Winter 12-2019

Time Series Event Forecasting in Consumer Electronic Markets using Random Forests

Benjamin Buchwitz
Catholic University of Eichstätt-Ingolstadt, benjamin.buchwitz@ku.de

Anne Falkenberg
Catholic University of Eichstätt-Ingolstadt, anne.falkenberg@ku.de

Ulrich Küsters
Catholic University of EichstÃ¤tt-Ingolstadt, ulrich.kuesters@ku.de

Time Series Event Forecasting in Consumer Electronic Markets using Random Forests

Completed Research Paper

Benjamin Buchwitz
Catholic University
of Eichstätt-Ingolstadt
benjamin.buchwitz@ku.de

Anne Falkenberg
Catholic University
of Eichstätt-Ingolstadt
anne.falkenberg@ku.de

Ulrich Küsters
Catholic University
of Eichstätt-Ingolstadt
ulrich.kuesters@ku.de

Abstract

Consumers are price-sensitive and opportunistic about the place of purchase when buying electronic goods. However, services that advise customers on their purchase time decisions for those products are missing. Given the objective to provide a binary signal to customers to either wait or purchase immediately, classification algorithms are a direct methodological choice. Approaches like random forests allow for the derivation of a probability and class prediction but are usually not used in time series contexts. This is due to missing or time-invariant regressors and unclear prediction settings. We show how classification methods can be used to generate reliable predictions of price events and analyze if they are subject to common market dependencies. Pooling univariate random forests and enhancing them with multivariate features shows that our approach generates stable and valuable recommendations. Because dependency structures between products are transferable, multivariate forecasting increases accuracy and issues recommendations where univariate approaches fail.

Keywords

Price Event Forecasting, Multivariate Time Series, Random Forest, E-Commerce

Introduction

During the last decade, the e-commerce market experienced double-digit revenue growth every year and is expected to reach a sales volume of 58.5 bn € in 2019 in Germany alone (HDE 2019; Statista 2018). While the revenue is steadily growing, this is also true for the number of retailers that offer their product portfolios online. For online shoppers, this is boon and bane. While almost everything can be purchased online, the vast number of online outlets increases searching costs for consumers dramatically. Due to the price heterogeneity in the e-commerce market, searching costs are especially high for strategic and bargain-hunting customers, who are willing to invest time to acquire knowledge about the pricing structure for the products of interest. This non-transparent and complex market gave rise to market aggregation platforms such as price comparison sites (PCS) that list a variety of products for a given product category and provide an overview of the retail landscape to allow customers comparing retailers and prices for standardized, homogeneous goods. The sustained popularity and the increasing number of users of price comparison sites show that these services fulfill a crucial need for and are appreciated by customers. However, while PCS support customers in all pre-purchase stages of the buying process, the information they present is static and only valuable at a particular point in time.

When shopping electronic goods, customers have proven to be particularly price-sensitive (Greenstein-Messica et al. 2017; Johnson et al. 2013), meaning they are generally willing to delay their buying decisions in case the price level is expected to drop. The strong focus of their customers on prices has led PCS to display historical minimum prices, meaning the best available price over all listed retailers for a single good over time. Drechsler and Natter (2011) show that these historical minimum prices influence customers when scheduling their purchase. Thus, historical prices are taking the role of reference prices that users compare to current price levels (Dutta 2012; Han et al. 2001; Moon et al. 2006). Practically speaking, historical prices are the only structured information source to form expectations about future price developments when buying electronic consumer goods online. The willingness to reschedule buying decisions in combination with unknown future prices creates uncertainty about the purchase time point. While it is obvious that customers need assistance to structurally and effectively utilize the provided information to their benefit, price comparison sites for electronic goods have not yet provided a solution that supports their customers when it comes to scheduling buying decisions.

PCS seem to be not as prevalent in the US as in Germany. However, there has been a new service introduced for consumer goods recently called *Shopbrain*, which searches and compares prices when consumers visit certain shopping sites. *Shopbrain* also tries to provide recommendations for the price development within the next 10 days. However, it seems that they focus on a comparison of the current price to the average price over a range of participating retailers (Cunningham and Zhu 2019). Therefore, the number of offers is limited, and the development of the average price is not necessarily meaningful to the customer. Services offered for the airfare industry are by far more common and spread. However, when comparing aggregation platforms for consumer goods to those for airfares, one notices that the latter recently started to offer buying recommendations to their customer base. These recommendations indicate if a customer should buy a product or service immediately for the advertised price or if s/he can expect a more competitive price in the future. The respective service, therefore, directly addresses price-sensitive customers, who are one of PCS' largest visitor groups (Meierhoff 2018). Surprisingly, a comparable service for items such as smartphones, tablets or laptops is not widely available. The reason for this is threefold. First, it must be noted that flight prices exhibit different time series characteristics than consumer product prices because airfares are usually managed by some form of a quantity-oriented revenue management system (Klein and Steinhardt 2008). The behavior of these systems creates patterns that can be detected and exploited to generate recommendations (Udachny 2015). Second, flights are reoccurring, meaning that the same flight from one destination to another is offered multiple times a day or week, which dramatically increases the data basis available for the prediction generation. Third, this difference in the data basis is further intensified by the fact that platforms such as *Kayak*, *Hopper, AirHint* and *Momondo* bundle multiple carriers and routes e.g. with and without stop-overs when generating buying recommendations for their visitors (Etzioni et al. 2003). Electronic consumer goods, on the other hand, can only be treated individually because even though a customer may be willing to reschedule her/his buying decision for a given smartphone, s/he may not be willing to buy a substitute with a different color, display size or memory configuration – not even for a drastically reduced price. This means that prediction engines are confronted with a comparably small data basis and – considering the statistical properties of the given time series data – a complex modeling and decision problem.

Possible methodological candidates that exploit the inherent price dynamics to form expectations about future price developments are time series forecasting methods. However, while univariate approaches, that rely solely on past observations to forecast values for future time points, have shown good performance in forecasting competitions (Makridakis et al. 2018; Makridakis and Hibon 2000), the examined task is not necessarily a pure time series forecasting one (Weiß 2009). One possible modeling approach is to signal customers to delay their buying decisions if any of the predicted future prices in a given time frame is (substantially) lower than the price at the forecasting origin. While this perspective focuses on extrapolations along the temporal dimension of the data, the respective decision can also be formulated as a binary classification question: Will the price fall in the given horizon or not?

Restraining from the use of time series forecasting methods and turning to classification algorithms has numerous advantages. First, modern classification algorithms such as random forests, support vector machines, and neural networks have benefitted from huge methodological improvements within the last three decades and show extraordinary performance in classification tasks. Besides, their implementation is widely available and present in a lot of numerically stable standard software packages (Liaw and Wiener 2002; Pedregosa et al. 2011). Second, combining heterogeneous information from different entities, products, and time points is straightforward from a conceptual perspective. This allows building a recommendation service to advise customers on their purchase time decision, even based on historical information for multiple series belonging to numerous products with non- or only partially overlapping time spans. Third, classification algorithms enable flexible and adaptive implementations, which not only allow staying in the same methodological family but also changing and evaluating the setting depending on the application context. Fourth, multivariate time series forecasting methods often require manual guidance concerning the context-specific, theoretical basis of interdependencies, and intensive tuning of hyperparameters. While, this is also true for some classification algorithms, especially random forests are known to rely on few hyperparameters and are robust to overfitting (Efron and Hastie 2016). Additionally, numeric price forecasts for each and every day in the decision horizon are not needed. If – from the beginning – the objective is to display a binary signal to the customer to either wait or purchase directly, then using classification algorithms is the more direct approach. Fifth, relationships in the consumer goods industries are non-transparent, complex and uncharted. Therefore, and due to the large number of retailers as well as items, it is not beneficial to identify causal interdependencies that have to be estimated on the basis of limited data. As items, retailers and (partially) brands enter and exit the market frequently, it may be more plausible and helpful to use methods that allow learning patterns and dependencies using all past data from both active and inactive products to describe pricing dynamics. This makes classification methods a natural candidate for predicting future price events and creating flexible and adaptive recommendations systems. However, while many modern machine learning methods gained popularity due to their strong performance in cross-sectional, classification-based tasks, forecasting competitions have revealed that the prediction performance is often rather disappointing so that they are frequently at par or even outperformed by naïve time series forecasting methods (Makridakis et al. 2018).

