a purchase. Consumers' uncertainty about product quality is relatively higher when they purchase an experience good than a search good, because they may not know the product quality before they make a purchase. Online product reviews written by prior experienced consumers may reduce inexperienced consumers' uncertainty and search costs by providing information on the product quality.

However, one possible challenge of using online review data is potential noise, bias, or promotional reviews (Luca 2017.) As shown in Table 1, some previous studies (Luca and Zervas 2016; Mayzlin, Dover, and Chevalier 2014) have investigated the impact of ownership, reputation, and market competition on firms' incentives to write a promotional review by analyzing aggregated product level summary data.

Many previous studies have focused on the impact of reviews on sales (Anderson and Magruder 2012; Chen 2018; Chevalier and Mayzlin 2006; Cui, Lui, and Guo 2012; Hu, Liu, and Zhang 2008; Liu, Lee, and Srinivasan 2019; Luca 2016; Mayzlin, Dover, and Chevalier 2014; Reimers and Waldfogel 2020). Most studies have used summary statistics of aggregated review data on the product level (e.g., the average rating for a product, the volume of reviews for a product, and the average review length for a product).

In particular, Cui, Lui, and Guo (2012) extracted product content dimensions from individual review text by using topic modeling, demonstrated the classification of each product content dimension by using deep learning, and measured the effect of each product content dimension on sales. Further, Timoshenko and Hauser (2019) identified consumer needs from individual review text from Amazon.com data by using deep learning. Furthermore, these two papers showed how to apply a convolutional neural network (CNN) model on top of word2vec embedding vectors to classify individual review text.

Table 1. Previous literature

Source	Data	Target	Method*	Related findings
Anderson and Magruder (2012)	Yelp	Restaurant	RDD	1. If the average rating increases, the frequency of consumer flows will increase 2. The effect of ratings is high when consumers have fewer alternative information sources
Chen (2018)	Yelp, Medicare	Physician	DiD LDA	1. Increasing the average ratings for a physician increases the physician's revenues and patient volume
Chevalier and Mayzlin (2006)	Amazon Barnes and Noble	Book	DiD	1. A higher rating of reviews may increase relative sales. 2. The impact of a one-star rating on relative sales is greater than that of a five-star rating

				3. The statistical significance of the review length variable indicates that consumers read the text in the reviews
Cui, Lui, and Guo (2012)	Amazon	Video game electronics	Panel model	1. The volume of reviews is more important for the sales of new experience goods than those of search goods. 2. The impact of the volume of reviews decreases over time.
Hu, Liu, and Zhang (2008)	Amazon	Book, DVD video	Regression	1. The impact of reviews on sales is larger when (a) the reviewer has a better reputation (b) the items were less reviewed by prior reviewers 2. The impact of reviews decreases as the item ages
Liu, Lee, and Srinivasan (2019)	Online retailer in the UK	Home and garden, technology	CNN RDD LDA	1. The effect of review content on sales is high when the average rating increases, the variance of the rating decreases, and the market is more competitive
Luca (2016)	Yelp, WA department of revenue	Restaurant	RDD	1. If the average rating increases, the revenue will increase
Mayzlin, Dover, and Chevalier (2014)	Expedia TripAdvisor	Hotel	Panel model	1. Hotels may have different levels of incentive to write promotional reviews based on their competition and ownership condition
Reimers and Waldfogel (2020)	New York Times, Amazon	Book	Panel model Nested logit	1 Professional critics' and crowd' star ratings affect sales and consumer surplus
Susan and David (2010)	Amazon	CD, MP3, video game	Tobit model	1. Five- or one-star rating reviews are less helpful for experience goods than mild rating reviews
Luca and Zervas (2016)	Yelp	Restaurant	Regression	1. A restaurant that has a weak reputation is more likely to write negative fake reviews of competitors 2. Fraud involving negative fake reviews may increase when the market becomes more competitive
Zhao et al. (2013)	US companies	Book	Bayesian model	1. Consumers learn product quality from product reviews compared with their own experience with similar products 2. Fake reviews enhance the uncertainty of consumers
Timoshenko and Hauser (2019)	Amazon	Oral care	CNN W2V	1. Deep learning methods increase the performance of identifying consumer needs from user-generated review sentences
Hu, Pavlou, and Zhang (2006)	Amazon	Book, DVD, video	Theory	1. Average ratings from reviewers may mislead consumers regarding the quality of the products because ratings often follow a bimodal distribution

* Note: RDD: Regression discontinuity design; DiD: difference in difference; CNN: Convolutional neural network; W2V: word2vec; LDA: Latent Dirichlet Allocation.

However, there is currently little research about how to:

(1) identify potential suspicious one-time or biased reviewers;

(2) estimate unobserved individual reviewers' characteristics from user DFs;

(3) evaluate the effect of prior other reviewers' DFs on the target reviewers' ratings;

(4) extract latent product content dimensions from review text;

(5) predict potential consumers' ratings before they make a purchase or write a review;

(6) classify reviewers' sentiment toward a product content dimension in the review.

There areas are the focus of this paper. Work is undertaken extract the digital footprints (DFs) of individual target reviewers and other prior reviewers (the crowd) from all the reviews in all categories over the entire sample period and use this information to identify and predict latent consumer preferences and sentiment.

To the best of the author's knowledge, this is the first study about identifying consumer preferences for programmable thermostats that require technology knowledge and skills. The required technical knowledge and skills to install, set up, and use programmable thermostats often raise concerns and become a source of difficulty. Therefore, the ease of usage and consumer support services are essential for inexperienced consumers to mitigate their concerns and difficulties. Even after adequately installing a programmable thermostat, it requires time for the purchasing consumers to know how well the thermostat saves energy and controls the home temperature. In addition, programmable thermostats are not frequently purchased and malfunctioning could cause flaws in other connected devices, additional repair costs, and physical discomfort. The frequency of and exposure to thermostat advertisements are relatively lower than in other popular research subject products (e.g., movies, music, and books); therefore, the sources of information on thermostats' product quality are less diverse than those on books, music, and movies.

Overall, inexperienced consumers' uncertainty and potential loss may be high. Therefore, online product reviews written by other prior reviewers will be more helpful for potential consumers of programmable thermostats than reviews of other usual products.

Liu, Lee, and Srinivasan (2019) showed that online product review data could be more influential for consumers when the product group has more competition, a shorter product history, and weaker brand power. Therefore, relatively new entry firms and disruptive products offer an opportunity to study how online product reviews affect consumer choice and preferences when consumers' uncertainty is high. In particular, an innovative new entry firm, the Nest, entered the market by releasing the first generation of its learning smart-thermostat on October 25, 2011, and it has been available to purchase from Amazon.com since December 15, 2011. The Nest released the second generation on October 2, 2012, and it was available from Amazon.com on the day of release. The Nest's first learning thermostat is an example of disruptive innovation except for the high price level of around $249 (Yang and Newman 2012.) In addition, the Nest's thermostat is an example of the internet of things (IoTs) for smart homes (Mäkinen 2014) due to its advanced features such as learning consumer preference, auto-scheduling of heating, and remote control of heating and cooling devices through a WiFi connection. The Nest rapidly grew into as a competitive market player, and Google acquired it for $3.2 billion on January 14, 2014.

Overall, inexperienced consumers may have high uncertainty not only due to the required technological knowledge and skills but also to changes in the market structure and competition. Therefore, information from prior online product reviews may be useful for inexperienced consumers.

3. Data

The Amazon review data used in this study are secondary (He and McAuley 2016.) The dataset has 142.8 million reviews that generated from May 1996 to July 2014. This data set does not have duplicate reviews for the same products. Data pre-processing consisted of the following:

Step. 1: Selecting reviews with no missing values, which results in a set of 110 PTs

Step. 2: Cleaning “suspicious one-time reviewers” and “always-the-same-rating reviewers”

Step. 3: Deleting reviewers and reviews for products with no DFs

Step. 4: Selecting the top 6 from 26 brands

Step. 5: Identifying five latent product content dimensions in the review text using LDA

Step. 6: Modifying the PCDs by leveraging a domain expert’s knowledge

Detailed descriptions for each step are shown below:

Step 1: Selecting reviews with no missing values, which results in a set of 110 PTs.

The programmable thermostats (PTs) belong to the “tools and home improvement” category. Clarifying a specific product group (programmable thermostats) based only on the category may lead to noisy or missing samples. Therefore, the set of programmable thermostats is carefully defined through the following processes:

1. Selecting the category to which the product belongs from the following list

A. [[“Tools & Home Improvement”, “Building Supplies”, “Heating & Cooling”, “Thermostats & Accessories”, “Thermostats”, “Programmable’]]

2. Removing the products that contain “non-programmable” in the title

3. Selecting the products that contain “programmable” in the product description.

4. Removing the products that contain “non-programmable”, “non programmable”, or “programmable no” in the product description.

5. Removing the products that have a missing value in the brand or price variables.

6. Evaluating the image of each product to verify the robustness of the product set.

The PT set without missing values in either brand or price variables will henceforth be called “programmable thermostats”; there are 110 thermostats in this set. Although the total number of initial reviews of the 110 PTs was 8,817, the total number of reviewers was 8,694, because some reviewers wrote multiple reviews.

This study considers only inexperienced consumers’ first review of the PTs, because inexperienced consumers may become experienced consumers after they have written their first review. Second and third reviews of PTs from the same reviewer are deleted. Therefore, the total number of reviews of PTs used in this research is 8,694, the same as the number of reviewers.

Step. 2: Cleaning “suspicious one-time reviewers” and “always-the-same-rating reviewers”

Step 2.1 Cleaning “suspicious one-time reviewers”

Zhao et al. (2013) indicated that fake reviews increase consumers’ uncertainty about products and that more believable online reviews of experience goods have a larger effect on consumer choice. Some firms may write positive reviews about their products and negative ones about their rivals’ products (Donaker et al. 2019; Luca and Zervas 2016; Mayzlin, Dover, and Chevalier 2014). Accordingly, deleting potential fake reviews is essential to improve the credibility of review and reduce consumer uncertainty.

Mayzlin, Dover, and Chevalier (2014) defined the “suspicious reviewer” as one who writes a review for a hotel for the first time only during the sample period (October 2011) and showed that their rating distribution is more polarized than that of the entire sample. This study takes this into account by accessing individual reviewers’ prior reviews in different categories over the entire sample period, defining a “suspicious one-time reviewer” as one who writes only a review for a PT as a first review and does not write reviews for any other products over the entire sample period.

This cleaning process assumes that suspicious one-time reviewers are less likely to write reviews of other products in different categories, excluding specific target product groups (own products or other competitors in the same product group), to minimize costs. In other words, suspicious one-time reviewers may be unlikely to post reviews outside of their product area. It is possible that they are actual reviewers. However, it is still reasonable to delete potential suspicious one-time reviewers to remove possible bias. In addition, suspicious one-time-reviewers do not have any digital footprints (DFs); therefore, these reviewers are supposed to be deleted in step 3 (deleting reviewers and reviews for products with no DFs.) A total of 1,165 reviews for 80 PTs are detected, written by 1,165 suspicious one-time reviewers.

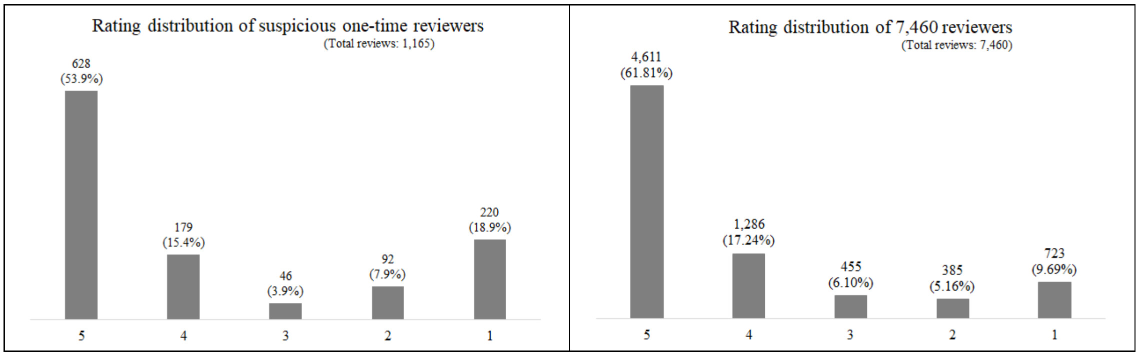
Step 2.2 Cleaning "always-the-same-rating reviewers (ASRs)"

Some reviewers always give a star-rating at the same level for all reviewed products in all categories, regardless of the product quality. Such reviewers may not respond to product quality and previous reviews written by the crowd. Consequently, these reviews do not reflect the product quality. It may also be possible that the reviewers give the same rating level because the number of reviews is simply small. Over the sample period, 1,970 reviewers rated products in all categories at the same level; however, 1,165 reviewers wrote only 1 review and 316 reviewers wrote 2 reviews.

In this study, an "always-the-same-rating reviewers (ASR)" is a reviewer who writes more than 8 reviews with the same rating level. In detail, a 5-star rating shows the highest probability of 0.595 in the "tool and home improvement" category. The probability of 9 consecutive 5-star rating is 0.00934, which is less than 0.01. Only 69 reviewers write more than 8 reviews at the same star rating level (5 stars), surprisingly designating them as "always happy reviewers (AHRs)"; these 69 reviews for 25 PTs are removed.

There is no overlap between 1,165 suspicious 1-time reviewers and 69 ASR reviewers. The number of reviewers become 7,460 after removing 1,234 reviewers. As can be seen in Figure 3, the share of 1-star ratings of suspicious reviewers (18.9%) is about twice as large as that of reviewers after cleaning the suspicious 1-time reviewers and ASRs (9.69%). Therefore, there is potential for negative promotional reviews in the suspicious 1-time reviewers' reviews.

Figure 3. Rating distributions of suspicious 1-time reviewers and reviewers after cleaning.



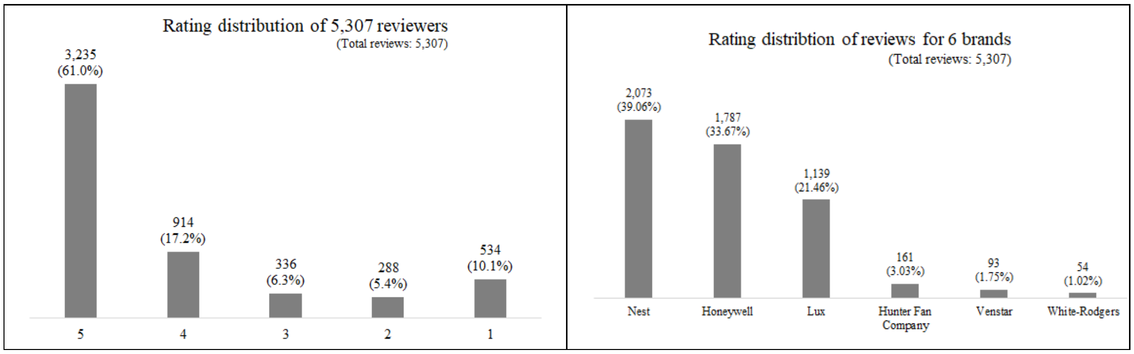
Step. 3: Deleting reviewers and reviews for products with no digital footprints (DFs.)

Without DFs, it is impossible to measure the effect of DFs on a reviewer's rating for a PT when the reviewer writes a review for a PT for the first time. Accordingly, this procedure is followed: (1) 1,965 reviewers do not have any previous reviews of other products excluding PTs in all categories before the first day of writing a review for PTs; (2) 91 reviewers write a review for a PT that does not have any previous reviews written by other prior reviewers. The overlap between the 1,965 reviewers and the 91 reviewers is 28 reviewers; therefore, 1,234 reviewers are removed.

Step. 4: Selecting the top 6 major brands

This procedure restricts the reviewers who write a review for 6 brands that have more than 50 reviews. After this restriction, 5,307 reviewers write a review for the 6 major players, specifically Nest (2,073, 39.06%), Honeywell (1,787, 33.67%), Lux(1,139, 21.46%), the Hunter Fan Company (161, 3.3%), Venstar (93, 1.75%), and White Roger (54, 1.02%). Finally, the number of reviewers and reviews for 71 PTs is 5,307.

Figure 4. Rating distributions of reviews of six major brands



Step 5: Identifying five latent product content dimensions in the reviews using LDA.

What is LDA (Latent Dirichlet allocation)?

LDA (Blei, Ng, and Jordan. 2003) is a Bayesian unsupervised learning model used to identify latent topics in each review and the distribution of these topics in each review. The terminology for LDA in this study is defined as follows:

- $w_{i,n}$ is the nth word in the ith review and it follows a multinomial distribution.

- V (vocabulary) is the total number of unique words in the set of all review data

- K is the total number of topics in each review and is a hyperparameter

- The ith review is a sequence of N words as $\mathrm{r_i} = (\mathrm{w_{i,1}}, \ldots, \mathrm{w_{i,N}})$

- A corpus is a set of M reviews as $\mathrm{R} = (\mathrm{r_1}, \ldots, \mathrm{r_M})$

As a generative probabilistic model, LDA assumes that each review is represented as a distribution over K topics as θ_i. θ_i is a vector in $\mathrm{R^K}$ that represents the proportion of each topic in the ith review. θ_i follows a Dirichlet distribution that has α as a Dirichlet parameter. In addition, φ_k is the kth topic vector in $\mathrm{R^V}$ that represents the proportion of each word that belongs to V in the kth topic. φ_k follows a Dirichlet distribution that has β as a topic hyperparameter. $\mathrm{z_{i,n}}$ is a vector in $\mathrm{R^K}$ that maps the nth word in the ith review to topic k. $\mathrm{z_{i,n}}$ and $\mathrm{w_{i,n}}$ follow a multinomial distribution. Overall, θ_i, φ_k, and $\mathrm{z_{i,n}}$ are latent variables and $\mathrm{w_{i,n}}$ is an observable variable.

In addition, LDA assumes that $\mathrm{w_R}$ (words in reviews) is generated from the joint distribution of θ_R (the review's topic distribution) and φ_K (the topic's word distribution). The joint distribution indicates the word generation process in reviews as follows:

$$\mathrm{p}(\varphi_\mathrm{K}, \theta_\mathrm{R}, \mathrm{z_R}, \mathrm{w_R}|\alpha, \beta) = \prod_{k=1}^{K} \mathrm{p}(\varphi_\mathrm{K}|\beta) \prod_{i=1}^{R} \mathrm{p}(\theta_\mathrm{i}|\alpha) \sum_{n=1}^{N} \mathrm{p}(z_{i,n}|\theta_\mathrm{i})\mathrm{p}(\mathrm{w_{i,n}}|\varphi_\mathrm{k}, \mathrm{z_{i,n}}|\theta_\mathrm{i})$$

Excluding $\mathrm{w_{i,n}}$, the other variables are latent variables. During the training process of LDA, the optimal values of the latent variables maximize the posterior probability.

The posterior probability is denoted as follows:

$$\mathrm{p}(\varphi_{\mathrm{K}}, \theta_{\mathrm{R}}, \mathrm{z}_{\mathrm{R}} |\ \mathrm{w}_{\mathrm{R}}) \ = \frac{\mathrm{p}(\varphi_{\mathrm{K}}, \theta_{\mathrm{R}}, \mathrm{z}_{\mathrm{R}}, \mathrm{w}_{\mathrm{R}})}{\mathrm{p}(\mathrm{w}_{\mathrm{R}})}$$

However, the denominator of the posterior probability is intractable for exact inference because $\varphi_{\mathrm{K}}, \theta_{\mathrm{R}},$ and z_{R} are unobserved variables. In fact, various approximate inference methods are applicable for estimating posterior probability such as variational inference and Gibbs sampling.

LDA Application in This Study

LDA is often called topic modeling. Topics in online product reviews indicate the product content dimensions for the products. The product review text for a specific product group contains finite product content dimensions (topics of product reviews) for the product group. Liu et al. (2019) divided the product content dimension for products from the online product review text into six dimensions as —(1) esthetics, (2) conformance, (3) durability, (4) feature, (5) brand, and (6) price—based on the empirical results of the LDA model and the theory (Garvin 1984.)

Though the theoretical framework is useful in general, this paper uses the LDA model to define the product content dimensions in online product reviews for a specific target product group (programmable thermostats) instead of the general category of goods.

After pre-processing, the number of unique words in 5,307 reviews (the review summary and the body of the review) for LDA is 4,554. The LDA model in this study contains 5 topic dimensions (Table 2.) The number of optimal topics is determined by the coherence score (Syed and Spruit 2017). As can be seen in Table 2, the author, who is a domain expert in the power industry interprets, 5 subjective product content dimensions.

Table 2. Topics in reviews after LDA

Topic dimensions	Interpretation	Top 15 keywords in each topic
1. Connectivity	The review describes WiFi, wireless connection issues with software (e.g., App) and hardware (e.g., HVAC)	wire, WiFi, power, device, connected, connect, wireless, Issue, common, update, app, router, software, hvac, connection,
2. Easiness	The review mentions ease of use, including simplicity of installation, programming, and use.	easy, work, install, program, installation, instruction, installed, simple, programming, nice, programmable, well, took, product, set
3. Saving	The review talks about energy savings, including money savings by reducing energy consumption.	energy, control, save, away, money, saving, heater, month, app, bill, iphone, electric, temperature, feature, best
4. Setting	The review contains content related to setting and control, and information related to temperature, time, scheduling, heating, and other devices.	temperature, time, set, heat, turn, day, back, go, temp, setting, system, need, want, work, change
5. Support	The review focuses on consumer support services before, during, and after they make a purchase.	support, customer, call, product, service, called, tech, told, said, company, hvac, issue, worked, working, customer_service

Step 6: Modifying the PCDs by leveraging the domain expert's knowledge.

The expert extends the five product content dimensions from the LDA model to nine dimensions based on domain knowledge and the purpose of the research design. The dimensions are (1) smart connectivity, (2) easiness, (3) energy saving, (4) functionality, (5) support, (6) price value, (7) privacy, (8) the Amazon effect, and (9) environmental friendliness.

Passonneau et al. (2009) suggested that annotation by experts transfers domain knowledge to machines for better prediction performance. Accordingly, the author manually annotates 47,763 labeling tasks for the reviewers' sentiment toward each product content dimension to transfer domain knowledge to the models as follows:

Dimension 1. Smart connectivity

This dimension indicates the reviewers' sentiment toward programmable thermostats' (PTs') remote control of other home appliances through a Wi-Fi connection using apps and software. Wireless connectivity is a key component of thermostats' smartness as an Internet of Things (IoT) device because it enables consumers to control their home appliances with smartphones, tablets, and computers wherever and whenever they want.

Features related to remote control, Wi-Fi accessibility, and software quality for wireless control belong to this dimension. Firmware for Wi-Fi thermostats can update itself periodically and display customized pictures on the touch screen. For example, reviewers present positive sentiments like the following: "It is nice to monitor & adjust home temperature remotely on iPhone." and "I love the automatic updates that I have been receiving."

Dimension 2. Easiness

This dimension indicates the reviewers' sentiment toward PTs' simplicity and convenience of installation, set up, programming, and usage. Unlike other experience goods, PTs require technical knowledge and skills. A lack of the required knowledge and skills may become a source of difficulty and failure of usage. The easiness of understanding the instruction manual, making the wiring connections, and controlling the device (including programming with a better user interface) belong to this dimension. Some reviewers posted “Easy to Install and Use” and “so easy to use and so easy to see in the dark.”

Dimension 3. Energy saving

This dimension indicates the reviewers’ sentiment toward programmable thermostats’ actual or expected energy saving and/or money saving due to better energy efficiency and cost-effectiveness than other thermostats or their previous one. The reviewers’ comments about features related to better energy saving belong to this dimension along with the reduction of utility bills for electricity or gas. For example, reviews in this dimension include “A much lower price in your electric bill.” and “My gas bill dropped by 30% the first month”

Dimension 4. Functionality

The purpose of thermostats is to control energy usage for heating and cooling. Accurate and precise control for temperature and time are therefore essential for a better programmable thermostat. This dimension presents the quality of controlling and performance of features. The discomfort caused by thermostats’ quality of functionality belongs to this dimension. For example, a clicking noise from thermostats during setting or programming indicates reviewers' negative sentiment toward this dimension. The reviews in this dimension include "Temperature not accurate but does the job." and "Makes a clicking noise."

Dimension 5. Support

This dimension is related to consumer and technical support service, replacement and return service, warranty, packing quality, additional support service on the website, and other helpful materials for consumers. Consumer support services are vital for consumer satisfaction because thermostats require technical knowledge and skill during installation, setting up, and programming.

Consumer support services are vital for consumer satisfaction because thermostats require technical knowledge and skill during installation, setting up, and programming. Consumer support services may also mitigate inexperienced consumers' concerns, technical difficulties, and dissatisfaction during the pre- and post-purchase periods. Some reviews in this dimension are "customer service is amazing! Tweet them for help even!" and "They sent mine in 2 days in perfect condition, plus they appear to have a fair return policy."

However, the expert disregards the reviewers' sentiment toward Amazon's quality of consumer support service. Without separately considering the online market platform's service quality, the reviewers' sentiment toward this dimension for the PTs will be biased.

Dimension 6. Price value

This dimension is a reviewer's subjective evaluation about the price level compared with the quality, future benefits, and other factors. Written comments related to the price value, all positive or negative events affecting the price (such as a discount), and repair costs belong to this dimension.

The prices on Amazon.com change very often and differ for consumers due to different promotions and memberships. The true price of reviewed products in the past may be different from the price at the time of web scraping. In this case, the observed price variables at the time of web scraping could be biased. Therefore, this study extracts the reviewers' sentiment toward this dimension from review text data. Some example reviews for this dimension are "this is money well spent.", "Gold box deal makes it worth", "Too expensive to justify the benefit", and "running a promo to give you a $40 gift card with your purchase."

Dimension 7. Privacy

This dimension is about privacy concerns related to thermostats. Wi-Fi thermostats provide remote control through the Internet, which may cause consumers to have concerns about privacy and data security. Wi-Fi thermostats can store and transform user information and consumption data. Most of the negative privacy concerns occurred for the Nest when Google purchased it on January 13, 2014. Some reviews are “Since Google's Nest buyout raises privacy concerns” and “Unless and until clear, unequivocal, irrevocable legal guarantees are in place that Google doesn't get Nest data, I would say that any Nest user must expect that, ultimately, Google will have all that data.”

Dimension 8. The Amazon effect

This dimension is the reviewers’ sentiment caused by Amazon’s service quality, such as Amazon’s delivery, consumer support, and refund and replacement policy. Reviews on Amazon.com describe not only the product quality but also Amazon’s service quality. If researchers do not account for the effect of Amazon’s service quality on the reviewers’ ratings, it may cause a bias. To the best of the author’s knowledge, this is the first paper to measure the effect of Amazon’s service quality on reviewers’ star ratings.

Some reviews for this dimension are “Amazon's return policy is great!”, “I am very pleased with this purchase and with Amazon customer service.”, “Amazon is really good about their customer service”, and “super fast Amazon delivery for free (overnight).”

Dimension 9. Environmental friendliness

Since programmable thermostats are a home energy control device requiring energy consumption for heating and cooling, some researchers may be interested in the issues related to carbon emissions and climate change.

This dimension is a binary variable indicating whether reviews contain comments about the environmental friendliness of thermostats. Only nine reviews contain comments related to this dimension, including “it helps save the environment!”, “I feel all environmentally friendly for wasting less energy, too.”, and “thanks to this environmentally friendly thermostat. I am also helping to save the world.”

4. Identifying Consumer Preferences by Using HETOP

4.1 The HETOP Model and its Marginal Effect

Amazon.com uses five-star ratings from one (negative) to five (positive). Ratings could censor the strength of reviewers’ latent utility; therefore, reviewers’ observable ratings indicate the range of their unobservable continuous preference (Green 2012) as follows:

$$\mathrm{R_{ipt}} = 1, \text{if } -\infty < \mathrm{U^*_{ipt}} \leq \mathrm{c_1}$$

$$\mathrm{R_{ipt}} = 2, \text{if } \mathrm{c_1} < \mathrm{U^*_{ipt}} \leq \mathrm{c_2},$$

$$\mathrm{R_{ipt}} = 3, \text{if } \mathrm{c_2} < \mathrm{U^*_{ipt}} \leq \mathrm{c_3},$$

$$\mathrm{R_{ipt}} = 4, \text{if } \mathrm{c_3} < \mathrm{U^*_{ipt}} \leq \mathrm{c_4},$$

$$\mathrm{R_{ipt}} = 5, \text{if } \mathrm{c_4} < \mathrm{U^*_{ipt}} < \infty.$$

The ordered dependent variable, $\mathrm{R_{ipt}} \in [1, 5]$, is reviewer i’s first star rating for a PT on day t. $\mathrm{U^*_{ipt}}$ denotes the unobservable continuous utility of reviewer i for product p on day t. The unknown cutting points (thresholds) are denoted as $\mathrm{c_k}$ and assume $\mathrm{c_1} < \mathrm{c_2} < \mathrm{c_3} < \mathrm{c_4}$. $\mathrm{U^*_{ipt}}$ can be represented as follows:

$$\mathrm{U^*_{ipt}} = \mathrm{x'_{ipt}}\beta + \rho\varepsilon_{\mathrm{it}}, \qquad \varepsilon_{\mathrm{it}} \sim \text{i.i.d Normal } (0, 1)$$

where $\mathrm{x_{it}}$ indicates a vector of independent variables and $\varepsilon_{\mathrm{it}}$ is an error term following a standard normal distribution. Hu, Pavlou, and Zhang (2006) showed that the star rating distribution of some experience goods (books, DVDs, and videos) follows bi-modal distribution on Amazon.com. The frequency of observed star ratings in this study follows bi-modal distribution, that is non-normal distribution. However, the cutting points adjust each rating probability (following normal distribution) to match the observed rating distribution (Greene and Hensher 2010b). $\rho > 0$ is a scale function to adjust the variance, and $\varepsilon_{\mathrm{it}}$ is a homoskedastic error (Chen and Khan 2003; Williams 2009.)

The ordered probit (OP) model assumes that $\rho = 1$, so there is no scaling effect on the underlying preferences. Some researchers have studied or applied heteroskedasticity to ordered response models (Chen and Khan 2003; Chen and Kockelman 2012; Greene and Hensher 2010a; Keele and Park 2006; Lemp, Kockelman, and Unnikrishnan 2011; Litchfield, Reilly, and Veneziani 2012; Wang and Kockelman 2005; Williams 2009).

In contrast to linear regression models, the existence of latent heteroskedasticity will cause inconsistency of the maximum likelihood estimators in OP models (Greene and Hensher 2010). The heteroskedasticity ordered probit (HETOP) model assumes its scaling function as $\rho_{\mathrm{i}} = \exp(\mathrm{Z'}_{\mathrm{it}}\gamma)$, where $\mathrm{Z_i}$ denotes the regressors for the scaling function and γ are unknown coefficients for $\mathrm{Z_{it}}$. The probability of a reviewer's star rating for a PT can be derived as follows:

$$P(R_{ipt} = 1|x_{it}) = P(\infty < U^{*}_{ipt} \leq c_1|x_{it}) = \Phi\left(\frac{c_1 - x_{it}\beta}{\rho_i}\right)$$

$$P(R_{ipt} = 2|x_{it}) = P(c_1 < \ U^{*}_{ipt} \leq c_2|x_{it}) = \Phi\left(\frac{c_2 - x_{it}\beta}{\rho_i}\right) - \Phi\left(\frac{c_1 - x_{it}\beta}{\rho_i}\right)$$

$$P(R_{ipt} = 3|x_{it}) = P(c_2 < \ U^{*}_{ipt} \leq c_3|x_{it}) = \Phi\left(\frac{c_3 - x_{it}\beta}{\rho_i}\right) - \Phi\left(\frac{c_2 - x_{it}\beta}{\rho_i}\right)$$

$$P(R_{ipt} = 4|x_{it}) = P(c_4 < \ U^{*}_{ipt} \leq c_3|x_{it}) = \Phi\left(\frac{c_4 - x_{it}\beta}{\rho_i}\right) - \Phi\left(\frac{c_3 - x_{it}\beta}{\rho_i}\right)$$

$$P(R_{ipt} = 5|x_{it}) = P(c_4 < \ U^{*}_{ipt} \leq \infty|x_{it}) = 1 - \Phi\left(\frac{c_4 - x_{it}\beta}{\rho_i}\right)$$

where Φ is the cumulative distribution function (CDF) of the standard normal distribution. The log-likelihood (LL) function for N reviewers and reviews is:

$$\ln LL(\theta) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{5} I(R_{ipt} = j)\ln P(R_{ipt} = j|x_i)$$

where $\theta = \{\beta, \gamma, c_1, c_2, c_3, c_4\}$.

This LL function is maximized with respect to unknown parameters θ. $I(\cdot)$ denotes an indicator function and θ can be estimated through the maximum likelihood estimation.

In contrast to linear regression models, the sign and size of the coefficients for an OP model cannot deliver a direct interpretation due to non-linearity. Generally, marginal effect analysis is a way to interpret each parameter in OP models. In addition, the variables in x_{it} can overlap with those in Z_{it}; therefore, x^{a}_{it} denotes the variables involved in both x_{it} and Z_{it} while x^{b}_{it} denotes the variables that only belong to x_{it}. In the case of continuous variables, Table 3 shows the marginal effect of both x^{a}_{it} and x^{b}_{it} as follows:

Table 3. The marginal effect of the HETOP model

	(1) case of $x_{it}^{a} \in x_{it} \cap Z_{it}{}^{c}$	(1) case of $x_{it}^{b} \in x_{it} \cap Z_{it}$
The marginal effect of x_{it} at R_{ipt}= 1	$\emptyset\left(\frac{c_1 - x'_{it}\beta}{\exp(Z'_{it}\gamma)}\right)\frac{-\beta_{x_{it}^{a}}}{\exp(Z'_{it}\gamma)}$	$\emptyset\left(\frac{c_1 - x'_{it}\beta}{\exp(Z'_{it}\gamma)}\right)\left(\frac{-\beta_{x_{it}^{b}} - (c_1 - x'_{it}\beta)\gamma_{x_{it}^{b}}}{\exp(Z'_{it}\gamma)}\right)$
The marginal effect of x_{it} at R_{ipt}= j where j $\in$ $\{2,3,4\}$	$\left[\emptyset\left(\frac{c_j - x'_{it}\beta}{\exp(Z'_{it}\gamma)}\right) - \emptyset\left(\frac{c_{j-1} - x'_{it}\beta}{\exp(Z'_{it}\gamma)}\right)\right]\frac{-\beta_{x_{it}^{a}}}{\exp(Z'_{it}\gamma)}$	$\left[\emptyset\left(\frac{c_j - x'_{it}\beta}{\exp(Z'_{it}\gamma)}\right)\left(\frac{-\beta_{x_{it}^{b}} - (c_j - x'_{it}\beta)\gamma_{x_{it}^{b}}}{\exp(Z'_{it}\gamma)}\right)\right]$ $-\left[\emptyset\left(\frac{c_{j-1} - x'_{it}\beta}{\exp(Z'_{it}\gamma)}\right)\left(\frac{-\beta_{x_{it}^{b}} - (c_{j-1} - x'_{it}\beta)\gamma_{x_{it}^{b}}}{\exp(Z'_{it}\gamma)}\right)\right]$
The marginal effect of x_{it} at R_{ipt}= 5	$\emptyset\left(\frac{c_4 - x'_{it}\beta}{\exp(Z'_{it}\gamma)}\right)\frac{\beta_{x_{it}^{a}}}{\exp(Z'_{it}\gamma)}$	$\emptyset\left(\frac{c_4 - x'_{it}\beta}{\exp(Z'_{it}\gamma)}\right)\left(\frac{\beta_{x_{it}^{b}} + (c_4 - x'_{it}\beta)\gamma_{x_{it}^{b}}}{\exp(Z'_{it}\gamma)}\right)$

$\emptyset(\cdot)$ indicates the probability density function (PDF) of standard normal distribution. The sign of a coefficient positively reflects the sign of the marginal effect only in the marginal effect of x_{it}^{a} at $R_{ipt} = 5$ and negatively reflects the sign of the marginal effect only in the marginal effect of x_{it}^{a} at $R_{ipt} = 1$. In all other cases, the coefficient sign does not guarantee the direction of the marginal effect for the parameter.

The marginal effect of the binary dummy at each level of $R_{ipt} = j \in [1,5]$ can be derived as follows (Mallick 2008):

$$\Delta P(R_{ipt} = j \mid x) = P(R_{ipt} = j \mid x_{it}, d_{it} = 1) - P(R_{ipt} = j \mid x_{it}, d_{it} = 0)$$

where d_{it} is a binary dummy variable and d_{it}= 0 indicates the base group.