The contribution of this paper, therefore, is twofold. We first aim to answer the questions if and how classification methods can be used to predict future price movements and generate meaningful buying recommendations. An essential part when answering this question is which variables should be included in the classification. We suspect that the poor performance of modern machine learning algorithms in time series contexts is at least partially grounded in incomplete or insufficient data or feature engineering. We, therefore, present a range of measures to adequately reflect the characteristics of the underlying time series to enable classification algorithms to generate beneficial decision recommendations. Second, research in time series forecasting suggests that forecasts for a group or collection of time series can be improved, when multivariate knowledge from the entire database is incorporated in the analysis (Hyndman et al. 2008; Smyl et al. 2018). In the context of classification, this leads to possible settings ranging from purely univariate considering only information from a single product to fully multivariate incorporating features constructed on the basis of multiple products. Consequently, following that path, the model estimation can be performed over a single, a subset or all available items in the market when generating buying recommendations. These implementation options facilitate examining the effect on the recommendation performance when changing from a product-centered univariate setting to a market- or brand-oriented multivariate one. Besides, it allows suggesting solutions for the inherent univariate-multivariate modeling tradeoff when applying classification methods to collections of time series and helps to generate valuable recommendations for online-shoppers that enables them to make better, more informed and less costly buying decisions. This paper is to the best of our knowledge the first study that investigates the effect of univariate and multivariate forecasting settings on classification performance for consumer-oriented decision support systems used for price event forecasting.

Data Properties and Sample Description

The primary source of data when dealing with price-oriented consumer decision support services are the historical price time series of the respective technological consumer goods. We, therefore, illustrate and evaluate our feature engineering and modeling approach using a large sample from the German consumer electronics market. Each time series consists of minimum prices, for which a customer could have bought the respective product from a retailer at the given time. The minimum price on a specific date is defined by the lowest price over all retailers listed on the comparison site. Discounts including coupons and promo codes known to the PCS are also considered in the price sequences. To make descriptive statistics and different prediction approaches comparable and allow generalization over the products, we restrict ourselves to the first two years of data, representing the typical life cycle of consumer electronic goods. This results in 730 observations for each item. Please note that the truncation of each product time series to two years is performed to stabilize features and to make the analysis more robust to aberrant price deviations often found after that period as products sell out and gradually leave the retailer landscape. Thus, in the following, we will use the terms market, sample, and data basis interchangeably. To allow the reader to gain a deep understanding of the data set, we first briefly focus on the properties present in every series before discussing the inter- and intratemporal structure of the time series in the sample and its associated characteristics in more detail.

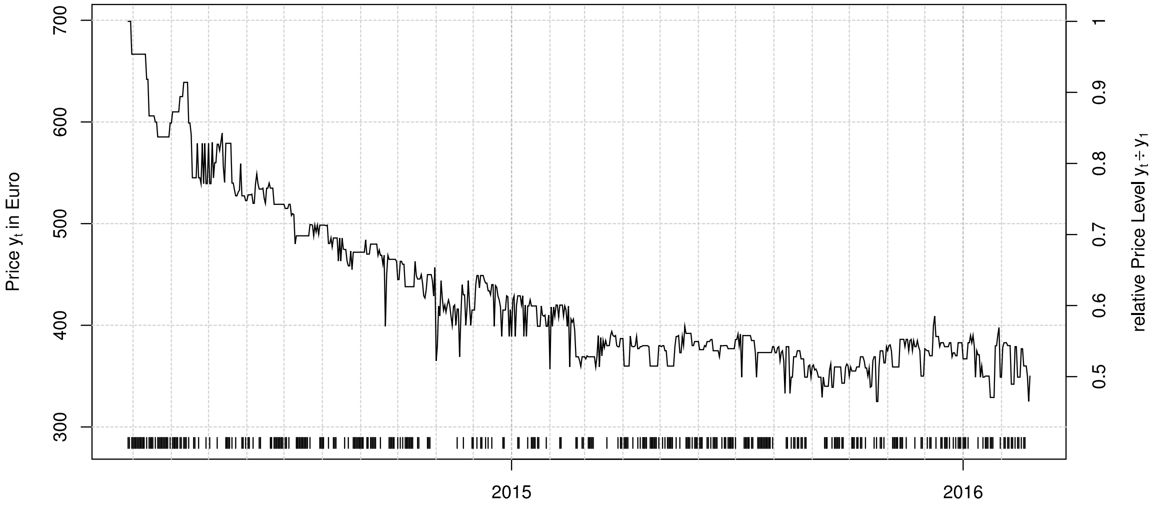

Figure 1. Price Time Series of a Smartphone between 25 Feb. 2014 and 24 Feb. 2016

Figure 1 shows a representative minimum price time series of a smartphone from the data set. It clearly exhibits three distinctive characteristics for technological consumer goods: price deterioration, time-varying price changes and phases with constant prices. First, as expected, the steepness of the time series decreases with the level, while the level declines to approximately half of the starting value. The lowest price in the series of 324.95€ is reached in February 2016 and corresponds to a relative price level of 46.49%. Second, there are visible and varying market phases with substantial volatility differences. Along these phases, price changes vary strongly in frequency and magnitude. While the nominal biggest singular price change is an increase from 356.99€ to 418.00€ that occurred in February 2015, the seven consecutive days with the absolute largest sum of price reductions of 220.09€ started in October 2014. Third, periods with constant prices make up for a significant proportion of the time series. In the price time series in figure 1, 45% of the 730 observations show no price change compared to the previous day, which is indicated by tick marks on the abscissa. For the sequence of price changes, this property leads to many zeros and is therefore referred to as zero-inflation (Kömm and Küsters 2015; Rydberg and Shephard 2003; Sucarrat and Grønneberg 2016).

The data set consists of 238 smartphone price time series between 2008 and 2017 from well-known and established brands, yielding a sample with 173,740 daily prices in total. The most frequently represented brand in the sample is Samsung with 71 different devices, followed by Apple with 46 devices. The brand with the least devices in the sample is LG with 11 smartphones. Brands that do not have an extensive smartphone lineup (less than ten devices in the sample) such as Google, Lenovo-Motorola, Blackberry or Microsoft are bundled in the "Others" group. The number of products per brand and the descriptive statistics are shown in table 1. Each time series represents a single stock keeping unit with completely homogeneous properties and features. Products that differ from each other by a single attribute such as color, display or memory size, therefore, constitute different time series. As the sample represents a wide range of phones available in Germany at the time of the data collection, it considers items from all market segments, ranging from reasonably priced beginner smartphones to expensive high-end devices. Subsequently, the time series have different initial prices, show different price developments and can be characterized by different rates of price deterioration.

The average initial price as shown in table 1 is calculated as the arithmetic mean of the first observed price of all products of the respective brand. The initial item-specific price often but not in all cases coincides with the manufacturer's suggested retail price. The highest average brand-specific initial price is recorded for the Apple smartphone lineup with 834.98€. Apple also contributes the product with the single highest initial price of 1299.00€. The most affordable lineup is offered by Huawei with an average price of 359.83€. Only the mean initial price from the "Others" group is lower due to a large share of budget phones in this group. The overall average initial price in the sample is 545.03€. When looking at the average zero-inflation, meaning the share of zeroes in the sequence of price changes, one can see that they show less variation by brand as all values are roughly around 60%. The smallest average zero-inflation is found in the price changes of Apple products with 58.49%, while the single highest zero-inflation (64.88%) in the sample stems from a product offered by Sony. The average zero-inflation for all 238 products is 59.48%. The average minimum relative price is a measure of price stability and is given by the quotient of the lowest price and the starting price of each product, averaged by brand. The phones with the most stable prices are manufactured by Apple, where the average minimum price amounts to 57.50% of the starting prices. The brand with the lowest value stability is LG, where the minimum price in the data only accounts for 36.68% of the starting prices. Simultaneously, the phone with the lowest relative price level is also offered by LG, here the relative price level drops to 21.54% of its initial price of 599€. On average, smartphones in the sample reach minimum price levels of 46.99% of their starting values.

Brand	N	Average Initial Price	Average Zero-Inflation	Average Min. rel. Price
Apple	46	834.98€	58.49%	57.50%
HTC	17	528.63€	60.64%	44.39%
Huawei	13	359.83€	59.00%	45.84%
LG	11	568.09€	61.32%	36.68%
Nokia	16	413.44€	61.89%	46.88%
Samsung	71	547.33€	59.05%	41.92%
Sony	33	495.24€	59.62%	43.49%
Others	31	308.92€	59.42%	52.35%
All	238	545.03€	59.48%	46.99%

Table 1. Data Overview and Descriptive Statistics

Obviously, neither were all phones in the sample introduced to the market at the same time nor have all of them been available at each given date. This, in turn, means that the descriptive statistics shown in table 1 are too simplistic. The positioning structure in the timeline, meaning the entrance and exit dates of each smartphone, as well as the number of smartphones that are available at a given day, are shown in figure 2. The abscissa in figure 2 shows the time, starting with the introduction of the first product on December, 18th 2008 by Nokia, whereas the last price observation in the sample is contributed by a smartphone manufactured by Microsoft and thus located in the "Others" group on October, 6th 2017. In total, the sample stretches over 3215 days. The left ordinate shows the average price of all products in the database and corresponds to the solid blue line in the plot. The average price ranges from roughly 200€ to slightly over 500€. Additionally, it can be seen that the average price of all smartphones in the sample exhibits a positive global trend overlaid by two effects. First, on the day of product launches the average price level shifts upwards. Second, immediately after product launches, prices deteriorate locally, and older products leave the sample. The right ordinate indicates the number of available products in the data, illustrated by the solid black line. It ranges from 1 directly at the beginning to a maximum of 126 simultaneously available phones in 2015. In addition to these two graphs, the starting and end dates of each product are shown by the stacked colored lines. Colors are chosen according to the brand assignment of the respective time series. While the sample, in the beginning, shows that the market is fairly diversified with many manufacturers contributing roughly the same number of products, especially Apple (green stacked lines) and Samsung (brown stacked lines) are quickly building up items in the sample. It becomes clear that since the end of 2013, manufacturers have been starting to increasingly differentiate products that are now frequently introduced in blocks. The largest blocked product launch in the sample can be observed on September, 10th 2014, where 16 Apple smartphones entered the market simultaneously, directly followed by 13 Samsung products on March, 2nd 2015 causing large upwards shifts of both the available number of products (black line) as well as the average price of smartphones (blue line). Obviously, entering several phones as a block result in a simultaneous leave after 730 days.

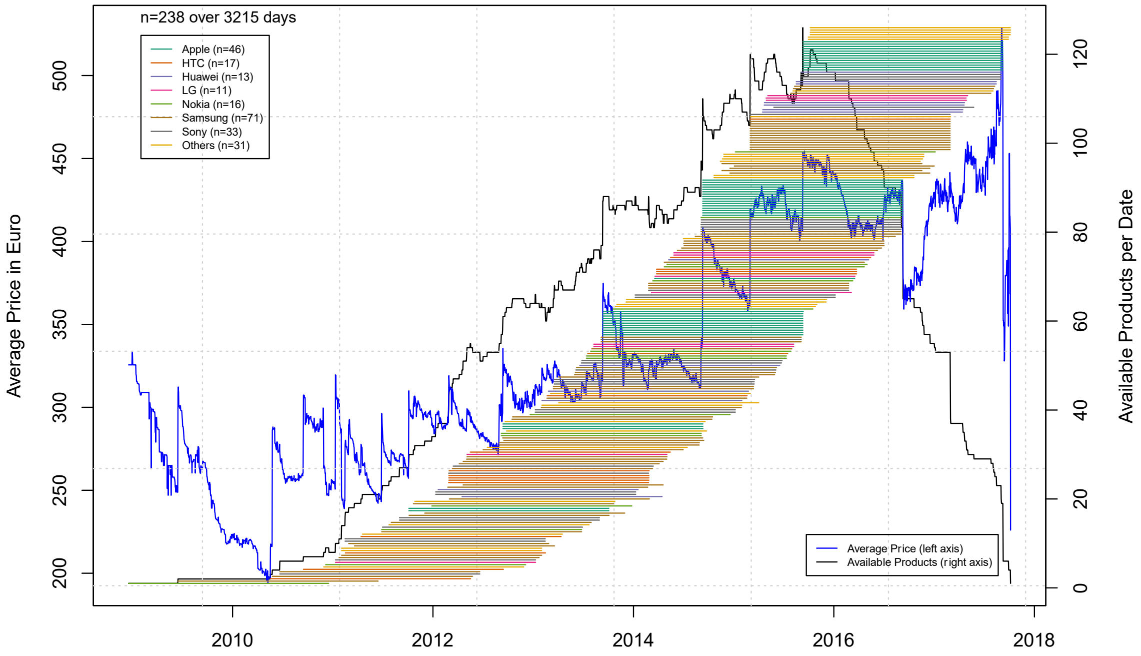

Figure 2. Temporal Structure of the Data Set

Feature Engineering

While feature engineering is a natural step in analyses involving machine learning, it is rather uncommon in the context of time series forecasting. There is some literature that develops time series features, but these are often descriptive and subsequently used for unsupervised learning such as clustering or visualization tasks (Kang et al. 2017). Which time series features possess predictive power and can be used to forecast price decline events for consumer goods, therefore, cannot be extracted solely from academia. As the main goal of this paper is to investigate the effects of multivariate learning for price event predictions, it should be noted that switching to a multivariate perspective also enables generating multivariate features for properties from a group of items in addition to measures that are derived from single series.

Prior to discussing the features, we introduce the target variable *event* E_T that should be predicted. While not being a feature in a narrow sense, it is still a variable that needs to be constructed based on the observed minimum price time series. The goal of our model is to predict if the price for a given product declines within an a priori defined time span of H days, which can be considered the decision horizon of the respective customer for whom the prediction is issued. To allow formalizing the event definition and the upcoming feature generation process, we refer to the sequence of prices as $\{y_t\}$ with $t \in 1, \dots, 730$. The rolling forecast origin is denoted by T so that the last price available for prediction is defined as y_T. The dichotomous price decline event or categorically coded event indicator variable is given by

$$E_T = \mathrm{I}(\min\{y_{T+1}, \dots, y_{T+H}\} < y_T),$$

where $I(\cdot)$ resembles the indicator function that takes the value 1 if the condition $(\cdot)$ is fulfilled and 0 otherwise. For the decision horizon, we set $H = 7$. Thus, we assume that the customer's rate of time preference is zero within the decision horizon of seven days. Therefore, we disregard the customer's potential utility gain from earlier purchases. The reason for this is that a customer's utility function is unique and possibly time-variant. To incorporate and evaluate the performance of the decision support considering each customer's utility gain, we would need to make multiple assumptions about the user-specific utility function (Huang 2011; Keeney and Raiffa 1993; Wang and Zhang 2011). We are aware that it is possible to set the time frame covered by the event variable to different values. Thus, we recommend customizing this parameter according to the user requirements for practical applications. It must be noted that a prediction cannot be evaluated until one week (for $H = 7$) after issuing the recommendation, as all prices in the decision horizon are required for the decision whether the advice was correct or not.

Univariate Features

Before describing the univariate feature set that represents the information baseline for both settings, it should be noted that some features are univariate derivable but only usable in a multivariate setting. This is the case for all attributes that do not vary along the temporal axis like the technological particularities of a given smartphone or the respective initial price. To keep the feature description compact, we focus on the derivation perspective of the features and discuss features that rely solely on data from a single item in this section. Besides, we omit an additional subscript that would not vary throughout the univariate feature section. However, obviously, all features described below are calculated for all items and given time points to form the feature space that is used in the upcoming model estimation.

returns: A natural starting point for the feature engineering emerges from the comparison with univariate time series forecasting models. Many models including the widely used autoregressive models estimate the dependency structure over time. To avoid distortions due to different price levels, we rely on relative price changes operationalized by returns instead of nominal values. As the data consist of daily price observations, we include variables for the last week (seven days) in the process. This resembles the idea of a seven order autoregressive process AR(7) based on the values r_T to r_{T-6}, obtained from the sequence of returns

$$r_t = \frac{y_t - y_{t-1}}{y_{t-1}}.$$

level and mean: To embed the return values and allow differentiation for varying market segments of the smartphones, we include the last available price observation at the forecasting origin y_T in the set of regressors as well as a measure for the price level. While the item-specific price level fluctuates intensively over time, the global average provides a more stable measure of the central tendency, which is why we add $\bar{y}_T$ to the feature set as well

$$\bar{y}_T = \frac{1}{T}\sum_{t=1}^{T} y_t.$$

drift: Market phases with many consecutive price reductions can often be characterized by their stable downward movements. These phases where prices show rapid and strong deterioration can be identified by trend tendencies, which we measure by a drift coefficient that is equal to the average price change

$$drift_T = \frac{y_T - y_1}{T - 1}.$$

zeroinflation: As described, constant segments between consecutive price observations constitute a large portion of the data. Given the fact that consumer goods prices generally deteriorate, customers can only profit from price reductions if there is price movement in the first place. We, therefore, include the zero-inflation as an indicator for the intensity of price setting activities of retailers. The zero-inflation can be calculated on the basis of the indicator function to signal if the current observation is equal to the prior price.