The digital footprints (DFs) and sentiment variables in this study are defined as follows:

Table 4. Variables generated from user and crowd DFs (N= 5,307)

Variable	Description	Mean	SD	Min	Max
rating (dependent)	i (the reviewer)' five-scale star-rating for a PT at t_i*	4.136	1.33	1	5
sum_len	i's length of review summary (headline) at t_i	28.62	17.78	2	134
rev_len	i's length of review body at t_i	796.84	1,007.17	0	11,981
title_len	The length of tittle for the PT reviewed by i at t_i	55.67	10.20	31	107
desc_len	The length of description for the PT reviewed by i at t_i	1,298.43	1,526.47	0	4,788
nest	Brand dummy for the Nest (base group is White Roger)	.39	.49	0	1
honey	Brand dummy for the Honeywell	.34	.47	0	1
hunter	Brand dummy for the Hunter Fan	.03	.17	0	1
lux	Brand dummy for the Lux	.21	.41	0	1
venstar	Brand dummy for the Venstar	.02	.13	0	1
price	p (the PT reviewed by i at t_i)'s price (at the time of web scrapping)	156.53	114.78	14.99	350.3
u_avg_p_dfs	i's average price for reviewed products in all category by t_i^b*	62.31	70.81	0	899.99
u_sd_p_dfs	i's SD of price for reviewed products in all category by t_i^b	55.15	67.63	0	629.32
u_max_p_dfs	i's the highest price among reviewed products in all category by t_i^b	194.50	212.68	0	999.99
u_help_dfs	The number of helpfulness upvote for i in all categories by t_i^b	4.14	15.70	0	911
u_no_help_dfs	The number of helpfulness downvote for i in all categories by t_i^b	1.03	3.49	0	170
u_avg_len_sum	i's average length of summary in all categories by t_i^b	25.35	11.12	1	125
u_sd_len_sum	i's SD of length of summary in all categories by t_i^b	8.76	7.09	0	64.35
u_avg_len_rev	i's average length of review body in all categories by t_i^b	558.21	504.42	71	7,354
u_sd_len_rev	i's SD of length of review body in all categories by t_i^b	305.07	435.72	0	8,139.37
sum_amz_video	i's number of reviews in the amazon instant video category by t_i^b	.00	.08	0	3
sum_appliance	i's number of reviews in the appliance category by t_i^b	.05	.26	0	4
sum_apps	i's number of reviews in the apps for android category by t_i^b	.00	.03	0	1
sum_arts_crafts	i's number of reviews in the art crafts category by t_i^b	.06	.54	0	30
sum_automotive	i's number of reviews in the automotive category by t_i^b	.48	1.69	0	35
sum_baby	i's number of reviews in the baby category by t_i^b	.18	1.14	0	28
sum_beauty	i's number of reviews in the beauty category by t_i^b	.24	1.90	0	93
sum_books	i's number of reviews in the book category by t_i^b	2.46	17.99	0	857
sum_buyakindle	i's number of reviews in the kindle category by t_i^b	.02	.21	0	8
sum_cdsvinyl	i's number of reviews in the cds and vinyl category by t_i^b	.33	2.06	0	92
sum_cellphone	i's number of reviews in the cell phones category by t_i^b	.62	2.04	0	55
sum_clothes	i's number of reviews in the clothes, shoes, jewelry category by t_i^b	.25	1.01	0	44
sum_computers	i's number of reviews in the computer category by t_i^b	.00	.08	0	2
sum_digit_music	i's number of reviews in the digital music category by t_i^b	.03	.31	0	11
sum_electronics	i's number of reviews in the electronics category by t_i^b	3.20	9.88	0	386
sum_giftcards	i's number of reviews in the gift cards category by t_i^b	.00	.08	0	2
sum_grocery	i's number of reviews in the grocery gourmet food category by t_i^b	.38	3.90	0	218
sum_healthcare	i's number of reviews in the health personal care category by t_i^b	.734	3.88	0	196
sum_home_kitch	i's number of reviews in the home kitchen category by t_i^b	1.08	3.72	0	137
sum_industry_spe	i's number of reviews in the industry specific category by t_i^b	.10	.57	0	23
sum_kindle_store	i's number of reviews in the kindle store category by t_i^b	.00	.06	0	3
sum_magazine	i's number of reviews in the magazine subscription category by t_i^b	.01	.19	0	10
sum_movies_tv	i's number of reviews in the move and tv category by t_i^b	.78	7.13	0	199
sum_musical_ins	i's number of reviews in the musical instrument category by t_i^b	.11	1.10	0	58
sum_office_prod	i's number of reviews in the office products category by t_i^b	.53	2.66	0	127
sum_patio_lawn	i's number of reviews in the patio, lawn, and garden category by t_i^b	.36	1.37	0	33
sum_pet_supp	i's number of reviews in the pet supplies category by t_i^b	.26	1.38	0	39
sum_software	i's number of reviews in the software category by t_i^b	.17	1.13	0	42

sum_sports_out	i's number of reviews in the spots and outdoors category by t_i^b	.57	3.99	0	260
sum_tools_home	i's number of reviews in the tools & home category by t_i^b	1.13	3.88	0	109
sum_toys_games	i's number of reviews in the tops and games category by t_i^b	.30	1.84	0	67
sum_video_games	i's number of reviews in the video games category by t_i^b	.32	2.26	0	66
u_cum_reviews	i's number of reviews in all categories by t_i^b	14.81	53.39	1	2,429
u_cate_diversity	Shanon index for i's category diversity of reviews posted by t_i^b	.98	.74	0	2.74
u_avg_rating	i's average star-rating in all categories by t_i^b	3.98	.99	1	5
u_sd_rating	i's SD of star-rating in all categories by t_i^b	.83	.72	0	2.83
c_cum_reviews	p's number of crowd's reviews by t_i^b	524.74	639.35	1	2,425
c_avg_rating	p's average rating of crowd by $t_{j\neq i}^b$*	4.20	.31	1	5
c_sd_rating	p's SD of crowd's rating by $t_{j\neq i}^b$	1.23	.30	0	2.83
c_avg_len_sum	p's average length of review summary written by crowd until $t_{j\neq i}^b$	27.55	2.99	4	55
c_sd_len_sum	p's SD of review summary written by crowd until $t_{j\neq i}^b$	16.24	3.07	0	36.89
c_avg_len_rev	p's average length of review body written by crowd until $t_{j\neq i}^b$	826.66	334.08	103	4,384.67
c_sd_len_rev	p's SD for the length of review body written by crowd until $t_{j\neq i}^b$	951.83	489.97	0	5,676.47
c_rating_rec	p's average rating of crowd at $t_{j\neq i}^b$	4.13	1.34	1	5
c_len_sum_rec	p's the length of review summary written by a crowd at $t_{j\neq i}^b$	27.31	17.19	1	134
c_len_rev_rec	p's the length of review body written by a crowd at $t_{j\neq i}^b$	704.78	920.48	0	11,981
day	Day dummies for t_i and base day is Monday (0)	2.88	1.98	0	6
month	Month dummies for t_i and base month is January (1)	2012.45	1.50	2005	2014
year	Year dummies for t_i and base year is 2005	6 .35	3.87	1	12
holiday	US holiday dummies and base is not holiday (0)	.03	.17	0	1
interval	The time interval between p's the day of first review and t_i	990.38	841.96	1	3,399
nest_avail	Dummy for the first day of the Nest's PT on Amazon (Dec 15, 2011)	.82	.38	0	1
smart_con	i's sentiment of p's smart connectivity in i's review at t_i	.19	.49	-1	1
easy	i's sentiment of p's easiness in i's review at t_i	.41	.67	-1	1
save	i's sentiment of p's energy saving in i's review at t_i	.18	.43	-1	1
func	i's sentiment of p's functionality in i's review at t_i	.16	.80	-1	1
support	i's sentiment of p's support in i's review at t_i	-.00	.39	-1	1
price value	i's sentiment of p's perceived price value in i's review at t_i	.10	.48	-1	1
privacy	i's sentiment of p's privacy issues in i's review at t_i	-.00	.07	-1	1
amazon	i's sentiment of p's Amazon effect in i's review at t_i	.01	.16	-1	1
env	i's sentiment of p's environmental friendliness in i's review at t_i	.00	.04	0	1

Note : t_i = the day when reviewer i wrote a review about a PT (p) for the first time; ** $t_i^b = \underset{t_i^b < t_i}{\operatorname{argmin}} |t_i - t_i^b|$, the most recent day when reviewer i wrote a review before t_i; $t_{j\neq i}^b = \underset{t_j^b < t_i}{\operatorname{argmin}} |t_i - t_j^b|$, the most recent day when the reviewer j wrote a review before t_i; and symbol u in front of the variables (e.g., u_avg_rating) indicates user DFs while c indicates crowd DFs (e.g., c_avg_rating.)

The author derives these variables by analyzing the dataset of 141 million Amazon product reviews that contain an individual-level reviewer ID, product ID, and time stamp for each review. This study assumes that the reviewers' different prior review experiences and patterns reflect their unobserved characteristics and preferences. The variables are divided into "at time" variables extracted from DFs at t_i; "user DF" variables extract reviewer i's prior reviews across all categories by t_i^b or at t_i^b; and "crowd DF" variables extract the reviews written by other prior reviewers on the PT by $t_{j\neq i}^b$ or at $t_{j\neq i}^b$. The number of prior reviews written by i in each subcategory by t_i^b denoted as "sum_+ subcategory name" and 32 subcategories are defined by merging similar subcategories during the pre-processing. The category diversity is the Shannon index, for which higher values mean that reviewer i writes reviews in more diverse subcategories by t_i^b.

$$\text{Diverisity index}_{i,t_i\,=} - \sum\nolimits_{c=1}^{C} P_{c,t_i} \ln P_{c,t_i}, \text{ where } P_{c,t_i} = \frac{N_{c,t_i^b}}{\sum_{c=1}^{C} N_{c,t_i^b}}$$

Here, N_c is the number of prior reviews in subcategory c by t_i^b.

As can be seen in Table 5, each model in this section contains a different combination of variables to identify the effect of DFs, sentiments, prices, and the volume of prior reviews on the consumers' star ratings. In particular, the review text data are divided into "review summary (headline)" and "review body." "Review" in this study denotes both the review summary and the review body text. In addition, other ex post reviewers' helpfulness votes for reviewer i's review after t_i are an ex post variable that does not affect the reviewers' star rating at t_i; therefore, this study disregards helpfulness votes for reviews after t_i.

Table 5. Empirical results from the HETOP and OP models

Variable	model_o1	model_h2	model_o2	model_h3	model_o3	model_h4	model_o4	model_h5	model_o5
sum_len	-0.010***	-0.008***	-0.006***	-0.007***	-0.006***	-0.008***	-0.005***	-0.008***	-0.005***
rev_len	-0.000***	-0.000***	-0.000***	-0.000***	-0.000***	-0.000***	-0.000***	-0.000***	-0.000***
title_len	0.003	0.003	0.003	0.002	0.003	0.002	0.003	0.002	0.003

desc_len	-0.000	-0.000**	-0.000**	-0.000**	-0.000**	-0.000**	-0.000**	-0.000**	-0.000**
nest	0.463***	0.170	0.016	0.033	-0.064	0.191	0.015	0.050	-0.055
honey	0.426***	0.431**	0.252	0.393*	0.234	0.492**	0.248	0.460**	0.231
hunter	0.235	-0.008	-0.124	-0.022	-0.134	-0.034	-0.141	-0.050	-0.152
lux	0.469***	0.555**	0.350*	0.531**	0.341*	0.594**	0.335*	0.578**	0.326*
venstar	0.648***	0.428	0.333	0.324	0.280	0.488	0.325	0.386	0.277
holiday	0.028	0.188	0.170	0.187	0.170	0.240	0.186	0.241	0.187
help_dfs		0.004	0.003	0.004	0.003	0.005	0.003	0.005	0.003
no_help_dfs		-0.011	-0.007	-0.011	-0.007	-0.013	-0.007	-0.013	-0.007
u_avg_len_sum		-0.002	-0.002	-0.002	-0.002	-0.002	-0.002	-0.002	-0.002
u_sd_len_sum		0.009*	0.006	0.010*	0.006	0.010	0.005	0.011*	0.006
u_avg_len_rev		-0.000	-0.000	-0.000	-0.000	0.000	-0.000	0.000	-0.000
u_sd_len_rev		0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
cum_reviews		-0.001**	-0.001**	-0.001**	-0.001**				
cate_diversity		0.002	0.003	-0.016	-0.009	-0.016	-0.009	-0.022	-0.012
u_avg_rating		0.225***	0.169***	0.223***	0.169***	0.254***	0.167***	0.255***	0.167***
u_sd_rating		0.006	-0.004	0.009	-0.001	0.014	0.000	0.015	0.002
c_avg_rating		0.120	0.100	0.102	0.089	0.137	0.092	0.123	0.084
c_sd_rating		-0.140	-0.104	-0.141	-0.106	-0.159	-0.113	-0.160	-0.113
c_cum_reviews		0.000	0.000	-0.000	0.000	0.000	0.000	-0.000	0.000
c_avg_len_sum		-0.022*	-0.015*	-0.020*	-0.014	-0.028**	-0.016*	-0.027*	-0.016*
c_sd_len_sum		0.030**	0.022**	0.028**	0.021**	0.034**	0.022**	0.032**	0.021**
c_avg_len_rev		0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
c_sd_len_rev		0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
c_rating_rec		0.008	0.004	0.008	0.004	0.010	0.004	0.010	0.004
c_len_sum_rec		-0.001	-0.001	-0.001	-0.001	-0.001	-0.001	-0.001	-0.001
c_len_rev_rec		-0.000	-0.000	-0.000	-0.000	-0.000	-0.000	-0.000	-0.000
interval		-0.000	-0.000	-0.000	-0.000	-0.000	-0.000	-0.000	-0.000
nest_avail		0.152	0.106	0.133	0.096	0.134	0.088	0.125	0.084
smart_con		0.699***	0.473***	0.689***	0.473***	0.826***	0.486***	0.824***	0.485***
easy		0.806***	0.580***	0.796***	0.580***	0.937***	0.589***	0.937***	0.589***
save		0.713***	0.494***	0.704***	0.494***	0.858***	0.512***	0.859***	0.513***
func		1.407***	1.030***	1.390***	1.031***	1.621***	1.042***	1.621***	1.042***
support		1.147***	0.793***	1.133***	0.793***	1.326***	0.801***	1.328***	0.801***
price_value		0.675***	0.487***	0.673***	0.491***	0.765***	0.488***	0.769***	0.491***
privacy		1.915***	1.352***	1.889***	1.352***	2.337***	1.431***	2.333***	1.429***
amazon		0.298*	0.203*	0.296*	0.203*	0.331*	0.195*	0.333*	0.196*
env		0.058	0.079	0.050	0.072	0.170	0.146	0.184	0.151
price				0.001	0.000			0.001	0.000
u_avg_p_dfs				0.000	0.000			0.000	0.000
u_sd_p_dfs				-0.000	-0.000			-0.000	-0.000
u_max_p_dfs				0.000	0.000			0.000	0.000
sum_amz_video						-0.667*	-0.388*	-0.664*	-0.386*
sum_appliance						0.227*	0.141*	0.224*	0.140*
sum_apps						-1.995*	-1.225*	-1.998*	-1.226*
sum_cellphone						-0.051*	-0.030*	-0.049*	-0.029*
sum_clothes						-0.091*	-0.061**	-0.091*	-0.061**
sum_grocery						-0.043**	-0.027**	-0.043**	-0.026**
sum_healthcare						0.055**	0.036**	0.056**	0.036**
sum_magazine						-0.367*	-0.211*	-0.367*	-0.212*
sum_pet_supp						-0.053*	-0.028	-0.052*	-0.027
sum_software						-0.052	-0.036*	-0.052	-0.036*
/cut1	-1.430***	-0.386	-0.310	-0.431	-0.339	-0.538	-0.392	-0.566	-0.405
/cut2	-1.157**	0.317	0.191	0.264	0.162	0.274	0.113	0.246	0.100
/cut3	-0.909*	1.063	0.728	1.000	0.700	1.133	0.654	1.105	0.641

/cut4	-0.386	2.579***	1.828***	2.497**	1.800***	2.875**	1.762**	2.848**	1.749**
Z. u_avg_rating		-0.032*		-0.032*		-0.031		-0.030	
Z. nest		0.544***		0.529***		0.681***		0.679***	
Z. honey		0.398**		0.381**		0.531***		0.528***	
Z. lux		0.386**		0.369**		0.491**		0.489**	
Z. hunter		0.582***		0.569***		0.726***		0.725***	
Z. venstar		0.229		0.200		0.364		0.351	
LR tests, $X^2(6)$		28.69***		28.93***		34.37***		34.48***	
Loglikelihood	-6046.585	-4003.836	-4018.181	-4002.670	-4017.135	-3977.845	-3995.028	-3977.223	-3994.463
AIC	12171.169	8159.673	8176.362	8165.341	8182.270	8169.689	8192.055	8176.446	8198.925
BIC	12427.656	8659.494	8636.724	8691.468	8668.938	8873.385	8856.291	8906.448	8889.468

Notes: *P-value* = *p < .1; **p < .05; ***p < .01; statistically insignificant variables represent the volume of prior reviews in each sub category by $\mathrm{t_i^b}$ and time dummies are not presented; "Z.variable" indicates a regressor of the variation function; LR test indicates likelihood ratio tests for the existence of heteroskedasticity in the model; AIC = Akaike information criterion; BIC = Bayesian information criterion. The sample size in this section is 5,306 as the number of samples in 2005 is 1; however, the sample size in the prediction sections is 5,307.

Unobserved omitted variables and the existence of heteroskedasticity may cause inconsistency of parameters in OP models (Greene and Hensher 2010b.) The models in this section contain the variables extracted from DFs and the reviewers' sentiment toward each product's content dimensions (PCDs) to reduce the omitted variable problem.

The misspecification of the variation function in HETOP models leads to biased parameters (Keele and Park 2006.) The author compares the empirical results between the HETOP and the OP models with different sets of regressors to check the variation function's misspecification in the HETOP models.

The AIC and BIC are used to evaluate the models. The AIC and BIC regard the model with fewer parameters and smaller sample sizes as a better-fitted model (Greene and Hensher 2010b). A smaller AIC or BIC value means a better model fit. The notation "model_o" indicates an OP model and "model_h" indicates a HETOP model. Model_o1 is the base model, which contains only observable variables at $\mathrm{t_i}$.

All the HETOP models show better model fits than the OP models with the same set of regressors. In addition, all the HETOP models show the existence of heteroskedasticity in the likelihood ratio (LR) test.

All the other models containing digital footprint (DF) variables and/or sentiment variables show a better model fit than the base model. This result indicates that mining DFs from user-generated online reviews and sentiment analysis for a specific product group could improve model fits. In particular, model_h2 shows the best model fit; therefore, model_h2 is the main model for the interpretation of coefficients and marginal effects.

Surprisingly, the models with price (at the time of web scraping) variables show a lower model fit than the models without price variables. Product prices on Amazon.com frequently change due to promotions, memberships, and other factors. Therefore, the actual price of reviewed products may often differ from the price at the time of web scraping. Further, the actual price at the time of purchasing could be different from the price at the time of writing a review. This price gap between the actual price and the price at the time of web scraping might be a source of inherent bias in the price variables. This study uses the reviewers' sentiment toward the perceived price value dimension as a sentiment variable.

The sign of coefficients for variables in OP models reflect the sign of the marginal effect with the extreme star ratings ($\mathrm{R}_{\mathrm{ipt}} = 5$ and $\mathrm{R}_{\mathrm{ipt}} = 1$). In the HETOP models, the sign of the coefficients for $\mathrm{x}_{\mathrm{it}}^{\mathrm{a}}$ variables reflect the sign of the marginal effects for the $\mathrm{x}_{\mathrm{it}}^{\mathrm{a}}$ variables with the extreme ratings. However, the sign of the coefficients for $\mathrm{x}_{\mathrm{it}}^{\mathrm{b}}$ variables does not directly reflect the sign of the marginal effects with any star ratings. In this study, all the variables in the HETOP models are $\mathrm{x}_{\mathrm{it}}^{\mathrm{a}}$ variables, excluding six $\mathrm{x}_{\mathrm{it}}^{\mathrm{b}}$ variables consisting of the reviewer's average star rating by $\mathrm{t}_{\mathrm{i}}^{\mathrm{b}}$ and five brand dummies.

In the empirical results, the sign of the coefficients for statistically significant variables in the OP and HETOP models are always the same in all cases. The coefficients for statistically significant variables in model_h2 (the main model for interpretation) show the same sign as and a similar magnitude to all the other HETOP models.

Firms often want to know about the characteristics of the most satisfied consumers (five-star reviewers), the most satisfied consumers' responses to the prior reviews written by other prior reviewers, and product attributes' influence on consumers' satisfaction. Based on the user DF variables in model_h2, the probability that a reviewer will give a five-star rating to the reviewed PT will decrease if the reviewer writes a longer length of review summary or body and has a greater volume of prior reviews in all categories.

In contrast, the probability of a reviewer giving a five-star rating will increase if the reviewer has a higher variance of review summary length in prior reviews. In addition, the reviewer's average star rating in prior reviews has a positive influence on the probability of the reviewer giving a five-star rating. Even though the direct economic interpretation is limited, the coefficient of the reviewer's average star rating is the biggest among the statistically significant variables in model_h2.

Model_h4 contains thirty-two variables for the volume of prior reviews in each subcategory instead of the volume of prior reviews in all categories, like model_h2. However, only the statistically significant variables are reported in Table 5.

The probability of a reviewer's five-star rating will increase if the reviewer writes a larger volume of prior reviews for products in the "appliance" and "health care and personal care" categories by t_i^b. For example, reviewers who have a high volume of prior reviews for products in the "appliance" category might have had more technical knowledge and experience with hardware devices. In addition, programmable thermostats are home energy control devices to keep the ideal temperature for consumers' comfort in their homes. Therefore, consumers who have a greater volume of prior reviews for products in the "health care and personal care" category may have better knowledge related to thermostats. In contrast, the probability of a reviewer giving a five-star rating will decrease if the reviewer writes a higher volume of reviews for products in the "Amazon instant video," "apps," "cell phones," "clothes," "groceries," "magazine subscriptions," and "pet supplies" categories. All these subcategories are not directly related to home devices. These data-driven interpretations are subjective, however, they show how to use DFs to understand latent consumer characteristics.

With other prior reviewers' DF variables in model_h2, the probability of a reviewer giving a five-star rating for a PT will increase if the PT has a higher variance of the length of review summaries written by other prior reviewers. In contrast, the probability that a reviewer will give a five-star rating for a PT will decrease if the PT has a longer length of average review summary written by other prior reviewers. Chevalier and Mayzlin (2006) suggested that the statistical significance of the review length variable indicates that consumers read the text in the reviews. Here, this point suggests that a reviewer who gives the extreme ratings (a 1-star or 5-star rating) for a PT may respond to prior reviewers' review summary and not their review body.

Based on the reviewers' sentiment toward product content dimensions (PCDs) extracted from the review text, the probability of a reviewer giving a five star-rating will increase if the reviewer has a positive attitude toward "smart connectivity," "easiness," "energy saving," "functionality," "support," "price value," "privacy," and "Amazon effect" dimensions. The results of the sentiment variables indicate that consumers prefer smarter and easier PTs to others. In addition, these consumers prefer PTs made by a firm that provides better support for consumers, such as technical support. Therefore, firms need to consider not only developing smarter products but also making them easier for consumers to use with better consumer support programs. These consumers also consider a PT's energy saving, functionality, and perceived price value. Interestingly, privacy also affects these consumers' preferences, and this may be caused by wireless smart thermostats. Firms may need to mitigate consumers' concerns about their privacy for energy consumption and life pattern data.

To the best of the author's knowledge, this is the first study to investigate the effect of online retail market service quality on consumers' sentiment. Amazon's better service quality, such as faster delivery, better consumer service, and flexible refund policy, will increase the probability of a reviewer giving a five-star rating. This result supports the idea that the online retail market service quality may influence consumers' preferences as well. Therefore, without considering the effect of the online market service quality on the reviewers, the estimation of consumer preferences may lead to upward or downward bias based on the online retail market's service quality. Meanwhile, the "environmental friendliness" dimension is statistically insignificant.

The above interpretations for the most satisfied consumers (five-star reviewers) are based on statistically significant variables in model_h2 (the main model for interpretation) and model_h4 (the model for interpretation of the volume of prior reviews in each subcategory). The interpretation of the models for unsatisfied consumers (one-star reviewers) is opposite to the above satisfied consumer case.

In contrast to the conventional marginal analysis, this study considers counterfactual scenarios. If a firm has sample data in this study and is making a business decision for the next month, it may want to analyze consumers' preferences in the most recent month. Table 6 shows the marginal effect of key variables (model_h2) at the average value of the Nest's reviewers during June 2014. This result provides the marginal effect of key variables for the Nest's representative consumers in June 2014.

Table 6. Marginal effect of the key variables in model_h2

	Rating 1	Rating 2	Rating 3	Rating 4	Rating 5
sum_len	.0000965*** (.0000324)	.0001758*** (.0000482)	.0003727*** (.0000897)	.0011106*** (.000243)	-.0017556*** (.0003798)
rev_len	.0000019*** (.00000065)	.0000035*** (.00000096)	.0000075*** (.00000178)	.0000223*** (.00000477)	-.0000353*** (.00000749)
desc_len	.00000096** (.00000041)	.0000017* (.00000068)	.0000037*** (.00000137)	.000011*** (.00000403)	-.0000174 (.00000627)
nest	.0069402*** (.0021822)	.0155956*** (.0036217)	.0359091*** (.0104679)	-.0036256 (.093566)	-.0548193 (.0988674)
honey	.0256201 (.0234841)	.0141422 (.0098896)	.0085583 (.0095073)	-.0587262*** (.0202738)	.0104056 (.042016)
lux	.0203594 (.0212434)	.0107719 (.0100643)	.0041348 (.0104108)	-.0669442*** (.0195704)	.0316781 (.0447294)
hunter	.0782568* (.0431248)	.0320287*** (.0073512)	.0247976*** (.0080534)	-.0532644 (.0327548)	-.0818186* (.0428057)
venstar	.0076163 (.0175559)	.0043517 (.0131642)	-.0007767 (.0157755)	-.0537036** (.0261312)	.0425123 (.056631)
u_sd_len_sum	-.0001197 (.0000742)	-.0002181* (.0001274)	-.0004624* (.0002612)	-.0013777* (.0007574)	.0021779* (.0012023)
cum_review	.0000123* (.00000654)	.0000224** (.0000111)	.0000475** (.0000225)	.0001416** (.000065)	-.0002238** (.0001029)
u_avg_rating	-.0044397*** (.0014089)	-.0072649*** (.0017163)	-.0139481*** (.0024774)	-.032701*** (.0051835)	.0583536*** (.007571)
c_avg_len_sum	.0002848* (.00016)	.000519* (.000275)	.0011001* (.0005657)	.0032782** (.0016594)	-.0051821** (.0026128)
c_sd_len_sum	-.0003876** (.0001869)	-.0007063** (.0003109)	-.0014971** (.0006273)	-.004461** (.0018085)	.0070518** (.0028552)
smart_con	-.0089893*** (.002545)	-.0163805*** (.0032834)	-.034722*** (.0050567)	-.1034653*** (.0104058)	.1635571*** (.0162277)
easy	-.0103636*** (.0027898)	-.0188848*** (.0035047)	-.0400303*** (.0051561)	-.1192833*** (.0090109)	.188562*** (.0134368)
save	-.0091721*** (.0025736)	-.0167137*** (.0033788)	-.0354282*** (.0054014)	-.1055698*** (.0121838)	.1668838*** (.0185038)
func	-.0180985*** (.0047887)	-.0329797*** (.005948)	-.0699073*** (.0085658)	-.2083114*** (.0140342)	.3292969*** (.01997)
support	-.01475*** (.0040622)	-.0268779*** (.005184)	-.0569733*** (.0077551)	-.1697703*** (.0138356)	.2683714*** (.0217078)
price_value	-.0086866*** (.0024035)	-.015829*** (.0031331)	-.0335529*** (.0049424)	-.0999817*** (.0105698)	.1580502*** (.0160569)
privacy	-.0246247*** (.0077094)	-.044872*** (.0110772)	-.0951157*** (.0198971)	-.2834277*** (.0531793)	.4480401*** (.0820835)
amazon	-.0038388* (.0022045)	-.0069952* (.0037641)	-.0148277* (.0077086)	-.044184** (.0224601)	.0698457** (.03551)
env	-.00075 (.0089839)	-.0013668 (.0163656)	-.0028971 (.0346832)	-.008633 (.1033051)	.0136469 (.1633324)

Notes: *P-value* = *p < .1; **p < .05; ***p < .01; only consider statistically significant variables or related variables.

The sign of the marginal effect of x^{a}_{it} for the extreme ratings shows an equal sign to the coefficient of those variables in model h2. Accordingly, the average star rating of the reviewers by $\mathrm{t_i^b}$ (only one continuous $\mathrm{x_{it}^b}$ variable) shows the same sign as the coefficient of this variable for the extreme ratings in model_h2. In contrast, the marginal effect of binary dummy variables for each brand (dummy type of $\mathrm{x_{it}^b}$) shows different signs from the coefficient for these dummies over the star ratings.

In terms of the user DF variables, the length of the review summary, the length of the review body, and the reviewer's volume of prior reviews in all categories negatively affect the probability of a reviewer giving a five-star rating. In contrast, the effect of the three variables on the probability of the reviewer giving a one-star rating is positive. The reviewers' average star rating in prior reviews positively affects the probability of the reviewer giving a five-star rating and negatively affects other star ratings.

In terms of other prior reviewers' (crowd) DF variables, the brand dummy variables show different patterns of marginal effects for each star rating. The marginal effect of the Nest brand dummy shows a negative influence on the probability of a reviewer giving a five-star rating; otherwise, it shows a positive influence on the probability of the reviewer's other star ratings. Increasing the crowd's average length of review summary for the PT will decrease the probability of the reviewer giving a five-star rating. In contrast, increasing the crowd's variance of the review summary length for the PT will increase the probability of a five-star rating.

In terms of the reviewers' sentiment toward each product content dimension (PCD), eight sentiment variables are statistically significant, excluding the environmental friendliness dimension. The sentiment variables show a positive relationship with the probability of a five-star rating; however, the sentiment variables have a negative relationship with the other star ratings. If a reviewer has more positive sentiment toward smart connectivity, easiness, energy saving, functionality, support, pricy value, and privacy for programmable thermostats and Amazon's service quality, the probability of a five-star rating will increase while those of the other star ratings will decrease.

All the models (containing digital footprints (DFs) and sentiment variables) show a much better model fit than the base model_o1 (containing only observable variables at t_i). Nonetheless, latent omitted variable bias is still a concern because a one-sided review system cannot provide actual socio-demographic information about the reviewers.

The robustness test in this study follows Altonji et al.'s (2005) and Mayzlin, Dover, and Chevalier's (2014) approaches. The first step is to compare the coefficients of the key variables between the model without control variables (the base model) and the model with control variables (the control model). If the signs of the coefficients for the key variables are the same and the magnitudes of the coefficients for the key variables are similar between the base and the control model, the effect of omitted variables on the coefficients of the key variables may be relatively small. In this case, the omitted variable problem might be neglectable for estimating the coefficients of the key variables.

As shown in Table 7, the sign of the coefficients for the statistically significant key variables is the same in the control and the base model. The magnitudes of the coefficients for the key variables are also similar in the control and the base model. These empirical results indicate that the omitted variable problem might be lessened by adding digital footprints (DFs) and sentiment variables for each product content dimension.

Even though there is still the possibility of selection on unobservable factors, at least, the models using DF and sentiment variables show a much better model fit than model_o1 and the same sign and a similar magnitude of coefficients for key variables across the HETOP models. This point indicates the importance of digital footprint mining and sentiment analysis to estimate consumer preference.

Table 7. Robustness test for the HETOP models

Variable	Base (47 variables)	model_h with control (66 variables)
sum_len	-0.008*** (0.002)	-0.008*** (0.002)
rev_len	-0.000*** (0.000)	-0.000*** (0.000)
desc_len	-0.000** (0.000)	-0.000** (0.000)
nest	0.286 (0.199)	0.170 (0.248)
honey	0.425** (0.206)	0.431** (0.213)
hunter	-0.104 (0.237)	-0.008 (0.265)
lux	0.498** (0.220)	0.555** (0.234)
venstar	0.491* (0.271)	0.428 (0.278)
u_sd_len_sum	0.009** (0.004)	0.009* (0.005)
cum_reviews	-0.001** (0.000)	-0.001** (0.000)
u_avg_rating	0.230*** (0.052)	0.225*** (0.053)
c_avg_len_sum	-0.025** (0.011)	-0.022* (0.012)
u_sd_len_sum	0.032** (0.013)	0.030** (0.013)
smart_con	0.708*** (0.149)	0.699*** (0.147)
easy	0.808*** (0.156)	0.806*** (0.154)
save	0.700*** (0.152)	0.713*** (0.154)
func	1.408*** (0.264)	1.407*** (0.263)
support	1.148*** (0.224)	1.147*** (0.223)
price value	0.665*** (0.138)	0.675*** (0.139)
privacy	1.938*** (0.500)	1.915*** (0.496)
amazon	0.291* (0.162)	0.298* (0.162)
env	0.022 (0.686)	0.058 (0.698)
Z.u_avg_rating	-0.033* (0.019)	-0.032* (0.019)
Z.nest	0.544*** (0.174)	0.544*** (0.174)
Z.honey	0.401** (0.174)	0.398** (0.174)
Z.lux	0.382** (0.174)	0.386** (0.174)

Z.hunter	0.594*** (0.196)	0.582*** (0.196)
Z.venstar	0.175 (0.224)	0.229 (0.224)
Time Fixed Effect	Yes	Yes
LR tests, $X^2(6)$	30.60***	28.69***
Loglikelihood	-4014.864	-4003.836
AIC	8143.728	8159.673
BIC	8518.593	8659.494

Notes: *P-value* = *p < .1; **p < .05; ***p < .01; only statistically significant variables and related variables are reported; standard deviation in parentheses.

4.2 Ex Ante Prediction of Potential Consumer Ratings Using Machine Learning

4.2.1 Research design for the ex ante prediction.

If firms know who will be satisfied and who will be unsatisfied with their products, they would be better able to target potential positive consumers. Therefore, seven different machine learning models (decision tree, support vector machine, random forest, extreme gradient boosting, artificial neural network, and long–short-term memory) are applied here to predict potential consumers' ratings before they make a purchase and write a review.

Classification in machine learning is a prediction task for a discrete dependent variable (i.e., label). For example, predicting a five-star rating from online product reviews involves multiclass classification, which is often a more difficult task than binary classification. Bouazizi and Ohtsuki (2019) showed that the accuracy of sentiment classification of a balanced dataset from Twitter decreased from 81.3% in a binary classification to 60.2% in seven different sentiment classifications. Even though some scholars (Haque, Saber, and Shah 2018) have simplified multiclass classification into binary classification, there are few studies about the effect of class range in a multiclass classification on prediction performance. This section provides each classifier's prediction performance in five-star ratings and three-class (positive, neutral, or negative) and binary class (positive or negative) classifications.

As shown in Table 8, the rating distribution in this study is skewed to the five-star rating (majority class); therefore, it is an imbalanced dataset. Classification of imbalanced data is a challenge in machine learning because classification results tend to be biased toward the majority class. Class weighting is a popular approach to mitigate the imbalanced class problem (Chen, Liaw, and Breiman 2004). In detail, class weighting puts more weight on minority classes (two- and three-star ratings) than majority classes in a machine learning model's loss function; therefore, the loss function becomes more sensitive to minority classes and less sensitive to majority classes. The class weighting is applied to each machine learning model in this section as a hyperparameter.

The ex ante classification of potential reviewers' star rating is divided into ex ante and partial ex ante classification. First, ex ante classification is the prediction of potential consumers' ratings before they make a purchase. In this case, firms do not know reviewers' ratings, reviews, and reviewed or purchased thermostats; therefore, these ex post variables are excluded from the ex ante model.

Second, the partial ex ante classification is a prediction of potential consumers' star ratings before they write a review for purchased thermostats. In this case, firms know the type of thermostats that consumers have purchased; however, they do not know the consumers' ratings and reviews for the purchased thermostats because the consumers have not posted a review yet. Therefore, reviewers' ratings and reviews are excluded from the partial ex ante model, but the programmable thermostat dummy variables are included in the partial ex ante model. These product dummies are not compatible with a heteroskedastic ordered probit (HETOP) model due to the perfect prediction and multicollinearity issues, and are therefore the HETOP models are excluded from the partial ex ante model.

If the machine learning model is too fitted to the training data, the fitted model's prediction performance for new data points in the validation set will decrease. This modeling error is usually called overfitting in machine learning (Dietterich 1995.) Each machine learning model has hyperparameters. The optimal hyperparameter values for each prediction machine are selected when the optimal values mitigate the overfitting problems during the hyperparameter tuning process.

In the training step, the original dataset is split into a total training set and a test set, and the total training set is also divided into a training set and a validation set for hyperparameter tuning. Each machine learning model is trained on the training set and predicts new data points in the validation set. The optimal hyperparameter values are selected when the validation loss stops decreasing while the train loss keeps decreasing.

In the test set prediction step, each prediction model is also trained on the total training data with the optimal hyperparameters selected during the training step. The model trained on the total training data predicts the label in the test set. In particular, review rating classification in the test set can be interpreted as predicting the strength of potential consumers' preferences regarding programmable thermostats (PTs).

The total sample period is from October 12, 2005 to July 17, 2014, and the total sample size is 5,307 reviews (reviewers). Programmable thermostats are home energy devices that control heating and cooling devices in the home. Therefore, the demand for programmable thermostats may differ in different weather conditions and seasons. This study defines the validation and test set with similar sample sizes (301 and 303 reviews) and time intervals (about a month in subsequent months). This study assumes that the weather and seasonality are similar in the validation and test datasets.

Table 8. Rating distribution over datasets

	Total Set		Total Training Set		Training Set		Valid Set		Test Set	
Rating	Count	Shares	Count	Shares	Count	Share	Count	Share	Count	Share
5	3,235	60.96%	3,039	60.73%	2,841	60.41%	198	65.78%	196	64.69%
4	914	17.22%	872	17.43%	829	17.63%	43	14.29%	42	13.86%
3	336	6.33%	322	6.43%	308	6.55%	14	4.65%	14	4.62%
2	288	5.43%	268	5.36%	258	5.49%	10	3.32%	20	6.60%
1	534	10.06%	503	10.05%	467	9.93%	36	11.96%	31	10.23%
Total	5,307	100.00%	5,004	100.00%	4,703	100.00%	301	100.00%	303	100.00%
Period	Oct 12, 2005 – July 17, 2014		Oct 12, 2005 – May 17, 2014		Oct 12, 2005 –Mar 16, 2014		Mar 17, 2014 –May 17, 2014		May 18, 2014 – July 17, 2014	

4.2.2 Machine learning models for ex ante prediction.

Six popular machine learning models are applied to ex ante prediction tasks. The support vector machine and decision tree models are base models used to compare their prediction performance with more complex models. Random forest and extreme gradient boosting are tree ensemble models. The artificial neural net and long–short-term memory models are deep learning models. A high-level overview of each model is presented below.

A. Kernel support vector machine (Kernel SVM)

The support vector machine (SVM) model finds the linear separable hyperplane in the feature space to classify labels (Schiilkop, Burgest, and Vapnik 1995.) To deal with non-linearly separable, noisy, and outlier data, Cortes and Vapnik (1995) introduced a slack variable as $\xi_i \geq 0, \forall i$ and a parameter C. ξ_i is the distance between the linear hyper-plane and the misclassified x_i, while C is a weight for the sum of ξ_i in the sample as $\sum_{i=1}^{N} \xi_i$ (Papadimitriou, Gogas, and Stathakis 2014.)

In particular, kernel SVM is applied in this study to consider the non-linearity of data. A kernel function K implicitly maps original data to a high-dimensional functional feature space Φ: $x \rightarrow \varphi(x)$, such that $K(x, x') =< \varphi(x), \varphi(x') >$ for two samples x and x'. The Gaussian radial basis function (RBF) is the kernel function, as follows:

$$K_{rbf}(x, x') = \exp(-\gamma ||x - x'||_2^2))$$

where $\gamma > 0$ and $||x - x'||^2$ is the squared Euclidean distance between x and x'. The RBF is a similarity measure ranging between zero and one, and $\varphi(x)$ has an infinite number of dimensions (Vert, Tsuda, and Schölkopf 2004).

Overall, the dual problem of kernel SVM can be expressed as follows:

$$\max_{\alpha_i} \sum_{i=1}^{N} \xi_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K_{rbf}(x_i, x_j),$$

$$\text{where } C \geq \alpha_i \geq 0 \text{ and } \sum_{i=1}^{N} \alpha_i y_i = 0$$

α_i denotes the Lagrange multipliers, and $\{x_i \mid C > \alpha_i > 0, \forall i\}$ are the support vectors deciding the decision boundary. C is an upper bound of ξ_i in this kernel SVM optimization setting. In addition, C and γ are two hyperparameters of SVM.

One-vs-rest (OvR) is a popular method for multiclass classification (Pal 2008). In the OvR approach to multiclass classification, five binary SVMs classify each star rating in an online product review against the rest of the ratings as {five-star, the others}, {four-star, the others}, {three-star, the others}, {two-star, the others}, and {one-star, the others}. The SVM shows the largest margins among the five SVMs and determines the star rating of new review data in the test set.

B. Decision tree (DT)

The decision tree (DT) model recursively partitions the feature space into a disjointed set of rectangle regions such that each region contains the same classes (Figure 5).

For multiclass classification, the DT model has K classes (K > 2). The feature space at each node n is divided into two sub-regions based on $\theta_n \in \{x_j , t_j \mid \text{node} = n \}$, where x_j denotes splitting variable j and t_j denotes the splitting value for x_j at node n. θ_n splits the data at node n into $\{D_{\text{left}}(\theta_n) \mid x_j \leq t_j \text{ at node } = n\}$ and $\{D_{\text{right}}(\theta_n) \mid x_j > t_j \text{ at node } = n\}$. R_n represents the region corresponding to node n in the feature space, and $N_n = \sum_{i=1}^{N} I(x_i \in R_n)$ means the total number of instances in R_n. Node m denotes the terminal node (i.e., leaf). The hyperparameter of DT is the maximum number of the tree depth in this study.

In DT, impurity means the heterogeneity of classes in a node and H (·) denotes the impurity function. The optimal value of $\theta_n{}^*$ minimizes the impurity at the given node n as follows:

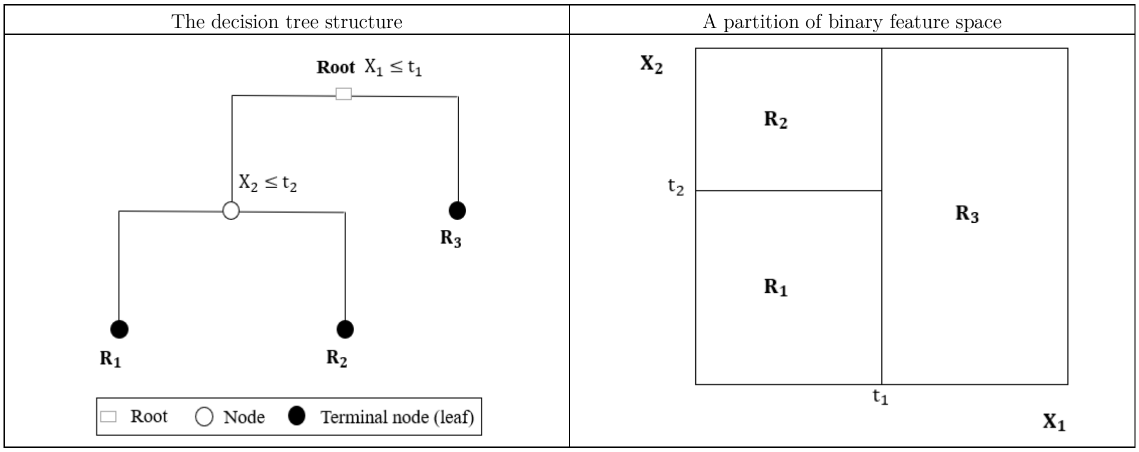
$$\theta_n{}^* = \underset{\theta_n}{\mathrm{agmin}} \frac{[N_{left|n} H\ (\ \{D_{left}(\theta_n)\ )+N_{right|n}\ H\ (\{D_{right}(\theta_n)\ )]}{N_n}\ , \text{where } N_n = N_{left|n} + N_{right|n}$$

Entropy is the impurity measure in this study and can be expressed as follows:

$$H\ (\ D(\theta_n)\ ) = -\sum_{k=1}^{K} p_{kn}(1-p_{kn})\ , \text{where } p_{kn} = \frac{1}{N_n} \sum_{\substack{k=1 \\ x_i \in R_n}}^{K} I(\,y_i = k)$$

The decision tree is simple, interpretable, applicable for regression and classification with continuous and/or categorical variables, and acceptable for a dataset containing missing values. However, the decision tree has high variance due to its hierarchical structure so that a small change of features can cause different split results. Further, the classification of the DT on imbalanced data could be biased toward the majority class. Therefore, the tree ensemble models are applied to mitigate these problems.

Figure 5. Decision tree structure


The decision tree structure
A partition of binary feature space
Root $X_1 \le t_1$
$X_2 \le t_2$
R_3
R_1
R_2
□ Root ○ Node ● Terminal node (leaf)
X_2
R_2
t_2
R_3
R_1
t_1
X_1

C. Random forest (RF)

Ensemble methods use a set of base classifiers. The random forest (RF) is a tree ensemble model called bootstrap aggregating. Dietterich (2000) suggested that ensemble models often perform better than single classifiers because (1) averaging each classifier may reduce the probability of using the wrong classifier; (2) different starting points of each classifier's optimization may reduce the possible local optima; and (3) combining each classifier may represent the correct function for mapping features to labels.

In particular, the RF is able not only to improve the prediction performance by reducing variation but also to maintain robust prediction performance with an increasing number of noisy variables (Friedman, Hastie, and Tibshirani 2001.)

The RF's procedure is: (1) generating an independent training set $\mathrm{s_i}$ by selecting a subset of the sample from training set S with replacement; (2) creating de-correlated RF $\mathrm{rf_i}$, by selecting a subset of features; (3) training $\mathrm{rf_i}$ with $\mathrm{s_i}$ and using fitted $\mathrm{rf_i}$ to classify new data x; and (4) repeating the above steps B times and classifying new data by using majority voting as follows:

$$\hat{\mathrm{y}} = \frac{1}{\mathrm{B}}\sum_{i=1}^{\mathrm{B}} \mathrm{rf_i}(\mathrm{x}; \theta_{i})$$

θ_{i} indicates the parameters determining the structure of $\mathbf{rf_i}$, including the subset of features, splitting variables and points at each node, and the values at each terminal node. The hyperparameters are the number of trees and the depth of the trees.

Breiman (2001) argued that the RF's prediction performance depends on individual DTs' performance and the correlation between DTs. Chen, Liaw, and Breiman (2004) suggested the weighted RF for imbalanced data. The minority classes could be less represented in the sub-samples due to resampling, and this may cause lower prediction performance for the minority classes.

D. **Extreme gradient boosting (XGB)**

Boosting combines multiple weak classifiers to build a strong classifier. However, boosting does not involve bootstrap resampling (James et al. 2013). Extreme gradient boosting (XGB; Che, and Guestrin 2016) implements gradient boosting (Friedman 2001) by regularizing the complexity of the tree structure. The prediction of a tree ensemble model is the sum of K DTs:

$$\widehat{y_i} = \sum_{k=1}^{K} f_k(x_i),\ f_k \in F$$

$$\text{where } F = \{f(x) = w_{q(x)} \mid q(x) \in \{1,..,T\} \text{ and } w \in R^T\}$$

F is a possible functional space of DTs. q is a leaf index function and represents the structure of the tree. T is the number of leaves in the tree. w is the weight of each leaf.

Each DT has an objective function (OF). A smaller value of the OF means a better tree structure. The optimization of each tree structure minimizes the OF:

$$\text{OF} = \text{training loss} + \text{regularization term} = \textstyle\sum_i^N L\ (y_i, \widehat{y_i})\ + \sum_{k=1}^K [\ \gamma T + \frac{1}{2}\lambda||w||^2\ ]$$

OP contains additive tree functions; therefore, it cannot be optimized by the conventional methods. Therefore, additive training is applied for the optimization by adding a new function $f_t(x_i)$ in each iteration t and using second-order Taylor approximation:

$$\mathrm{OF}^{(t)} \approx \sum_{i}^{N} L(y_i, \widehat{y_i}^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \ + \sum_{k=1}^{K} [\gamma T + \frac{1}{2}\lambda ||w||^2]$$

$$\text{where } g_i = \frac{\partial L(y_i, \widehat{y_i}^{(t-1)})}{\partial \widehat{y_i}^{(t-1)}} \text{ and } h_i = \frac{\partial^2 L(y_i, \widehat{y_i}^{(t-1)})}{\partial \ \widehat{y_i}^{(t-1)}}$$

For the multiclass classification, the softmax loss (cross entropy loss) is applied:

$$L(y_i, \widehat{y_i}) \ = \ -\alpha_k \sum_{k=1}^{K} I(y_i \ = k) \ \log \Pr(\widehat{y_i} \ = y_i \mid x)$$

For imbalanced data, α_k becomes $\frac{N}{K \times N_k}$ to put more weight on the minority class and less on the majority class in the loss function (Chen at el. 2017.) If XGB is not weighted, α_k becomes 1. The hyperparameters of XGB in this study are the number of trees, tree depth, learning rate, and class weight.

F. Artificial neural network (ANN)

An ANN is a deep learning (DL) model. DL automatically learns a representation of data for required tasks (LeCun, Bengio, and Hinton 2015.) Recently, deep learning has shown dramatic progress in diverse areas including natural language processing (NLP). Deep learning also has the potential to improve business analytics (Urban et al. 2020.)

Deep learning relies on the universal approximation theorem (Cybenko 1989; Hornik 1991). In this theorem, ANN, $\hat{F}(x, w)$ can approximate any Borel measurable function f(x) (any continuous function on a compact subset of finite Euclidean space is Borel measurable) with any desired degree of accuracy (LeCun, Bengio, and Hinton 2015; Strang 2019) as follows:

$$\mathrm{If}\ \forall\ \mathrm{f(x)\ is\ continous\ in\ R^n, there\ is\ weight\ vector\ w\ in\ |\hat{F}(x,w)\ -f(x)\ \ | < \varepsilon, \forall x}$$

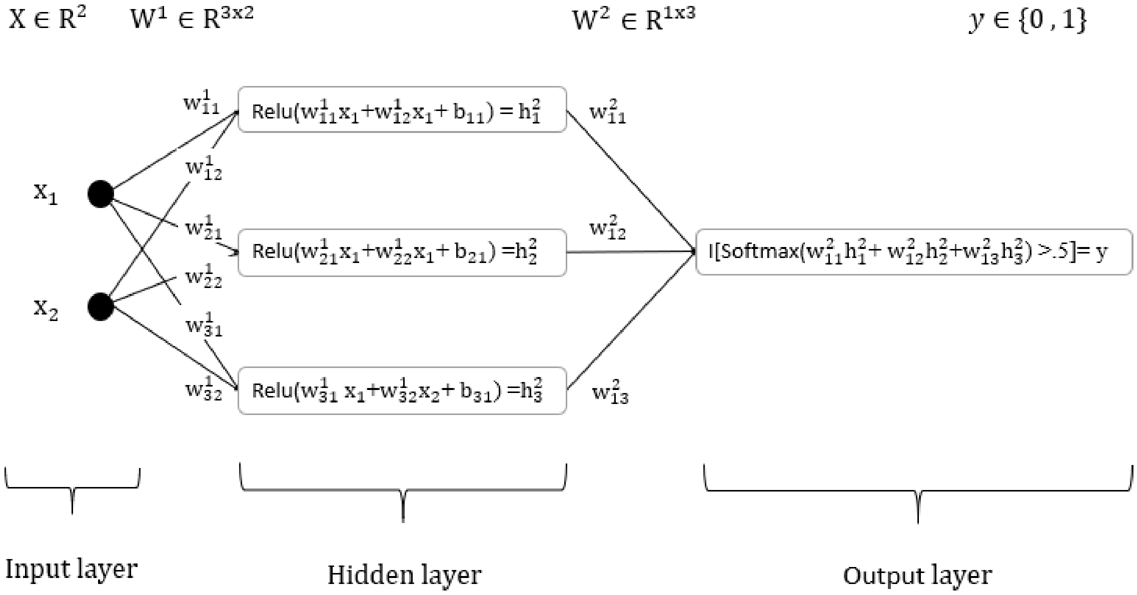
The ANN will also be useful for approximating E(Y|X) by mitigating functional form misspecification (Bergtold and Ramsey 2020; Kuan and White 1994).

The ANN has a multilayer structure with input, hidden, and output layers. Figure 6 shows the basic structure of the ANN for binary classification. The ANN example has the input layer with two input variables, one hidden layer with three neurons, and one output layer. Each neuron in the hidden layer receives a weighted input value from the input layer and the received input values enter the activation function (continuous non-linear function) in each neuron. In this example, the activation function is the rectified linear unit (ReLU) as f(x) = max (0, x). The weighted sum of output values from the hidden layer enters the output layers. The softmax function, $\mathrm{f(x_i) = \frac{exp\ (x_i)}{\sum_j exp\ (x_i)}}$, turns the output values from the previous hidden layer into the probability of class one. If P(class = 1) > .5, the label will be one; otherwise, it will be zero. The ANN learns optimal weights by backpropagation (Chauvin and Rumelhart 1995.)

In this study, the ANN structure contains two hidden layers. The activation functions are ReLU. The optimization method for minimizing cross-entropy loss is Adam (Kingma and Ba 2014). Dropout is regularization to prevent overfitting during the training steps.

The hyperparameters are the optimal training iteration, dropout rate, learning rate, and number of neurons in two hidden layers. The class weight is also a hyperparameter; however, the class-weighted ANN shows lower prediction performance than the un-weighted one.

Figure 6. Example of the ANN structure



$X \in R^2$
$W^1 \in R^{3x2}$
$W^2 \in R^{1x3}$
$y \in \{0, 1\}$
x_1
x_2
w^1_{11}
w^1_{12}
w^1_{21}
w^1_{22}
w^1_{31}
w^1_{32}
$\mathrm{Relu}(w^1_{11}x_1+w^1_{12}x_1+b_{11}) = h^2_1$
$\mathrm{Relu}(w^1_{21}x_1+w^1_{22}x_1+b_{21}) = h^2_2$
$\mathrm{Relu}(w^1_{31}x_1+w^1_{32}x_2+b_{31}) = h^2_3$
w^2_{11}
w^2_{12}
w^2_{13}
$I[\mathrm{Softmax}(w^2_{11}h^2_1+w^2_{12}h^2_2+w^2_{13}h^2_3) > .5] = y$
Input layer
Hidden layer
Output layer

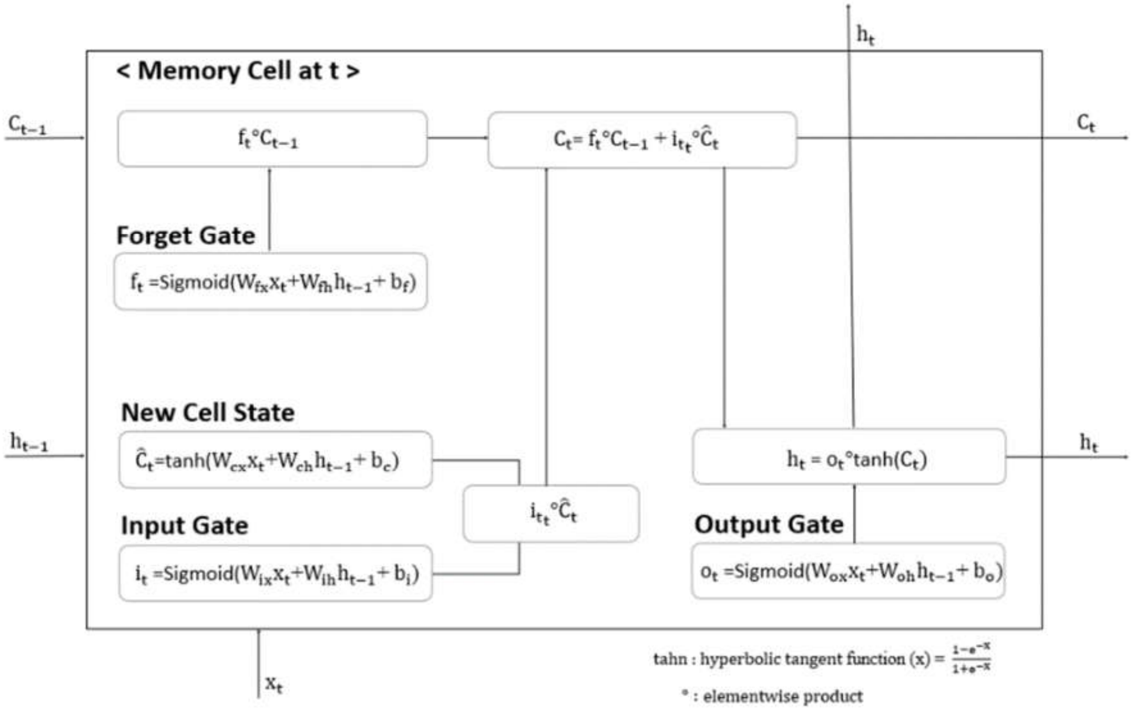
G. LSTM

The recurrent neural net (RNN) is a DL model for sequence data; however, the RNN may suffer from the vanishing gradient problem during the training of long sequence data (Hochreiter 1998). LSTM mitigates the vanishing gradient problem by introducing the memory cell structure (Hochreiter and Schmidhuber 1997; Rao at el. 2018).

LSTM has a multilayer structure with input, hidden, and output layers. In particular, the hidden layer(s) contains memory cells. Each memory cell is controlled by three gates (the input $\mathrm{i_t}$, forget gate $\mathrm{f_t}$, and output gate $\mathrm{o_t}$). The memory cell at time t receives the input value $\mathrm{x_t}$, hidden state $\mathrm{h_{t-1}}$ and previous cell state at t-1 $\mathrm{C_{t-1}}$. The input gate $\mathrm{i_t}$ decides whether the information in $\mathrm{x_t}$ and $\mathrm{h_{t-1}}$ is useful for $\mathrm{C_t}$. The forget gate $\mathrm{f_t}$ decides whether the information in $\mathrm{h_{t-1}}$ is useful for $\mathrm{C_t}$. The output $\mathrm{o_t}$ decides which information in $\mathrm{C_t}$ will be preserved in $\mathrm{h_t}$.

Figure 7 shows the structure of the memory cell. The hyperparameters of the LSTM model in this study are the learning rate, training epochs, and number of neurons.

Figure 7. The structure of the memory cell (Fischer and Krauss 2018; Olah 2015)


< Memory Cell at t >
C_{t-1}
$f_t \circ C_{t-1}$
$C_t = f_t \circ C_{t-1} + i_{t_t} \circ \hat{C}_t$
C_t
h_t
Forget Gate
f_t =Sigmoid($W_{fx} x_t + W_{fh} h_{t-1} + b_f$)
New Cell State
h_{t-1}
$\hat{C}_t$=tanh($W_{cx} x_t + W_{ch} h_{t-1} + b_c$)
$i_{t_t} \circ \hat{C}_t$
$h_t = o_t \circ \tanh(C_t)$
h_t
Input Gate
i_t =Sigmoid($W_{ix} x_t + W_{ih} h_{t-1} + b_i$)
Output Gate
o_t =Sigmoid($W_{ox} x_t + W_{oh} h_{t-1} + b_o$)
x_t
tahn : hyperbolic tangent function (x) = $\frac{1-e^{-x}}{1+e^{-x}}$
° : elementwise product

4.2.3 Ex ante prediction performance in five-star rating classification.

The prediction performance criteria for classification are: 1) “accuracy,” 2) “precision,” 3) “recall,” and 4) “F1 score,” as described below:

1. Accuracy: the ratio of the total number of correctly classified reviews over the total number of reviews
2. Precision: the fraction of reviews correctly classified for a given star rating over the total number of reviews classified as the star rating.
3. Recall: the fraction of reviews correctly classified for a given star rating over the true number of reviews belong to the star rating
4. F-measure: the weighted average of precision and recall in the following format:

$$\text{F1 score} = \frac{2 \times (\text{precision} \times \text{recall})}{\text{precision} + \text{recall}}$$

Accuracy could mislead the prediction performance of classifiers for an imbalanced dataset. For example, the share of five-star ratings in the test set is 0.647 (196 five-star ratings over 303 reviews in the test set). In this case, if a classifier is biased toward the five-star rating and subsequently predicts all the samples in the test set to be five-star ratings, the macro accuracy of the classifier will be 0.647. Even though the classifier cannot predict other star ratings at all, the macro accuracy will still be 0.647. If a researcher wants to evaluate the prediction machines by using accuracy, the ML model's test set accuracy should be more than 0.647.

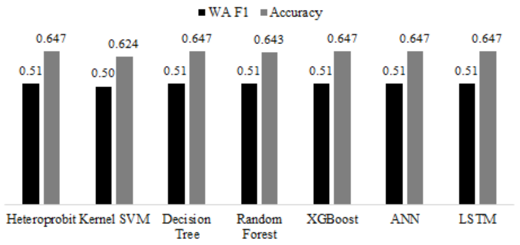
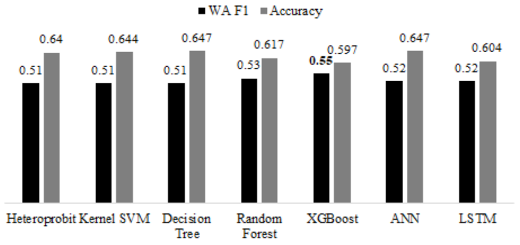
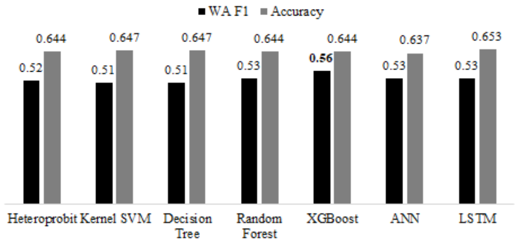
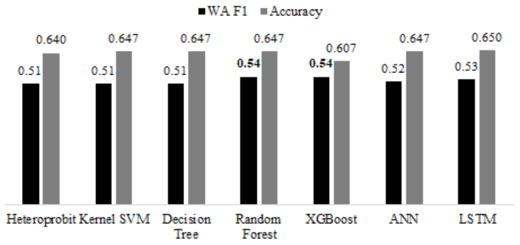
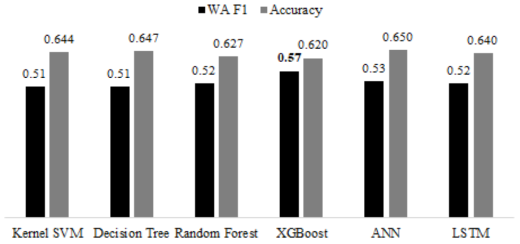
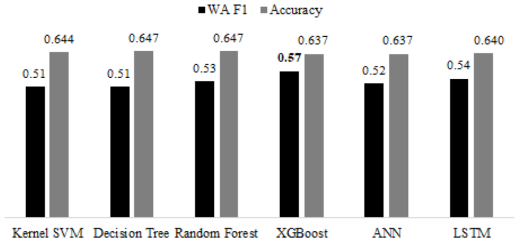
According to the studies conducted by Ibrahim, Torki, and El-Makky (2018) and Jeni, Cohn, and De La Torre (2013), the F1 score may be a better evaluation criterion for this imbalanced dataset. The weighted average macro F1 score (WA F1) is the evaluation criterion for each model's prediction performance in this study as follows:

$$\text{Weighted average macro F1 score (WA F1)} = \sum_{k=1}^{K} \frac{N_k}{N} \times \text{ k class's F1} - \text{score}$$

The predictive performance of six popular prediction machines with six different feature sets can be seen in Table 9. Model 1 ("at time model") is the base model that only contains 37 observable variables at t_i. This model is a base model (feature set) for the prediction performance of the six machine learning models with different models (i.e., different feature sets.) Without digital footprintd and sentiment variables, as in the case of model 1, the machine learning models' prediction performance in the WA F1 score is not better than that of the econometric model (HETOP). In this case, there is no reason to apply machine learning models to predict potential consumers' review ratings instead of the conventional econometric model. In addition, the prediction performance of machine learning and econometric models with this feature set is very low.

As can be seen in Table 16 (Appendix C), most of the classifiers' prediction results in model 1 are biased toward the majority class (5-star rating); therefore, the accuracy of each model is often 0.647, because these classifiers always predict 5-star ratings for the test dataset and the share of five-star reviews is 0.647. In this majority-biased classification case, multiclass classification is simply a binary classification indicating whether a review's star rating is a 5-star rating.

Table 9. Ex ante prediction results

<table>
<tr><th>Models</th><th>variables</th><th>Weighted Average (WA) Macro F1-score and accuracy</th></tr>
<tr><td>Model 1:
at time model</td><td>37 variables including:
1. variables when the reviewers write a review
2. time fixed effects</td><td>
</td></tr>
<tr><td>Model 2:
ex ante model</td><td>59 variables including:
1. 37 variables from at time model
2. 22 DFs variables including the reviewer's volume of prior reviews in all category</td><td>
</td></tr>
<tr><td>Model 3:
ex ante-sub-model</td><td>90 variables including:
1. variables from 'At time' model
2. 32 variables for the reviewer's volume of prior reviews in each sub-category</td><td>
</td></tr>
<tr><td>Model 4:
ex ante-sub-price model</td><td>94 variables including:
1. 90 variables from ex ante-sub-model
2. price and price DFs (4 variables)</td><td>
</td></tr>
<tr><td>Model 5:
partial
ex ante-sub-model</td><td>161 variables including:
1. 90 variables from ex ante-sub-model
2. 71 product dummies</td><td>
</td></tr>
<tr><td>Model 6:
partial
ex ante-sub-price model</td><td>165 variables including:
1. partial ex ante-sub- model (161 variables)
2. price and price DFs (4 variables)</td><td>
</td></tr>
</table>

Models 2, 3, and 4 (Table 9) are ex ante models used to predict consumers' potential ratings for programmable thermostats (PTs) before they make a purchase. Model 3 (the "ex ante sub-model") shows the highest prediction performance of the best classifier in each model among the three models. XGB in model 3 is the best prediction machine among the six classifiers in the three different models (models 2, 3, and 4) with a WA F1 score of 0.56 (Table 9). LSTM in model 3 shows the highest accuracy among the six classifiers in all the models (including the three models) with a score of 0.657.

The base classifiers (HETOP, SVM, and DT) do not show better prediction performance than model 1 (the base model). The base classifiers always predict test data as the majority class (five-star rating) in models 1, 2, 3, and 4. Therefore, there is no improvement in prediction performance for the base classifiers. The base classifiers may not be suitable for multiclass classification for the imbalanced dataset in this study.

Surprisingly, adding more price variables to model 3 does not improve the classifiers' prediction performance in model 4, excluding the random forest case. This point indicates that adding a potentially biased variable (price at the time of web scraping) to prediction machines may not improve the prediction performance.

Table 10 provides the detailed model structure, the optimal hyperparameters for each model, and the confusion matrix for each classifier's prediction. Notably, all the classifiers in models 2, 3, and 4 show a zero WA F1 score for the minority classes (three- and two-star ratings. This point shows the biased prediction problem in the imbalanced data.

Table 10. Model 3: Ex ante-sub-model (90 variables)

<table>
<tr><th>Models</th><th>Hyperparameter</th><th>Accuracy</th><th>Precision</th><th>Recall</th><th>F1-score</th><th>Confusion matrix</th></tr>
<tr><td>Heteroprobit</td><td>None</td><td>0.644</td><td>1: 0.33
2: 0.00
3: 0.00
4: 0.00
5: 0.65
WA: 0.45</td><td>1: 0.06
2: 0.00
3: 0.00
4: 0.00
5: 0.98
WA: 0.64</td><td>1: 0.11
2: 0.00
3: 0.00
4: 0.00
5:0.78
WA: 0.52</td><td><table>
<tr><th></th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th></tr>
<tr><td>1</td><td>2</td><td>0</td><td>0</td><td>0</td><td>29</td></tr>
<tr><td>2</td><td>1</td><td>0</td><td>0</td><td>0</td><td>19</td></tr>
<tr><td>3</td><td>0</td><td>0</td><td>0</td><td>0</td><td>14</td></tr>
<tr><td>4</td><td>0</td><td>0</td><td>0</td><td>0</td><td>42</td></tr>
<tr><td>5</td><td>3</td><td>0</td><td>0</td><td>0</td><td>193</td></tr>
</table></td></tr>
<tr><td>Kernel SVM</td><td>Kernel: RGB
Gamma: 8
C:0.1</td><td>0.647</td><td>1: 0.00
2: 0.00
3: 0.00
4: 0.00
5: 0.65
WA: 0.42</td><td>1: 0.00
2: 0.00
3: 0.00
4: 0.00
5: 1.00
WA: 0.65</td><td>1: 0.00
2: 0.00
3: 0.00
4: 0.00
5: 0.79
WA: 0.51</td><td><table>
<tr><th></th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>31</td></tr>
<tr><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>20</td></tr>
<tr><td>3</td><td>0</td><td>0</td><td>0</td><td>0</td><td>14</td></tr>
<tr><td>4</td><td>0</td><td>0</td><td>0</td><td>0</td><td>42</td></tr>
<tr><td>5</td><td>0</td><td>0</td><td>0</td><td>0</td><td>196</td></tr>
</table></td></tr>
<tr><td>Decision Tree</td><td>Criteria: Entropy
Max depth: 1</td><td>0.647</td><td>1: 0.00
2: 0.00
3: 0.00
4: 0.00
5: 0.65
WA: 0.42</td><td>1: 0.00
2: 0.00
3: 0.00
4: 0.00
5: 1.00
WA: 0.65</td><td>1: 0.00
2: 0.00
3: 0.00
4: 0.00
5: 0.79
WA: 0.51</td><td><table>
<tr><th></th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>31</td></tr>
<tr><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>20</td></tr>
<tr><td>3</td><td>0</td><td>0</td><td>0</td><td>0</td><td>14</td></tr>
<tr><td>4</td><td>0</td><td>0</td><td>0</td><td>0</td><td>42</td></tr>
<tr><td>5</td><td>0</td><td>0</td><td>0</td><td>0</td><td>196</td></tr>
</table></td></tr>
<tr><td>Random Forest</td><td>Tree numbers: 46
Depth: 43</td><td>0.644</td><td>1: 0.67
2: 0.00
3: 0.00
4: 0.14
5: 0.66
WA: 0.51</td><td>1: 0.06
2: 0.00
3: 0.00
4: 0.02
5: 0.98
WA: 0.64</td><td>1: 0.12
2: 0.00
3: 0.00
4: 0.04
5: 0.79
WA: 0.53</td><td><table>
<tr><th></th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th></tr>
<tr><td>1</td><td>2</td><td>0</td><td>0</td><td>2</td><td>27</td></tr>
<tr><td>2</td><td>1</td><td>0</td><td>0</td><td>0</td><td>19</td></tr>
<tr><td>3</td><td>0</td><td>0</td><td>0</td><td>1</td><td>13</td></tr>
<tr><td>4</td><td>0</td><td>0</td><td>1</td><td>1</td><td>40</td></tr>
<tr><td>5</td><td>0</td><td>0</td><td>1</td><td>3</td><td>192</td></tr>
</table></td></tr>
<tr><td>Xgboost</td><td>Tree number: 50
Depth : 3
Learning rate: 0.2</td><td>0.644</td><td>1: 0.40
2: 0.00
3: 0.00
4: 0.50
5: 0.66
WA: 0.54</td><td>1: 0.19
2: 0.00
3: 0.00
4: 0.12
5: 0.94
WA: 0.64</td><td>1: 0.26
2: 0.00
3: 0.00
4: 0.19
5: 0.78
<b>WA: 0.56</b></td><td><table>
<tr><th></th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th></tr>
<tr><td>1</td><td>6</td><td>1</td><td>0</td><td>0</td><td>24</td></tr>
<tr><td>2</td><td>1</td><td>0</td><td>0</td><td>1</td><td>18</td></tr>
<tr><td>3</td><td>0</td><td>0</td><td>0</td><td>0</td><td>14</td></tr>
<tr><td>4</td><td>0</td><td>0</td><td>0</td><td>5</td><td>37</td></tr>
<tr><td>5</td><td>8</td><td>0</td><td>0</td><td>4</td><td>184</td></tr>
</table></td></tr>
<tr><td>ANN</td><td>Epoch: 9
Drop out: 0.5
Learning rate: 0.0001
Hidden layer 1 node: 270
Hidden layer 2 node: 270</td><td>0.637</td><td>1: 0.50
2: 0.00
3: 0.00
4: 0.25
5: 0.65
WA: 0.51</td><td>1: 0.03
2: 0.00
3: 0.00
4: 0.07
5: 0.96
WA: 0.64</td><td>1: 0.06
2: 0.00
3: 0.00
4: 0.11
5: 0.78
WA: 0.53</td><td><table>
<tr><th></th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>2</td><td>28</td></tr>
<tr><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>20</td></tr>
<tr><td>3</td><td>0</td><td>0</td><td>0</td><td>0</td><td>14</td></tr>
<tr><td>4</td><td>1</td><td>0</td><td>0</td><td>3</td><td>38</td></tr>
<tr><td>5</td><td>0</td><td>0</td><td>0</td><td>7</td><td>189</td></tr>
</table></td></tr>
<tr><td>LSTM</td><td>Epoch: 70
Learning rate: 0.0002
Hidden layer node: 70</td><td><b>0.653</b></td><td>1: 0.20
2: 0.00
3: 0.00
4: 0.67
5: 0.66
WA: 0.54</td><td>1: 0.03
2: 0.00
3: 0.00
4: 0.05
5: 0.99
WA: 0.65</td><td>1: 0.06
2: 0.00
3: 0.00
4: 0.09
5: 0.79
WA: 0.53</td><td><table>
<tr><th></th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>30</td></tr>
<tr><td>2</td><td>1</td><td>0</td><td>0</td><td>0</td><td>19</td></tr>
<tr><td>3</td><td>0</td><td>0</td><td>0</td><td>1</td><td>13</td></tr>
<tr><td>4</td><td>2</td><td>0</td><td>0</td><td>2</td><td>38</td></tr>
<tr><td>5</td><td>1</td><td>0</td><td>0</td><td>0</td><td>195</td></tr>
</table></td></tr>
</table>

* WA indicates weighted average macro values.

* The horizontal labels from 1 (left) to 5 (right) are the predictive star ratings, while the vertical labels from 1 (top) to 5 (bottom) are the true star ratings. The values on the diagonal are the number of correct predictions for the star ratings mapped to the horizontal or vertical star ratings.

Models 5 and 6 (Table 9) are “partial ex ante” models to predict consumers’ potential product review ratings for the programmable thermostats (PTs) purchased before they write a review. These models contain the product dummies for 71 programmable thermostats; therefore, firms know the type of programmable thermostats purchased by the consumers.

The difference between model 5 and model 6 is the existence of price variables. Model 6 is simply model 5 with price variables; therefore, model 5 does not contain price variables. Adding price variables to model 5 does not show a certain pattern of prediction performance improvement. The prediction performance of the best classifier (XGB) in each model is the same or higher in model 5 (without price variables). At least, this result indicates that adding potentially biased variables (price variables) does not guarantee better prediction performance for classifiers. In particular, the best classifier (XGB) in model 5’s prediction performance for the WA F1 score of 0.57 is the highest among all the models (including models 5 and 6). In particular, the class-weighted XGB has a better prediction performance than XGB without the class weight in both model 5 and model 6 (Table 11). However, the other classifiers show a lower performance with the class weighting.