$$zeroinflation_T = \frac{1}{T - 1} \sum_{t=2}^{T} I(y_t = y_{t-1})$$

ratioyTmean7: In order to provide evidence whether the given price is a good offer, we expand the feature set by the ratio of the price at the forecast origin to the average price of the last seven days. We include this variable to allow implicit non-linear interactions between variables like $\bar{y}_T$ and *drift* as well as relative variables like *zeroinflation* and r_T. This provides information about whether the price level at the forecast origin is comparably high or low and is, therefore, a proxy for the local over- or underpricing degree. Abstracting from the human perspective *ratioyTmean7,* thus, ensures that no waiting recommendation will be issued when high discounts are observed and vice versa. The measure is formally defined as

$$ratioyTmean7_T = \frac{y_T}{\frac{1}{7} \cdot \sum_{t=T}^{T-6} y_t}$$

sumlastreductions7: While the different calm and active market phases can be characterized, on the one hand, by their price setting intensities, the magnitude of these price changes also provides insights. While we are especially interested in price reductions, we include the nominal sum of price reductions that occurred over the period of the last seven days as a local measure of single-sided dispersion.

$$sumlastreductions7_T = \sum_{t=T}^{T-6} (y_t - y_{t-1}) \cdot I(y_t < y_{t-1})$$

daysinmarket: In addition to the pricing-oriented variables presented above, *daysinmarket* provides a product-specific time axis for interactions and to model the progress in a product's life cycle. It, therefore, takes the value 1 on the day a product is introduced into the market and ranges up to a maximum of 730, which marks the last observation in the corresponding minimum price time series. It is operationalized by T.

Multivariate Features

As mentioned before, the multivariate features are derived for a set of interrelated product time series. The cardinality of these groups inevitably affects the corresponding multivariate measures and therefore needs to be discussed. While there are many possible grouping options, we chose to focus on two naturally emerging ones. First, we group all products together and create a market group. As not all 238 products are purchasable at the same time (see figure 2), the number of products that are active varies. Obviously, only active products are incorporated into the feature calculation for a specific date. Observations that are not available due to a product being inactive are ignored. This is also true for the second group, which we construct based on the brand affiliation. To avoid having very small groups that contain only a few products, we revert to the same logic that we applied when calculating and presenting descriptive statistics in table 1 and combine brands with less than ten products in an "Others" group. This yields the variable *brandcluster*, which we use in addition to the market group to construct the following features. However, we also embed *brandcluster* (cardinality=8) as categorical variable directly in the upcoming multivariate modeling process to account for brand-wise heterogeneity in the products.

productcount: The simplest multivariate measure is the number of active products in each of the groups. The realizations of the variable *productcount_market* are identical to the number of items illustrated by the solid black line in figure 2. The measure *productcount_brandcluster* is an additional partition, which indicates how many items of a brand are active on a given date. Both measures illustrate the competition within the market or brand, indicating the range of choices customers face. The obtained values are assigned to all products of the concerning group, that are active on the given day. It has to be noted that especially in the beginning of the sample when no item from a specific brand is released so far, the counts for this brand can be zero. However, this does not affect the variable as there is no observation to which this realization could be assigned. The upper boundary for the brand-wise variable is the total number of items presented in table 1.

avgprice: Based on the allocation of products to their *brandcluster,* one can not only determine the number of active products but can also calculate statistics over the respective prices for each day and group. An obvious measure is given by the arithmetic mean of the price observations up to the forecast origin, which yields the variables *avgprice_brandcluster* and *avgprice_market*. The latter one was already informally introduced when discussing the general sample properties. The corresponding realizations over time are given by the blue solid line in figure 2.

avgpricedet: The average price deterioration is another measure that allows describing the properties of the respective group. The product-specific price deterioration $pdet_T = y_T/y_1$ is defined as the quotient of the price at a given date in relation to the initial product price. The variable *avgpricedet* can be obtained by averaging these $pdet_T$ values over the products that belong to the respective group. This yields the variables *avgpricedet_brandcluster* and *avgpricedet_market*. Both variables measure the price-wise erosion in the market or brand.

daysinsample: Additionally to the quantities derived based on the product price, a multivariate version of the *daysinmarket* variable is given by a time axis that spans the whole sample, comparable to the one displayed in figure 2. We, therefore, include the difference from the given day to the date of the first product launch as an additional variable in the analysis. To avoid that the variable starts with the value 0 on December, 18th 2008, we shift all values by one unit. Therefore, *daysinsample* ranges from 1 to 3215.

Methodology

Generally, any of the well-known statistical learning and machine learning techniques for binary classification can be used to generate predictions in the context of price event forecasting. The number of suitable methods is manifold and includes support vector machines, linear and quadratic discriminant analysis, naive Bayes, kernel methods, logit models (linear and non-linear like additive models), neural nets (for shallow not for deep learning), decision trees as well as ensemble-based methods like adaboost, gradient boosting machines (GBM), bagging and random forests (Cortes and Vapnik 1995; Efron and Hastie 2016; Goodfellow et al. 2016; Hastie et al. 2017; James et al. 2013).

All these supervised models use a training set, where each observation $i = 1, \dots, N$ consists of p features. In the univariate analysis each observation i is equivalent to a single, past time point, while in the multivariate approach N is determined by the sum of date-product pairs. The features are collected in a p-dimensional feature vector $\mathbf{x} \in \mathcal{X}$, which is constructed based on the univariate ($p = 14$) and multivariate features ($p = 22$) described above. Combining the feature vectors for all N observations yields the feature matrix $\mathbf{X}$. Each feature vector is associated with a categorical binary label $\mathrm{c_i} \in \{0,1\}$ that corresponds to the *event* E_T. The label vector $\mathbf{c}$ results from concatenating all N realizations of c_i. The purpose of most, while not all models mentioned, is the computation of a parametric or non-parametric probabilistic mapping $f: \mathcal{X} \to [0,1]$, which indicates the probability of the event $c = 1$. Using a decision rule, this probability estimate can be transferred to an event prediction, so that $f: \mathcal{X} \to \{0,1\}$. In the case of binary classification, using a majority vote is the most common option to determine a threshold (Kuhn and Johnson 2013). Both, the categorical prediction as well as the estimated event probabilities will be presented in the result section to allow the reader to gain an understanding of the modeling procedure and its respective performance.

The basic idea of our multivariate extension is the use of a common prediction function, which is shifted and modified by model- and brand-specific features to describe the brand and market environment for the given smartphone specifically. However, nothing is known about the direction and magnitude of the implicit autoregressive dependency structure. It could be time series-specific, common for some groups or even uniform for the entire sample. There might also be time dependencies either with regard to the internal time scale of a series (*daysinmarket*) or to the complete time scale (*daysinsample*). With the chosen settings, we collect empirical evidence to answer the fundamental question: Is the cohesive structure of consumer electronic goods identical for all products (respectively product groups) or is the function on which the indicator E_T depends product-specific and cannot be generalized?

For this purpose, most of the mentioned methods face severe challenges or a large increase in complexity, computational effort or manual configuration need. For example, the described knowledge gaps result in enormous difficulties when parametrizing classical logit models common in micro-econometrics (Winkelmann 2008). Traditional tree models, on the other hand, have the advantage of delivering explainable structures but suffer from low accuracy and high data dependencies. Fortunately, several machine learning methods have been developed in the last three decades that leverage and improve the idea of simple trees. The most prominent approaches are gradient boosting machines and random forests. GBMs are heavily dependent on an appropriate setting of hyperparameters. The modeling process, therefore, often incorporates excessive cross-validation and supplementary statistical approaches are required for the hyperparameter optimization. This additional effort often becomes an essential and dominating part of the modeling process. However, suboptimal hyperparameter configurations frequently result in over- or underfitting, making results unreliable. As our main objective in this paper is to show effects based on the general prediction setting, we choose to restrict our model portfolio to the random forest approach developed by Breiman (2001), which is more robust to overfitting, as well as to suboptimal hyperparameter settings, and delivers a comprehensive baseline applicable without excessive model tuning. This allows focusing on the empirical research question without effects being masked by methodological overhead and enables us to clearly isolate effects originating from the multivariate perspective.