The prediction performance for the minority classes (three and two stars) has a zero F1 score in both model 5 and 6. This point indicates that adding product dummy variables to models cannot solve the majority class bias in these imbalanced data.

Table 11. Model 6: Partial ex ante sub-model (161 variables)

Models	Hyperparameter	Accuracy	Precision	Recall	F1-score	Contingent Metrix					
Kernel SVM	Kernel: RGB Gamma: 0.001 C: 1.0	0.644	1: 0.00	1: 0.00	1: 0.00		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	0	0	0	0	31
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.33	4: 0.02	4: 0.04	3	0	0	0	0	14
			5: 0.65	5: 0.99	5: 0.78	4	0	0	0	0	42
			WA: 0.46	WA: 0.64	WA: 0.51	5	0	0	0	0	196
Decision Tree	Criteria: Entropy Max depth: 1	0.647	1: 0.00	1: 0.00	1: 0.00		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	0	0	0	0	31
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.00	4: 0.00	4: 0.00	3	0	0	0	0	14
			5: 0.65	5: 1.00	5: 0.79	4	0	0	0	0	42
			WA: 0.42	WA: 0.65	WA: 0.51	5	0	0	0	0	196
Random Forest	Tree numbers: 9 Depth: 13	0.627	1: 0.30	1: 0.10	1: 0.15		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	3	0	0	1	27
			3: 0.00	3: 0.00	3: 0.00	2	2	0	0	0	18
			4: 0.00	4: 0.00	4: 0.00	3	0	0	0	0	14
			5: 0.66	5: 0.95	5: 0.78	4	1	1	1	0	39
			WA: 0.46	WA: 0.63	WA: 0.52	5	4	0	1	4	187
Xgboost	Tree number: 60 Depth: 14 Learning rate: 0.2 Class weighted*	0.620	1: 0.28	1: 0.23	1: 0.25		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	7	0	0	6	18
			3: 0.00	3: 0.00	3: 0.00	2	4	0	0	2	14
			4: 0.34	4: 0.24	4: 0.28	3	0	0	0	0	14
			5: 0.70	5: 0.87	5: 0.78	4	2	2	0	10	28
			WA: 0.53	WA: 0.62	**WA: 0.57**	5	12	1	1	11	171
ANN	Epoch: 146 Drop out: 0.4 Learning rate: 0.0002 Hidden layer 1 node: 483 Hidden layer 2 node: 483	**0.650**	1: 0.50	1: 0.06	1: 0.11		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	2	0	0	0	29
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.50	4: 0.02	4: 0.05	3	0	0	0	0	14
			5: 0.65	5: 0.99	5: 0.79	4	1	0	0	1	40
			WA: 0.54	WA: 0.65	WA: 0.53	5	1	0	0	1	194
LSTM	Epoch: 127 Learning rate: 0.0002 Hidden layer node: 242	0.640	1: 0.25	1: 0.03	1: 0.06		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	1	0	0	0	30
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.50	4: 0.02	4: 0.00	3	0	0	0	0	14
			5: 0.65	5: 0.98	5: 0.05	4	0	0	0	1	41
			WA: 0.51	WA: 0.64	WA: 0.52	5	3	0	0	1	192

* Heteroskedastic ordered probit (HETOP) is excluded because the model is incompatible with product dummies due to multicollinearity.

* Class weight $= \frac{N}{K \times N_k}$, where N is the number of sample; K is number of classes; and, N_k *is* the number of sample belong to class k.

Class weight values [2.014, 3.646, 3.054, 1.135, 0.331] for each class (1, 2, 3, 4, and 5 star)

4.2.3 Ex ante prediction performance in the three-class classification.

The three-class classification in this section contains “positive (five- and four-star ratings),” “neutral (three-star rating),” and “negative (two- and one-star ratings).” Table 12 shows that the distribution of the three classes is skewed toward the positive class.

However, the class distribution is more balanced than the five-star rating classification. If a machine learning model classifies all the instances in the test set as a positive class, the accuracy will be 0.7855. Therefore, the minimum reasonable accuracy of a classifier is 0.7855.

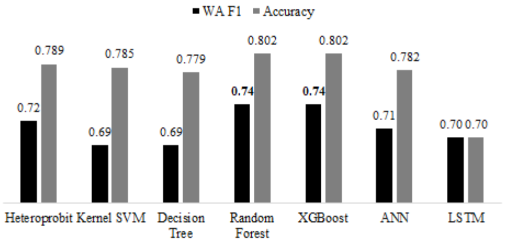
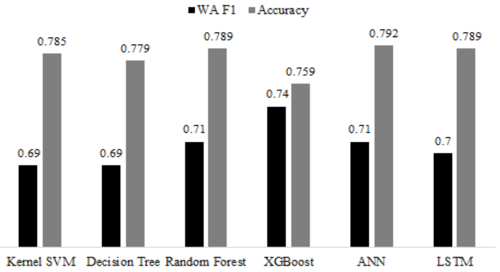
Table 12. Class distribution in the three-class classification

	Total Set		Total Training Set		Sub Training Set		Valid Set		Test Set	
1	4,149	78.18%	3,911	78.16%	3670	78.04%	241	80.07%	238	78.55%
0	336	6.33%	322	6.43%	308	6.55%	14	4.65%	14	4.62%
-1	822	15.49%	771	15.41%	725	15.42%	46	15.28%	51	16.83%
Total	5,307	100.00%	5,004	100.00%	4703	100.00%	301	100.00%	303	100.00%

As shown in Table 13, the RF and XGB show the highest WA F1 score (of 0.74) and accuracy (of 0.802) in the ex ante sub-model (90 variables without product dummies). The prediction performance of the best classifier (XGB) in the three-class classification (0.74 in the WA F1 score) is higher than the five-star rating classification (0.56 in the WA F1 score) with the feature set of the ex ante sub-model. This point indicates that the reduction of the class range may improve the prediction performance of machine learning models with imbalanced data.

Surprisingly, adding 71 product dummies to the feature set in the ex ante sub-model does not improve the WA F1 score of most of the classifiers, excluding the RF (Table 13.) This result indicates that information about purchased programmable thermostat may not much useful to improve machine learning models' prediction performance.

Table 13. Ex ante prediction results in the three-class case

Models	Variables	Weighted Average (WA) Macro F1 score and accuracy
Model 3: ex ante-sub-model	90 variables including: 1. variables from 'At time' model 2. 32 variables for the reviewer's volume of prior reviews in each sub-category	
Model 5: partial ex ante-sub-model	161 variables including: 1. 90 variables from ex ante-sub-model 2. 71 product dummies	

Even though adding product dummies to the feature set in the ex ante sub-model does not improve the WA F1 score of the deep learning models (ANN and LSTM), the accuracy of these deep learning models is higher with product dummies (Table 13). As can be seen in Table 14, another interesting finding is that all the classifiers show a zero F1 score for the minority class (three-star rating). This means that the reduction of the range of classes from the five-class (five-star ratings) to the three-class classification does not improve the prediction performance for the minority class in this imbalanced dataset.

Overall, adding product dummies does not improve the best classifier's prediction performance in the three-class classification. XGB is the best classifier in the WA F1 score. Furthermore, XGB shows stable prediction performance with and without product dummies variables.

Table 14. Three-class classification: Ex ante-sub model (90 variables)

Models	Hyperparameter	Accuracy	Precision	Recall	F1-score	Confusion matrix
Heteropobit		0.789	1: 0.56 2: 0.00 3: 0.80 WA: 0.72	1: 0.10 2: 0.00 3: 0.98 WA: 0.79	1: 0.17 2: 0.00 3: 0.88 WA: 0.72	1 2 3 1 5 0 46 2 0 0 14 3 4 0 234
Kernel SVM	Kernel: RGB Gamma: 1.0 C: 0.1	0.785	1: 0.00 2: 0.00 3: 0.79 WA: 0.62	1: 0.00 2: 0.00 3: 1.00 WA: 0.79	1: 0.00 2: 0.00 3: 0.88 WA: 0.69	1 2 3 1 0 0 51 2 0 0 14 3 0 0 238
Decision Tree	Criteria: entropy Max depth: 4	0.779	1: 0.25 2: 0.00 3: 0.79 WA: 0.66	1: 0.02 2: 0.00 3: 0.99 WA: 0.78	1: 0.04 2: 0.00 3: 0.88 WA: 0.69	1 2 3 1 1 0 50 2 0 0 14 3 3 0 235
Random Forest	Tree numbers: 16 Depth: 42	0.802	1: 0.73 2: 0.00 3: 0.80 WA: 0.75	1: 0.16 2: 0.00 3: 0.99 WA: 0.80	1: 0.26 2: 0.00 3: 0.89 **WA: 0.74**	1 2 3 1 6 2 43 2 0 0 14 3 5 0 233
Xgboost	Tree number: 100 Depth: 4 Learning rate:0.2	0.802	1: 0.78 2: 0.00 3: 0.80 WA: 0.76	1: 0.14 2: 0.00 3: 0.99 WA: 0.80	1: 0.23 2: 0.00 3: 0.89 **WA: 0.74**	1 2 3 1 7 0 44 2 0 0 14 3 2 0 236
ANN	Epoch: 3 Drop out: 0.4 Learning rate: 0.0002 Hidden layer 1 node:180 Hidden layer 2 node:180	0.782	1: 0.38 2: 0.00 3: 0.79 WA: 0.69	1: 0.06 2: 0.00 3: 0.98 WA: 0.78	1: 0.10 2: 0.00 3: 0.88 WA: 0.71	1 2 3 1 3 0 48 2 1 0 13 3 4 0 234
LSTM	Epoch: 232 Learning rate: 0.0002 Hidden layer node: 322	0.700	1: 0.50 2: 0.00 3: 0.79 WA: 0.70	1: 0.02 2: 0.00 3: 1.00 WA: 0.79	1: 0.04 2: 0.00 3: 0.88 WA: 0.70	1 2 3 1 1 0 50 2 0 0 14 3 1 0 237

4.2.4 Ex ante prediction performance in the binary classification.

For binary classification, the range of the five-star ratings is reduced to the binary class as positive (five- and four-star ratings) or negative (others). The class distribution is skewed toward the positive class. If a machine learning model classifies all the instances in the test set as a positive class, the accuracy will be 0.7855. Therefore, the minimum reasonable level of accuracy for a classifier is 0.7855 (Table 15).

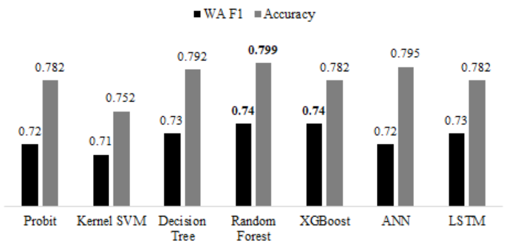
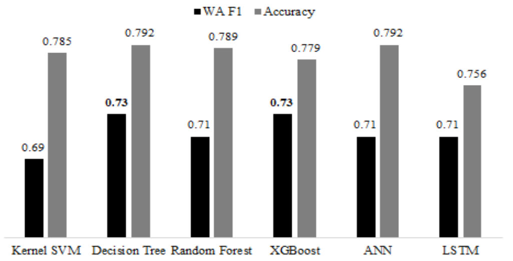
Table 15. Class distribution in the binary classification

	Total Set		Total Training Set		Sub Training Set		Valid Set		Test Set	
1	4,149	78.18%	3,911	78.16%	3670	78.04%	241	80.07%	238	78.55%
0	1,158	21.82%	1,093	21.84%	1033	21.96%	60	19.93%	65	21.45%
Total	5,307	100.00%	5,004	100.00%	4703	100.00%	301	100.00%	303	100.00%

The WA F1 score of the best classifier in the binary classification is the same as in the three-class classification, 0.74 (Table 16). This point indicates that reducing the class ranges from three-class to binary classification does not improve the prediction performance of classifiers. However, the prediction performance of the binary classification is better than the five-star rating classification in terms of the WA F1 score.

Table 16 also shows that the accuracy of the best classifiers in the binary classification (0.799 with the RF) is less than that of the three-class classification (0.802 with XGB and the RF). In addition, adding product dummies to the feature set does not improve binary classifiers' prediction performance, excluding the DT.

Table 16. Ex ante prediction results in the binary class case

Models	Variables	Weighted Average (WA) Macro F1-score and Accuracy
Model 3: ex ante-sub-model	90 variables including: 1. variables from ‘At time’ model 2. 32 variables for the reviewer’s volume of prior reviews in each sub-category	
Model 5: partial ex ante-sub-model	161 variables including: 1. 90 variables from ex ante-sub-model 2. 71 product dummies	

* The binary probit model is applied; however, the probit model is incompatible with product dummies due to the multicollinearity problem.

Table 17. Binary classification for ex ante prediction

Models	Hyperparameter	Accuracy	Precision	Recall	F1-score	Confusion matrix
PROBIT	Binary classification	0.782	1: 0.47 2: 0.80 WA: 0.73	1: 0.11 2: 0.97 WA: 0.78	1: 0.18 2: 0.87 WA: 0.72	1 2 1 7 58 2 8 230
Kernel SVM	Kernel: RGB Gamma: 0.01 C: 10	0.752	1: 0.29 2: 0.79 WA: 0.68	1: 0.11 2: 0.93 WA: 0.75	1: 0.16 2: 0.88 WA: 0.71	1 2 1 7 58 2 17 221
Decision Tree	Criteria: Entropy Max depth: 3	0.792	1: 0.60 2: 0.80 WA: 0.76	1: 0.09 2: 0.98 WA: 0.79	1: 0.16 2: 0.88 WA: 0.73	1 2 1 6 59 2 4 234
Random Forest	Tree numbers: 31 Depth: 38	**0.799**	1: 0.64 2: 0.81 WA: 0.77	1: 0.14 2: 0.98 WA: 0.80	1: 0.23 2: 0.88 **WA: 0.74**	1 2 1 9 56 2 5 233
Xgboost	Tree number: 100 Depth: 3 Learning rate: 0.1	0.782	1: 0.48 2: 0.80 WA: 0.73	1: 0.15 2: 0.95 WA: 0.78	1: 0.23 2: 0.87 **WA: 0.74**	1 2 1 10 55 2 11 227
ANN	Epoch: 146 Drop out: 0.4 Learning rate: 0.0002 Hidden layer 1 node: 270 Hidden layer 2 node: 270	0.795	1: 0.71 2: 0.80 WA: 0.78	1: 0.08 2: 0.99 WA: 0.80	1: 0.14 2: 0.88 WA: 0.72	1 2 1 5 60 2 2 236
LSTM	Epoch: 8 Learning rate: 0.0002 Hidden layer node: 135	0.782	1: 0.47 2: 0.80 WA: 0.73	1: 0.12 2: 0.96 WA: 0.78	1: 0.20 2: 0.87 WA: 0.73	1 2 1 8 57 2 9 229

To summarize the prediction of consumers' potential star ratings with different feature sets and class ranges, XGB is the best and most stable prediction machine. The prediction performance is the highest in the three-class classification. Surprisingly, adding price variables (potentially biased variables) and product dummies does not improve much or does not improve the prediction performance of the best classifiers among the classifiers in each case.

In addition, the minority class (three-star rating) prediction performance is nearly zero in this imbalanced dataset. A firm often wants to classify potential happy consumers for target marketing. Following this assumption, firms may pay less attention to predicting potential three-star-rating consumers. However, if a three-star-rating reviewer group is the minority group in a society, it may cause unfairness and inequality issues.

4.4 Sentiment Classification in the Product Content Dimension Using NLP

Labeling text data for sentiment analysis often requires high-cost, time-consuming, and labor-intensive work. For example, the domain expert took about 45 days to complete the 37,149 labeling tasks in this study. If the volume of review data is larger, the required time, labor, and financial cost for annotation will increase as well. In this case, firms could reduce these labeling costs by leveraging natural language processing (NLP).

Due to recent innovation in the NLP methods, firms could apply deep learning methods to identify semantic meanings from review text. In particular, after training NLP models on an expert-annotated training dataset, the trained NLP models could classify the reviewers' sentiment toward a specific product content dimension in a new review text dataset. Firms could apply these sentiment analyses to heuristic, fast, data-driven business decision making for better consumer support and feedback.

As a digital experiment to examining NLP's potential for sentiment analysis, diverse NLP methods are applied to classify reviewers' sentiment toward a specific product content dimension (functionality) because the functionality dimension contains the least imbalanced data among the nine product content dimensions (PCDs) for programmable thermostats (PTs). As shown in Table 18, the reviewers' sentiment regarding the functionality is distributed as follows: positive (1) with 41.70%, neutral (0) with 32.77%, and negative (-1) with 25.53%. This dataset is relatively balanced compared with the previous datasets.

Table 18. Sentiment distribution in the functionality dimension

Sentiment	Nest	Honeywell	Lux	Hunter Fan Com	Venstar	White Roger	Total	Percent
-1	584	424	269	45	18	15	1,355	25.53%
0	700	619	315	46	32	27	1,739	32.77%
1	789	744	555	70	43	12	2,213	41.70%
Total	2,073	1,787	1,139	161	93	54	5,307	100.00%

Word embedding is a way to map words, sentences, and documents to the real vector space. Word embedding assumes that numerical vectors generated from review text contain the semantic information in the review text. Following this assumption, the quality of word embedding vectors is essential for sentiment classification performance. Three different word-embedding approaches are applied in this study to convert review text into numerical input vectors: (1) word frequency-based embedding, (2) word distribution-based embedding, and (3) context-based embedding.

In particular, transfer learning has shown success in different NLP tasks and has become an important approach in NLP (Devlin et al. 2018; Erhan et al. 2010; Pan and Yang 2009). Transfer learning assumes that, when the training dataset is relatively small, using parameters in pre-trained models trained with big data could improve NLP models' performance in a new target task.

Two popular transfer learning approaches are fine-tuning (Devlin et al. 2018) and further pre-training (Gururangan et al. 2020). The fine-tuning approach simply reuses a pre-trained model for new target tasks. A further pre-training approach is to train a pre-trained model with domain data to update the weights in the pre-trained model to reflect domain contextual information. The fine-tuning and further pre-training methods are applied to the W2V and BERT models in this study.

On top of each word-embedding vector generated from the review text, tree-based ensemble models (RF, XGB) and a deep learning model (CNN) are applied to classify reviewers' sentiment toward the functionality dimension. Each classification model is combined with a suitable word-embedding method for each classifier's characteristics.

4.4.1 Word embedding: Mapping text to numerical vectors

4.4.1.1. Term frequency–inverse document frequency (TF-IDF)

Frequency-based embedding is a simple way to map each review text to numerical vectors. Term frequency–inverse document frequency (TF-IDF) is a frequency-based type of word embedding and penalizes the high-frequency words in the entire review (Haque, Saber, and Shah 2018). For example, “the” may have a low TF-IDF value because many reviews contain “the.”

The pre-processing for TF-IDF in this study is conducted as follows:

Step 1. Putting all words into lower case;

Step 2. Splitting the review text into words;

Step 3. Removing stopwords, punctuation, numbers, and single characters;

Step 4. Lemmatizing words (converting words into the base form, e.g., writing → write).

After the above steps, the number of unique words in 5,307 review texts (vocabulary) is 15,843. This is a spare high-dimension matrix containing many zero values. TF-IDF represents how frequently a word appears in the entire review as follows:

$$\mathrm{TF-IDF\ score\ (unique\ word_{n,i})} = \mathrm{tf_{n,i}} \times \log\frac{\mathrm{N}}{\mathrm{df_n}}$$

$\mathrm{tf_{n,i}}$: the frequency of word n in review i (term frequency)

$\mathrm{df_n}$: the frequency of reviews containing word n (document frequency)

N : the number of total reviews (N = 5,307)

In this equation, low-frequency words in review i will have a low TF-IDF score due to low term frequency; common words that occur in many reviews will also have a low TF-IDF score due to low document frequency (Gentzkow, Kelly, and Taddy 2019). On top of the TF-IDF embedding vectors from the review text data, tree ensemble models (RF and XGB) are applied for sentiment analysis. TF-IDF has a high-dimensional spare matrix and cannot represent similarity, ambiguity, and contextual meaning in a text.

4.4.1.2. Word2Vec (W2V)

The Word2Vec (W2V) model is a word distribution-based embedding method and generates dense embedding vectors representing each word's semantic meaning. For example, the W2V model may generate similar embedding vectors for "pen" and "pencil" because the two words contain similar semantic meanings.

As a pre-process, the following steps are applied:

Step 1. Converting emoticon and $ symbols into related words;

Step 2. Splitting the review text into words (tokenization);

Step 3. Removing stopwords, punctuation, numbers, and single characters;

Step 4. Lemmatizing words (converting words into the base form, e.g., writing → write).

After the above steps, the W2V model generates embedding vectors from each review text. The skip-gram W2V model (Mikolov et al. 2013; Timoshenko and Hauser 2019) generates k-dimensional real-vector word embedding $\mathrm{v_n}$ for the nth word in all reviews by maximizing the following objective function:

$$\frac{1}{\mathrm{N}}\sum_{\mathrm{n=1}}^{\mathrm{N}}\sum_{\mathrm{-c<s<c;\ s>0}}\log \mathrm{p}(\mathrm{word}_{\mathrm{n+s}}|\mathrm{word}_{\mathrm{n}})$$

$$\text{where } \mathrm{p}(\mathrm{word_s}|\mathrm{word_n}) = \frac{\exp(\mathrm{v\prime_s v_n})}{\sum_{\mathrm{t=1}}^{\mathrm{T}}\exp(\mathrm{v\prime_t v_{n)}}}$$

N is the number of words in all the reviews (the entire corpus); c is the window size for selecting neighboring words around the center word n; and T is the number of unique words (vocabulary) in all the reviews. In this study, the W2V model is trained with all the reviews (N = 1,926,047) in the "tool and home improvement" category and the number of unique words is 73,856. The hyperparameters are the W2V embedding dimension, window size, and training dataset. After hyperparameter tuning, the optimal W2V embedding dimension is 100 and the optimal window size is 5.

4.4.1.3. Bidirectional Encoder Representations from Transformers (BERT).

Bidirectional Encoder Representations from Transformers (BERT; Devlin et al. 2018) is a state-of-the-art context-based embedding method. BERT can represent the same word in a sentence with different embedding vectors by reflecting the contextual meaning of each word in the sentence. For example, in the sentences "I did not like this thermostat in the past. Now, I love this thermostat," the word "thermostat" occurs twice, in the first and in the second sentence. BERT generates different embedding vectors for “thermostat” in the first and second sentences based on the contextual information in them. Meanwhile, context-free embedding models (e.g., TF-IDF and W2V) generate the same embedding vectors for “thermostat” in both sentences.

In particular, the domain expert in this study reads and annotates all 5,307 reviews for PTs and finds that the review text often contains a comparison between the previously owned PT and the newly purchased PT; therefore, the same word in the review often represents different contexts based on its position in the review. For example, “I disliked the previous thermostat. However, I love this new thermostat.” In this text, even though the word “thermostat” occurs both in the first and in the second sentence, the first one may contain a negative sentiment and the second one may contain a positive sentiment. However, context-free embedding models (e.g., TF-IDF and W2V) cannot capture different semantic meanings of the same word in different positions in the review sentence. In contrast to the context-free embedding models, BERT (context-based embedding) can find the contextual difference between occurrences of the same word in different positions in the review sentence.

The pre-trained BERT embedding model is trained with 800 million words using a book corpus (Zhu et al. 2015) and 2,500 million words from Wikipedia data. BERT uses the WordPiece tokenizer (Wu et al. 2016), which splits each word into sub-words to deal with out-of-vocabulary words.

BERT’s structure is based on multilayered transformer encoders (Vaswani et al. 2017).

BERT is trained for two objectives: masked language modeling (MLM) and next sentence prediction (NSP). MLM is a prediction task for randomly masked tokens in the sentences to learn about the contextual information in the text. NSP is a binary classification indicating whether the second sentence is a subsequent sentence to the first one to learn about the relationship between sentences.

This study uses the BERT-based model, which contains 30,522 unique tokens with 768 embedding dimensions for fine-tuning and further pre-training. With a fine-tuned BERT, the CNN is applied on top of the pre-trained embedding from the original BERT model. Having further pre-trained BERT, the BERT embedding is updated by training on the review text data and is used as input vectors for the CNN classifier. Recently, Gururangan et al. (2020) and Sun et al. (2019) showed that further pre-training with domain data could improve machine learning models' performance.

4.4.2 Convolutional neural network (CNN) for sentiment classification

Many studies (Kalchbrenner, Grefenstette, and Blunsom 2014; Kim 2014; Zhang and Wallace 2015) have applied a convolutional neural network (CNN) for text classification and shown good performance. In particular, Liu, Lee, and Srinivasan (2019) and Timoshenko and Hauser (2019) applied CNN text classification on top of W2V embedding trained on online product review data. In this study, the CNN classifier on top of BERT or W2V embedding is applied for sentiment analysis. Figure 8 provides an example of a simplified CNN model for the binary classification model. The structure of the CNN in this example has four layers. The first layer is the input word embedding generated from the review text. Each review text is split into tokens (e.g., words in a W2V model and sub-words in a BERT model) and becomes a sequence of the tokens with length n. The tokenized review is denoted as $\mathrm{x}_{1:n}$. Each token x_i is mapped to a word-embedding vector R^d. The embedded sequence of tokens $\mathrm{x}_{1:n}$ is expressed as follows:

$$\mathrm{x}_{1:n} \; = \; \mathrm{x}_1 \oplus \; \mathrm{x}_2 \; ..\oplus \; \mathrm{x}_n, \text{ where } \mathrm{x}_i \in \mathrm{R}^d, \mathrm{i} \in \{1, \ldots, \mathrm{n}\}$$

$\oplus$ denotes the concatenation operator. After concatenation of the sequence of n embedded tokens from a review, $\mathrm{x}_{1:n}$ becomes the word embedding matrix in $\mathrm{R}^{n \times d}$.

The second layer is the convolutional layer. A convolution operation (filter) is applied to input word embedding from the first layer to generate a feature map. A filter f has a word window size h and the same embedding dimension d with input word embedding as $\mathrm{f} \in \; \mathrm{R}^{h \times d}$. A filter is applied to each window size of words in the review sentence $\{\mathrm{x}_{1:h}, \mathrm{x}_{2:h}, \ldots, \mathrm{x}_{n-h+1:n}\}$ and generates a feature map $\mathrm{c} = \{\mathrm{c}_1, \mathrm{c}_2, \ldots, \mathrm{c}_{n-h+1}\}$ as $\mathrm{c} \in \mathrm{R}^{n-h+1}$. Here, $\mathrm{c}_{i, i \in \{1,..,n-h+1\}}$ is an output of the filter as follows:

$$\mathrm{c}_i \; = \; \mathrm{f}(\text{weight vector} \cdot \mathrm{x}_{i:i+h-1} + \mathrm{b}).$$

where f is the non-linear activation function (the ReLU function in this study), $\cdot$ is an inner product, and b is a bias term. Filters with different window sizes are applied m times and generate m feature maps per filter.

The third layer is the pooling layer. The 1-max-pooling operation (Boureau, Ponce, and LeCun 2010) is applied to each feature map and generates scalar values as follows:

$$\hat{c} \ = \ \max c = \max\{c_1, c_2, \ldots, c_{n-h+1}\}, \text{ where } \hat{c} \ \in \ R^1$$

The idea of 1-max-pooling is the selection of the most important feature values in a feature map, and each feature map generating one scalar value. If there are e feature maps, 1-max-pooling generates a feature vector as $o = \ [\hat{c}_1, \ldots, \hat{c}_e]$.

The last layer is the fully connected layer. A feature vector o generated from text data is an input vector for the last layer. Dropout (Srivastava et al. 2014) is applied to feature vector o to prevent overfitting problems. This feature vector o can be combined with z variables from structured data as $o_{full} = \ [\hat{c}_1, \ldots, \hat{c}_e, \ X_1, \ldots, X_z]$. In this study, the CNN model is defined as the "partial model" when the feature vector is o (text only) or the "full model" when the feature vector is o_{full} (text and numerical variables).

The activation function is the softmax function and the output of the last layer y is:

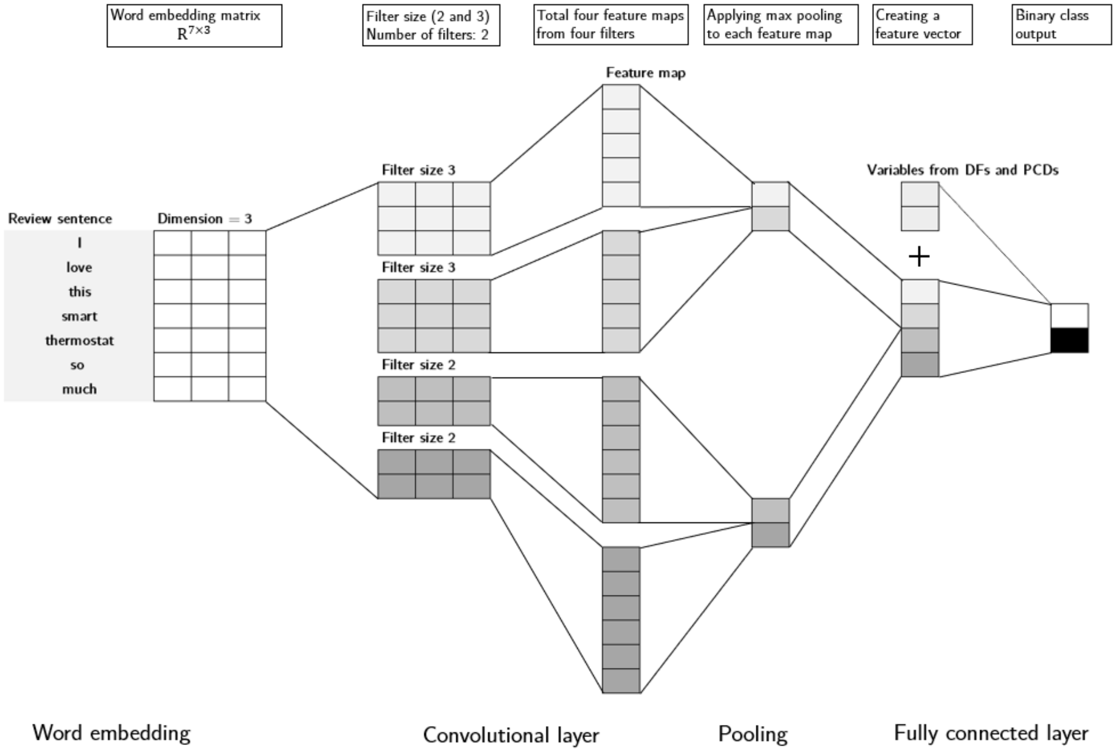
$$y \ (\text{class } = k) = \text{softmax}(w \cdot o + b), \text{ where } y = [0, \ 1]$$

Here, each class has a predicted probability, and the class showing the highest predicted probability will be a predicted class.

According to Zhang and Wallace (2015), the filter size and number of filters are key hyperparameters for a CNN model; 1-max pooling is better than other pooling methods; and regularization has little influence on the performance of the CNN classification. This study applies multiple different feature sizes and filters to find the optimal parameters.

Input embedding vectors are generated from multiple different versions of the W2V and BERT models. For structured data, 161 variables are selected from the partial ex ante sub-model as input variables for the full model (text and structured data model.)

Figure 8. The structure of the CNN (Kim 2014; Zhang and Wallace 2015)



4.4.3 Sentiment classification experiment design.

Table 19 shows the distribution of the three classes in the functionality decision in the review text. The dataset is relatively less imbalanced than the previous datasets; therefore, the prediction performance of the minority class (-1) may be better than in other previous cases. The bottom line of the test set accuracy is 0.3795.

Table 19. Class distribution in the functionality dimension

	Total Set		Total Training Set		Training Set		Valid Set		Test Set	
Class	Count	Shares	Count	Shares	Count	Share	Count	Share	Count	Share
-1	1,355	25.53%	1,281	25.60%	1211	25.75%	70	23.26%	74	24.42%
0	1,739	32.77%	1,625	32.47%	1523	32.38%	102	33.89%	114	37.62%
1	2,213	41.70%	2,098	41.93%	1969	41.87%	129	42.86%	115	37.95%
Total	5,307	100.00%	5,004	100.00%	4,703	100.00%	301	100.00%	303	100.00%
Period	Oct 12, 2005 – July 17 2014		Oct 12, 2005 – May 17, 2014		Oct 12, 2005 –Mar 16, 2014		Mar 17, 2014 –May 17 2014		May 18, 2014 – July 17 2014	

This study defines the partial and full models based on the type of features in the model. The partial model simplifies the feature engineering by excluding digital footprint (DF) mining from user-generated content (UGC) to generate numerical input variables. In general, DF mining requires intensive manual coding and enough computing resources (e.g., mass storage space and big-memory computers). Generating input variables from DFs also requires a large online product review dataset that contains an individual user ID, product ID, and time stamp. Firms often want to reduce feature engineering by focusing only on review text data (the partial-model approach). In contrast, the full-model approach demonstrates how to combine unstructured review text data and structured data to improve a classifier's prediction performance.

In this section, tree ensemble models (RF and XGB) are selected as a baseline model to compare their prediction performance with more complex models. The TF-IDF embedding method is applied to the RF and XGB models because the RF and XGB are incompatible with the two-dimensional word-embedding vectors generated from the W2V and BERT models.

The CNN model is a popular deep learning model for text classification. In particular, the CNN models on top of BERT or W2V embedding vectors are the main classifiers in this section. The CNN model's hyperparameters are the length of the review text, training epochs, number of filters, filter sizes, dropout rate, and learning rate in this study.

The W2V embedding models are trained on different review dataset with different window sizes and embedding dimensions. The CNN classifier on top of Google's pre-trained W2V embedding[2] (trained on three million words and phrases from Google News) shows lower prediction performance than the CNN classifier on top of W2V embedding (trained on online product review data in this study). In particular, two different online product review datasets are used for training the W2V models: (1) W2V_S (N = 169,809 reviews), containing all reviews of the target reviewers across all categories over the entire sample period; and (2) W2V_L (N = 1,926,047 reviews), consisting of all reviews in the "tool and home improvement category" over the entire sample period. The W2V model trained on W2V_L shows better performance for sentiment analysis in this section than the W2V model trained on W2V_S and Google's pre-trained model.

The BERT models are applied to word-embedding methods with two different approaches, the fine-tuning and further pre-training approaches. The fine-tuning approach simply reuses the pre-trained embedding vectors from the original model as input-

[2] Google code archive: https://code.google.com/archive/p/word2vec/

embedding vectors for a classifier. This approach relies on transferring learning and has recently shown successful performance in NLP tasks.

A further pre-training approach updates the pre-trained embedding vectors by training the pre-trained model on domain data to adapt domain context information to embedding vectors. However, there is no ground truth or theoretical proof for ensuring the better performance of further pre-training with noisy online product review data. Two different online product review datasets are applied for further pre-training: (1) BERT_S (N = 169,809 reviews), containing all reviews of the target reviewers across all categories over the entire sample period; and (2) BERT_L (N = 1,926,047 reviews), consisting of all reviews in the "tool and home improvement category." For further pre-training of the BERT model on domain-specific review data, the hyperparameters are the learning rate, batch size, and further training steps. In this study, the optimal hyperparameters for further training BERT are the learning rate 0.00001, batch size 32, and 1,926,047 training steps. In the BERT model, the maximum length of tokens is fixed as 512 (510 without special tokens); therefore, 512 is the maximum length of review tokens for the BERT model in this study.

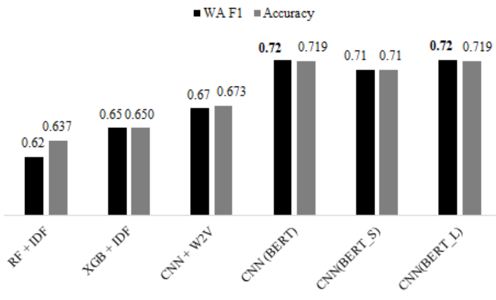
4.4.4 Sentiment classification of online product reviews.

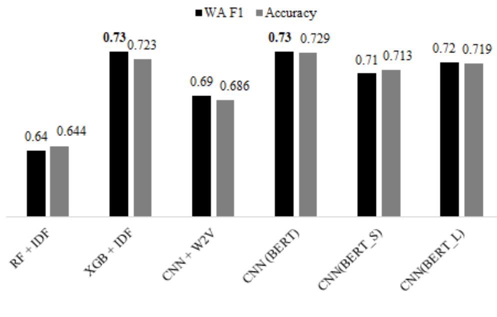
Table 20 presents the results of the sentiment classification of reviews about a specific product content dimension. The classification models are divided into the partial model (using text only) and the full model (using text and structured data). In the partial model, the CNN models on top of fine-tuned BERT or further pre-trained BERT_L embedding show the highest WA F1 score and accuracy. Accuracy is an important evaluation metric to measure the prediction performance because the dataset in this section is relatively more balanced than the datasets in the previous sections.

All the CNN models on top of BERT embedding shows better prediction performance than the tree ensemble models and the CNN models on top of context-free embedding (TF-IDF and W2V embedding). This point indicates that BERT is a better embedding method for sentiment classification in this section. It demonstrates that the identification of contextual information from online product review text is a critical factor for the sentiment classification of online product reviews (Table 20).

The CNN model on top of further pre-trained BERT on the BERT_S dataset shows lower prediction performance that the CNN models on top of the pre-trained BERT or further pre-trained BERT on the BERT_L dataset (Table 20).

Table 20. Sentiment classification results

Models	Word embedding	Weighted Average (WA) Macro F1-score and Accuracy
Partial model (Text only)	TF-IDF: embedding on 5,307 reviews W2V: embedding on W2V_L (dimension: 100, size: 5) BERT: pre-trained embedding BERT_S: further pre-trained embedding on BERT_S BERT_L: further pre-trained embedding on BERT_L	

Full model (Text + partial ex ante-sub-model)	Word embedding conditions in 'Text only' and 161 variables from the partial ex ante-sub-model	

In the full model, the CNN model on top of the fine-tuned BERT embedding shows the highest WA F1 score and accuracy (Table 20), indicating that firms could easily implement sentiment analysis without intensive training steps for word-embedding models and accomplish high prediction performance by reusing pre-trained BERT embedding as input embedding vectors. The CNN models with further trained BERT embedding show lower prediction performance than the CNN model with pre-trained BERT embedding. Therefore, further pre-training of BERT may not be a suitable embedding method in this section. Surprisingly, the class-weighted XGB on top of TF-IDF embedding shows the same WA F1 score as the CNN on top of pre-trained BERT embedding (Table 21). The prediction performance of XGB with text and structure data is higher than that of XGB with text data only. Therefore, this may be due to the weighted XGB's good prediction performance with structured numerical variables.

In contrast to the previous sections, the dataset in this section is a relatively balanced dataset; therefore, the imbalanced classification problem is not a critical issue in this section and the classification performance for the minority class is not low. Overall, the

CNN on top of fine-tuned BERT is the best option in all cases, with high prediction performance and a low computational cost for training the embedding model. In addition, the full-model cases are mostly better than the partial-model cases.

Table 21. Full model for sentiment classification

<table>
<tr><th>Models</th><th>Word Embedding</th><th>Hyperparameter</th><th>Accuracy</th><th>Precision</th><th>Recall</th><th>F1-score</th><th>Confusion matrix</th></tr>
<tr><td>Random Forest</td><td>TD-IDF</td><td>Tree numbers: 29
Depth: 26</td><td>0.644</td><td>1: 0.76
2: 0.65
3: 0.59
WA: 0.66</td><td>1: 0.53
2: 0.67
3: 0.70
WA: 0.64</td><td>1: 0.62
2: 0.66
3: 0.64
WA:0.64</td><td><table><tr><th></th><th>1</th><th>2</th><th>3</th></tr><tr><td>1</td><td>39</td><td>12</td><td>23</td></tr><tr><td>2</td><td>6</td><td>76</td><td>32</td></tr><tr><td>3</td><td>6</td><td>29</td><td>80</td></tr></table></td></tr>
<tr><td>Xgboost</td><td>TD-IDF</td><td>Tree number: 100
Depth: 7
Learning rate: 0.2
Class weighted*</td><td>0.723</td><td>1: 0.84
2: 0.74
3: 0.67
WA:0.74</td><td>1: 0.80
2: 0.66
3: 0.77
WA: 0.73</td><td>1: 0.82
2: 0.69
3: 0.72
WA: 0.73</td><td><table><tr><th></th><th>1</th><th>2</th><th>3</th></tr><tr><td>1</td><td>59</td><td>5</td><td>10</td></tr><tr><td>2</td><td>6</td><td>75</td><td>33</td></tr><tr><td>3</td><td>5</td><td>22</td><td>88</td></tr></table></td></tr>
<tr><td>CNN</td><td>W2V*</td><td>Max length = 1800
Epoch: 22
Number of filters:200
Filter size = (3,4,5)
Dropout = 0.7
Learning rate = 0.0001</td><td>0.686</td><td>1: 0.81
2: 0.65
3: 0.65
WA:0.70</td><td>1: 0.74
2: 0.62
3: 0.71
WA:0.69</td><td>1: 0.77
2: 0.64
3: 0.68
WA: 0.69</td><td><table><tr><th></th><th>1</th><th>2</th><th>3</th></tr><tr><td>1</td><td>55</td><td>12</td><td>7</td></tr><tr><td>2</td><td>6</td><td>71</td><td>37</td></tr><tr><td>3</td><td>7</td><td>26</td><td>82</td></tr></table></td></tr>
<tr><td>CNN</td><td>BERT</td><td>Max length: 512
Epoch: 15
Number of filters: 200
Filter sizes: (2,3,4)
Dropout: 0.7
Learning rate: 0.00001
Class weighted*</td><td>**0.729**</td><td>1: 0.94
2: 0.63
3: 0.78
WA:0.76</td><td>1: 0.62
2: 0.58
3: 0.68
WA:0.73</td><td>1: 0.75
2: 0.72
3: 0.73
WA:0.73</td><td><table><tr><th></th><th>1</th><th>2</th><th>3</th></tr><tr><td>1</td><td>46</td><td>21</td><td>7</td></tr><tr><td>2</td><td>2</td><td>97</td><td>15</td></tr><tr><td>3</td><td>1</td><td>36</td><td>78</td></tr></table></td></tr>
<tr><td>CNN</td><td>BERT further pre-training (BERT_S*)</td><td>Max length: 512
Epoch: 49
Number of filters:200
Filter sizes: (2,3,4)
Dropout: 0.6
Learning rate: 0.00001</td><td>0.713</td><td>1: 0.88
2: 0.64
3: 0.74
WA:0.73</td><td>1: 0.68
2: 0.84
3: 0.61
WA:0.71</td><td>1: 0.76
2: 0.72
3: 0.67
WA: 0.71</td><td><table><tr><th></th><th>1</th><th>2</th><th>3</th></tr><tr><td>1</td><td>50</td><td>14</td><td>10</td></tr><tr><td>2</td><td>3</td><td>96</td><td>15</td></tr><tr><td>3</td><td>4</td><td>41</td><td>70</td></tr></table></td></tr>
<tr><td>CNN</td><td>BERT further pre-training (BERT_L*)</td><td>Max length: 512
Epoch: 11
Number of filters:300
Filter sizes: (3,4,5)
Dropout:0.7
Learning rate:0.0001</td><td>0.719</td><td>1: 0.73
2: 0.69
3: 0.75
WA:0.72</td><td>1: 0.77
2: 0.72
3: 0.69
WA:0.72</td><td>1: 0.75
2: 0.70
3: 0.71
WA:0.72</td><td><table><tr><th></th><th>1</th><th>2</th><th>3</th></tr><tr><td>1</td><td>57</td><td>11</td><td>6</td></tr><tr><td>2</td><td>11</td><td>82</td><td>21</td></tr><tr><td>3</td><td>10</td><td>26</td><td>79</td></tr></table></td></tr>
</table>

* Class weight: class [-1, 0, 1], weights for each class [1.2945, 1.0293, 0.7962]; W2V: trained on W2V_L and embedding dimension is 100 with window size 5; BERT further training on BERT_S: further pre-trained with target reviewers' reviews across all categories (169,809 reviews) and further pre-trained with 849,045 steps (5 epochs with 169,809 steps per epoch); BERT further training on BERT_L: further pre-trained with all reviews in the "tool and home improvement" category and further pre-trained with 1,926,047 reviews (1,926,027 steps with 1 epoch).

5. Conclusion

This paper proposes novel approaches (1) to identify unobserved consumer characteristics and preferences by analyzing the target consumers' and other prior reviewers' digital footprints (DFs); (2) to extract product-specific product content dimensions from review text data by using topic modeling and domain expert annotation; (3) to predict individual consumers' potential preferences by using machine learning models; (4) to classify consumers' sentiment toward a specific product content dimension (PCD) by using context-based word embedding and state-of-the-art deep learning models.

This study finds that all heteroskedastic ordered probit (HETOP) models containing DFs and sentiment variables show a higher model fit than the base model containing no DFs or sentiment variables. Furthermore, machine learning models containing DFs and sentiment variables show better prediction performance than the base model. These points indicate the importance of DF mining and sentiment analysis for estimation and prediction tasks. The HETOP models' results show that a consumer is less likely to give a five-star rating for a reviewed programmable thermostat (PT) if he or she: (1) writes a longer review summary and body, (2) has a lower variance of review summary length in prior reviews, a larger volume of prior reviews across all categories, and a higher average rating in prior reviews across all categories, (3) writes a review for the PT that has a higher average length of review summary and/or lower variance of review summary length in prior reviews, (4) writes a larger volume of prior reviews in specific product categories ("Amazon instant video," "apps for Android," "cell phones," "clothes, shoes, jewelry," "grocery gourmet food," "health and personal care," "magazine subscriptions," and "software") and a smaller volume of reviews in the "appliance" category.

The eight sentiment variables positively affect the probability of a 5-star rating. The sentiment variables represent the target consumers' sentiment toward product content dimensions (PCDs). The dimensions are (1) smart connectivity, (2) easiness, (3) energy and money saving, (4) functionality, (5) support, (6) perceived price value, (7) privacy, and (8) the Amazon effect. These results suggest that firms could identify latent PCDs from user-generated online product reviews and measure the effect of these dimensions on the consumers' preferences for a specific target product group. In addition, to the best of the author's knowledge, this study is the first study about the effect of the online retail market platform's service quality on the consumers' star ratings. Without considering the online platform service quality effect, empirical results will be biased.

This study also finds that extreme gradient boosting (XGB) is the best prediction machine among six popular machine learning algorithms to predict individual consumers' potential preferences regarding the target product group. In addition, the models containing DFs and sentiment variables show higher prediction performance than the model without these variables. Adding potentially biased price variables does not improve the prediction performance. Interestingly, the prediction performance is the highest in the three-class classification, the second highest in the binary classification, and the lowest in the original five-class (five-star ratings) classification. Interestingly, the prediction performance of machine learning models for the minority class (the three-star rating) is extremely low. The machine learning models' prediction with the imbalanced dataset tends to be skewed toward the majority class, while the machines show low prediction performance for the minority class.

This study applies natural language processing (NLP) to classify the target consumers' sentiment toward a specific product content dimension from the review text. Firms could apply this approach to reduce expensive domain expert annotation costs and implement data-driven business decisions. The proposed convolutional neural network (CNN) model on top of pre-trained BERT embedding shows higher classification performance than the CNN model on top of word2vec embedding. This point indicates that contextual information in the online product review text is critical for improving the sentiment classification performance. The CNN model on top of further pre-trained BERT embedding shows lower performance than the CNN model on top of pre-trained BERT embedding. This may be due to the noise in the online review text data.

This paper contributes to the literature on consumer preference in digital economics and quantitative marketing. Anyone can voluntarily write a review without any fees in the one-sided review system (e.g., Amazon.com). The one-sided review system also does not provide detailed information about the reviewers; therefore, conventional revealed- and stated-preference analysis may be limited in identifying latent consumer preferences from online product reviews. This study identifies latent consumer characteristics and preferences by (1) mining the target consumers' and other prior reviewers' digital footprints (DFs) and (2) extracting the target consumers' sentiment toward product content dimensions from the target reviewers' review text.

This study also contributes to the literature on energy economics because the target product of this study is programmable thermostats, which require technical knowledge and skills. To the best of the author's knowledge, this is the first paper to analyze consumers' preferences regarding home energy control devices by using online product reviews. In particular, the innovative new firm, the Nest, entered the market and became a competitive market player with disruptive innovation. The uncertainty of inexperienced consumers may be high due to competition and disruptive innovation, hence the value of experienced consumers' prior online product reviews of the thermostats.

This study extracts product-specific product content dimensions from user-generated online product review text. It provides detailed product-specific contents and the effect of consumers' sentiment toward these product content dimensions on their preferences. The results suggest that consumers consider not only the smartness of programmable thermostats but also the easiness of using the device. Surprisingly, consumers also consider the value of privacy. Without extracting the latent product content dimension from the user-generated online product reviews, firms may not find these latent factors that affect consumer preferences from the summary statistics of online review data. The dimensions are very specific for home energy control devices. This approach could apply to designing the promotion of energy-efficient products, measuring the effect of an energy policy (such as energy star certification) on consumers' preferences in the online retail market platform, and identifying the factors that affect consumer satisfaction or dissatisfaction. In particular, online product reviews are free, easy to access, and reflect the actual consumer voices.

This study also contributes to the literature on biased online product review detection. It defines "suspicious one-time reviewers" as reviewers who write only one review for a programmable thermostat during the entire sample period. The suspicious one-time reviewers' share of the one-star ratings is higher than that of the target reviewers. In addition, this study defines "always-the-same raters" as reviewers who write reviews with the same star rating more than eight times. Surprisingly, all the always-the-same raters' ratings are five-star ratings. The processes could be applied as pre-processing of online product review analysis to mitigate the potential bias in the reviews.

This study contributes to the literature on classification in the machine learning field. It shows how to combine variables generated from text and other numerical variables for classification. This study also shows the effect of class ranges on each machine learning algorithms' prediction performance with the imbalanced dataset, finding that all the machine learning algorithms show low prediction performance for the minority class with the imbalanced dataset.

Another contribution of this study is related to the literature on natural language processing (NLP). It provides empirical evidence of outperformance of the context-based embedding (BERT) approach compared with context-free embedding models (TF-IDF and Word2Vec). In particular, this study applies transfer learning concepts by applying pre-trained BERT embedding as input embedding of the CNN classifier. It also suggests that the further pre-training of BERT with domain review text data may not guarantee the improvement of prediction performance.

Finally, this study contributes to the literature on the online review and recommendation system. It shows how to use online product reviews to identify and predict consumers' characteristics and preferences regarding a specific target product group. Therefore, this approach could be implemented in an online review system to identify the factors that influence consumer preferences and design better recommendation systems.

In sum, the approaches in this study are interpretable, applicable, and scalable to a wide range of goods, allowing for the identification and prediction of unobserved consumer preferences and sentiment associated with product content dimensions for a specific target product group.

7. Limitations and Future Study

The Amazon review system is a one-sided review system; therefore, reviewers' true information is not available. The asymmetric information problem is an inherent limitation in this research area (Mayzlin, Dover, and Chevalier 2014). Furthermore, anyone can write a review as a buyer in Amazon's review system. Although the pre-processing steps in this study may remove potentially biased reviews, researchers cannot know whether biased reviews are still present. Therefore, a study with true and fake review data would be useful to identify the differences between true and fake reviews. In addition, the detection of "suspicious one-time reviewers" and/or "always-the-same raters" in each product category will be an interesting topic because different categories may have different characteristics.

This study is based on a one-sided review system for online product reviews. Online product reviews in different online review systems will be a good topic for better online product review system design. The target product group in this study is experience goods requiring technical knowledge and skills. Applying the approaches in this study to specific search goods (e.g., organic or non-organic milk) or credible goods (e.g., wine) will be a good extension of this study. The effect of expensive domain expert annotation and relatively inexpensive crowd sourcing annotation (e.g., Amazon Mechanical Turk) for sentiment analysis of machines' classification performance will also be a valuable topic for future research. The size of the review data may influence machine learning models' prediction performance. Therefore, applying the approaches in this study to a product group with a larger review dataset will be useful in the future.

For now, the interpretability of each variable relies on the discrete choice model. In the prediction tasks, the interpretability of out-of-sample prediction models has not been discussed because complex models are often black-box models. However, the recent interesting research trends are the interpretability of the machine learning models after prediction. Therefore, the interpretability of machine learning models would be an interesting topic for future study.

The imbalanced classification problem could cause social inequality or unfairness issues if the majority class group belongs to the minority groups in a society. This study applies the class weighting approach to mitigate the problem of imbalance; however, critical questions remain: what is the optimal class weight for minority groups, what is a suitable evaluation metric or process to reduce potential bias, and what are the unexpected consequences of biased classification for an imbalanced dataset? These questions need to be answered in the future.

8. References

Anderson, M., and J. Magruder (2012), "Learning from the Crowd: Regression Discontinuity Estimates of the Effects of an Online Review Database," *Economic Journal*, 122 (563), 957-989.

Blei, D. M., A. Y. Ng, and M. I. Jordan (2003), "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, 3 (January), 993–1022.

Bouazizi, M., and T. Ohtsuki (2019), "Multi-Class Sentiment Analysis on Twitter: Classification Performance and Challenges," *Big Data Mining and Analytics*, 2 (3), 181–94.

Boureau, Y. L., J. Ponce, and Y. LeCun (2010), "A Theoretical Analysis of Feature Pooling in Visual Recognition," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 111–8.

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

Chauvin, Y., and D. E. Rumelhart, eds. (1995), *Backpropagation: Theory, Architectures, and Applications*. Psychology Press.

Chen, C., A. Liaw, and L. Breiman (2004), "Using Random Forest To Learn Imbalanced Data," *University of California, Berkeley*, 110 (1–12), 24.

Chen, S., and S. Khan (2003), “Rates of Convergence for Estimating Regression Coefficients in Heteroskedastic Discrete Response Models,” *Journal of Econometrics*, 117 (2), 245–78.

Chen, T., and C. Guestrin (2016, August), “Xgboost: A Scalable Tree Boosting System,” In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–94.

Chen, T. D., and K.M. Kockelman (2012), “Roles of Vehicle Footprint, Height, and Weight in Crash Outcomes: Application of a Heteroscedastic Ordered Probit Model,” *Transportation Research Record*, 2280 (1), 89–99.

Chen, W., K. Fu, J. Zuo, X. Zheng, T. Huang, and W. Ren (2017), “Radar Emitter Classification for Large Data Set Based on Weighted-Xgboost,” *IET Radar, Sonar & Navigation*, 11 (8), 1203–7.

Chen, Y., (2018), “User-generated physician ratings: Evidence from Yelp,” *url: https://www. softwareadvice. com/resources/how-patientsuse*.

Chevalier, J. A., and D. Mayzlin (2006), “The Effect of Word of Mouth on Sales: Online Book Reviews,” *Journal of Marketing Research*, 43 (3), 345–54.

Cui, G., H. K. Lui, and X. Guo (2012), “The Effect of Online Consumer Reviews on New Product Sales,” *International Journal of Electronic Commerce*, 17 (1), 39–58.

Cybenko, G. (1989), “Approximation by Superpositions of a Sigmoidal Function,” *Mathematics of Control, Signals and Systems*, 2 (4), 303–14.

Devlin, J., M. W. Chang, K. Lee, and K. Toutanova (2018), “Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding,” arXiv preprint arXiv:1810.04805.

Dietterich, T. (1995), “Overfitting and Undercomputing in Machine Learning,” *ACM Computing Surveys (CSUR)*, 27 (3), 326–7.

Dietterich, T. G. (2000, June), “Ensemble Methods in Machine Learning.” In *International Workshop on Multiple Classifier Systems*, 1–15. Berlin, Heidelberg: Springer.

Donaker, G., H. Kim, M. Luca, M. A. Weber, S. E. House Rich, G. J. Duhon, R. Berman, S. Melumad, C. Humphrey, R. Meyer, and D. Ngwe (2019), “Designing Better Online Review Systems.” *Harvard Business Review*, November/December, 97 (6), 3.

Erhan, D., A. Courville, Y. Bengio, and P. Vincent (2010, March), “Why Does Unsupervised Pre-training Help Deep Learning?,” in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 201–8.

Fischer, T., and C. Krauss (2018), “Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions,” *European Journal of Operational Research*, 270 (2), 654–69.

Friedman, J., T. Hastie, and R. Tibshirani (2001), *The Elements of Statistical Learning*: Series in Statistics (Vol. 1, No. 10). New York: Springer.

Friedman, J. H. (2001), "Greedy Function Approximation: A Gradient Boosting Machine," *Annals of Statistics*, 1189–232.

Garvin, D.A. (1984), "What does product quality really mean?" *Sloan management review*, *25*.

Gentzkow, M., B. Kelly, and M. Taddy (2019), "Text as Data," *Journal of Economic Literature*, 57 (3), 535–74.

Green, W. H. (2012), *Econometric Analysis*, 7th ed, *Harlow: Pearson Education.*

Greene, W. H., and D. A. Hensher (2010a), "Ordered Choices and Heterogeneity in Attribute Processing," *Journal of Transport Economics and Policy (JTEP)*, 44 (3), 331–64.

Greene, W. H., and D. A. Hensher (2010b), *Modeling Ordered Choices: A Primer*. Cambridge University Press.

Gururangan, S., A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith (2020), "Don't Stop Pretraining: Adapt Language Models to Domains and Tasks," arXiv preprint arXiv:2004.10964.

Haque, T. U., N. N. Saber, and F. M. Shah (2018, May), "Sentiment Analysis on Large Scale Amazon Product Reviews," in *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*, 1–6. IEEE.

He, R. and J. McAuley (2016, April), "Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering," In *Proceedings of the 25th International Conference on World Wide Web*, 507–17.

Hochreiter, S. (1998), "The Vanishing Gradient Problem during Learning Recurrent Neural Nets and Problem Solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6 (2), 107–16.

Hochreiter, S., and J. Schmidhuber (1997), "Long Short-Term Memory," *Neural Computation*, 9 (8), 1735-80.

Hornik, K. (1991), "Approximation capabilities of multilayer feedforward networks," *Neural networks*, 4(2), 251-7.

Hu, N., Liu, L. and Zhang, J.J. (2008), "Do online reviews affect product sales? The role of reviewer characteristics and temporal effects.", *Information Technology and management*, 9(3), 201-14.

Hu, N., P. A. Pavlou, and J. Zhang (2006), "Can Online Reviews Reveal a Product's True Quality? Empirical Findings and Analytical Modeling of Online Word-of-Mouth Communication," in *Proceedings of the 7th ACM Conference on Electronic Commerce*, 324–30.

Ibrahim, M., M. Torki, and N. El-Makky (2018, December), "Imbalanced Toxic Comments Classification Using Data Augmentation and Deep Learning," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 875–8. IEEE.

James, G., D. Witten, T. Hastie, and R. Tibshirani (2013), *An Introduction to Statistical Learning* (Vol. 112, p. 18). New York: Springer.

Jeni, L. A., J. F. Cohn, and F. De La Torre (2013, September), "Facing Imbalanced Data—Recommendations for the Use of Performance Metrics," in *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction* (pp. 245–51). IEEE.

Kalchbrenner, N., E. Grefenstette, and P. Blunsom (2014, June), "A Convolutional Neural Network for Modelling Sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics* (Volume 1: Long Papers), 655–65.

Keele, L. and D. K. Park (2006, September), “Difficult Choices: An Evaluation of Heterogeneous Choice Models,” in *Paper for the 2004 Meeting of the American Political Science Association*, 2–5.

Kim, Y. (2014, October), “Convolutional Neural Networks for Sentence Classification,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746–51.

Kingma, D. P., and J. Ba (2014), “Adam: A Method for Stochastic Optimization,” arXiv preprint arXiv:1412.6980.

Kuan, C. M., and H. White (1994), “Artificial Neural Networks: An Econometric Perspective,” *Econometric Reviews*, 13 (1), 1–91.

LeCun, Yann, Bengio Yoshua, and Geoffrey Hinton (2015), “Deep Learning,” *Nature*, 521 (7553), 436.

Lemp, J. D., K. M. Kockelman, and A. Unnikrishnan (2011), “Analysis of Large Truck Crash Severity Using Heteroskedastic Ordered Probit Models,” *Accident Analysis & Prevention*, 43 (1), 370–80.

Litchfield, J., B. Reilly, and M. Veneziani (2012), “An Analysis of Life Satisfaction in Albania: An Heteroscedastic Ordered Probit Model Approach,” *Journal of Economic Behavior & Organization*, 81 (3), 731–41.

Liu, X., D. Lee, and K. Srinivasan (2019), "Large-Scale Cross-Category Analysis of Consumer Review Content on Sales Conversion Leveraging Deep Learning," *Journal of Marketing Research*, 46 (6), 918–43.

Luca, M. (2016), "Reviews, Reputation, and Revenue: The Case of Yelp.com," Harvard Business School NOM Unit Working Paper 12-016 (March 15, 2016).

Luca, M. (2017), "Designing Online Marketplaces: Trust and Reputation Mechanisms," *Innovation Policy and the Economy*, 17 (1), 77–93.

Luca, M., and G. Zervas (2016), "Fake It Till You Make It: Reputation, Competition, and Yelp Review Fraud," *Management Science*, 62 (12), 3412–27.

Mäkinen, S. J. (2014, December), "Internet-of-Things Disrupting Business Ecosystems: A Case in Home Automation," in *2014 IEEE International Conference on Industrial Engineering and Engineering Management*, 1467–70. IEEE.

Mallick, D. (2008), "Marginal and Interaction Effects in Ordered Response Models," Economics and Econometrics Research Institute (EERI), Brussels

Mayzlin, D., Y. Dover, and J. Chevalier (2014), "Promotional Reviews: An Empirical Investigation of Online Review Manipulation," *American Economic Review*, 104 (8), 2421–44.

McCluskey, J. J. (2000), “A Game Theoretic Approach to Organic Foods: An Analysis of Asymmetric Information and Policy,” *Agricultural and Resource Economics Review*, 29 (1), 1–9.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean (2013), “Distributed Representations of Words and Phrases and Their Compositionality. In *Advances in Neural Information Processing Systems*, 3111–9.

Nowak, J., A. Taspinar, and R. Scherer (2017, June), “LSTM Recurrent Neural Networks for Short Text and Sentiment Classification,” in *International Conference on Artificial Intelligence and Soft Computing*, 553–62. Cham: Springer.

Olah, C. (2015), “Understanding LSTM Networks.”

Pal, M. (2008), “Multiclass Approaches for Support Vector Machine Based Land Cover Classification”, arXiv preprint arXiv:0802.2411.

Pan, S. J., and Q. Yang (2009), “A Survey on Transfer Learning,” *IEEE Transactions on Knowledge and Data Engineering*, 22 (10), 1345–59.

Papadimitriou, T., P. Gogas, and E. Stathakis (2014), “Forecasting Energy Markets Using Support Vector Machines,” *Energy Economics*, 44, 135–42.

Passonneau, R. J., C. Rudin, A. Radeva, and Z. A. Liu (2009, March), "Reducing Noise in Labels and Features for a Real World Dataset: Application of NLP Corpus Annotation Methods," in *International Conference on Intelligent Text Processing and Computational Linguistics*. Berlin, Heidelberg: Springer, 86–97.

Ramsey, S. M., and J. S. Bergtold (2020), "Examining Inferences from Neural Network Estimators of Binary Choice Processes: Marginal Effects, and Willingness-to-Pay," *Comput Econ*. https://doi.org/10.1007/s10614-020-09998-w

Rao, G., W. Huang, Z. Feng, and Q. Cong (2018), "LSTM with Sentence Representations for Document-Level Sentiment Classification," *Neurocomputing*, 308, 49–57.

Reimers, I.C. and Waldfogel, J. (2020), "Digitization and Pre-Purchase Information: The Causal and Welfare Impacts of Reviews and Crowd Ratings." *NBER Working Paper* w26776, National Bureau of Economic Research.

Schiilkop, P. B., C. Burgest, and V. Vapnik (1995, August), "Extracting Support Data for a Given Task," in *Proceedings, First International Conference on Knowledge Discovery & Data Mining*. Menlo Park, CA: AAAI Press, 252–7.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014), "Dropout: A Simple Way To Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, 15 (1), 1929–58.

Strang, G. (2019), *Linear Algebra and Learning from Data*. Wellesley-Cambridge Press.

Sun, C., X. Qiu, Y. Xu, and X. Huang (2019, October), "How To Fine-Tune BERT for Text Classification?," in *China National Conference on Chinese Computational Linguistics*. Cham: Springer, 194–206.

Susan, M. M., & S. David (2010), "What Makes a Helpful Online Review? A Study of Customer Reviews on Amazon.com," *MIS Quarterly*, 34 (1), 185–200.

Syed, S., & M. Spruit (2017, October), "Full-Text or Abstract? Examining Topic Coherence Scores Using Latent Dirichlet Allocation," in *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 164–74.

Tadelis, S. (2016), "Reputation and Feedback Systems in Online Platform Markets," *Annual Review of Economics*, 8, 321–40.

Timoshenko, A., and J. R. Hauser (2019), "Identifying Customer Needs from User-Generated Content," *Marketing Science*, 38 (1), 1–20.

Train, K. E. (2009), *Discrete Choice Methods with Simulation*. Cambridge University Press.

Urban, G., A. Timoshenko, P. Dhillon, and J. R. Hauser (2020), "Is Deep Learning a Game Changer for Marketing Analytics?," *MIT Sloan Management Review*, 61 (2), 70-6.

Vaswani, A., N. Shazeer, U. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin (2017), “Attention Is All You Need,” in *Advances in Neural Information Processing Systems*, 5998–6008.

Vert, J. P., K. Tsuda, and B. Schölkopf (2004), “A Primer on Kernel Methods,” *Kernel Methods in Computational Biology*, 47, 35–70.

Wang, X. and K. M. Kockelman (2005), “Use of Heteroscedastic Ordered Logit Model To Study Severity of Occupant Injury: Distinguishing Effects of Vehicle Weight and Type,” *Transportation Research Record*, 1908 (1), 195–204.

Williams, R. (2009), “Using Heterogeneous Choice Models To Compare Logit and Probit Coefficients across Groups,” *Sociological Methods & Research*, 37 (4), 531–59.

Wu, Y., M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, and J. Klingner (2016), “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation,” arXiv preprint arXiv:1609.08144.

Yang, R., and M. W. Newman (2012, September), “Living with an Intelligent Thermostat: Advanced Control for Heating and Cooling Systems,” in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 1102–7.

Zhang, Y., and B. Wallace (2015), “A Sensitivity Analysis of (and Practitioners’ Guide to) Convolutional Neural Networks for Sentence Classification,” arXiv preprint arXiv:1510.03820.

Zhao, Y., S. Yang, V. Narayan, and Y. Zhao (2013), “Modeling Consumer Learning from Online Product Reviews,” *Marketing Science*, 32 (1), 153–69.

Zhu, Y., R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler (2015), “Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books,” in *Proceedings of the IEEE International Conference on Computer Vision*, 19–27.

Appendix A. Computing Environment

1. High Performance Computing

The author used WSU Kamiak HPC (https://hpc.wsu.edu/). The most of codes were written by JupterLab(https://jupyterlab.readthedocs.io/en/stable/). The most cases, GPU is Tesla K80 and there were the maximum number of GPUs was four. For heavy optimization, the author submits a SLURM jobscript (i.e., XGB.) The Google Colab and Colab pro were also used for deep learning models, including BERT. All the code and results were recorded in each Jupyter notebook file.

Table 22. Computing Package

Method	Python package
Data pre-processing	Pandas, Numpy, Dask, Multiprocessing
Feature engineering for text data	NLTK (https://www.nltk.org/)
HETOP and OP.	Stata 16 (https://www.stata.com/)
LDA	Gensim (https://radimrehurek.com/gensim/)
SVM	Sklearn (https://scikit-learn.org/stable/)
DT	Sklearn (https://scikit-learn.org/stable/)
RF	Sklearn (https://scikit-learn.org/stable/)
XGB	Xgboost (https://xgboost.readthedocs.io/en/latest/)
ANN	Pytorch (https://pytorch.org/)
LSTM	Pytorch (https://pytorch.org/)
CNN	Pytroch (https://pytorch.org/)
TF-IDF	Sklearn (https://scikit-learn.org/stable/)
W2V	Gensim (https://radimrehurek.com/gensim/)
BERT	Huggingface transformer (https://huggingface.co/transformers/)

Appendix B. Ordered Response Model

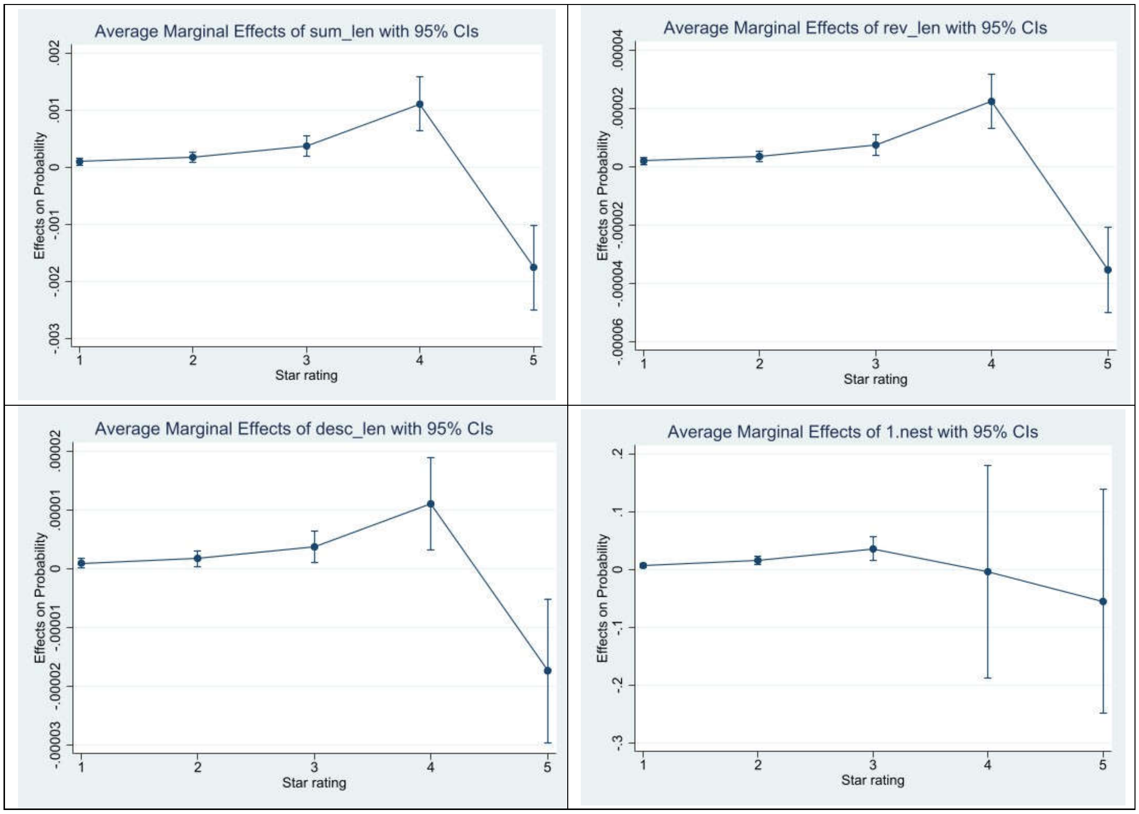
Table 23. Empirical results of the OP and HETOP models.