Classification Algorithm – Random Forest

A random forest is based on the computation of a large and usually predefined number of binary classification trees where the inputs of each tree are changed randomly in two ways. First, for each tree, a subset of observations is drawn from $(\mathbf{X}, \mathbf{c})$ with replacement, which is referred to as bootstrap aggregation or bagging (Breiman 1996). Random forests where randomization is restricted to this step are referred to as bagged trees. Second, in each tree, the number of considered splitting variables in each node is restricted randomly to m variables with $m \ll p$.

In traditional tree methods like ID3 (Quinlan 1986) or CART (Breiman et al. 1984), small deviations of data often result in substantially changed trees, which is disadvantageous when using a single decision tree. This can be overcome and exploited by using many trees in ensemble methods. Beyond this, non-ensemble methods often require either termination criteria or full-tree pruning methods, which are cumbersome and error-prone. The main idea behind a random forest is to reduce these disadvantages by creating an ensemble of predictors and attaining the decorrelation of individual trees. While the first bagging step reduces the dependence on individual data points, the second step rectifies the remaining issue that the most influential features dominate the construction for all trees causing them to still be correlated. To give less important variables a chance to influence a given tree (e.g. by embedding interactions), a random selection procedure is applied in each splitting step. This reduces correlation further and also increases the variance over the ensemble. The bagging step includes asymptotically $1 - e^{-1} \approx 2/3$ of all observations in every tree (Efron 1979). This allows the calculation of an out-of-bag (OOB) error estimate, which is based on the simulated out-of-sample prediction of all observations that have not been used when constructing a specific tree of the ensemble (Efron and Hastie 2016).

The random forest results in many full trees, which are averaged to a combined prediction function. This means that for each prediction target, the feature vector is passed to all trees in the forest. For each tree, a binary prediction is generated resulting in multiple, (potentially) different predictions for the entire forest. The share of trees that vote for a positive outcome, meaning predict $c = 1$, yields the estimated probability of a price decrease within the considered horizon. Furthermore, the contribution of each feature can be assessed by computing the gains in accuracy or the reduction of heterogeneity when including the variable. The feature importance allows validating the usage of variables in the modeling process as it provides a measure of (predictive) influence.

These explanations underline that the random forest algorithm mainly relies on just two hyperparameters - the number of trees to grow B when constructing the forest and number of variables m, which are randomly considered for each split in the decision trees. While B should be sufficiently large, candidate values for m can be extracted from the literature. We, therefore, follow the recommendation of Liaw and

Wiener (2002), which is more rigorous than the widely used implementation of Pedregosa et al. (2011; Scikit-Learn 2019), and set $B = 500$ and $m = \lfloor\sqrt{p}\rfloor$.

Uni- and Multivariate Forecasting Settings

To answer the central question of this study, it is necessary to elaborate on the two different settings that allow us to draw conclusions on whether it is favorable to switch from a univariate to a multivariate setting when predicting price decline events. We test our modeling approach using a time series cross-validation scheme that is based on a rolling forecast simulation (Küsters 2012; Tashman 2000). For this purpose, we use an extending sliding window, meaning that the data basis of the model estimation grows when moving forward in time (Hyndman and Athanasopoulos 2018).

Because enough observations are needed to train the initial model, for the univariate model estimation, we reserve the first 89 time points as an initial training set. Having completed the first iteration that generated a prediction for one product, the forecast origin is moved further one observation extending the training set to 90 time points. The univariate, as well as the multivariate approaches, learn their model based on all observations that are available up to a certain point in time. However, the univariate model is limited to observations from one particular product, while the multivariate model unifies all observations and uses them for training. After each model estimation, one class prediction for the next decision recommendation is generated – either prices are expected to drop within the decision horizon or not. However, this prognosis is only valid for the day it was issued on, because a new model estimation is carried out the next day including the information from the day before using incremental data augmentation. While it would save computing resources to increase the step size from one to multiple days, this would ignore the effects of new observations on the parameter and prediction quality. We, therefore, chose not to assume time-constant dependencies, which would allow model transferability over time, but perform a full evaluation with forecast origin-dependent parameter estimates instead.

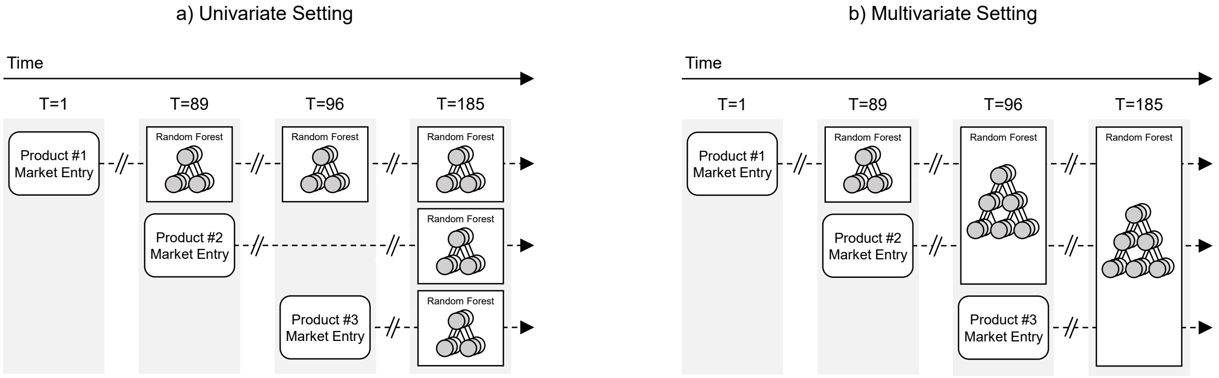

Figure 3. Forecasting Configurations

Figure 3 illustrates the forecasting settings for a hypothetical market with three available products. The univariate configuration is shown in figure 3a (left) and the multivariate forecasting setting is presented in figure 3b (right). For both configurations, four exemplarily chosen time points are displayed. It should be noted that the variable T refers to a shared time axis for the whole sample. It is, therefore, the illustrative analogon to the variable *daysinsample* opposed to the product-specific time axis represented by *daysinmarket*. Figure 3a shows that the univariate models are isolated and, therefore, can be estimated independently from each other. Product #1 is introduced into the market at $T = 1$, while two other products enter the market at $T = 89$ and $T = 96$. Each product needs an initial calibration data set of 89 days; thus, the first model is estimated at $T = 89$ for product #1. This is also the reason why no model can be calculated for product #2 at $T = 96$. At $T = 185$ three random forests – one for each product – are estimated. Keeping in mind that the whole sample in our study consists of 238 products and each time series contains two years, respectively 730 days of data, this leads to 150,892 univariate random forest models that are required to generate the necessary buying recommendations. This number results from the fact that 89 observations are withheld prior to the initial estimation and the last 7 predictions cannot be evaluated. Figure 3b shows that the number of estimated models is much smaller in the multivariate setting as only one random forest estimation per day is needed, which leads to only 3,119 required model estimations. However, the first product still requires an initial calibration set, wherefore in our exemplary setting, the multivariate model at $T = 89$ is identical to the univariate one because only one product exists at this day. Yet, all subsequent models can be estimated after the feature set can be calculated, which in our case requires price observations to compute the last seven returns. Then, new products such as product #2 can be added instantly to the model estimation at $T = 96$. Thus, the multivariate setting has three consequences: First, predictions for new products can be generated earlier. Second, more data is available for the estimation of one random forest (indicated by the bigger pictograms in figure 3b). Third, the total number of generated predictions per random forest grows because each multivariate model generates several predictions on a given day. Considering the number of available products per day and, therefore, the variable *productcount_market*, it reveals that one multivariate model generates between one and 126 predictions simultaneously, while the univariate models are strictly limited to one prediction per model and time point.