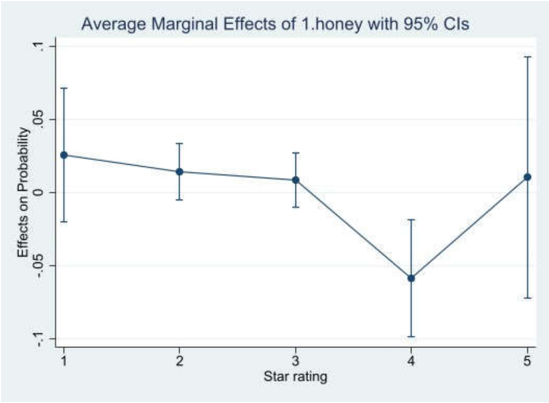
Variable	model_o1	model_h2	model_o2	model_h3	model_o3	model_h4	model_o4	model_h5	model_o5
sum_len	-0.010*** (0.001)	-0.008*** (0.002)	-0.006*** (0.001)	-0.007*** (0.002)	-0.006*** (0.001)	-0.008*** (0.002)	-0.005*** (0.001)	-0.008*** (0.002)	-0.005*** (0.001)
rev_len	-0.000*** (0.000)	-0.000*** (0.000)	-0.000*** (0.000)	-0.000*** (0.000)	-0.000*** (0.000)	-0.000*** (0.000)	-0.000*** (0.000)	-0.000*** (0.000)	-0.000*** (0.000)
title_len	0.003 (0.002)	0.003 (0.003)	0.003 (0.002)	0.002 (0.003)	0.003 (0.002)	0.002 (0.003)	0.003 (0.002)	0.002 (0.003)	0.003 (0.002)
desc_len	-0.000 (0.000)	-0.000** (0.000)	-0.000** (0.000)	-0.000** (0.000)	-0.000** (0.000)	-0.000** (0.000)	-0.000** (0.000)	-0.000** (0.000)	-0.000** (0.000)
nest	0.463*** (0.162)	0.170 (0.248)	0.016 (0.216)	0.033 (0.277)	-0.064 (0.238)	0.191 (0.273)	0.015 (0.218)	0.050 (0.311)	-0.055 (0.239)
honey	0.426*** (0.158)	0.431** (0.213)	0.252 (0.184)	0.393* (0.211)	0.234 (0.186)	0.492** (0.232)	0.248 (0.186)	0.460** (0.231)	0.231 (0.187)
hunter	0.235 (0.184)	-0.008 (0.265)	-0.124 (0.223)	-0.022 (0.263)	-0.134 (0.223)	-0.034 (0.294)	-0.141 (0.224)	-0.050 (0.295)	-0.152 (0.225)
lux	0.469*** (0.164)	0.555** (0.234)	0.350* (0.191)	0.531** (0.231)	0.341* (0.191)	0.594** (0.252)	0.335* (0.193)	0.578** (0.251)	0.326* (0.193)
venstar	0.648*** (0.203)	0.428 (0.278)	0.333 (0.234)	0.324 (0.282)	0.280 (0.242)	0.488 (0.310)	0.325 (0.236)	0.386 (0.319)	0.277 (0.244)
holiday	0.028 (0.100)	0.188 (0.163)	0.170 (0.115)	0.187 (0.161)	0.170 (0.115)	0.240 (0.189)	0.186 (0.115)	0.241 (0.189)	0.187 (0.115)
help_dfs		0.004 (0.003)	0.003 (0.002)	0.004 (0.003)	0.003 (0.002)	0.005 (0.003)	0.003 (0.002)	0.005 (0.003)	0.003 (0.002)
no_help_dfs		-0.011 (0.010)	-0.007 (0.007)	-0.011 (0.010)	-0.007 (0.007)	-0.013 (0.012)	-0.007 (0.007)	-0.013 (0.012)	-0.007 (0.007)
u_avg_len_sum		-0.002 (0.003)	-0.002 (0.002)	-0.002 (0.003)	-0.002 (0.002)	-0.002 (0.003)	-0.002 (0.002)	-0.002 (0.003)	-0.002 (0.002)
u_sd_len_sum		0.009* (0.005)	0.006 (0.004)	0.010* (0.005)	0.006 (0.004)	0.010 (0.006)	0.005 (0.004)	0.011* (0.006)	0.006 (0.004)
u_avg_len_rev		-0.000 (0.000)	-0.000 (0.000)	-0.000 (0.000)	-0.000 (0.000)	0.000 (0.000)	-0.000 (0.000)	0.000 (0.000)	-0.000 (0.000)
u_sd_len_rev		0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)
cum_reviews		-0.001** (0.000)	-0.001** (0.000)	-0.001** (0.001)	-0.001** (0.000)				
cate_diversity		0.002 (0.048)	0.003 (0.035)	-0.016 (0.053)	-0.009 (0.039)	-0.016 (0.061)	-0.009 (0.039)	-0.022 (0.066)	-0.012 (0.042)
u_avg_rating		0.225*** (0.053)	0.169*** (0.020)	0.223*** (0.052)	0.169*** (0.020)	0.254*** (0.063)	0.167*** (0.020)	0.255*** (0.063)	0.167*** (0.020)
u_sd_rating		0.006 (0.048)	-0.004 (0.034)	0.009 (0.048)	-0.001 (0.034)	0.014 (0.055)	0.000 (0.034)	0.015 (0.055)	0.002 (0.034)
c_avg_rating		0.120 (0.106)	0.100 (0.082)	0.102 (0.105)	0.089 (0.082)	0.137 (0.120)	0.092 (0.082)	0.123 (0.121)	0.084 (0.083)
c_sd_rating		-0.140 (0.130)	-0.104 (0.104)	-0.141 (0.128)	-0.106 (0.104)	-0.159 (0.145)	-0.113 (0.105)	-0.160 (0.145)	-0.113 (0.105)
c_cum_reviews		0.000 (0.000)	0.000 (0.000)	-0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	-0.000 (0.000)	0.000 (0.000)
c_avg_len_sum		-0.022* (0.012)	-0.015* (0.009)	-0.020* (0.012)	-0.014 (0.009)	-0.028** (0.014)	-0.016* (0.009)	-0.027* (0.014)	-0.016* (0.009)
c_sd_len_sum		0.030** (0.013)	0.022** (0.009)	0.028** (0.013)	0.021** (0.009)	0.034** (0.015)	0.022** (0.009)	0.032** (0.015)	0.021** (0.010)
c_avg_len_rev		0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)
c_sd_len_rev		0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)
c_rating_rec		0.008 (0.021)	0.004 (0.015)	0.008 (0.021)	0.004 (0.015)	0.010 (0.024)	0.004 (0.015)	0.010 (0.024)	0.004 (0.015)

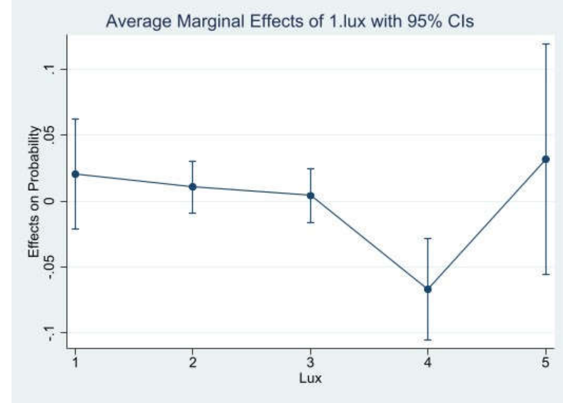
c_len_sum_rec		-0.001 (0.002)	-0.001 (0.001)	-0.001 (0.002)	-0.001 (0.001)	-0.001 (0.002)	-0.001 (0.001)	-0.001 (0.002)	-0.001 (0.001)
c_len_rev_rec		-0.000 (0.000)	-0.000 (0.000)	-0.000 (0.000)	-0.000 (0.000)	-0.000 (0.000)	-0.000 (0.000)	-0.000 (0.000)	-0.000 (0.000)
interval		-0.000 (0.000)	-0.000 (0.000)	-0.000 (0.000)	-0.000 (0.000)	-0.000 (0.000)	-0.000 (0.000)	-0.000 (0.000)	-0.000 (0.000)
nest_avail		0.152 (0.338)	0.106 (0.255)	0.133 (0.333)	0.096 (0.255)	0.134 (0.384)	0.088 (0.255)	0.125 (0.384)	0.084 (0.256)
smart_con		0.699*** (0.147)	0.473*** (0.043)	0.689*** (0.145)	0.473*** (0.043)	0.826*** (0.186)	0.486*** (0.044)	0.824*** (0.187)	0.485*** (0.044)
easy		0.806*** (0.154)	0.580*** (0.028)	0.796*** (0.153)	0.580*** (0.028)	0.937*** (0.195)	0.589*** (0.028)	0.937*** (0.196)	0.589*** (0.028)
save		0.713*** (0.154)	0.494*** (0.051)	0.704*** (0.153)	0.494*** (0.051)	0.858*** (0.198)	0.512*** (0.052)	0.859*** (0.199)	0.513*** (0.052)
func		1.407*** (0.263)	1.030*** (0.028)	1.390*** (0.261)	1.031*** (0.028)	1.621*** (0.330)	1.042*** (0.028)	1.621*** (0.331)	1.042*** (0.028)
support		1.147*** (0.223)	0.793*** (0.051)	1.133*** (0.222)	0.793*** (0.051)	1.326*** (0.278)	0.801*** (0.051)	1.328*** (0.279)	0.801*** (0.051)
price_value		0.675*** (0.139)	0.487*** (0.042)	0.673*** (0.139)	0.491*** (0.042)	0.765*** (0.167)	0.488*** (0.043)	0.769*** (0.169)	0.491*** (0.043)
privacy		1.915*** (0.496)	1.352*** (0.231)	1.889*** (0.491)	1.352*** (0.231)	2.337*** (0.626)	1.431*** (0.234)	2.333*** (0.627)	1.429*** (0.235)
amazon		0.298* (0.162)	0.203* (0.112)	0.296* (0.160)	0.203* (0.112)	0.331* (0.187)	0.195* (0.113)	0.333* (0.187)	0.196* (0.113)
env		0.058 (0.698)	0.079 (0.471)	0.050 (0.689)	0.072 (0.471)	0.170 (0.826)	0.146 (0.484)	0.184 (0.827)	0.151 (0.484)
price				0.001 (0.001)	0.000 (0.001)			0.001 (0.001)	0.000 (0.001)
u_avg_p_dfs				0.000 (0.000)	0.000 (0.000)			0.000 (0.001)	0.000 (0.000)
u_sd_p_dfs				-0.000 (0.001)	-0.000 (0.001)			-0.000 (0.001)	-0.000 (0.001)
u_max_p_dfs				0.000 (0.000)	0.000 (0.000)			0.000 (0.000)	0.000 (0.000)
sum_amz_video						-0.667* (0.385)	-0.388* (0.230)	-0.664* (0.386)	-0.386* (0.231)
sum_appliance						0.227* (0.132)	0.141* (0.078)	0.224* (0.132)	0.140* (0.078)
sum_apps						-1.995* (1.019)	-1.225* (0.632)	-1.998* (1.021)	-1.226* (0.633)
sum_cellphone						-0.051* (0.027)	-0.030* (0.016)	-0.049* (0.027)	-0.029* (0.016)
sum_clothes						-0.091* (0.049)	-0.061** (0.029)	-0.091* (0.049)	-0.061** (0.029)
sum_grocery						-0.043** (0.019)	-0.027** (0.011)	-0.043** (0.019)	-0.026** (0.011)
sum_healthcare						0.055** (0.026)	0.036** (0.015)	0.056** (0.026)	0.036** (0.016)
sum_magazine						-0.367* (0.202)	-0.211* (0.115)	-0.367* (0.202)	-0.212* (0.115)
sum_pet_supp						-0.053* (0.030)	-0.028 (0.018)	-0.052* (0.030)	-0.027 (0.018)
sum_software						-0.052 (0.036)	-0.036* (0.021)	-0.052 (0.036)	-0.036* (0.022)
/cut1	-1.430*** (0.513)	-0.386 (0.873)	-0.310 (0.682)	-0.431 (0.863)	-0.339 (0.683)	-0.538 (0.991)	-0.392 (0.685)	-0.566 (0.992)	-0.405 (0.685)
/cut2	-1.157** (0.513)	0.317 (0.872)	0.191 (0.682)	0.264 (0.862)	0.162 (0.683)	0.274 (0.986)	0.113 (0.685)	0.246 (0.987)	0.100 (0.685)
/cut3	-0.909* (0.513)	1.063 (0.892)	0.728 (0.682)	1.000 (0.882)	0.700 (0.683)	1.133 (1.010)	0.654 (0.685)	1.105 (1.011)	0.641 (0.685)
/cut4	-0.386 (0.513)	2.579*** (0.993)	1.828*** (0.682)	2.497** (0.982)	1.800*** (0.683)	2.875** (1.143)	1.762** (0.685)	2.848** (1.145)	1.749** (0.685)
Z. u_avg_rating		-0.032* (0.019)		-0.032* (0.019)		-0.031 (0.019)		-0.030 (0.019)	

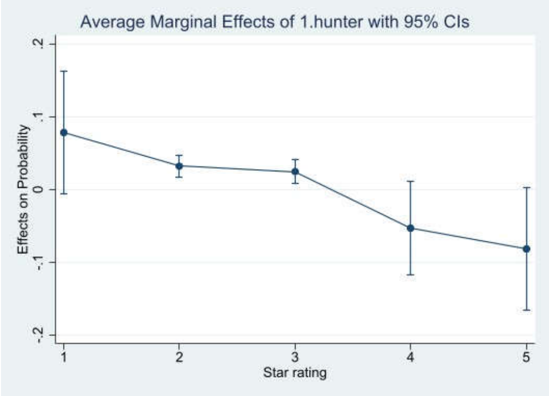
Z. nest		0.544*** (0.174)		0.529*** (0.175)		0.681*** (0.193)		0.679*** (0.194)	
Z. honey		0.398** (0.174)		0.381** (0.175)		0.531*** (0.194)		0.528*** (0.194)	
Z. lux		0.386** (0.174)		0.369** (0.176)		0.491** (0.193)		0.489** (0.194)	
Z. hunter		0.582*** (0.196)		0.569*** (0.197)		0.726*** (0.214)		0.725*** (0.215)	
Z. venstar		0.229 (0.224)		0.200 (0.225)		0.364 (0.240)		0.351 (0.242)	
LR tests, $X^2(6)$		28.69***		28.93***		34.37***		34.48***	
Loglikelihood	-6046.585	-4003.836	-4018.181	-4002.670	-4017.135	-3977.845	-3995.028	-3977.223	-3994.463
AIC	12171.169	8159.673	8176.362	8165.341	8182.270	8169.689	8192.055	8176.446	8198.925
BIC	12427.656	8659.494	8636.724	8691.468	8668.938	8873.385	8856.291	8906.448	8889.468

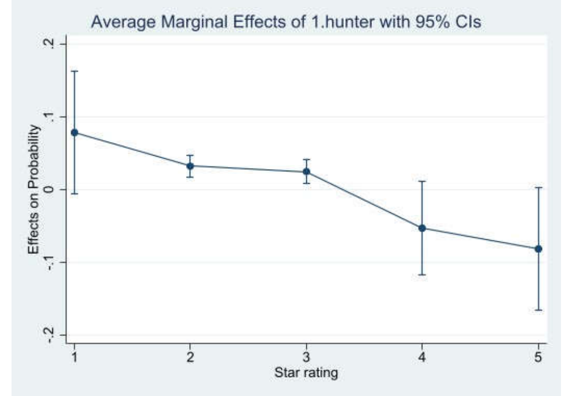
Figure 9. Marginal effect (statistically significant variables or related variables)

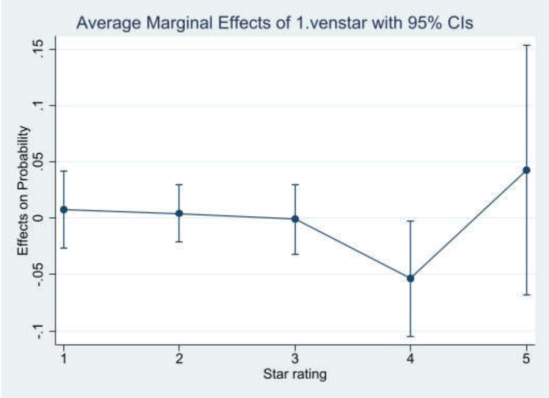


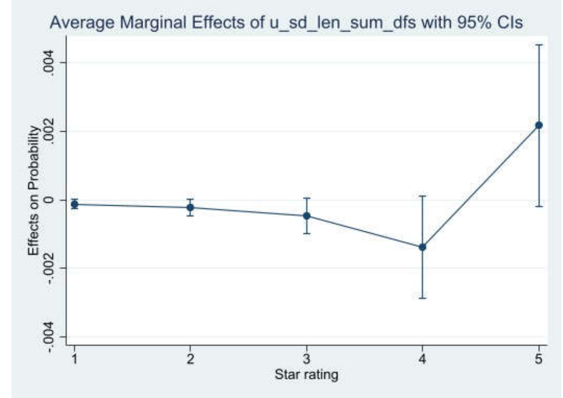

Average Marginal Effects of 1.honey with 95% CIs
Effects on Probability
-.1
-.05
0
.05
.1
1
2
3
4
5
Star rating

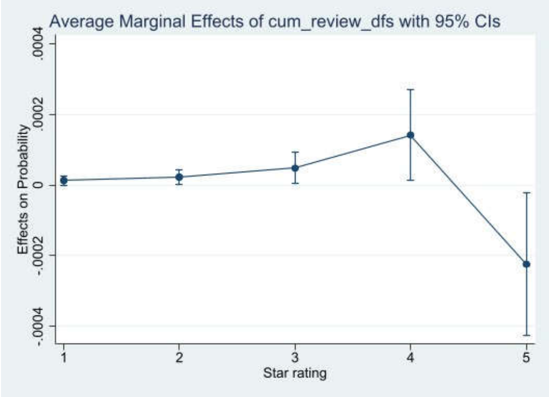

Average Marginal Effects of 1.lux with 95% CIs
Effects on Probability
-.1
-.05
0
.05
.1
1
2
3
4
5
Lux

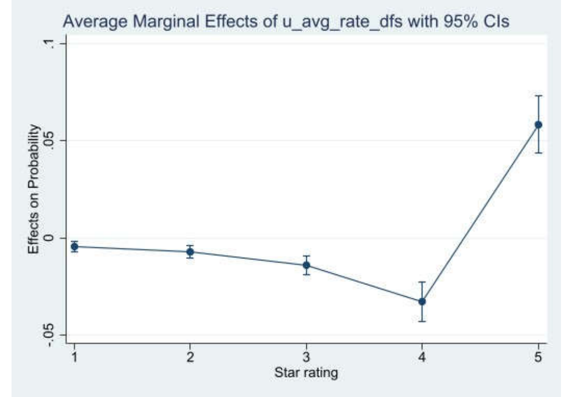

Average Marginal Effects of 1.hunter with 95% CIs
Effects on Probability
-.2
-.1
0
.1
.2
1
2
3
4
5
Star rating

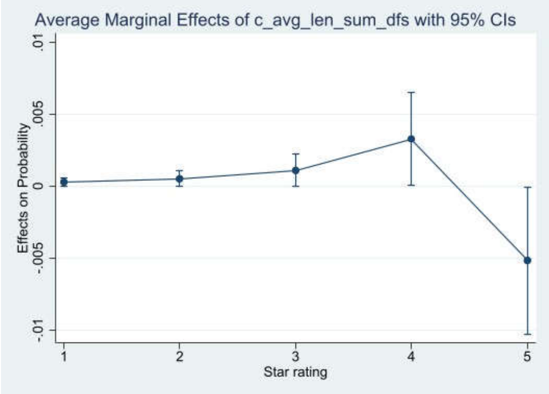

Average Marginal Effects of 1.hunter with 95% CIs
Effects on Probability
-.2
-.1
0
.1
.2
1
2
3
4
5
Star rating

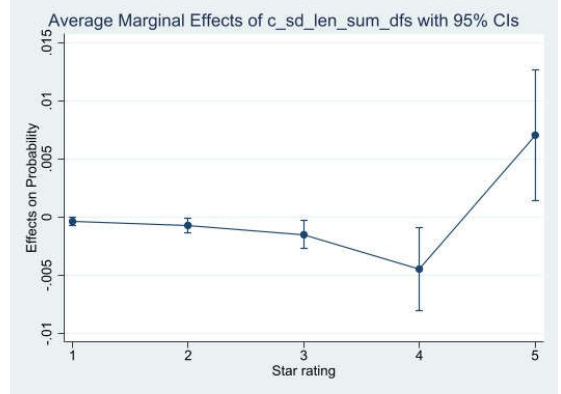

Average Marginal Effects of 1.venstar with 95% CIs
Effects on Probability
-.1
-.05
0
.05
.1
.15
1
2
3
4
5
Star rating

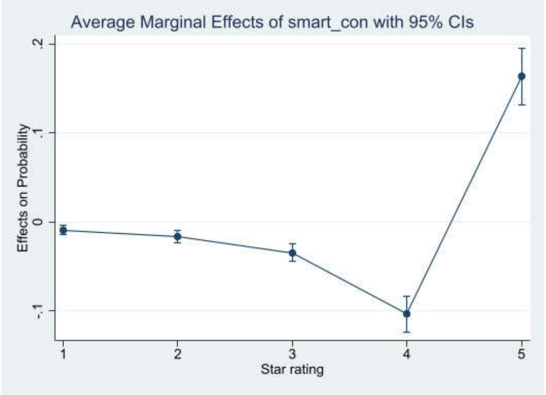

Average Marginal Effects of u_sd_len_sum_dfs with 95% CIs
Effects on Probability
-.004
-.002
0
.002
.004
1
2
3
4
5
Star rating

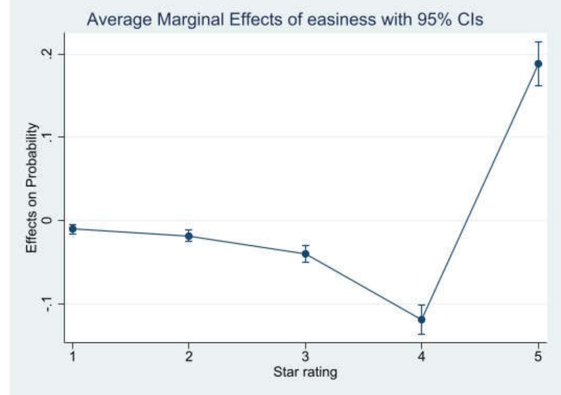

Average Marginal Effects of cum_review_dfs with 95% CIs
Effects on Probability
-.0004
-.0002
0
.0002
.0004
1
2
3
4
5
Star rating

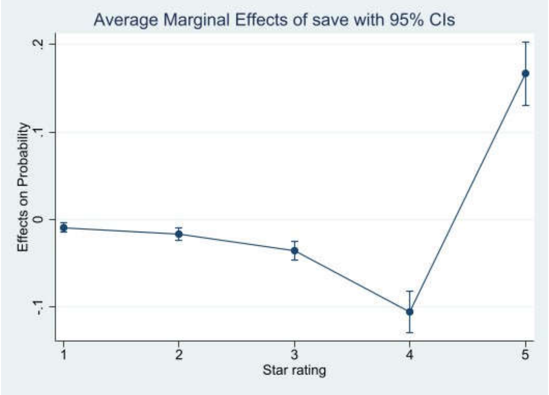

Average Marginal Effects of u_avg_rate_dfs with 95% CIs
Effects on Probability
-.05
0
.05
.1
1
2
3
4
5
Star rating

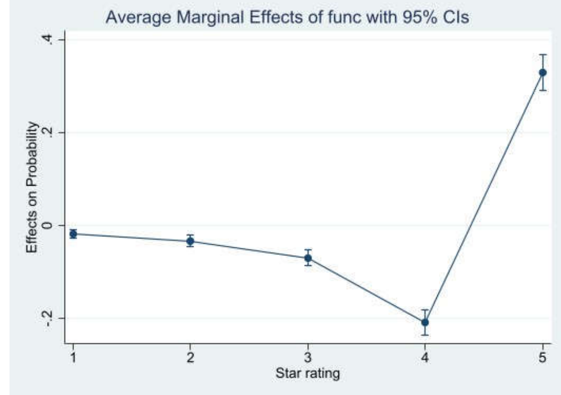

Average Marginal Effects of c_avg_len_sum_dfs with 95% CIs

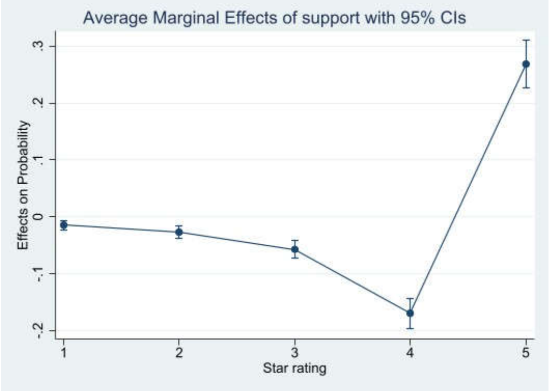

Average Marginal Effects of c_sd_len_sum_dfs with 95% CIs

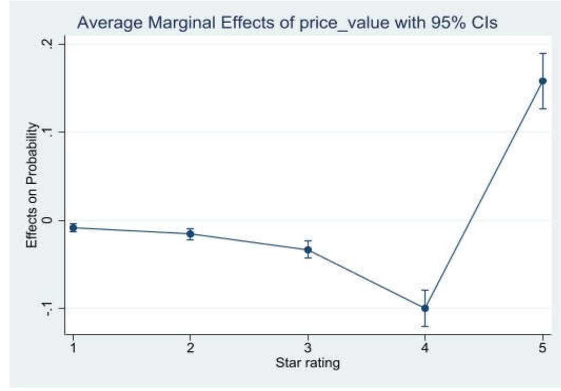

Average Marginal Effects of smart_con with 95% CIs

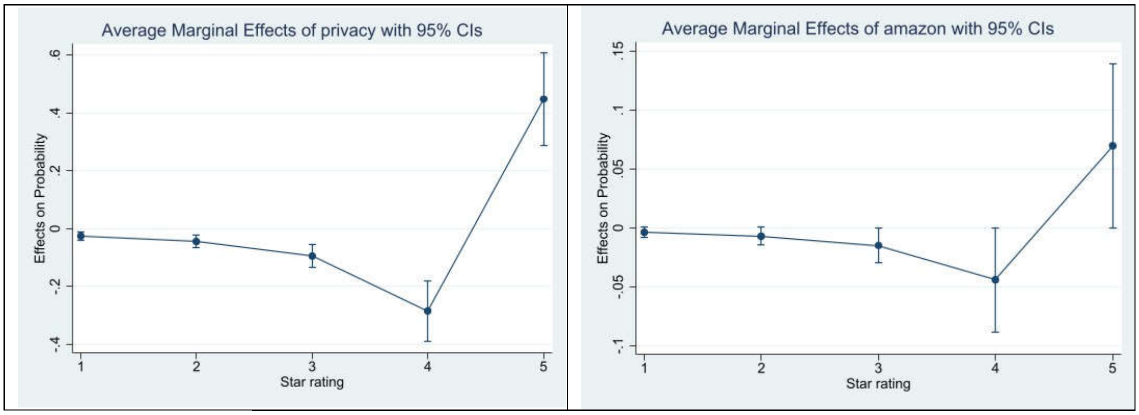

Average Marginal Effects of easiness with 95% CIs


Average Marginal Effects of save with 95% CIs


Average Marginal Effects of func with 95% CIs


Average Marginal Effects of support with 95% CIs


Average Marginal Effects of price_value with 95% CIs



* Note: based on model_h2

Appendix C. Ex Ante Prediction Results

Table 24. Train, valid, and test set rating distribution

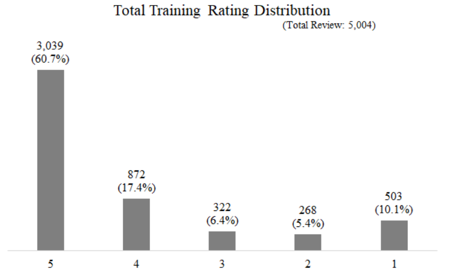
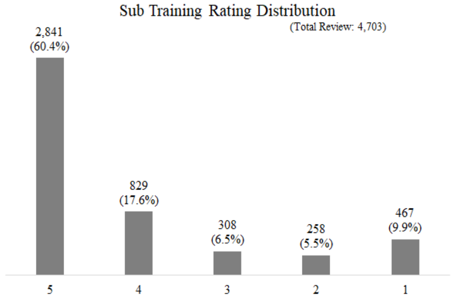
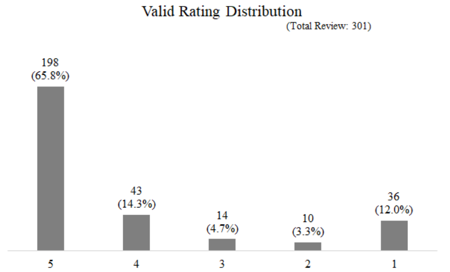
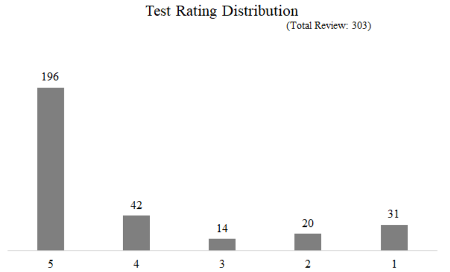
Data set	Sample size	Period	Rating distribution
Total Training	5004	Oct 12, 2005 – May 17, 2014	
Sub Training	4703	Oct 12, 2005 –Mar 16, 2014	
Validation	301	Mar 17, 2014 –May 17 2014	
Test	303	May 18, 2014 – July 17 2014	

Table 25. Ex-ante prediction model 1: At-time (37 variables)

Models	Hyperparameter	Acc*	Precision	Recall	F1-score	Confusion matrix					
Heteroprobit	None	0.647	1: 0.00	1: 0.00	1: 0.00		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	0	0	0	0	31
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.00	4: 0.00	4: 0.00	3	0	0	0	0	14
			5: 0.65	5: 1.00	5: 0.79	4	0	0	0	0	42
			WA: 0.42	WA: 0.65	WA: 0.51	5	0	0	0	0	196
Kernel SVM	Kernel: RGB Gamma:0.006 C:50	0.624	1: 0.09	1: 0.03	1: 0.05		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	1	0	1	0	29
			3: 0.00	3: 0.00	3: 0.00	2	2	0	0	0	18
			4: 0.00	4: 0.00	4: 0.00	3	0	0	0	0	14
			5: 0.65	5: 0.96	5: 0.77	4	0	0	0	0	42
			WA: 0.43	WA:0.62	WA: 0.50	5	8	0	0	0	188
Decision Tree	Criteria: Entropy Max depth: 1	0.647	1: 0.00	1: 0.00	1: 0.00		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	0	0	0	0	31
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.00	4: 0.00	4: 0.00	3	0	0	0	0	14
			5:0.65	5: 1.00	5: 0.79	4	0	0	0	0	42
			WA: 0.42	WA: 0.65	WA: 0.51	5	0	0	0	0	196
Random Forest	Tree numbers: 6 Depth: 10	0.643	1: 0.00	1: 0.00	1: 0.00		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	0	0	0	0	31
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.00	4: 0.00	4: 0.00	3	0	0	0	0	14
			5: 0.65	5: 0.99	5: 0.78	4	0	0	0	0	42
			WA: 0.42	WA: 0.64	WA: 0.51	5	0	0	0	1	195
XGboost	Tree number: 70 Depth: 2 Learning rate: 0.30	0.647	1: 0.00	1: 0.00	1: 0.00		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	0	0	0	0	31
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.00	4: 0.00	4: 0.00	3	0	0	0	0	14
			5: 0.65	5: 1.00	5: 0.79	4	0	0	0	0	42
			WA: 0.42	WA: 0.65	WA: 0.51	5	0	0	0	0	196
ANN	Epoch: 150 Drop out: 0.4 Learning rate: 0.0002 Hidden layer 1 node:148 Hidden layer 2 node:148	0.647	1: 0.00	1: 0.00	1: 0.00		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	0	0	0	0	31
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.00	4: 0.00	4: 0.00	3	0	0	0	0	14
			5: 0.65	5: 1.00	5: 0.79	4	0	0	0	0	42
			WA: 0.42	WA: 0.65	WA: 0.51	5	0	0	0	0	196
LSTM	Epoch: 318 Learning rate: 0.0002 Hidden layer node: 37	0.647	1: 0.00	1: 0.00	1: 0.00		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	0	0	0	0	31
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.00	4: 0.00	4: 0.00	3	0	0	0	0	14
			5: 0.65	5:1.00	5: 0.79	4	0	0	0	0	42
			WA: 0.42	WA: 0.65	WA: 0.51	5	0	0	0	0	196

* WA indicated weighted average macro values

* Horizontal labels from 1 (left) to 5 (right) are the predictive star-rating while vertical labels from 1 (top) to 5 (bottom) are true star-rating, values in diagonal are the number of correct prediction for the star-rating mapped into horizontal or vertical star-rating.

* Acc: accuracy

Table 26. Ex ante prediction model 2: ex ante model (59 variables)

Models	Hyperparameter	Accuracy	Precision	Recall	F1-score	Confusion matrix					
Heteroprobit	None	0.640	1: 0.20	1: 0.03	1: 0.06		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	1	0	0	0	30
			3: 0.00	3: 0.00	3: 0.00	2	1	0	0	0	19
			4: 0.00	4: 0.00	4: 0.00	3	0	0	0	0	14
			5: 0.00	5: 0.98	5: 0.78	4	0	0	0	0	42
			WA: 0.44	WA: 0.64	WA: 0.51	5	.3	0	0	0	193
Kernel SVM	Kernel: RGB Gamma: 0.1 C:5.0	0.644	1: 0.00	1: 0.00	1: 0.00		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	0	0	0	0	31
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.00	4: 0.00	4: 0.00	3	0	0	0	0	14
			5: 0.65	5: 0.99	5: 0.78	4	0	0	0	0	42
			WA:0.42	WA:0.64	WA:0.51	5	1	0	0	0	195
Decision Tree	Criteria: Entropy Max depth: 1	0.647	1: 0.00	1: 0.00	1: 0.00		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	0	0	0	0	31
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.00	4: 0.00	4: 0.00	3	0	0	0	0	14
			5: 0.65	5: 1.00	5: 0.79	4	0	0	0	0	42
			WA: 0.42	WA: 0.65	WA: 0.51	5	0	0	0	0	196
Random Forest	Tree numbers: 21 Depth: 12	0.617	1: 0.54	1:0.23	1:0.32		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	7	0	0	3	21
			3: 0.00	3: 0.00	3: 0.00	2	2	0	0	2	16
			4: 0.00	4: 0.00	4: 0.00	3	0	0	0	0	14
			5: 0.66	5: 0.92	5: 0.77	4	1	1	0	0	40
			WA:0.48	WA:0.62	WA:0.53	5	3	0	0	13	180
Xgboost	Tree number: 80 Depth : 16 Learning rate: 0.4 Class weighted	0.597	1: 0.32	1: 0.32	1: 0.32		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	10	0	0	3	31
			3: 0.00	3: 0.00	3: 0.00	2	5	0	0	5	20
			4: 0.19	4: 0.14	4: 0.16	3	0	0	0	1	14
			5: 0.69	5: 0.84	5: 0.76	4	3	1	0	6	42
			WA:0.51	WA:0.60	WA:**0.55**	5	13	0	2	16	196
ANN	Epoch: 299 Drop out: 0.4 Learning rate: 0.0001 Hidden layer 1 node: 118 Hidden layer 2 node: 118	0.647	1: 0.33	1: 0.06	1: 0.11		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	2	0	0	0	29
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.00	4: 0.00	4: 0.00	3	0	0	0	0	14
			5: 0.65	5: 0.99	5: 0.79	4	2	0	0	0	40
			WA: 0.46	WA: 0.65	WA: 0.52	5	2	0	0	0	194
LSTM	Epoch: 127 Learning rate:0.0002 Hidden layer node: 300	0.604	1: 0.00	1:0.00	1: 0.00		1	2	3	4	5
			2: 0.00	2:0.00	2: 0.00	1	0	0	0	2	29
			3: 0.00	3:0.00	3: 0.00	2	0	0	0	2	18
			4: 0.23	4:0.17	4: 0.19	3	0	0	0	2	12
			5: 0.65	5:0.90	5: 0.76	4	1	0	0	7	34
			WA:0.46	WA:0.60	WA:0.52	5	3	0	0	17	176

* Note: XGboost: Class weight = # num of sample / (number of total class x frequency of each class) for all row level

Table 27. Ex ante prediction model 3: ex ante-sub-model (90 variables)

Models	Hyperparameter	Accuracy	Precision	Recall	F1-score	Confusion matrix					
Heteroprobit	None	0.644	1: 0.33	1: 0.06	1: 0.11		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	2	0	0	0	29
			3: 0.00	3: 0.00	3: 0.00	2	1	0	0	0	19
			4: 0.00	4: 0.00	4: 0.00	3	0	0	0	0	14
			5: 0.65	5: 0.98	5:0.78	4	0	0	0	0	42
			WA: 0.45	WA: 0.64	WA: 0.52	5	3	0	0	0	193
Kernel SVM	Kernel: RGB Gamma: 8 C:0.1	0.647	1: 0.00	1: 0.00	1: 0.00		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	0	0	0	0	31
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.00	4: 0.00	4: 0.00	3	0	0	0	0	14
			5: 0.65	5: 1.00	5: 0.79	4	0	0	0	0	42
			WA: 0.42	WA: 0.65	WA: 0.51	5	0	0	0	0	196
Decision Tree	Criteria: Entropy Max depth: 1	0.647	1: 0.00	1: 0.00	1: 0.00		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	0	0	0	0	31
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.00	4: 0.00	4: 0.00	3	0	0	0	0	14
			5: 0.65	5: 1.00	5: 0.79	4	0	0	0	0	42
			WA: 0.42	WA: 0.65	WA: 0.51	5	0	0	0	0	196
Random Forest	Tree numbers: 46 Depth: 43	0.644	1: 0.67	1: 0.06	1: 0.12		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	2	0	0	2	27
			3: 0.00	3: 0.00	3: 0.00	2	1	0	0	0	19
			4: 0.14	4: 0.02	4: 0.04	3	0	0	0	1	13
			5: 0.66	5: 0.98	5: 0.79	4	0	0	1	1	40
			WA: 0.51	WA: 0.64	WA: 0.53	5	0	0	1	3	192
Xgboost	Tree number: 50 Depth : 3 Learning rate: 0.2	0.644	1: 0.40	1: 0.19	1: 0.26		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	6	1	0	0	24
			3: 0.00	3: 0.00	3: 0.00	2	1	0	0	1	18
			4: 0.50	4: 0.12	4: 0.19	3	0	0	0	0	14
			5: 0.66	5: 0.94	5: 0.78	4	0	0	0	5	37
			WA: 0.54	WA: 0.64	WA: 0.56	5	8	0	0	4	184
ANN	Epoch: 9 Drop out: 0.5 Learning rate: 0.0001 Hidden layer 1 node: 270 Hidden layer 2 node: 270	0.637	1: 0.50	1: 0.03	1: 0.06		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	1	0	0	2	28
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.25	4: 0.07	4: 0.11	3	0	0	0	0	14
			5: 0.65	5: 0.96	5: 0.78	4	1	0	0	3	38
			WA: 0.51	WA: 0.64	WA: 0.53	5	0	0	0	7	189
LSTM	Epoch: 70 Learning rate: 0.0002 Hidden layer node: 70	0.653	1: 0.20	1: 0.03	1: 0.06		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	1	0	0	0	30
			3: 0.00	3: 0.00	3: 0.00	2	1	0	0	0	19
			4: 0.67	4: 0.05	4: 0.09	3	0	0	0	1	13
			5: 0.66	5: 0.99	5: 0.79	4	2	0	0	2	38
			WA: 0.54	WA: 0.65	WA: 0.53	5	1	0	0	0	195

Table 28. Ex ante prediction model 4: ex ante-sub-price model (94 variables)

Models	Hyperparameter	Accuracy	Precision	Recall	F1-score	Confusion matrix					
Heteroprobit	None	0.640	1: 0.20	1: 0.03	1: 0.06		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	1	0	0	0	30
			3: 0.00	3: 0.00	3: 0.00	2	1	0	0	0	19
			4: 0.00	4: 0.00	4: 0.00	3	0	0	0	0	14
			5: 0.65	5: 0.98	5: 0.78	4	0	0	0	0	42
			WA: 0.44	WA: 0.64	WA: 0.51	5	3	0	0	0	193
Kernel SVM	Kernel: RGB Gamma: 1.0 C: 0.1	0.647	1: 0.00	1: 0.00	1: 0.00		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	0	0	0	0	31
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.00	4: 0.00	4: 0.00	3	0	0	0	0	14
			5:0.65	5: 1.00	5: 0.79	4	0	0	0	0	42
			WA: 0.42	WA: 0.65	WA: 0.51	5	0	0	0	0	196
Decision Tree	Criteria: Entropy Max depth: 1	0.647	1: 0.00	1: 0.00	1: 0.00		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	0	0	0	0	31
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.00	4: 0.00	4: 0.00	3	0	0	0	0	14
			5: 0.65	5: 1.00	5: 0.79	4	0	0	0	0	42
			WA: 0.42	WA: 0.65	WA: 0.51	5	0	0	0	0	196
Random Forest	Tree numbers: 40 Depth:47	0.647	1: 0.33	1: 0.10	1: 0.15		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	3	0	0	2	26
			3: 0.00	3: 0.00	3: 0.00	2	1	0	0	0	19
			4: 0.00	4: 0.05	4: 0.08	3	0	0	0	0	14
			5: 0.67	5: 0.97	5: 0.79	4	4	0	0	2	36
			WA: 0.50	WA: 0.65	**WA: 0.54**	5	1	0	0	4	191
Xgboost	Tree number: 50 Depth: 15 Learning rate: 0.1	0.607	1: 0.32	1: 0.26	1: 0.29		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	8	0	0	2	21
			3: 0.00	3: 0.00	3: 0.00	2	4	0	0	3	13
			4: 0.20	4: 0.10	4: 0.13	3	0	0	0	0	14
			5: 0.67	5: 0.88	5: 0.76	4	1	0	0	4	37
			WA: 0.49	WA: 0.61	**WA: 0.54**	5	12	0	1	11	172
ANN	Epoch: 17 Drop out: 0.4 Learning rate: 0.0002 Hidden layer 1 node: 288 Hidden layer 2 node: 188	0.647	1: 0.29	1: 0.06	1: 0.11		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	2	0	0	0	29
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.00	4: 0.00	4: 0.00	3	0	0	0	0	14
			5: 0.66	5: 0.99	5: 0.79	4	4	0	0	0	38
			WA: 0.45	WA: 0.65	WA: 0.52	5	1	0	0	1	194
LSTM	Epoch: 96 Learning rate: 0.0002 Hidden layer node: 282	0.650	1: 0.50	1: 0.13	1: 0.21		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	4	0	0	1	26
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.00	4: 0.00	4: 0.00	3	0	0	0	0	14
			5: 0.66	5: 0.98	5: 0.79	4	1	0	0	0	41
			WA: 0.48	WA: 0.65	WA: 0.53	5	3	0	0	0	193

Table 29. Ex ante prediction model 5: ex ante-sub model (161 variables)

Models	Hyperparameter	Accuracy	Precision	Recall	F1-score	Confusion matrix					
Kernel SVM	Kernel: RGB Gamma: 0.001 C: 1.0	0.644	1: 0.00	1: 0.00	1: 0.00		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	0	0	0	0	31
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.33	4: 0.02	4: 0.04	3	0	0	0	0	14
			5: 0.65	5: 0.99	5: 0.78	4	0	0	0	0	42
			WA: 0.46	WA: 0.64	WA: 0.51	5	0	0	0	0	196
Decision Tree	Criteria: Entropy Max depth: 1	0.647	1: 0.00	1: 0.00	1: 0.00		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	0	0	0	0	31
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.00	4: 0.00	4: 0.00	3	0	0	0	0	14
			5: 0.65	5: 1.00	5: 0.79	4	0	0	0	0	42
			WA: 0.42	WA: 0.65	WA: 0.51	5	0	0	0	0	196
Random Forest	Tree numbers: 9 Depth: 13	0.627	1: 0.30	1: 0.10	1: 0.15		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	3	0	0	1	27
			3: 0.00	3: 0.00	3: 0.00	2	2	0	0	0	18
			4: 0.00	4: 0.00	4: 0.00	3	0	0	0	0	14
			5: 0.66	5: 0.95	5: 0.78	4	1	1	1	0	39
			WA: 0.46	WA: 0.63	WA: 0.52	5	4	0	1	4	187
Xgboost	Tree number: 60 Depth: 14 Learning rate: 0.2 Class weighted	0.620	1: 0.28	1: 0.23	1: 0.25		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	7	0	0	6	18
			3: 0.00	3: 0.00	3: 0.00	2	4	0	0	2	14
			4: 0.34	4: 0.24	4: 0.28	3	0	0	0	0	14
			5: 0.70	5: 0.87	5: 0.78	4	2	2	0	10	28
			WA: 0.53	WA: 0.62	**WA: 0.57**	5	12	1	1	11	171
ANN	Epoch: 146 Drop out: 0.4 Learning rate: 0.0002 Hidden layer 1 node: 483 Hidden layer 2 node: 483	0.650	1: 0.50	1: 0.06	1: 0.11		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	2	0	0	0	29
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.50	4: 0.02	4: 0.05	3	0	0	0	0	14
			5: 0.65	5: 0.99	5: 0.79	4	1	0	0	1	40
			WA: 0.54	WA: 0.65	WA: 0.53	5	1	0	0	1	194
LSTM	Epoch: 127 Learning rate: 0.0002 Hidden layer node: 242	0.640	1: 0.25	1: 0.03	1: 0.06		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	1	0	0	0	30
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.50	4: 0.02	4: 0.00	3	0	0	0	0	14
			5: 0.65	5: 0.98	5: 0.05	4	0	0	0	1	41
			WA: 0.51	WA: 0.64	WA: 0.52	5	3	0	0	1	192

* Heteroskedastic ordered probit is excluded because the model is not compatible with product dummies due to the multicollinearity.