For the implementation of the developed method, we use the statistical programming environment R (R Core Team 2018) and the open-source package "randomForest" (Liaw and Wiener 2002) for model estimation and prediction generation, which is based on the original Fortran implementation of Breiman (2001, 2004). While one may assume that estimating the models for the univariate configurations takes longer as the number of models to estimate is by far larger than the roughly 3,000 models that are required in the multivariate configuration, this is not the case. A full run of the univariate setting that estimates models for all product-time point pairs and generates all corresponding predictions takes about 45 minutes opposed to ~ 18 hours for the multivariate estimation on a Workstation (Dual Xeon E5-2630v4, 64 GB Ram, Ubuntu 18.04, R 3.5.1) due to the increased variable set and increasing tree complexity.

Evaluation and Performance Assessment

While the random forest produces a class prediction that can be used as a buying recommendation directly, the underlying mechanism also produces a probability estimate based on the ensemble votes. It is noteworthy that the probability predictions generated in our setting are genuine out-of-sample predictions that do not utilize any information available beyond the forecasting origin. These need to be distinguished from the within-sample OOB probability estimates that are also generated by the approach and sometimes reported and referred to as prediction probabilities. While Liaw and Wiener (2002) state that they found the OOB probabilities to be fairly accurate, it can be shown that they are biased under some circumstances (Bylander 2002; Mitchell 2011). However, their biggest drawback in time series event forecasting applications is that they are only available after the corresponding observations have been incorporated in the model estimation and thus cannot be obtained when the prediction is needed.

A common evaluation measure of binary classifiers is the receiver operator characteristic (ROC). The ROC for each product has been generated on the basis of the out-of-sample probability estimates by varying the probability cutoff to generate the respective class predictions (Fawcett 2006; Sing et al. 2005). Figure 4 shows the ROC whereby figure 4a (left) refers to the performance of the univariate setting, while figure 4b (right) displays results for the multivariate approach. Each part of the figure describes and summarizes the performance of the 238 products in the sample. The product-specific ROC curves are illustrated by the thin yellow lines that visualize the heterogeneity of the results. The solid red line represents the average ROC, obtained by calculating the arithmetic mean of all product-dependent performance measures. The red, as well as the yellow lines, can be compared to the blue-colored line that indicates the performance of a random classifier as a benchmark. The six solid black lines in each of the two plots represent the 99%, 95%, 80%, respectively 20%, 5% and 1% quantiles of all product-specific ROC curves. The median is very similar to the shown red average line and is, therefore, not displayed as it provides no additional information.

The curves in figure 4 indicate that the multivariate estimation on average yields more precise predictions and thus leads to more correct buying recommendations for customers of electronic consumer goods. The average area under the curve ($\overline{AUC}$) calculated over all product-specific AUC values is with 67.6% for the multivariate setting compared to 63.8% for the univariate setting significantly higher as implied by the paired t-test ($p<0.001$). The same can be seen when looking at the black solid lines, which show larger distances to the benchmark, which, in turn, means that recommendations for most products improve when they are generated based on the multivariate configuration. This can also be seen when inspecting the maximum product AUC values in the top left corner of the plots. It increases from 79.9% in the univariate setting to 82.9% for the multivariate configuration. However, as the respective minimums show, not all products improve when switching to the multivariate prediction approach. When analyzing the individual AUC values, improvements are generated for 200 of the 238 products (84.0%). For the products, where the random forest approach cannot leverage the multivariate information, a median performance decrease of 1.5% in AUC is generated. This, however, is overcompensated by improvements of up to 13.8% realized by the remaining products. The median enhancement in AUC for the 200 products where the performance improves is 4.7%.

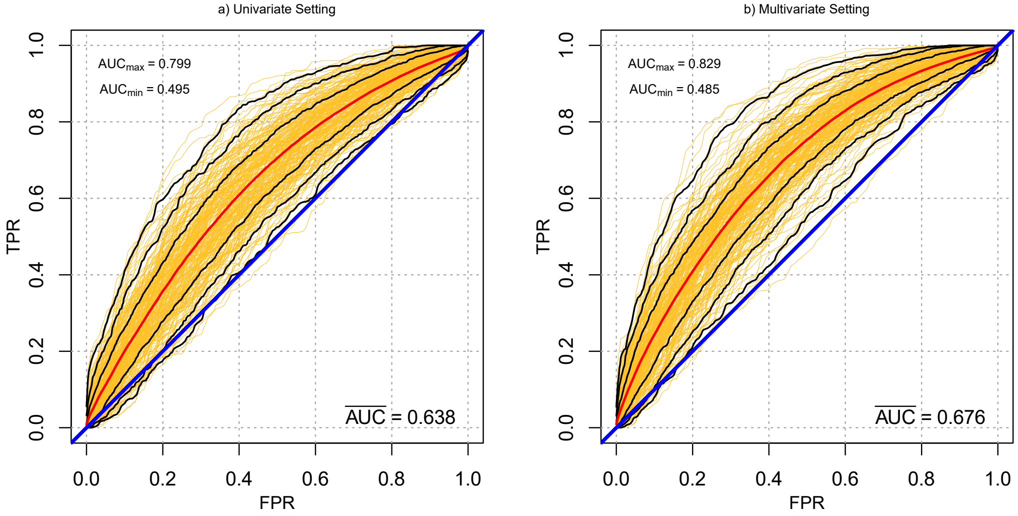

Figure 4. ROC Performance Plots

When the random forest algorithm generates class predictions directly, the underlying probability estimates are transferred using a static threshold as the cutoff value. In the case of a bivariate classification problem, the threshold is given by the Laplacian probability, which is the best cutoff without relying on further assumptions such as (potentially asymmetric) user- or product-dependent utility functions. However, in the case, that incorrect class predictions are subject to different misclassification costs the probability threshold can also be calibrated. Due to the lack of further information and for brevity and clarity, we revert to the class predictions generated by the majority vote for the remaining performance evaluation.

Table 2 shows the confusion matrices for the univariate as well as the multivariate approach when relying on the class predictions generated by the random forests using a static threshold of 0.5. The columns of table 2 represent the issued predictions, while the rows show the observed realizations of the event variable E_T. As the marginal distributions of the observations are identical for both confusion matrices, the last column is valid for both approaches. The last row shows, for both approaches individually, the sum of issued predictions to either buy the respective product immediately (FALSE, as $\hat{E}_T = 0$) or to wait with the purchase as the price is expected to drop (TRUE, as $\hat{E}_T = 1$). To make the results comparable, in total, each setting is evaluated based on 150,892 predictions.

Generally, it can be seen that the share of realized price decline events is larger with 65.43% than the portion of constant or increasing prices (34.57%). The ratio of generated predictions is in the univariate case (68.20% TRUE) more aligned with the marginal distribution of the empirical price event than the ones issued by the multivariate approach (77.56% TRUE), which slightly favors waiting recommendations. However, the multivariate approach delivers better performance for almost all performance measures, which are calculated based on the confusion matrix. The most common ones are accuracy, precision, false positive rate, true positive rate and F1 score (Han et al. 2012). In total, the share of correctly predicted events grows from 65.16% to 69.14% when extending the perspective from univariate to multivariate, leading to an accuracy gain of 3.98%, while the precision stays almost constant with a small absolute difference of 0.14%. It can be seen that the multivariate setting issues considerably fewer buying recommendations (FALSE) and the share of falsely issued purchase recommendations increases (true negative rate decreases from 45.61% to 37.85%). Another consequence is that the false positive rate rises slightly from 54.39% in case of the univariate approach to 62.15% for the multivariate perspective. However, the true positive rate increases by 10.18% to a value of 85.68% of correctly identified price decline events and the F1 score with 78.93% confirms the improvement when using the multivariate perspective (univariate F1 score 73.93%).