* class : [1, 2, 3, 4, 5], class weight: [2.014, 3.646, 3.054, 1.135, 0.331]

Table 30. Ex ante prediction model 6: ex ante-sub-price model (165 variables)

Models	Hyperparameter	Accuracy	Precision	Recall	F1-score	Confusion matrix					
Kernel SVM	Kernel: RGB Gamma: 0.001 C:1.0	0.644	1: 0.00	1: 0.00	1: 0.00		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	0	0	0	0	31
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.33	4: 0.02	4: 0.04	3	0	0	0	0	14
			5: 0.65	5: 0.99	5: 0.78	4	0	0	0	1	41
			WA: 0.46	WA: 0.64	WA: 0.51	5	0	0	0	0	196
Decision Tree	Criteria: Entropy Max depth: 1	0.647	1: 0.00	1: 0.00	1: 0.00		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	0	0	0	0	31
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.00	4: 0.00	4: 0.00	3	0	0	0	0	14
			5: 0.65	5: 1.00	5: 0.79	4	0	0	0	0	42
			WA: 0.42	WA: 0.65	WA: 0.51	5	0	0	0	0	196
Random Forest	Tree numbers: 30 Depth: 43 Class weighted	0.647	1: 0.40	1: 0.06	1: 0.11		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	2	0	0	1	28
			3: 0.00	3: 0.00	3: 0.00	2	1	0	0	1	18
			4: 0.25	4: 0.02	4: 0.04	3	0	0	0	0	14
			5: 0.66	5: 0.98	5: 0.79	4	0	0	0	1	41
			WA: 0.50	WA: 0.65	WA: 0.53	5	2	0	0	1	193
Xgboost	Tree number: 100 Depth: 14 Learning rate: 0.1 Class weighted	0.637	1: 0.31	1: 0.29	1: 0.30		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	9	1	0	4	17
			3: 0.00	3: 0.00	3: 0.00	2	6	0	0	3	11
			4: 0.32	4: 0.17	4: 0.22	3	0	0	0	1	13
			5: 0.71	5: 0.90	5: 0.79	4	3	0	0	7	32
			WA: 0.53	WA: 0.64	**WA: 0.57**	5	11	0	1	7	177
ANN	Epoch: 97 Drop out: 0.4 Learning rate: 0.0002 Hidden layer 1 node: 330 Hidden layer 2 node: 165	0.637	1: 0.33	1: 0.03	1: 0.06		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	1	0	1	1	28
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.20	4: 0.05	4: 0.08	3	0	0	0	1	13
			5: 0.66	5: 0.97	5: 0.78	4	2	0	0	2	38
			WA: 0.49	WA: 0.64	WA: 0.52	5	0	0	0	6	190
LSTM	Epoch: Learning rate: Hidden layer node:	0.640	1: 0.30	1: 0.10	1: 0.15		1	2	3	4	5
			2: 0.00	2: 0.00	2: 0.00	1	3	0	0	0	28
			3: 0.00	3: 0.00	3: 0.00	2	0	0	0	0	20
			4: 0.40	4: 0.10	4: 0.15	3	0	0	0	2	12
			5: 0.66	5: 0.95	5: 0.78	4	2	0	0	4	36
			WA: 0.51	WA: 0.64	WA: 0.54	5	5	0	0	4	187

Table 31. Three class classification: ex ante prediction

Models	Hyperparameter	Accuracy	Precision	Recall	F1-score	Confusion matrix			
Heteropobit	3 classes	0.789	1: 0.56	1: 0.10	1: 0.17		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	5	0	46
			3: 0.80	3: 0.98	3: 0.88	2	0	0	14
			WA: 0.72	WA: 0.79	WA: 0.72		4	0	234
Kernel SVM	Kernel: RGB Gamma: 1.0 C: 0.1	0.785	1: 0.00	1: 0.00	1: 0.00		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	0	0	51
			3: 0.79	3: 1.00	3: 0.88	2	0	0	14
			WA: 0.62	WA: 0.79	WA: 0.69		0	0	238
Decision Tree	Criteria: entropy Max depth: 4	0.779	1: 0.25	1: 0.02	1: 0.04		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	1	0	50
			3: 0.79	3: 0.99	3: 0.88	2	0	0	14
			WA: 0.66	WA: 0.78	WA: 0.69	3	3	0	235
Random Forest	Tree numbers: 16 Depth: 42	0.802	1: 0.73	1: 0.16	1: 0.26		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	6	2	43
			3: 0.80	3: 0.99	3: 0.89	2	0	0	14
			WA: 0.75	WA: 0.80	**WA: 0.74**	3	5	0	233
Xgboost	Tree number: 100 Depth: 4 Learning rate:0.2	0.802	1: 0.78	1: 0.14	1: 0.23		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	7	0	44
			3: 0.80	3: 0.99	3: 0.89	2	0	0	14
			WA: 0.76	WA: 0.80	**WA: 0.74**		2	0	236
ANN	Epoch: 3 Drop out: 0.4 Learning rate: 0.0002 Hidden layer 1 node:180 Hidden layer 2 node:180	0.782	1: 0.38	1: 0.06	1: 0.10		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	3	0	48
			3: 0.79	3: 0.98	3: 0.88	2	1	0	13
			WA: 0.69	WA: 0.78	WA: 0.71		4	0	234
LSTM	Epoch: 232 Learning rate: 0.0002 Hidden layer node: 322	0.700	1: 0.50	1: 0.02	1: 0.04		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	1	0	50
			3: 0.79	3: 1.00	3: 0.88	2	0	0	14
			WA: 0.70	WA: 0.79	WA: 0.70		1	0	237

Table 32. Three class classification: partial ex ante prediction

Models	Hyperparameter	Accuracy	Precision	Recall	F1-score	Confusion matrix			
Kernel SVM	Kernel: RGB Gamma: 1.0 C: 0.1	0.785	1: 0.00	1: 0.00	1: 0.00		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	0	0	51
			3: 0.79	3: 1.00	3: 0.88	2	0	0	14
			WA: 0.62	WA: 0.79	WA: 0.69	3	0	0	238
Decision Tree	Criteria: Entropy Max depth: 4	0.779	1: 0.25	1: 0.02	1: 0.04		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	1	0	50
			3: 0.79	3: 0.99	3: 0.88	2	0	0	14
			WA: 0.66	WA: 0.78	WA: 0.69	3	3	0	235
Random Forest	Tree numbers: 11 Depth: 14	0.789	1: 0.57	1: 0.08	1: 0.14		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	4	0	47
			3: 0.79	3: 0.99	3: 0.88	2	0	0	14
			WA: 0.72	WA: 0.79	WA: 0.71	3	3	0	235
Xgboost	Tree number: 90 Depth: 2 Learning rate:0.1 Class weighted	0.759	1: 0.42	1: 0.41	1: 0.42		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	21	0	30
			3: 0.83	3: 0.88	3: 0.85	2	0	0	14
			WA: 0.72	WA: 0.76	**WA: 0.74**	3	29	0	209
ANN	Epoch: 196 Drop out: 0.3 Learning rate: 0.0002 Hidden layer 1 node: 322 Hidden layer 2 node:322	0.792	1: 0.75	1: 0.06	1: 0.11		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	3	0	48
			3: 0.79	3: 1.00	3: 0.88	2	0	0	14
			WA: 0.75	WA: 0.79	WA: 0.71	3	1	0	237
LSTM	Epoch: 127 Learning rate: 0.0002 Hidden layer node: 242	0.789	1: 1.00	1: 0.02	1: 0.04		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	1	0	50
			3: 0.79	3: 1.00	3: 0.88	2	0	0	14
			WA: 0.79	WA: 0.79	WA: 0.70	3	0	0	238

Table 33. Binary classification for ex ante prediction

<table>
<tr><th>Models</th><th>Hyperparameter</th><th>Accuracy</th><th>Precision</th><th>Recall</th><th>F1-score</th><th>Confusion matrix</th></tr>
<tr><td>HETOP</td><td>Binary classification</td><td>0.782</td><td>1: 0.47
2: 0.80
WA: 0.73</td><td>1: 0.11
2: 0.97
WA: 0.78</td><td>1: 0.18
2: 0.87
WA: 0.72</td><td><table><tr><td></td><td>1</td><td>2</td></tr><tr><td>1</td><td>7</td><td>58</td></tr><tr><td>2</td><td>8</td><td>230</td></tr></table></td></tr>
<tr><td>Kernel SVM</td><td>Kernel: RGB
Gamma: 0.01
C: 10</td><td>0.752</td><td>1: 0.29
2: 0.79
WA: 0.68</td><td>1: 0.11
2: 0.93
WA: 0.75</td><td>1: 0.16
2: 0.88
WA: 0.71</td><td><table><tr><td></td><td>1</td><td>2</td></tr><tr><td>1</td><td>7</td><td>58</td></tr><tr><td>2</td><td>17</td><td>221</td></tr></table></td></tr>
<tr><td>Decision Tree</td><td>Criteria: Entropy
Max depth: 3</td><td>0.792</td><td>1: 0.60
2: 0.80
WA: 0.76</td><td>1: 0.09
2: 0.98
WA: 0.79</td><td>1: 0.16
2: 0.88
WA: 0.73</td><td><table><tr><td></td><td>1</td><td>2</td></tr><tr><td>1</td><td>6</td><td>59</td></tr><tr><td>2</td><td>4</td><td>234</td></tr></table></td></tr>
<tr><td>Random Forest</td><td>Tree numbers: 31
Depth: 38</td><td>0.799</td><td>1: 0.64
2: 0.81
WA: 0.77</td><td>1: 0.14
2: 0.98
WA: 0.80</td><td>1: 0.23
2: 0.88
WA: 0.74</td><td><table><tr><td></td><td>1</td><td>2</td></tr><tr><td>1</td><td>9</td><td>56</td></tr><tr><td>2</td><td>5</td><td>233</td></tr></table></td></tr>
<tr><td>Xgboost</td><td>Tree number: 100
Depth: 3
Learning rate: 0.1</td><td>0.782</td><td>1: 0.48
2: 0.80
WA: 0.73</td><td>1: 0.15
2: 0.95
WA: 0.78</td><td>1: 0.23
2: 0.87
WA: 0.74</td><td><table><tr><td></td><td>1</td><td>2</td></tr><tr><td>1</td><td>10</td><td>55</td></tr><tr><td>2</td><td>11</td><td>227</td></tr></table></td></tr>
<tr><td>ANN</td><td>Epoch: 146
Drop out: 0.4
Learning rate: 0.0002
Hidden layer 1 node: 270
Hidden layer 2 node: 270</td><td>0.795</td><td>1: 0.71
2: 0.80
WA: 0.78</td><td>1: 0.08
2: 0.99
WA: 0.80</td><td>1: 0.14
2: 0.88
WA: 0.72</td><td><table><tr><td></td><td>1</td><td>2</td></tr><tr><td>1</td><td>5</td><td>60</td></tr><tr><td>2</td><td>2</td><td>236</td></tr></table></td></tr>
<tr><td>LSTM</td><td>Epoch: 8
Learning rate: 0.0002
Hidden layer node: 135</td><td>0.782</td><td>1: 0.47
2: 0.80
WA: 0.73</td><td>1: 0.12
2: 0.96
WA: 0.78</td><td>1: 0.20
2: 0.87
WA: 0.73</td><td><table><tr><td></td><td>1</td><td>2</td></tr><tr><td>1</td><td>8</td><td>57</td></tr><tr><td>2</td><td>9</td><td>229</td></tr></table></td></tr>
</table>

Table 34. Binary classification for partial ex ante prediction

<table>
<tr><th>Models</th><th>Hyperparameter</th><th>Accuracy</th><th>Precision</th><th>Recall</th><th>F1-score</th><th>Confusion matrix</th></tr>
<tr><td>Kernel SVM</td><td>Kernel: RGB
Gamma: 0.1
C: 1.0</td><td>0.785</td><td>1: 0.00
2: 0.79
WA: 0.62</td><td>1: 0.00
2: 1.00
WA: 0.79</td><td>1: 0.00
2: 0.88
WA: 0.69</td><td><table><tr><th></th><th>1</th><th>2</th></tr><tr><td>1</td><td>0</td><td>65</td></tr><tr><td>2</td><td>0</td><td>238</td></tr></table></td></tr>
<tr><td>Decision Tree</td><td>Criteria: Entropy
Max depth: 3</td><td>0.792</td><td>1: 0.60
2: 0.80
WA: 0.76</td><td>1: 0.09
2: 0.98
WA: 0.79</td><td>1: 0.16
2: 0.88
<b>WA: 0.73</b></td><td><table><tr><th></th><th>1</th><th>2</th></tr><tr><td>1</td><td>6</td><td>59</td></tr><tr><td>2</td><td>4</td><td>234</td></tr></table></td></tr>
<tr><td>Random Forest</td><td>Tree numbers: 22
Depth: 18
Class balanced</td><td>0.789</td><td>1: 0.60
2: 0.79
WA: 0.75</td><td>1: 0.05
2: 0.99
WA: 0.79</td><td>1: 0.09
2: 0.88
WA: 0.71</td><td><table><tr><th></th><th>1</th><th>2</th></tr><tr><td>1</td><td>3</td><td>62</td></tr><tr><td>2</td><td>2</td><td>236</td></tr></table></td></tr>
<tr><td>Xgboost</td><td>Tree number: 70
Depth: 15
Learning rate: 0.4
Class weighted</td><td>0.779</td><td>1: 0.45
2: 0.80
WA: 0.73</td><td>1: 0.15
2: 0.95
WA: 0.78</td><td>1: 0.23
2: 0.87
<b>WA: 0.73</b></td><td><table><tr><th></th><th>1</th><th>2</th></tr><tr><td>1</td><td>10</td><td>55</td></tr><tr><td>2</td><td>12</td><td>226</td></tr></table></td></tr>
<tr><td>ANN</td><td>Epoch: 154
Drop out: 0.4
Learning rate: 0.0002
Hidden layer 1 node: 322
Hidden layer 2 node:161</td><td>0.792</td><td>1: 0.75
2: 0.79
WA: 0.78</td><td>1: 0.05
2: 1.00
WA: 0.79</td><td>1: 0.09
2: 0.88
WA: 0.71</td><td><table><tr><th></th><th>1</th><th>2</th></tr><tr><td>1</td><td>3</td><td>62</td></tr><tr><td>2</td><td>1</td><td>237</td></tr></table></td></tr>
<tr><td>LSTM</td><td>Epoch: 127
Learning rate: 0.0002
Hidden layer node: 242</td><td>0.756</td><td>1: 0.30
2: 0.79
WA: 0.69</td><td>1: 0.11
2: 0.93
WA: 0.76</td><td>1: 0.16
2: 0.86
WA: 0.71</td><td><table><tr><th></th><th>1</th><th>2</th></tr><tr><td>1</td><td>7</td><td>58</td></tr><tr><td>2</td><td>16</td><td>222</td></tr></table></td></tr>
</table>

Appendix D. Sentiment analysis

Table 35. Partial model for sentiment classification

<table>
<tr><th>Models</th><th>Word Embedding</th><th>Hyperparameter</th><th>Accuracy</th><th>Precision</th><th>Recall</th><th>F1-score</th><th>Confusion matrix</th></tr>
<tr><td>Random Forest</td><td>TD-IDF</td><td>Tree numbers: 29
Depth: 31</td><td>0.637</td><td>1: 0.79
2: 0.67
3: 0.58
WA:0.66</td><td>1: 0.30
2: 0.75
3: 0.74
WA:0.64</td><td>1: 0.43
2: 0.71
3: 0.65
WA:0.62</td><td><table><tr><td></td><td>1</td><td>2</td><td>3</td></tr><tr><td>1</td><td>22</td><td>18</td><td>34</td></tr><tr><td>2</td><td>1</td><td>86</td><td>27</td></tr><tr><td>3</td><td>5</td><td>25</td><td>85</td></tr></table></td></tr>
<tr><td>Xgboost</td><td>TD-IDF</td><td>Tree number: 90
Depth: 8
Learning rate:0.4</td><td>0.650</td><td>1: 0.73
2: 0.64
3: 0.62
WA: 0.65</td><td>1: 0.61
2: 0.67
3: 0.66
WA: 0.65</td><td>1: 0.66
2: 0.66
3: 0.64
WA: 65</td><td><table><tr><td></td><td>1</td><td>2</td><td>3</td></tr><tr><td>1</td><td>45</td><td>11</td><td>18</td></tr><tr><td>2</td><td>9</td><td>76</td><td>29</td></tr><tr><td>3</td><td>8</td><td>31</td><td>76</td></tr></table></td></tr>
<tr><td>CNN</td><td>W2V*</td><td>Max length: 1800
Epoch: 61
Number of filter: 200
Filter size: (3,4,5)
Dropout: 0.7
Learning rate: 0.0001</td><td>0.673</td><td>1: 0.74
2: 0.66
3: 0.66
WA:0.68</td><td>1: 0.57
2: 0.75
3: 0.66
WA:0.67</td><td>1: 0.64
2: 0.70
3: 0.66
WA:0.67</td><td><table><tr><td></td><td>1</td><td>2</td><td>3</td></tr><tr><td>1</td><td>42</td><td>16</td><td>16</td></tr><tr><td>2</td><td>5</td><td>86</td><td>23</td></tr><tr><td>3</td><td>10</td><td>29</td><td>76</td></tr></table></td></tr>
<tr><td>CNN</td><td>BERT (Fining tunning)</td><td>Max length: 512
Epoch 16
Number of filters: 200
Filter sizes: (3,4,5)
Dropout: 0.7
Learning rate: 0.00001</td><td>0.719</td><td>1: 0.85
2: 0.65
3: 0.75
WA:0.74</td><td>1: 0.64
2: 0.84
3: 0.65
WA:0.72</td><td>1: 0.73
2: 0.73
3: 0.70
<b>WA:0.72</b></td><td><table><tr><td></td><td>1</td><td>2</td><td>3</td></tr><tr><td>1</td><td>47</td><td>18</td><td>9</td></tr><tr><td>2</td><td>2</td><td>96</td><td>16</td></tr><tr><td>3</td><td>6</td><td>34</td><td>75</td></tr></table></td></tr>
<tr><td>CNN</td><td>BERT further training with OV (5epochs)</td><td>Max length: 512
Epoch: 15
Number of filters:
Filter sizes: (3,4,5)
Dropout: 0.7
Learning rate: 0.0001</td><td>0.710</td><td>1: 0.75
2: 0.69
3: 0.71
WA:0.71</td><td>1: 0.58
2: 0.82
3: 0.68
WA:0.71</td><td>1: 0.66
2: 0.75
3: 0.69
WA:.71</td><td><table><tr><td></td><td>1</td><td>2</td><td>3</td></tr><tr><td>1</td><td>43</td><td>14</td><td>17</td></tr><tr><td>2</td><td>5</td><td>94</td><td>15</td></tr><tr><td>3</td><td>9</td><td>28</td><td>78</td></tr></table></td></tr>
<tr><td>CNN</td><td>BERT further training with OV (1epochs)</td><td>Max length: 512
Epoch: 13
Number of filters:300
Filter sizes: (2,3,4)
Dropout:0.7
Learning rate:0.0001</td><td>0.719</td><td>1: 0.77
2: 0.68
3: 0.75
WA:0.73</td><td>1: 0.68
2: 0.82
3: 0.64
WA:0.71</td><td>1: 0.72
2: 0.74
3: 0.69
<b>WA:0.72</b></td><td><table><tr><td></td><td>1</td><td>2</td><td>3</td></tr><tr><td>1</td><td>50</td><td>14</td><td>10</td></tr><tr><td>2</td><td>5</td><td>94</td><td>15</td></tr><tr><td>3</td><td>10</td><td>31</td><td>74</td></tr></table></td></tr>
</table>

* class : [-1, 0, 1], class weight: [1.302, 1.026, 0.795], W2V: optimal dimension size is 100 and the optimal model is trained on W2V_L.

Table 36. Full model for sentiment classification

<table>
<tr><th>Models</th><th>Word Embedding</th><th>Hyperparameter</th><th>Accuracy</th><th>Precision</th><th>Recall</th><th>F1-score</th><th>Confusion matrix</th></tr>
<tr><td>Random Forest</td><td>TD-IDF</td><td>Tree numbers: 29
Depth: 26</td><td>0.644</td><td>1: 0.76
2: 0.65
3: 0.59
WA: 0.66</td><td>1: 0.53
2: 0.67
3: 0.70
WA: 0.64</td><td>1: 0.62
2: 0.66
3: 0.64
WA:0.64</td><td><table><tr><th></th><th>1</th><th>2</th><th>3</th></tr><tr><td>1</td><td>39</td><td>12</td><td>23</td></tr><tr><td>2</td><td>6</td><td>76</td><td>32</td></tr><tr><td>3</td><td>6</td><td>29</td><td>80</td></tr></table></td></tr>
<tr><td>Xgboost</td><td>TD-IDF</td><td>Tree number: 100
Depth: 7
Learning rate: 0.2
Class weighted</td><td>0.723</td><td>1: 0.84
2: 0.74
3: 0.67
WA:0.74</td><td>1: 0.80
2: 0.66
3: 0.77
WA: 0.73</td><td>1: 0.82
2: 0.69
3: 0.72
WA: 0.73</td><td><table><tr><th></th><th>1</th><th>2</th><th>3</th></tr><tr><td>1</td><td>59</td><td>5</td><td>10</td></tr><tr><td>2</td><td>6</td><td>75</td><td>33</td></tr><tr><td>3</td><td>5</td><td>22</td><td>88</td></tr></table></td></tr>
<tr><td>CNN</td><td>W2V*</td><td>Max length = 1800
Epoch: 22
Number of filter:200
Filter size = (3,4,5)
Dropout = 0.7
Learning rate = 0.0001</td><td>0.686</td><td>1: 0.81
2: 0.65
3: 0.65
WA:0.70</td><td>1: 0.74
2: 0.62
3: 0.71
WA:0.69</td><td>1: 0.77
2: 0.64
3: 0.68
WA: 0.69</td><td><table><tr><th></th><th>1</th><th>2</th><th>3</th></tr><tr><td>1</td><td>55</td><td>12</td><td>7</td></tr><tr><td>2</td><td>6</td><td>71</td><td>37</td></tr><tr><td>3</td><td>7</td><td>26</td><td>82</td></tr></table></td></tr>
<tr><td>CNN</td><td>BERT</td><td>Max length: 512
Epoch: 15
Number of filters: 200
Filter sizes: (2,3,4)
Dropout: 0.7
Learning rate: 0.00001
Class weighted</td><td>0.729</td><td>1: 0.94
2: 0.63
3: 0.78
WA:0.76</td><td>1: 0.62
2: 0.58
3: 0.68
WA:0.73</td><td>1: 0.75
2: 0.72
3: 0.73
WA:0.73</td><td><table><tr><th></th><th>1</th><th>2</th><th>3</th></tr><tr><td>1</td><td>46</td><td>21</td><td>7</td></tr><tr><td>2</td><td>2</td><td>97</td><td>15</td></tr><tr><td>3</td><td>1</td><td>36</td><td>78</td></tr></table></td></tr>
<tr><td>CNN</td><td>BERT further training with OV (5epoch)</td><td>Max length: 512
Epoch: 49
Number of filters:200
Filter sizes: (2,3,4)
Dropout: 0.6
Learning rate: 0.00001</td><td>0.713</td><td>1: 0.88
2: 0.64
3: 0.74
WA:0.73</td><td>1: 0.68
2: 0.84
3: 0.61
WA:0.71</td><td>1: 0.76
2: 0.72
3: 0.67
WA: 0.71</td><td><table><tr><th></th><th>1</th><th>2</th><th>3</th></tr><tr><td>1</td><td>50</td><td>14</td><td>10</td></tr><tr><td>2</td><td>3</td><td>96</td><td>15</td></tr><tr><td>3</td><td>4</td><td>41</td><td>70</td></tr></table></td></tr>
<tr><td>CNN</td><td>BERT further training with TH (1epoch)</td><td>Max length: 512
Epoch: 11
Number of filters:300
Filter sizes: (3,4,5)
Dropout:0.7
Learning rate:0.0001</td><td>0.719</td><td>1: 0.73
2: 0.69
3: 0.75
WA:0.72</td><td>1: 0.77
2: 0.72
3: 0.69
WA:0.72</td><td>1: 0.75
2: 0.70
3: 0.71
WA:0.72</td><td><table><tr><th></th><th>1</th><th>2</th><th>3</th></tr><tr><td>1</td><td>57</td><td>11</td><td>6</td></tr><tr><td>2</td><td>11</td><td>82</td><td>21</td></tr><tr><td>3</td><td>10</td><td>26</td><td>79</td></tr></table></td></tr>
</table>

W2V: optimal dimension size is 100 and the optimal model is trained on W2V_L.

Appendix F. Data Preprocessing

Overall, data-pre-processing follows:

Step. 1: Selects the reviews about the 110 PTs contain no missing values.

Step. 2: Cleaning suspicious reviewers (1,165) and biased (69) reviewers.

Step. 3: Deleting 1,965 (had no DFs) and 91 (wrote a review about the PTs that had no DFs) reviewers n the closest day before writing the first review for PTs.

Step. 4: Selecting the top six brands (contained reviews more than 50) among 26 brands.

Step. 5: Identifying five latent topics in each review based on LDA.

Step. 6: Modifying PCDs by leverage the domain knowledge of expert.

The detail procedure for each step are follows:

Step. 1: Selects the reviews about the 110 PTs contain no missing values.

Programmable thermostats belong in the "Tools and Home Improvement" category. Clarifying a specific product group, programmable thermostats, based only on the category, may lead to noisy samples or missing samples. Therefore, the set of programable thermostats is carefully defined by the following processes:

1. Selecting the category to which the product belongs from the following list

A. [['Tools & Home Improvement', 'Building Supplies', 'Heating & Cooling', 'Thermostats & Accessories', 'Thermostats', 'Programmable']]

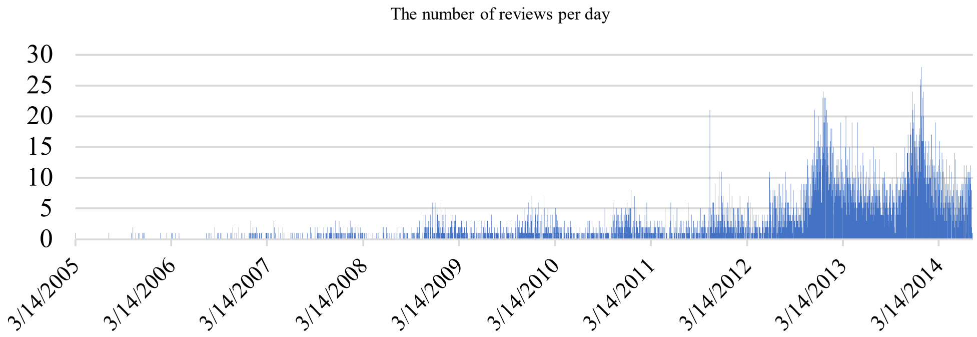
7. Removing the products that contain “non-programable” in the title
8. Selecting the products that contain “programmable” in the product description
9. Removing the products that contain “non-programable, non programmable, programable no” in the product description
10. Removing the products that have a missing value in brand or price variables
11. Evaluating the image of each product selected in step 5 to verify the robustness of the product set.

The set of PTs without missing values in either brand or price variables will be called “programmable thermostats” in the rest of this paper; there are110 thermostats in this set. Although the total number of reviews of the 110 PTs was 8,817, the total number of reviewers was 8,694, because some reviewers wrote more than one review. Six reviewers each posted 3 reviews (0.03% of reviewers), while 111 (1.28%) reviewers posted 2 reviews in the given period. This research aims to estimate unexperienced consumers' preference on PTs using unexperienced consumers’ digital footprint on other products and experienced consumer reviews on PTs. The importance of review content may be higher for inexperienced consumers due to their uncertainty about the new product. To restricts the review of the inexperienced consumer as their first review on the PTs, the second and third reviews from the same reviewer on PTs were deleted.

Six reviewers wrote three reviews for PTs, 111 reviewers writing two reviews. Therefore, only the first review of each more than one-time reviewers' review was removed. In addition, the number of reviewers writing more than one-time is 1.34% among 8694 reviewers in the entire sample period. This point indicates that this product is not repeatedly reviewed well as like book reviews. Therefore, the total number of reviews on programmable thermostats used in this research is 8,694, the same as the number of reviewers.

These 8,694 reviewers produced a total of 225,184 reviews (including the 8,817 reviews for programmable thermostats) for 155,704 unique products from 16,480 unique brands, found across all of Amazon's product categories. The reviews for all products were posted from August 14, 1997 to July 23, 2014, while the reviews for programmable thermostats were posted from March 14, 2005 to July 21, 2014.

Figure 10. Daily number of reviews of PTs without missing price or brand values



As can be seen in Figure 10, the reviews of programmable thermostats started on March 14, 2005 and ran through July 21, 2014. The daily number of reviews on programmable thermostats showed seasonal patterns. The highest number—31 reviews on a single day—occurred on October 24, 2011

Table 37. The number of reviews of programmable thermostats (without missing price or brand values) by brand (Only first review)

Brand	The number of reviews	Average rating	The number of products	Review Period
Nest	3163(36.38%)	3.978185 (1.524157)	2	2012-02-22 2014-07-16
Honeywell	2849(32.77%)	4.190593 (1.258888)	42	2005-03-14 2014-07-18
Lux	1906(21.92%)	4.321616 (1.113470)	9	2007-01-27 2014-07-12
Hunter Fan Company	291(3.35%)	3.900344 (1.457572)	6	2005-07-19 2014-07-09
Venstar	136(1.56%)	4.154412 (1.343730)	5	2012-04-03 2014-07-16
White-Rodgers	96(1.10%)	3.916667 (1.484422)	13	2008-01-09 2014-07-11
Robertshaw	46(0.53%)	3.760870 (1.566096)	7	2008-12-28 2014-06-16
Jensen	34(0.39%)	3.529412 (1.709781)	1	2005-11-26 2013-08-17
King Electric	30(0.35%)	4.000000 (1.339068)	1	2011-01-02 2014-05-04
2gig	30(0.35%)	3.966667 (1.401559)	1	2011-07-18 2014-04-12o
Aprilaire	23(0.26%)	4.347826 (1.368765)	2	2010-12-07 2014-07-08
LuxPro	14(0.16%)	4.428571 (1.222500)	1	2008-09-29 2013-09-06
RiteTemp	11(0.13%)	2.909091 (1.640399)	1	2009-05-06 2014-07-12
Filtrete	11(0.13%)	3.636364 (1.629278)	1	2012-05-10 2014-05-29
Ecobee	9(0.10%)	3.0 (2.0)	1	2012-05-18 2013-07-30
Hunter	7(0.08%)	4.857143 (0.377964)	1	2013-01-09 2014-02-22
Cadet	7(0.08%)	2.571429 (1.511858)	3	2009-06-22 2013-11-07
Garrison	7(0.08%)	4.142857 (1.069045)	1	2010-11-11 2013-12-09

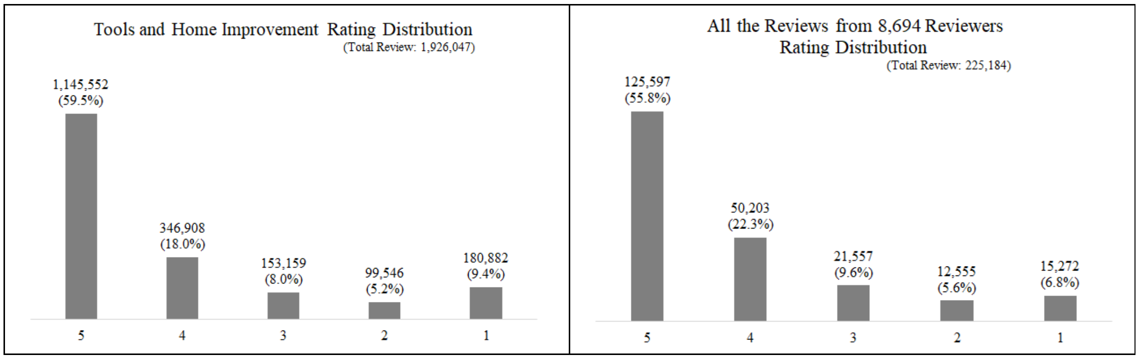
Easy Heat	4(0.05%)	1.75 (1.50)	1	2013-01-25 2014-02-27
Radio Thermostat Company of America	3(0.03%)	2.666667 (2.081666)	1	2014-07-11 2014-07-21
Econo-Heat	3(0.03%)	3.666667 (2.309401)	1	2013-12-11 2014-01-07
Ideal-Air	3(0.03%)	5.0 (0)	1	2011-03-28 2014-03-03
Braeburn	2(0.02%)	5.0 (0)	1	2013-02-06 2014-02-17
Rheem	2(0.02%)	4.0 (0)	1	2013-01-29 2013-11-29
Nuheat	2(0.02%)	4.500000 (0.707107)	1	2013-03-12 2013-04-07
Laticrete	2(0.02%)	4.000000 (1.414214)	1	2010-11-10 2013-11-25
WarmlyYours	2(0.02%)	5.0 (0)	1	2014-04-09 2014-06-27
M P Global Products LLC	1(0.01%)	5.0 (0)	1	2014-04-15 2014-04-15
Total	8694		108	

The Nest, Honeywell, and Lux companies were dominant players in the programmable thermostat market, with 3206 reviews for Nest (36.4%), 2896 reviews for Honeywell (32.8%), 1929 Reviews for Lux (21.9%). Nest's thermostats were smart thermostats contains a learning algorithm in the sample period. The 1st generation of Nest Smart Thermostat was released on 25 October 2011, and 2nd generation of Nest smart thermostat was released on 2 October 2012. The 3rd generation learning thermostat was released on 1 September 2015. Therefore, this dataset contains reviews for the first and second generation of reviews.