		Univariate Prediction		**Multivariate Prediction**		
		TRUE	FALSE	TRUE	FALSE	SUM
Obs.	TRUE	74,532	24,192	84,587	14,137	98,724
	FALSE	28,375	23,793	32,424	19,744	52,168
	SUM	102,907	47,985	117,011	33,881	150,892

Table 2. Confusion Matrix Results

Further insights can be obtained when discussing the variable importance that can be derived from the random forest models to validate the feature engineering process. The most common measure for variable importance is the mean decrease in accuracy, which is calculated based on the difference in OOB error rate when permutating the realizations for each variable, whose importance should be assessed after training the model (Breiman 2004). Averaging the decreases over all trees in the random forest yields the absolute (forecast origin-dependent) variable importance for each feature in the respective configuration. To make these configurations comparable, we normalized the sum of the accuracy decreases to unity and calculate the share for each variable as a relative measure of variable importance. However, we restrain from displaying detailed tables or images due to the length restrictions of this article and do not present a detailed discussion of the values over time. The shown values refer to the relative importance generated on the basis of the last model estimation. We, however, find that the importance stabilizes over time for both configurations after being more volatile for the first model estimations. In the univariate analysis, it can be observed that the individual returns show lower importance values compared to the remaining regressors. The variable r_T has the highest importance of all returns, while the importance of larger lags is subsequently lower. Univariate features that cause the largest accuracy decrease are *mean* and *drift*. This is identical for the multivariate perspective, where r_T accounts for (only) 3.02% of the average accuracy decrease of all regressors, while the *mean* as most important univariate feature accounts for 6.79% accuracy decrease. The remaining variables show importance values ranging from 4.42% for the categorical variable *brandcluster* up to 7.19% for the average price deterioration *avgpdet_brandcluster*. Features that describe the market environment rank in between these figures and show importance from 6.68% (*daysinsample*) to 5.79% (*avgpdet_market*), while the univariate features deliver comparable scores with values between 4.53% (*zeroinflation*) and 6.79% (*mean*). Concluding, one can say that the high importance of the brand and market features seems to contribute to the described superior performance of the multivariate setting.

Additionally to the increased performance on the given and comparable sample, the multivariate method has another advantage. Due to the multivariate model estimation, it is possible to generate predictions for new products that were recently launched without waiting 89 days. Usually, short or non-existent price histories would prohibit estimating a classification model. The multivariate perspective allows the transfer of the estimated dependencies to products shortly after they become available. The initialization period of a new product or time series is only bound to the construction of the feature vector, which in our case is one week after market entry because we included the past seven return values in our model. In the given setup, this leads to 170,326 instead of 150,892 generated predictions, an increase of 12.88%. While the fact that more predictions can be generated is interesting and potentially useful, the issued recommendations are only valuable to customers when they meet certain quality criteria. Additionally, reasonable performance would also indicate that the assumption of a constant and transferable dependency structure between products is valid. The multivariate model that is built on this assumption should be able to deliver roughly the same predictive performance independent of the stage in the life cycle. This means that the performance for the additionally generated predictions should strongly deviate from and be, thus, significantly higher than the performance of the univariate model. Besides, it should be more comparable to the performance of the multivariate model respectively deviate insignificantly.

Out of the 19,434 excess predictions, a total of 13,599 were correctly issued, leading to a prediction accuracy of 69.98%. The share of positively predicted values and, therefore, the precision amounts to 74.34%. The harmonic mean of the precision and the sensitivity leads to an F1 score of 79.69%. All measures indicate good performance exceeding the values from the univariate model. Testing for significant difference, we use the prediction results of all 238 products individually and test whether accuracy and F1 score are higher using a paired t-test. We found that the additional multivariate predictions perform significantly better than the univariate models ($p<0.05$). Furthermore, the t-test provides no evidence that the measures for the excess predictions systematically deviate from the performance of the multivariate model ($p>0.05$).

Conclusion

In this study, we outlined an approach to generate consumer-oriented buying decision recommendations on the basis of price time series events and presented evidence that the deterministic function used to model and predict the cohesive structure of price decline events for technological consumer goods can be generalized over groups of items. By developing features that extract characteristic properties of the data and takes the explicit temporal structure into account, we were able to use classification methods successfully in the time series context. We show that applying random forests to generate binary purchase recommendations delivers stable and valuable results. Enriching and pooling information of the isolated univariate perspective, where each item is treated individually, with multivariate features that describe the brand and market dynamics improves the quality and performance of predictive models significantly. Our work illustrates that powerful digital shopping assistant services can be developed to help customers schedule their purchase time decision and advocates the transition towards a data-driven service and business model landscape for price aggregation platforms. The practical implication of our work is that by combining information from different brands and products, services of PCS can be enhanced and by that they can help customers to save money.

We tested whether classification methods can be used to generate reliable predictions of future price events using a sample with 173,740 daily minimum price observations of 238 smartphones from the German consumer goods market with price histories between 2008 and 2017. A random forest generates both, price decline event probabilities as well as dichotomous buying recommendations indicating whether a product should be bought immediately or if the purchase should be delayed within a customizable decision horizon. The receiver operator characteristic, as well as the binary classifier evaluation, show that prediction accuracy rises significantly when adhering to the recommendations issued by the random forest configuration for multiple time series. This illustrates that dependency structures between products are similar and can be transferred between products and onto new time series. This finding is supported by the feature importance, showing that multivariate variables strongly contribute to the prediction accuracy. Additionally, the multivariate approach is capable of generating more predictions than the univariate setting, because it allows issuing recommendations for products that are new in the market. In accordance with the overall picture, these predictions align with the performance of the multivariate model providing further evidence for the value added by the generalization of the forecasting function.

The key limitation of this research is the lack of knowledge of the impact of the introduction of a recommendation service, meaning the (counter-) measures taken by retailers and the willingness to adapt by customers. Besides, there are multiple ways of improving the chosen methodological approach. Therefore, the findings in this paper raise several starting points for future research contributions. We restrained from applying excessive hyperparameter optimization methodologies to isolate the effects of the transition from a univariate to the multivariate prediction setting. However, while proceeding towards practical applications, it may be interesting and valuable to investigate procedures to optimize hyperparameters especially with respect to the time domain of the data. This would also make it possible to use more tuning intensive machine learning methods for the prediction and investigate their real-world performance when working with the developed and broadened feature space. Another extension of this paper would be to economically evaluation the added value of the generated predictions. This possibility results from the fact that every correct waiting recommendation yields a saving for the customer while every wrong delay of the purchase can cause a loss. This would practically illustrate the extent of the benefits of our approach.

References

Breiman, L. 1996. "Bagging Predictors," *Machine Learning* (24:2), pp. 123–140. (https://doi.org/10.1007/bf00058655).

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

Breiman, L. 2004. "Setting Up, Using And Understanding Random Forests V4.0." (https://www.stat.berkeley.edu/~breiman/Using_random_forests_v4.0.pdf).

Breiman, L., Friedman, J., Olshen, R. A., and Stone, C. J. 1984. *Classification and Regression Trees*, Chapman & Hall/CRC.

Bylander, T. 2002. "Estimating Generalization Error on Two-Class Datasets Using Out-of-Bag Estimates," *Machine Learning* (48:1–3), pp. 287–297. (https://doi.org/10.1023/A:1013964023376).

Cortes, C., and Vapnik, V. 1995. "Support-Vector Networks," *Machine Learning* (20), pp. 273–297. (https://doi.org/10.1109/64.163674).

Cunningham, J., and Zhu, N. 2019. "Shopbrain - Frequently Asked Questions," pp. 1–3. (https://www.shopbrain.com/faq, accessed October 28, 2019).

Drechsler, W., and Natter, M. 2011. "Do Price Charts Provided by Online Shopbots Influence Price Expectations and Purchase Timing Decisions?," *Journal of Interactive Marketing* (25:2), pp. 95–109. (https://doi.org/10.1016/j.intmar.2011.02.001).

Dutta, S. 2012. "Vulnerability to Low-Price Signals: An Experimental Study of the Effectiveness of Genuine and Deceptive Signals," *Journal of Retailing* (88:1), pp. 156–167. (https://doi.org/10.1016/j.jretai.2011.08.003).

Efron, B. 1979. "Bootstrap Methods: Another Look at the Jackknife," *The Annals of Statistics* (7:1), pp. 1–26. (https://doi.org/10.1214/aos/1176344552).

Efron, B., and Hastie, T. 2016. "Computer Age Statistical Inference: Algorithms, Evidence, and Data Science," *Computer Age Statistical Inference: Algorithms, Evidence, and Data Science*. (https://web.stanford.edu/~hastie/CASI/).

Etzioni, O., Tuchinda, R., Knoblock, C. A., and Yates, A. 2003. "To Buy or Not to Buy: Mining Airfare Data to Minimize Ticket Purchase Price," in *Proceedings of the Ninth ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pp. 119–128. (https://doi.org/10.1145/956750.956767).

Fawcett, T. 2006. "An Introduction to ROC Analysis," *Pattern Recognition Letters* (27:8), pp. 861–874. (https://doi.org/10.1016/j.patrec.2005.10.010).

Goodfellow, I., Bengio, Y., and Courville, A. 2016. *Deep Learning*, MIT Press.

Greenstein-Messica, A., Rokach, L., and Shabtai, A. 2017. "Personal-Discount Sensitivity Prediction for Mobile Coupon Conversion Optimization," *Journal of the Association for Information Science and Technology* (68:8), pp. 1940–1952. (https://doi.org/10.1002/asi.23838).

Han, J., Kamber, M., and Pei, J. 2012. *Data Mining: Concepts and Techniques*, (3rd ed.), Waltham, MA: Morgan Kaufmann. (https://doi.org/10.1016/B978-0-12-381479-1.00001-0).

Han, S., Gupta, S., and Lehmann, D. R. 2001. "Consumer Price Sensitivity and Price Threshold," *Journal of Retailing* (77:4), pp. 435–456. (https://doi.org/10.1016/S0022-4359(01)00057-4).

Hastie, T., Tibshirani, R., and Friedman, J. 2017. *The Elements of Statistical Learning - Data Mining, Inference, and Prediction*, (2nd ed.), Palo Alto, CA: Springer. (https://web.stanford.edu/~hastie/Papers/ESLII.pdf).

HDE. 2019. "Umsatz Durch E-Commerce (B2C) in Deutschland in Den Jahren 1999 Bis 2018 Sowie Eine Prognose Für 2019 (B2C-e-Commerce Revenue in Germany from 1999 to 2018)," *Handelsverband Deutschland - Statista ID: 3979*. (https://de.statista.com/statistik/daten/studie/3979/umfrage/e-commerce-umsatz-in-deutschland-seit-1999/).

Huang, S. L. 2011. "Designing Utility-Based Recommender Systems for e-Commerce: Evaluation of Preference-Elicitation Methods," *Electronic Commerce Research and Applications* (10:4), pp. 398–407. (https://doi.org/10.1016/j.elerap.2010.11.003).

Hyndman, R. J., and Athanasopoulos, G. 2018. *Forecasting : Principles and Practice*, (2nd ed.).

Hyndman, R. J., Koehler, A. B., Ord, K., and Snyder, R. D. 2008. *Forecasting with Exponential Smoothing - The State Space Approach*, Springer. (https://doi.org/10.1007/978-3-540-71918-2).

James, G., Witten, D., Hastie, T., and Tibshirani, R. 2013. *An Introduction to Statistical Learning with Applications in R*, (7th ed.), New York, NY: Springer. (http://www-bcf.usc.edu/~gareth/ISL/).

Johnson, J., Tellis, G. J., and Ip, E. H. 2013. "To Whom, When, and How Much to Discount? A Constrained Optimization of Customized Temporal Discounts," *Journal of Retailing* (89:4), pp. 361–373.

(https://doi.org/10.1016/j.jretai.2013.08.002).

Kang, Y., Hyndman, R. J., and Smith-Miles, K. 2017. "Visualising Forecasting Algorithm Performance Using Time Series Instance Spaces," *International Journal of Forecasting* (33:2), pp. 345–358. (https://doi.org/10.1016/j.ijforecast.2016.09.004).

Keeney, R. L., and Raiffa, H. 1993. *Decisions with Multiple Objectives - Preferences and Value Tradeoffs*, Victoria, Australia: Cambridge University Press.

Klein, R., and Steinhardt, C. 2008. *Revenue Management - Grundlagen Und Mathematische Modelle (Fundamentals and Mathematical Models)*, Augsburg, Germany: Springer.

Kömm, H., and Küsters, U. 2015. "Forecasting Zero-Inflated Price Changes with a Markov Switching Mixture Model for Autoregressive and Heteroscedastic Time Series," *International Journal of Forecasting* (31:3), pp. 598–608. (https://doi.org/10.1016/j.ijforecast.2014.10.008).

Kuhn, M., and Johnson, K. 2013. *Applied Predictive Modeling*, New York, NY: Springer. (https://doi.org/10.1007/978-1-4614-6849-3).

Küsters, U. 2012. "Evaluation, Kombination Und Auswahl Betriebswirtschaftlicher Prognoseverfahren (Evaluation, Combination and Selection of Business Forecasting Methods)," in *Prognoserechnung* (7th ed.), P. Mertens and S. Rässler (eds.), Heidelberg, Germany: Springer, pp. 423–467.

Liaw, A., and Wiener, M. 2002. "Classification and Regression by RandomForest," *R News* (2:3), pp. 18–22. (https://doi.org/10.1159/000323281).

Makridakis, S., and Hibon, M. 2000. "The M3-Competition : Results, Conclusions and Implications," *International Journal of Forecasting* (16), pp. 451–476.

Makridakis, S., Spiliotis, E., and Assimakopoulos, V. 2018. "The M4 Competition: Results, Findings, Conclusion and Way Forward," *International Journal of Forecasting* (34:4), pp. 802–808. (https://doi.org/10.1016/j.ijforecast.2018.06.001).

Meierhoff, E. 2018. *Data Based Overview of the German E-Commerce Market*, Berlin, Germany: Idealo.

Mitchell, M. W. 2011. "Bias of the Random Forest Out-of-Bag (OOB) Error for Certain Input Parameters," *Open Journal of Statistics* (01:03), pp. 205–211. (https://doi.org/10.4236/ojs.2011.13024).

Moon, S., Russell, G. J., and Duvvuri, S. D. 2006. "Profiling the Reference Price Consumer," *Journal of Retailing* (82:1), pp. 1–11. (https://doi.org/10.1016/j.jretai.2005.11.006).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, É. 2011. "Scikit-Learn : Machine Learning in Python," *Journal of Machine Learning Research* (12), pp. 2825–2830.

Quinlan, J. R. 1986. "Induction of Decision Trees," *Machine Learning* (1), pp. 81–106. (https://doi.org/10.1007/BF00116251).

R Core Team. 2018. "R: A Language and Environment for Statistical Computing," Vienna, Austria. (https://www.r-project.org/).

Rydberg, T. H., and Shephard, N. 2003. "Dynamics of Trade-by-Trade Price Movements: Decomposition and Models," *Journal of Financial Econometrics* (1:1), pp. 2–25. (https://doi.org/10.1093/jjfinec/nbg002).

Scikit-Learn. 2019. *Scikit-Learn User Guide - Release 0.20.3*, Scitkit-Learn Developers. (https://scikit-learn.org/stable/_downloads/scikit-learn-docs.pdf).

Sing, T., Sander, O., Beerenwinkel, N., and Lengauer, T. 2005. "ROCR: Visualizing Classifier Performance in R," *Bioinformatics* (21:20), pp. 3940–3941. (https://doi.org/10.1093/bioinformatics/bti623).

Smyl, S., Ranganathan, J., and Pasqua, A. 2018. "M4 Forecasting Competition: Introducing a New Hybrid ES-RNN Model," *Uber Engineering*. (https://eng.uber.com/m4-forecasting-competition/, accessed April 21, 2019).

Statista. 2018. "E-Commerce in Germany." (https://www.statista.com/study/32414/e-commerce-in-germany-statista-dossier/).

Sucarrat, G., and Grønneberg, S. 2016. "Models of Financial Return with Time-Varying Zero Probability," *MPRA Working Paper* (68931), pp. 1–25.

Tashman, L. J. 2000. "Out-of-Sample Tests of Forecasting Accuracy: An Analysis and Review," *International Journal of Forecasting* (16:4), pp. 437–450. (https://doi.org/10.1016/S0169-2070(00)00065-0).

Udachny, A. 2015. "When to Buy Ryanair Tickets? - Presenting AirHint.Com." (http://www.airhint.com/about).

Wang, J., and Zhang, Y. 2011. "Utilizing Marginal Net Utility for Recommendation in E-Commerce," *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in*

Information Retrieval, pp. 1003–1012. (https://doi.org/10.1145/2009916.2010050).

Weiß, C. H. 2009. *Categorical Time Series Analysis and Applications in Statistical Quality Control*, (1st ed.), Würzburg, Germany: Winter Industries.

Winkelmann, R. 2008. *Econometric Analysis of Count Data*, (5th ed.), Berlin, Germany: Springer. (https://doi.org/10.1007/978-3-540-78389-3).