By considering only the reviewers' the first review about PTs, the number of reviewers and review for PTs are the equal. However, the total number of reviews written by this review is around 225,184 reviews in the entire sample period. As shown in the Figure

11, compared to the overall rating distribution of reviews in ‘ 'Tools & Home Improvement', the reviewers’ (who post reviews for PTs) the overall rating distribution of reviews in all categories during entire sample period has less share of rating 5 and higher share of rating 4.

Figure 11. Rating distributions



Step. 2: Cleaning suspicious reviewers (1,165) and biased (69) reviewers.

Step 2.1 Cleaning Suspicious Review Data

Zhao et al. (2013) indicated that fake reviews increase consumer uncertainty about products and more believable online reviews on experience goods have a larger effect on consumer choice. In fact, some firms may want to write the positive reviews about own products and the negative about their rivals’ products. Accordingly, cleaning potential fake reviews, therefore, is important to improve the credibility of review and reduce consumer uncertainty. Mayzlin, D et al. (2014) defined “suspicious reviews” are written by the one-time reviewers. This study can access reviewers’ prior reviews in different categories. This study defines potential "suspicious reviewers" who wrote only one review for a PT and did not write any reviews in different product groups at all over the entire sample period. This cleaning process is called "suspicious reviewer cleaning" in this paper. This cleaning process assumes suspicious reviewers are less likely to write reviews of other products in different categories excluding specific target product groups (own products or other competitors in the same product group) to minimize costs. In other words, suspicious reviewers may be unlikely to post reviews out of their product area. It may possible that they were actual reviewers; however, it is still reasonable to clean the potential suspicious reviewers to remove possible bias. The total number of suspicious reviews were 1,165 written for 80 PTs and removed from sample groups.

Figure 12. suspicious reviewers' overall rating distribution

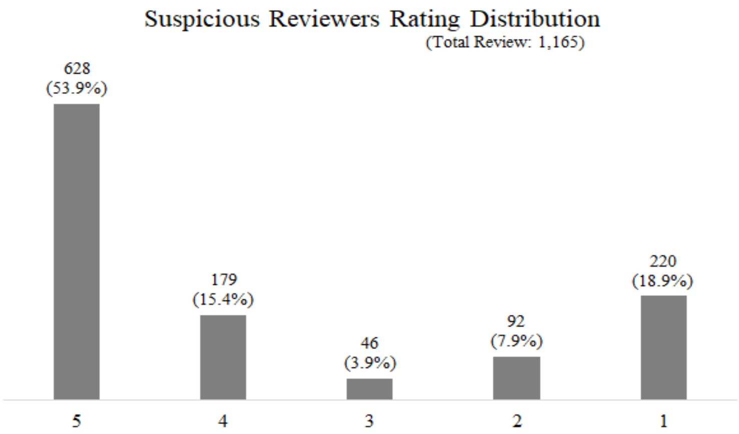
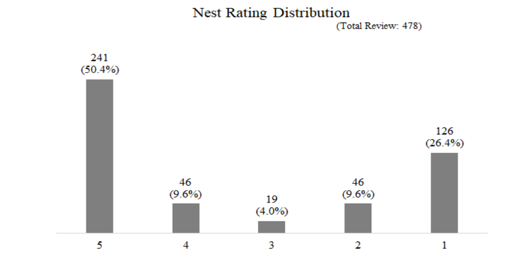
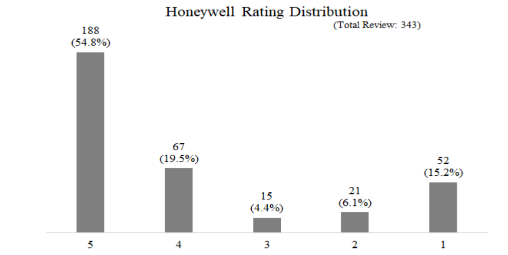
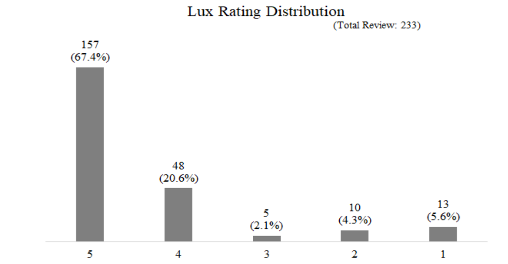
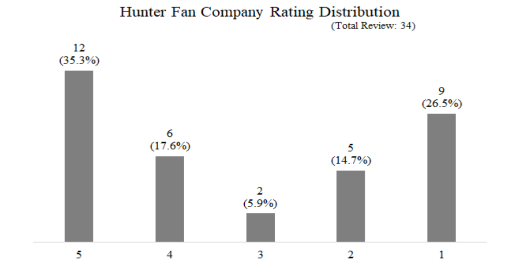
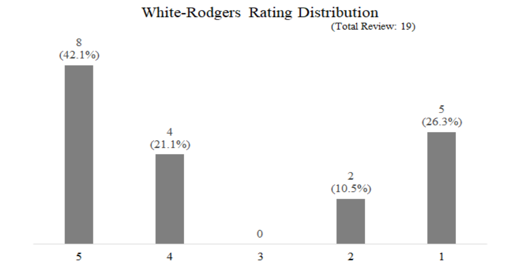
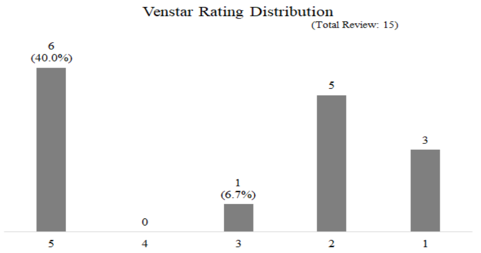


Table 38. Suspicious reviewers rating distribution per top six brands

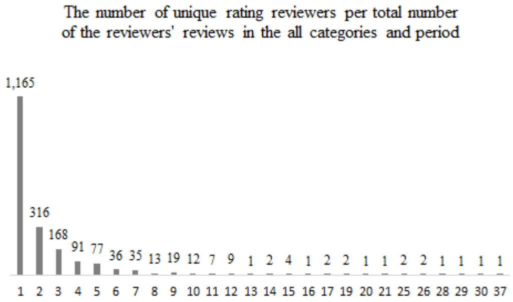
Brand	Review numbers	Period of reviews	Rating distribution
Nest (2)	478 (41.03%)	3/8/2012- 7/14/2014	
Honeywell (37)	343 (33.67%)	10/18/2005- 7/18/2014	
Lux (8)	233 (29.44%)	8/28/2007- 6/28/2014	
Hunter Fan Company (5)	34 (2.92%)	10/18/2005- 6/22/2013	
White-Rodgers* (8)	19 (1.63%)	2008-12-29- 6/29/2014	

Venstar (4)	15 (1.29%)	4/17/2012- 5/12/2014	

Step 2.2 Cleaning always same rater

Some reviewers may always rate at the same level for all reviewed products in all category, regardless of the product type, quality, brand, or other attributes. Such reviewers may not respond to the quality of products and prior reviews written by crowds. Consequently, these reviewers' reviews do not reflect the actual quality of products. However, it may possible that the reviewers wrote same rating level, because the number of reviews was just small. In fact, 1,970 reviewers rated at the same level for products in all categories over entire sample period; however, 1,165 reviewers wrote only one review and 316 reviewers wrote two reviews. Therefore, always same rater (ASR) is a reviewer who wrote more than eight times with same rating level. In detail, five-star rating showed the highest probability as of 0.595 in "Tool and home improvement" category. The probability of nine consecutive five-star ratings is 0.00934 which is less than 0.01. only 69 reviewers wrote more than eight reviews at the same level of star rating. Surprisingly, all 69 ASR are always happy raters (AHR) who rated five-star for all reviewed product and wrote more than eight reviews. Finally, 69 reviews for 25 PTs were removed.

Figure 13. The number of unique rating reviewers



Step. 3: Deleting 1,965 reviewers and 91 reviews for products had no DFs.

Without user and crowd DFs, it is impossible to measure the effect of DFs on the reviewers' rating for PTs when they wrote a review for the first time. In this regard, the procedures in step 3 is removing: (1) 1965 reviewers did not have any previous reviews on other products excluding PTs in all categories before the first day of writing a review for PTs (2) 91 reviewers wrote a review for the PTs that did not have any prior reviews written by crowds. The overlap of 1,965 reviewers and 91 reviewers is 28 reviewers; therefore, the number of reviewers became 5,376 after removing 1,234 reviewers

Step. 5: Selecting the top six major brands (reviews more than 50) among 26 brands

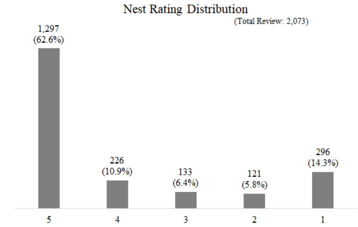
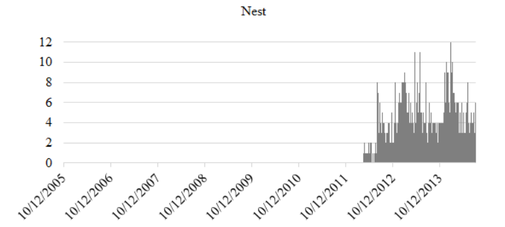
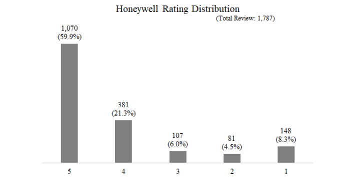
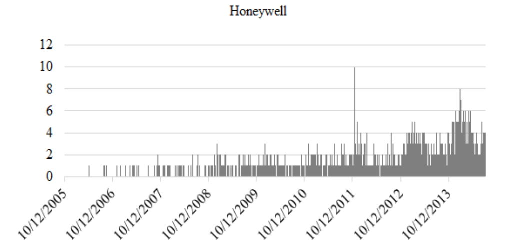
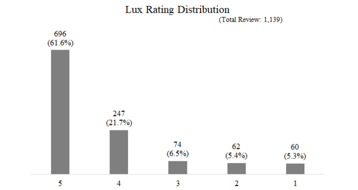
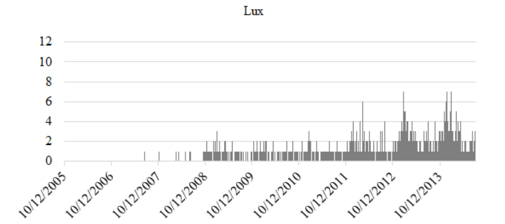
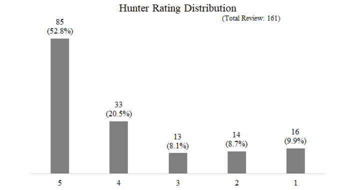
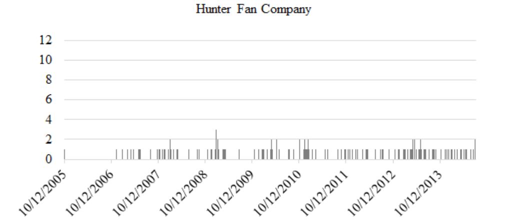
This procedure restricted the reviewers who wrote a review for major six brands that containing more than fifty reviews. After this restriction, 5,307 reviewers wrote a review for major six players including Nest (2,073, 39.06%), Honeywell (1,787, 33.67%), Lux(1139, 21.46%), Hunter Fan Company (161, 3.3%), Venstar (93, 1.75%), and White

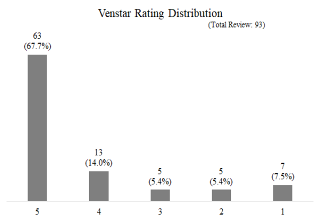
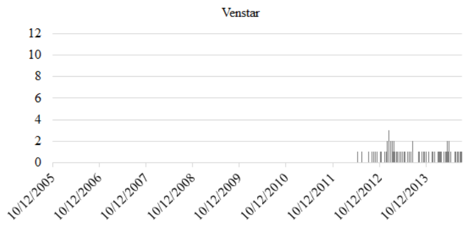
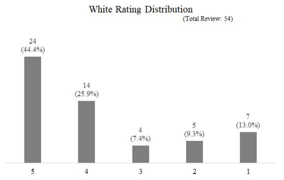
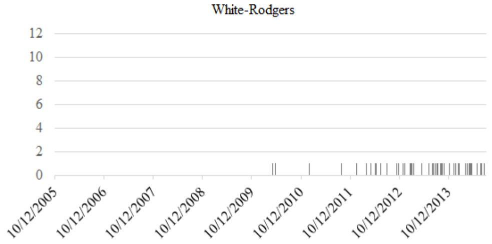
Roger (54, 1.02%). Finally, the number of reviewer and review for 71 PTs between October 12, 2005 and July 17, 2014 became 5,307.

Brand analysis

As shown in the Table 39, the PTs reviews are dominated by three big players, Nest, Honeywell, and Lux. Total number of three players' share of reviews is around 94.20%. In particular, the Nest had only two types of PTs which is known as learning thermostats.

Table 39. Brand analysis

Brand	Reviews	Asin rating distribution	Asin review per day
Nest (2)	2,073 (39.1%)		
Honeywell (40)	1,787 (33.7%)		
Lux (9)	1,139 (21.5%)		
Hunter Fan Company (6)	161 (3.3%)		

Venstar (5)	93 (1.7%)		
White-Rodgers* (9)	54 (1.0%)		

* White-Rodgers is a division of Emerson

* () indicates the number of unique products

Figure 14. Rating distribution over day

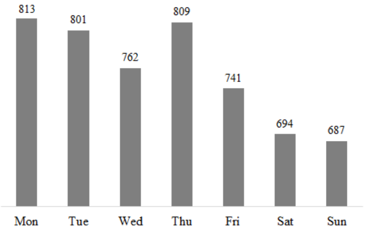
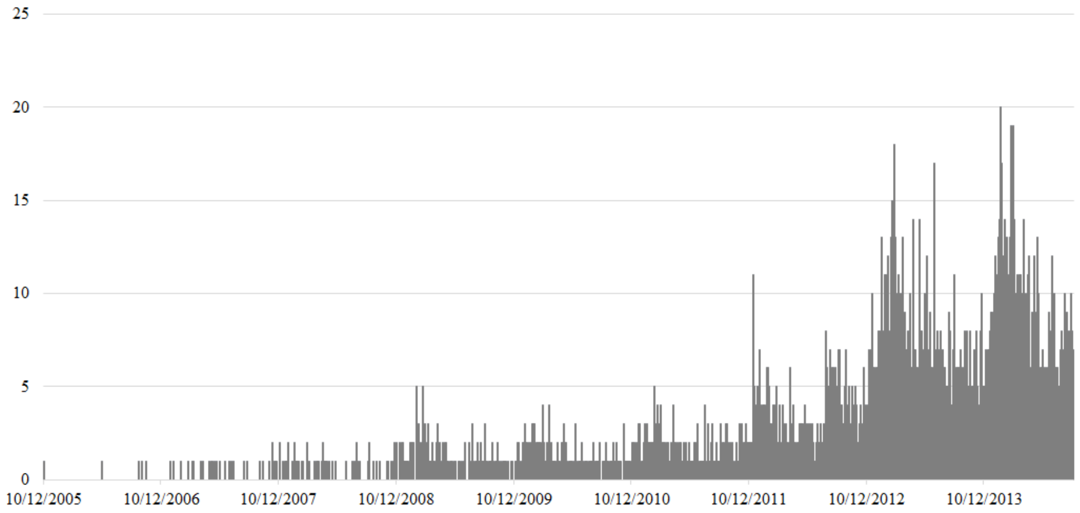


Figure 15. The number of reviews per day



Step. 6. Identifying latent topics in each review based on Latent Dirichlet allocation

What is LDA?

LDA (Blei 2003) is an unsupervised machine learning algorithm to identify latent topics in each review and distribution of these topics in each review. In other words, LDA is a dimension reduction from vocabulary dimension R^{V} to topic dimension R^{K}

In this study, the terminology for LDA is defined as follow:

- $w_{i,n}$ is the *n*th word in the *i*th review and it follows a multinomial distribution.
- V is the total number of unique words in the set of all review data
- K is the total number of topics in each review which is a hyperparameter of LDA
- The *i*th review is a sequence of N words as $r_i = (w_{i,1}, \ldots, w_{i,N})$
- A corpus is a set of M reviews as $R = (r_1, \ldots, r_M)$

As a generative probabilistic model, LDA assumes that each review is represented as a distribution over K topics as θ_i. θ_i is a vector in R^K that represents the proportion of each topic in the *i*th review. θ_i follows a Dirichlet distribution that has α as a Dirichlet parameter. In addition, φ_k is a *k*th topic vector in R^V that represents the proportion of each word belongs V in the *k*th topic. φ_k follows a Dirichlet distribution that has β as a topic hyperparameter. $z_{i,n}$ is a vector in R^K that maps *n*th word in *i*th review into topic *k*. $z_{i,n}$ follows a multinomial distribution. $w_{i,n}$ also follows a multinomial distribution. Overall, θ_i, φ_k, $z_{i,n}$ are latent variables and $w_{i,n}$ is an observable variable within a review.

In addition, LDA assumes that w_R (words in reviews) generated from the joint distribution of θ_R (review's topic distribution) and φ_K (topic's word distribution). The joint distribution indicates words generation process in reviews can be expressed as follows:

$$\mathrm{p}(\varphi_\mathrm{K}, \theta_\mathrm{R}, \mathrm{z}_\mathrm{R}, \mathrm{w}_\mathrm{R} | \alpha, \beta) \;=\; \prod_{\mathrm{k}=1}^{\mathrm{K}} \mathrm{p}(\varphi_\mathrm{k} |\; \beta) \prod_{\mathrm{i}=1}^{\mathrm{R}} \mathrm{p}(\theta_\mathrm{i} | \;\alpha) \sum_{\mathrm{n}=1}^{\mathrm{N}} \mathrm{p}(\;\mathrm{z}_{\mathrm{i,n}} | \theta_\mathrm{i}) \mathrm{p}(\mathrm{w}_{\mathrm{i,n}} | \varphi_\mathrm{k}, \mathrm{z}_{\mathrm{i,n}} | \theta_\mathrm{i})$$

Excluding $w_{i,n}$, the other variables are latent variables. During the training process of LDA, the optimal value of the latent variables that maximizes the posterior probability. The posterior probability denoted as follows:

$$p(\varphi_K, \theta_R, z_R |\; w_R) \;=\; \frac{p(\varphi_K, \theta_R, z_R, w_R)}{p(w_R)}$$

However, the denominator of the posterior probability is intractable for exact inference because φ_K, θ_R, z_R are unobserved variables. In fact, various approximate inference methods are applicable for estimating posterior probability such as variational inference. Gensim python package is used in this study and it uses a variational inference method.

Application of LDA in this study

LDA model (Blei, 2003) is a Bayesian unsupervised learning method to extract the latent topics from documents. In other words, LDA (topic modeling) is a technique to reduce the dimension of text to smaller topic dimensions. Topics in reviews about products indicate the product content dimensions for the products. Product review text for a specific product group contains finite product content dimensions (topics of product reviews) for the product group. The author conducted LDA to define product content dimensions in reviews for the programmable thermostats. The pre-process of LDA text analysis is conducted through the following procedure: the following manner:

1. Lowering characters if the review text contains capital letters
2. Splitting each sentence into words (tokens)
3. Removing stop-words (e.g. “”), punctuations (e.g. “”), single characters, and numbers, but not words that contain numbers
4. Lemmatization
5. Creating bi-grams and add to the dictionary
6. Removing words in less than nine reviews, or more than 40% of the documents
7. Selecting the optimal number of topics in LDA by using a coherence score

After pre-processing, the total number of unique words in all reviews (vocabulary dimension) for LDA is 4,554. Therefore, the dimension of vocabulary is $R^{4,554}$. This high dimensional word embedding dimension was reduced by LDA to five topic dimensions, R^5, The number of the optimal topics is determined by the coherence score (Syed& Spruit, 2017; Figure 17). A domain expert in power industry initially interpreted five subjective product content dimensions:1. connectivity, 2. easiness, 3. saving, 4. setting, and 5. support. These are based on top keywords in each dimension as follows:

Topic Dimension	Interpretation	Top 15 keywords in each topic
Connectivity	The review describes wifi, wireless connection issues with software (e.g. App) and hardware(e.g. HVAC)	wire, wifi, power, device, connected, connect, wireless, Issue, common, update, app, router, software, hvac, connection,
Easiness	The review mentions t ease of use, including simplicity of installation, programming, and use.	easy, work, install, program, installation, instruction, installed, simple, programming, nice, programmable, well, took, product, set
Saving	The review talks about energy savings, including money savings by reducing energy consumption, including billing issues.	energy, control, save, away, money, saving, heater, month, app, bill, iphone, electric, temperature, feature, best
Setting	The review contains content related to setting, control, and information related to temperature, time, scheduling, heating, and other devices.	temperature, time, set, heat, turn, day, back, go, temp, setting, system, need, want, work, change
Support	The review is regarding consumer support services after the purchase.	support, customer, call, product, service, called, tech, told, said, company, hvac, issue, worked, working, customer_service

Figure 16. Word cloud of each dimension

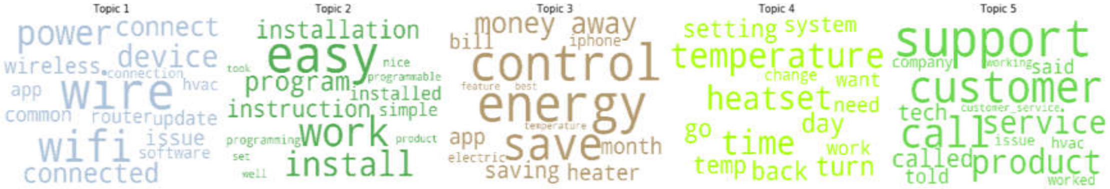
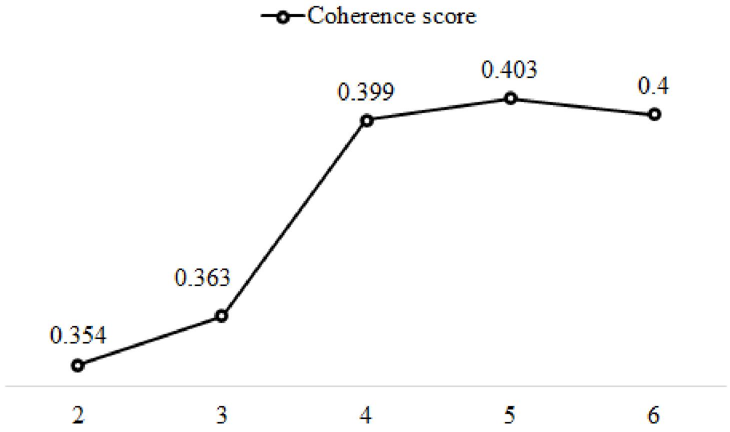


Figure 17. C_v coherence score over topic numbers



Step 7. Modifying and labeling product content dimensions in all review from the expert

the expert extended five dimensions from LDA to nine dimensions, which were 1. Smart-connection, 2. Easiness, 3. Energy and Money Saving, 4. Functionality, 5. Support, 6. Perceived price value 7. Privacy, 8. Amazon effect, and 9. Environment friendliness, based on domain knowledge and the purpose of research design.

Subsequently, the domain expert who had five years of working experience in the power industry annotated 47,763 labeling tasks for sentiment analysis (5307 reviews x 9 dimensions) over two months in a consistent manner and to transfer domain knowledge. Passonneau et al. (2009) mentioned that the expert's annotation transfer domain knowledge from the expert to the learning models to make a better predictive model.

1. Smart connectivity

The domain expert defined smart-connectivity as thermostats' remote control of other home appliances through Wi-Fi by using Apps and software; this dimension measures the reviewers' sentiment on each product's smart-connectivity. Wireless connectivity is a crucial component of the smartness of thermostats as an Internet of Things (IoT) device; it enables consumers to control home appliances with smartphones, tablets, and desktop computers wherever and whenever they want. In addition, a firmware for Wi-Fi thermostats can update itself periodically and display customized pictures on the touch screen. Additional features related to remote control, Wi-Fi accessibility, and software quality for wireless control belong to this dimension. For example, reviewers presented positive sentiments like: “It is nice to monitor & adjust home temperature remotely on iPhone.” and “I love the automatic updates that I have been receiving. They keep adding new features.”

2. Easiness

Easiness indicates simplicity and convenience of installation, set up, programming, and usage. Unlike other experience goods, thermostats require technical experience and knowledge. A lack of the required skills and knowledge may cause reviewers to failure in their attempts to install and use thermostats. In addition, the easiness of following the instruction manual, making the wiring connections, controlling the device (including programming with a better user-interface) belong to this dimension. For example, some reviewers posted: “Easy to Install and Use” and “so easy to use and so easy to see in the dark.”

3. Energy saving

This dimension indicates actual or expected energy-saving and indirect money saving through energy saving due to better energy efficiency and cost-effectiveness of a given thermostat than others or the previous one. Features related to better energy saving also belong to this dimension, such as energy reports that give consumers incentives to change their energy consumption behaviors in better energy-efficient ways. The reduction of utility bills for electricity and gas also belongs to this dimension. Some reviews in this dimension are “My gas bill dropped by 30% the first month” and “A much lower price in your electric bill.”

4. Functionality

The primary purpose of thermostats is to control energy usage for heating and cooling. In this regard, accurate and precise control for temperature and time is essential for becoming a better thermostat. In this dimension, functionality presents the quality of controlling and performance of features. The discomfort caused by thermostats’ quality of functionality belongs to this dimension, such as clicking noise from thermostats during the setting or programming as a negative sentiment. Some reviews in this dimension are “Temperature not accurate but does the job.”, “Makes a clicking noise”, and “Temperature controlled switch works great!”

5. Support

This dimension is related to consumer and technical support, replacement, and return service, including warranty, packing quality, additional support service on their website, and other helpful materials for consumers by the sellers. The expert disregarded the reviewers’ sentiment caused by the quality of Amazon’s support service to restrict the sentiment of reviewers for support services caused only by the quality of product or service from sellers. Consumer support services before and after purchase are vital for thermostats because thermostats require technical knowledge and skill during installation, setting, and programming. Consumer support services may mitigate inexperienced consumers' concerns, technical difficulty, and dissatisfaction during pre-and post-purchase periods. Some reviews in this dimension are “customer service is amazing! Tweet them for help even!” and “They sent mine in 2 days in perfect condition, plus they appear to have a fair return policy.”

6. Perceived price value

This perceived price value is a reviewer’s subjective evaluation about price level compared to the quality, future benefit, and other unobserved confounders. The reason for using sentiment of perceived price value instead of observed price variables at the time of web scrapping is that prices for a good on Amazon.com changes very often and different for consumers due to different promotions and memberships. Overall, written comments related to pricy value, all positive or negative events impacted on the price such as a discount, rebate programs for energy-efficient thermostats from utilities, and realized repair costs belong to this dimension. Some example reviews are “this is money well spent.”, “Gold box deal makes it worth”, “Too expensive to justify the benefit”, “running a promo to give you a $40 gift card with your purchase.” and “A bonus to us was a rebate offered by our power company on energy efficient thermostats.”

7. Privacy

This dimension is about privacy concerns related to thermostats. Wi-Fi smart-thermostats provide remote control through the internet, which can cause consumers' concern about privacy and data security because Wi-Fi thermostats can store and transform user information and consumption data through the internet. The most privacy concerns skewed to Nest after Google purchased Nest with $3.2 billion on January 13, 2014. Some reviews are "Since Google's Nest buyout raises privacy concerns" and "Unless and until clear, unequivocal, irrevocable legal guarantees are in place that Google doesn't get Nest data, I would say that any Nest user must expect that, ultimately, Google will have all that data."

8. Amazon effect

Amazon effect dimension is the reviewers' sentiment caused by Amazon's service quality. Reviews on Amazon.com contain not only product quality but also Amazon's service quality. There have not yet been studied about the effect of online retail market platforms' service quality on reviewers rating. However, if researchers do not count for the effect of Amazon's service quality on ratings, it may lead to potential bias to measure product quality based on reviews. Clearly, the author defined Amazon's effect as a sentiment of reviewers about Amazon's delivery, refund, and replacement policy in the review. Some reviews for this dimension are "Amazon's return policy is great!", "I am very pleased with this purchase and with Amazon customer service.", "Amazon is really good about their customer service so we dont expect any problems here.”, and “super fast Amazon delivery for free (overnight).”

9. Environment friendliness

Thermostats are an energy control device. Some researchers may wonder whether reviewers are concerned about carbon emission and climate change due to heating and cooling devices' energy consumption. This dimension is a binary variable whether reviews contain comments about the environment-friendliness of thermostats. Only nine reviews wrote about environmental issues. Some comments in this dimension are “it helps save the environment!” and “I feel all environmentally friendly for wasting less energy, too.”

Table 40. correlation among sentiment variables

	Smart connect	easy	save	func	sup	amazon	Price value	env	privacy	support
Smart connect	1.000									
easy	0.121	1.000								
save	0.183	0.117	1.000							
func	0.215	0.227	0.235	1.000						
sup	0.210	0.139	0.110	0.222	1.000					
amazon	0.016	0.024	-0.005	0.001	0.022	1.000				
Price value	0.043	0.094	0.097	0.181	0.132	0.054	1.000			
env	0.030	0.016	0.067	-0.014	0.024	0.025	0.011	1.000		
privacy	0.003	0.015	0.021	0.016	0.182	0.005	0.044	0.003	1.000	
support	0.214	0.139	0.108	0.224	0.983	0.022	0.126	0.023	0.013	1.000

Table 41. class distribution of smart connectivity PCD

Smart connect	Nest	Honeywell	Lux	Hunter Fan Com	Venstar	White Roger	Total	Percent
-1	184	39	3	0	3	0	229	4.32%
0	884	1,536	1,131	160	54	53	3,818	71.94%
1	1,005	212	5	1	36	1	1,260	23.74%
Total	2,073	1,787	1,139	161	93	54	5,307	100.00%

Table 42. class distribution of easiness PCD

Easiness	Nest	Honeywell	Lux	Hunter Fan Com	Venstar	White Roger	Total	Percent
-1	117	241	136	18	15	10	537	10.12%
0	963	537	444	52	34	13	2,043	38.50%
1	993	1,009	559	91	44	31	2,727	51.38%
Total	2,073	1,787	1,139	161	93	54	5,307	100.00%

Table 43. class distribution of energy saving PCD

Saving	Nest	Honeywell	Lux	Hunter Fan Com	Venstar	White Roger	Total	Percent
-1	64	25	3	2	0	1	95	1.79%
0	1,359	1,530	992	121	84	52	4,138	77.97%
1	650	232	144	38	9	1	1,074	20.24%
Total	93	1,787	1,139	161	54	2,073	5,307	100.00%

Table 44. class distribution of functionality PCD

Functionality	Nest	Honeywell	Lux	Hunter Fan Com	Venstar	White Roger	Total	Percent
-1	584	424	269	45	18	15	1,355	25.53%
0	700	619	315	46	32	27	1,739	32.77%
1	789	744	555	70	43	12	2,213	41.70%
Total	2,073	1,787	1,139	161	93	54	5,307	100.00%

Table 45. class distribution of support PCD

Support	Nest	Honeywell	Lux	Hunter Fan Com	Venstar	White Roger	Total	Percent
-1	250	119	27	4	6	4	410	7.73%
0	1,581	1,544	1,086	155	79	49	4,494	84.68%
1	242	124	26	2	8	1	403	7.59%
Total	2,073	1,787	1,139	161	93	54	5,307	100.00%

Table 46. class distribution of the Amazon effect PCD

Amazon	Nest	Honeywell	Lux	Hunter Fan Com	Venstar	White Roger	Total	Percent
-1	8	16	4	0	1	0	29	0.55%
0	2,040	1,722	1,114	157	91	51	5,175	97.51%
1	25	49	21	4	1	3	103	1.94%
Total	2,073	1,787	1,139	161	93	54	5,307	100.00%

Table 47. class distribution of pricy value PCD

Price value	Nest	Honeywell	Lux	Hunter Fan Com	Venstar	White Roger	Total	Percent
-1	306	59	18	2	6	3	394	7.42%
0	1,485	1,393	909	103	76	41	4,007	75.50%
1	282	335	212	56	11	10	906	17.07%
Total	2,073	1,787	1,139	161	93	54	5,307	100.00%

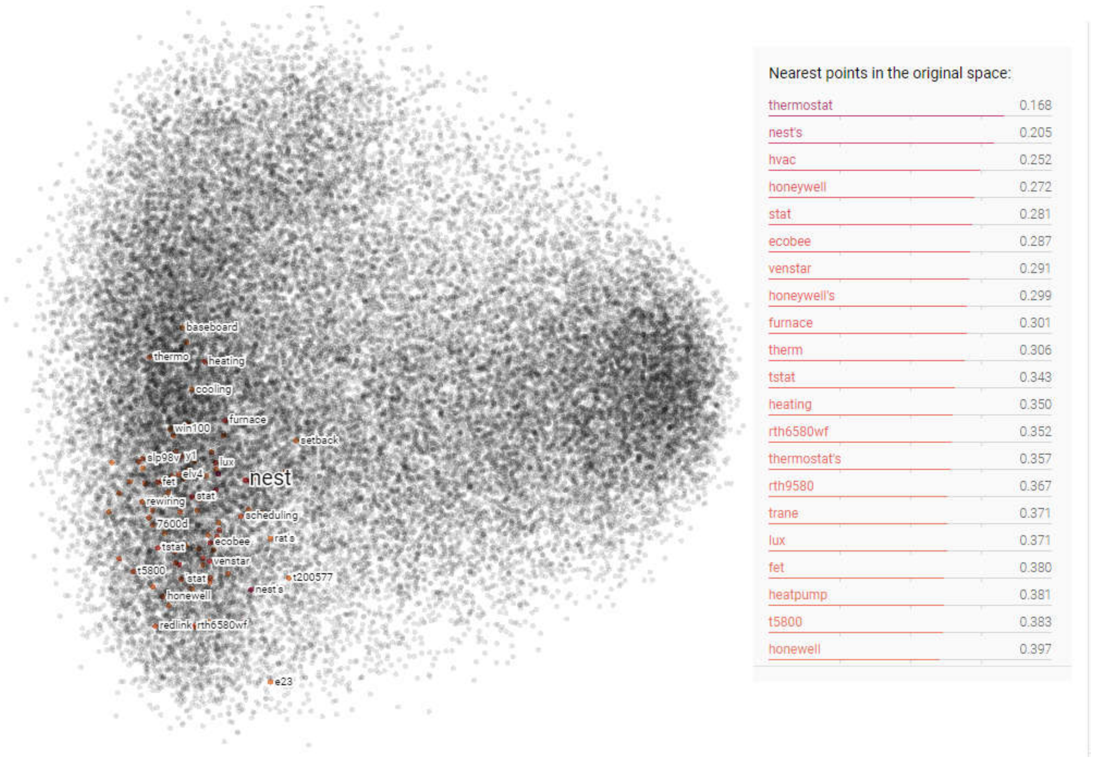
Table 48. class distribution of environmental friendliness PCD

Environment Friendly	Nest	Honeywell	Lux	Hunter Fan Com	Venstar	White Roger	Total	Percent
0	2,069	1,786	1,138	160	92	54	5,299	99.85%
1	4	1	1	1	1	0	8	0.15%
Total	2,073	1,787	1,139	161	93	54	5,307	100.00%

Table 49. class distribution of privacy PCD

Privacy	Nest	Honeywell	Lux	Hunter Fan Com	Venstar	White Roger	Total	Percent
-1	24	2	0	0	0	0	26	0.49%
0	2,049	1,783	1,139	161	93	54	5,279	99.47%
1	0	2	0	0	0	0	2	0.04%
Total	2,073	1,787	1,139	161	93	54	5,307	100.00%

Figure 18. Word2Vector visualization in 2 dimension



Nearest points in the original space:

thermostat	0.168
nest's	0.205
hvac	0.252
honeywell	0.272
stat	0.281
ecobee	0.287
venstar	0.291
honeywell's	0.299
furnace	0.301
therm	0.306
tstat	0.343
heating	0.350
rth6580wf	0.352
thermostat's	0.357
rth9580	0.367
trane	0.371
lux	0.371
fet	0.380
heatpump	0.381
t5800	0.383
honewell	0.397

Note: https://projector.tensorflow.org/, Initial word2vec dimension is 100 and is reduced to 2 dimensions by applying PCA. X-axis and Y-axis indicate principal component. Distance between words are based on cosine similarity score (=similarity $= \frac{A \cdot B}{||A||||B||}$ where A and B is a vector)